IT Industry Readiness Bootcamp Program

A Project of Information Science & Technology Department, Government of Sindh

Web Engineering

Course Manual

Prepared By:

Mr. Nisar Ahmed Siddiqui

Table of Contents

CHAPTER 1 – INTRODUCTION TO WEB

1.1 Web development

Web Development refers to the creating, building, and maintaining of websites. It includes aspects such as web design, web publishing, web programming, and database management. It is the creation of an application that works over the internet i.e. websites.

Web programming, also known as web development, is the creation of dynamic web applications. Examples of web applications are social networking sites like Facebook or e-commerce sites like Amazon

The word Web Development is made up of two words, that is:

- Web: It refers to websites, web pages or anything that works over the internet.
- Development: It refers to building the application from scratch.

1.2 Web Development Overview

There are two broad divisions of web development – front-end development (also called client-side development) and back-end development (also called server-side development).

Front-end development refers to constructing what a user sees when they load a web application – the content, design and how you interact with it. This is done with three codes – HTML, CSS and JavaScript.

HTML, short for Hyper Text Markup Language, is a special code for ‘marking up’ text in order to turn it into a web page. Every web page on the net is written in HTML, and it will form the backbone of any web application.

CSS, short for Cascading Style Sheets, is a code for setting style rules for the appearance of web pages. CSS handles the cosmetic side of the web. Finally, JavaScript is a scripting language that's widely used to add functionality and interactivity to web pages.

Back-end development controls what goes on behind the scenes of a web application. A back-end often uses a database to generate the front-end.

Back-end scripts are written in many different coding languages and frameworks, such as…

- PHP
- Ruby on Rails
- ASP.NET
- Perl
- Java
- Node.js
- Python

1.3 Who is a web developer?

A web developer is at heart an interactive artist. They're someone driven by a deep desire to create things. A web developer's canvas is a user's web browser.

Much like how a curious child takes pleasure in making toys by joining LEGO blocks—and then experiences a similar joy in taking things apart to see how they're made, a web developer's job is to use the basic building blocks of the web (like HTML, CSS and JavaScript) to create something complex like a webpage. Don't sweat it! We'll get into all of those terms soon.

It is also the web developer's job to diagnose problems in a website's functionality, to understand how something works by reading the code behind it, and to make changes to fix any issues. That essentially makes web developers the physicians of the world wide web. When we talk about the world wide web, we're mainly referring to websites and web applications. At this point, the curious aspiring developer in you might be wondering, "What's the difference between the two?"

Well, we're glad you asked! A website is usually a simple page or a group of pages (the popular web comic site, xkcd, for example). However, modern websites are much more than that. Take Google Docs, for example. It has a nice interface that, once opened, looks very much like a desktop software like Microsoft Office. Such complex websites are often termed web applications. A web developer today is expected to know how to create and work with web applications. Having said that, most people use the terms interchangeably, so don't get too hung up on the terminology!

1.3.1 Types of web developers

There are different types of web developers who focus on different areas. These include:

Frontend developers: Frontend developers implement web page designs using HTML and CSS. They make sure the website looks pretty on different devices, and that the forms and buttons work.

Backend developers: Backend developers create the backbone of the web application. They write code logic that handles a user's input (for example, what should happen when you click the signup button after filling in a form).

Full stack developers: Full stack developers do bits of both backend and frontend. Depending on the problem at hand, they can switch cape and move stacks. You can learn more about the differences between frontend and backend development in this guide.

There are many other specific roles in web development, like system architects, AI, machine learning and security engineers. These roles require more specialized knowledge of one or more of the above types of development, so many professionals in these roles will start by gaining some general web development experience.

1.4 What are the skills required to become a web developer?

What does it take to become a web developer? Essentially, just three things: HTML, CSS and JavaScript—the **three pillars** of the web, which we'll be learning about over the next few days. Together, these three pillars make every website work, defining the content to be displayed (HTML), telling a browser how to display that content (CSS), and making the content interactive (JavaScript), respectively.

A web developer is well versed in these three technologies. They can read other people's code and make changes to it. They can find and debug bugs (shortcomings in existing code). A web developer might, at times, work on a new project (a new website) from scratch, or may have to work on an existing website and make it better. A typical day in the life of a web developer involves fixing bugs, developing new features (that is, enhancements) and webpages, and working with other developers to discuss ways to solve problems.

Don't be overwhelmed by all of these details. You'll soon see that they're all very connected, and learning one of these automatically makes you good at a few others!

1.5 Network: A network is a group of two or more computer systems (Multiple gadgets, additionally called hosts), which are related through a couple of channels for the motive of sending and receiving data (records/media) in a shared environment. The community also can consist of serval gadgets/mediums that resource communique among or extra machines; those gadgets are called Network devices and consist of routers, switches, hubs, and bridges, amongst others.

1.6 The Internet: In simplest terms, the Internet is a global network comprised of smaller networks that are interconnected using standardized communication protocols. The Internet standards describe a framework known as the Internet protocol suite. This model divides methods into a layered system of protocols.

These layers are as follows:

1. **Application layer (highest) –** concerned with the data(URL, type, etc.). This is where HTTP, HTTPS, etc., comes in.

2. **Transport layer –** responsible for end-to-end communication over a network.
3. **Network layer –** provides data route.

The Internet provides a variety of information and communication facilities; contains forums, databases, email, hypertext, etc. It consists of private, public, academic, business, and government networks of local to global scope, linked by a broad array of electronic, wireless, and optical networking technologies.

1.7 History of Internet & Web

The internet is a global network of interconnected computers and servers that allows people to communicate, share information, and access resources from anywhere in the world. It was created in the 1960s by the "US Department of Defence" as a way to connect computers and share information between researchers and scientists.

Talking about the history of internet, the ARPANET (Advanced Research Projects Agency Network, later renamed the internet) established a successful link between the University of California Los Angeles and the Stanford Research Institute on October 29, 1969. Libraries automate and network catalogs outside of ARPANET in the late 1960s.

TCP/IP (Transmission Control Protocol and Internet Protocol) is established in the 1970s, allowing internet technology to mature. The development of these protocols aided in the standardization of how data was sent and received via the internet. NSFNET (National Science Foundation Network), the 56 Kbps backbone of the internet, was financed by the National Science Foundation in 1986. Because government monies were being used to administer and maintain it, there were commercial restrictions in place at the time.

In the year 1991, a user-friendly internet interface was developed. Delphi was the first national commercial online service to offer internet connectivity in July 1992. Later in May 1995, All restrictions on commercial usage of the internet are lifted. As a result, the internet has been able to diversify and grow swiftly. Wi-Fi was first introduced in 1997. The year is 1998, and Windows 98 is released. Smartphone use is widespread in 2007. The 4G network is launched in 2009. The internet is used by 3 billion people nowadays. By 2030, there are expected to be 7.5 billion internet users and 500 billion devices linked to the internet.

The World Wide Web, or simply the web, is a system of interconnected documents and resources, linked together by hyperlinks and URLs. It was created by Tim Berners-Lee in 1989 as a way for scientists to share information more easily. The web quickly grew to become the most popular way to access information on the internet.

Together, the internet and the web have revolutionized the way we communicate, do business, and access information. They have made it possible for people all over the world to connect with each other instantly and have transformed many industries, from media and entertainment to education and healthcare.

Computers and their structures are tough to approach, and it is made even extra tough while you want to recognize phrases associated with the difficulty this is already utilized in regular English, Network, and the net will appear to be absolutely wonderful from one some other, however, they may seem like identical

1.7.1 Uses of the Internet:

- **E-mail:** E-mail is an electronic message sent across a network from one computer user to one or more recipients. It refers to the internet services in which messages are sent from and received by servers.
- **Web Chat:** Web chat is an application that allows you to send and receive messages in real-time with others. By using Internet chat software, the user can log on to specific websites and talk with a variety of other users online. Chat software is interactive software that allows users to enter comments in one window and receive responses from others who are using the same software in another window.
- **World Wide Web:** The World Wide Web is the Internet's most popular information exchange service. It provides users with access to a large number of documents that are linked together using hypertext or hyperlinks.
- **E-commerce:** E-commerce refers to electronic business transactions made over the Internet. It encompasses a wide range of product and service-related online business activities.
- **Internet telephony:** The technique that converts analog speech impulses into digital signals and routes them through packet-switched networks of the internet is known as internet telephony.
- **Video conferencing:** The term "video conferencing" refers to the use of voice and images to communicate amongst users.

Web Client

The client (or user) side of the Internet. The Web browser on the user's computer or mobile device is referred to as a Web client. It could also apply to browser extensions and helper software that improve the browser's ability to support specific site services.

Web browser

A web browser is a software program software that searches for, retrieves, and presentations material which includes Web pages, photos, videos, and different files. The browser sends a request to the Webserver, which then transmits the statistics returned to the browser, which presentations the findings at the laptop. Example – Mozilla Firefox, Microsoft Edge, Google Chrome, Safari etc.

Webpage

An internet web page (additionally called a web page) is a report that may be regarded in an internet browser at the World Wide Web. HTML (HyperText Markup Language) and CSS (Cascading Style Sheet) are used to generate the primary shape of an internet web page. An internet web page is generally a segment of an internet site that carries statistics in plenty of formats, which includes textual content inside the shape of paragraphs, lists, tables, and so on.

The home web page is the beginning or first web page of an internet site. It gives trendy statistics and connections to all the internet pages which are associated. Every internet web page has its personal deal with. This may be visible withinside the deal with the bar. As a result, if we need to get admission to a selected internet web page, the deal needs to be placed inside the browser's deal with bar.

Website

An internet site, in trendy, is a group of statistics approximately statistics prepared into many internet pages. An internet site is probably made for a sure motive, subject matter, or to provide a service. An internet site (abbreviated as "website" or "site") is a group of online pages connected collectively through links and saved on an internet server. By clicking on links, a tourist can pass from one web page to the next. An internet site's pages also are connected below one area call and proportion a not unusual place subject matter and template.

Search Engine

Search engines are websites that search on the internet on behalf of users and show a listing of results. More than actually written may be discovered on seek engines. You can be capable of looking for different online content material which includes photographs, video content material, books, and news, in addition to gadgets and offerings, relying on the seek engine you are the use of.

To make use of the Internet, you do not always want to recognize the deal with an internet site. It is crucial to recognize the way to do a look for statistics. Using a seek engine is one of the only methods to seek. A seek engine can help you in finding what you are looking for. You also can appearance up net maps and instructions to help you to plot your adventure from one factor to some other. Example: Google, Bing, DuckDuckGo, yahoo, etc.

Web 1.0: The Beginning of the Internet

Web 1.0, which emerged with the invention of the Internet, featured static web pages that could only be viewed. These early websites were essentially digital brochures, with limited interactivity and no ability for users to contribute content. This era saw the rise of early browsers such as Mosaic and Netscape Navigator, which made it easier for users to access the Web.

Web 2.0: The Era of Social Media and User-Generated Content

Web 2.0, which emerged during the "Internet Boom" era of the late 1990s and early 2000s, introduced more dynamic web pages and websites, and allowed for user-generated content. This era saw the rise of social media platforms such as Facebook, Twitter, and YouTube, as well as the introduction of blogs, wikis, and other forms of user-generated content.

Web 3.0: The Future of the Internet

Web 3.0, which is still in development, promises to bring a more personalized and intuitive experience for users. It is expected to be more intelligent and able to understand the context and intent of the user's actions.

This will be achieved through the integration of Artificial Intelligence (AI) and Machine Learning (ML) technologies.

Many experts in the tech industry claimed that Web 3.0 is the next evolution of the internet that is focused on decentralization, intelligence, and security. It aims to create a more personalized, intuitive, and inclusive experience for users by leveraging new technologies such as blockchain and artificial intelligence. The goal of Web 3.0 is to make the internet more democratic, open-source, and secure.

The World Wide Web:

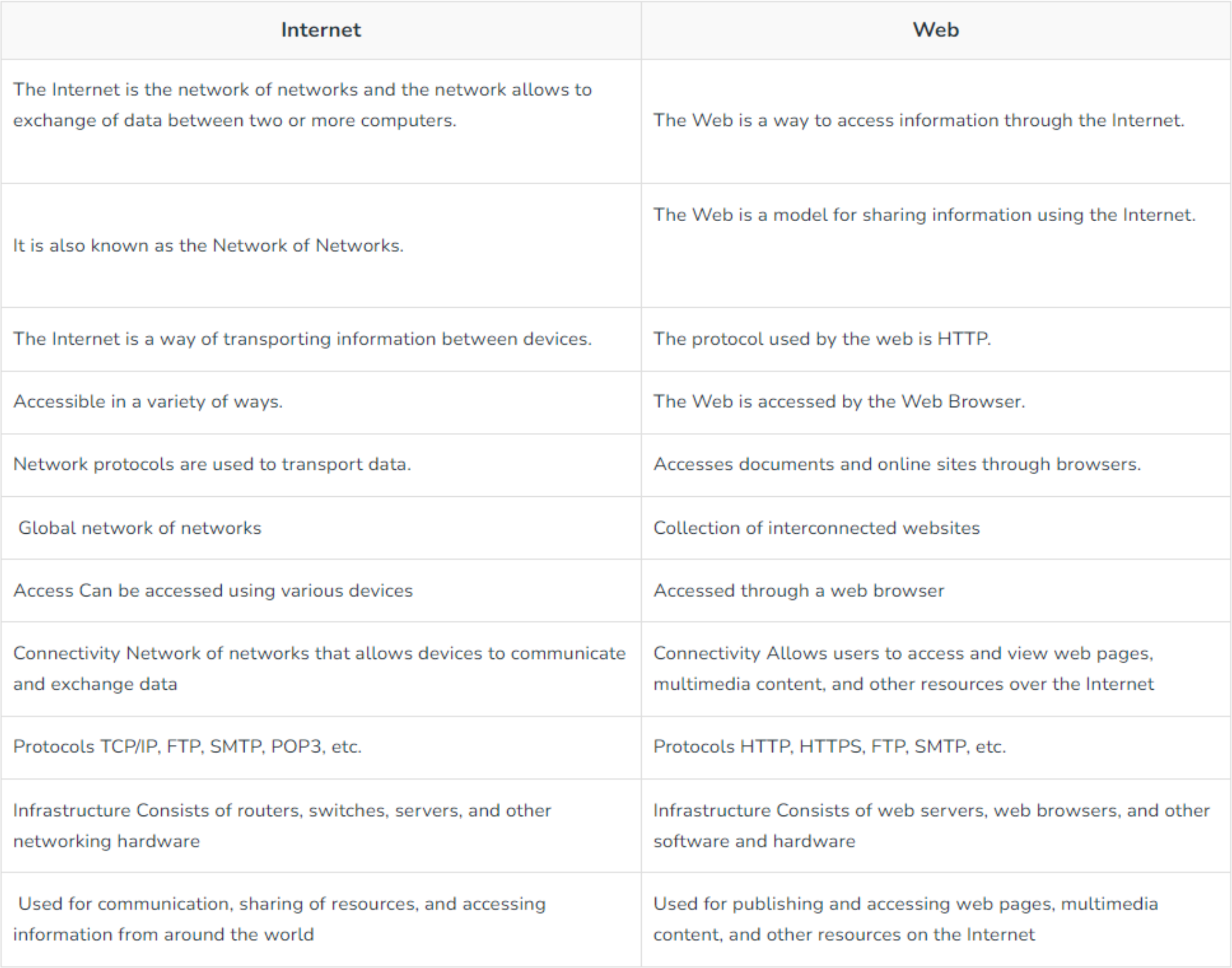
The Web is the only way to access information through the Internet. It's a system of Internet servers that support specially formatted documents. The documents are formatted in a markup language called **HTML**, or "Hyper Text Markup Language", which supports a number of features including links and multimedia. These documents are interlinked using hypertext links and are accessible via the Internet.

To link hypertext to the Internet, we need:

1. The markup language, i.e., HTML.
2. The transfer protocol, e.g., HTTP.
3. Uniform Resource Locator (URL), the address of the resource

We access the Web using Web browsers.

Difference between Web and Internet:



Internet	Web
The Internet is the network of networks and the network allows to exchange of data between two or more computers.	The Web is a way to access information through the Internet.
It is also known as the Network of Networks.	The Web is a model for sharing information using the Internet.
The Internet is a way of transporting information between devices.	The protocol used by the web is HTTP.
Accessible in a variety of ways.	The Web is accessed by the Web Browser.
Network protocols are used to transport data.	Accesses documents and online sites through browsers.
Global network of networks	Collection of interconnected websites
Access Can be accessed using various devices	Accessed through a web browser
Connectivity Network of networks that allows devices to communicate and exchange data	Connectivity Allows users to access and view web pages, multimedia content, and other resources over the Internet
Protocols TCP/IP, FTP, SMTP, POP3, etc.	Protocols HTTP, HTTPS, FTP, SMTP, etc.
Infrastructure Consists of routers, switches, servers, and other networking hardware	Infrastructure Consists of web servers, web browsers, and other software and hardware
Used for communication, sharing of resources, and accessing information from around the world	Used for publishing and accessing web pages, multimedia content, and other resources on the Internet

URI: URI stands for **'Uniform Resource Identifier'**. A URI can be a name, locator, or both for an online resource whereas a URL is just the locator. URLs are a subset of URIs. A URL is a human-readable text that was designed to replace the numbers (IP addresses) that computers use to communicate with servers.

A URL consists of a protocol, domain name, and path (which includes the specific subfolder structure where a page is located) like-

URL = protocol://WebSiteName.topLevelDomain/path

1. Protocol – HTTP or HTTPS.
2. WebSiteName – geeksforgeeks, google etc.
3. topLevelDomain- .com, .edu, .in etc.
4. path- specific folders and/or subfolders that are on a given website.

Who governs the Internet?

The Internet is not governed and has no single authority figure. The ultimate authority for where the Internet is going rests with the Internet Society, or ISOC.

ISOC is a voluntary membership organization whose purpose is to promote global information exchange through Internet technology.

- ISOC appoints the IAB- Internet Architecture Board. They meet regularly to review standards and allocate resources, like addresses.
- IETF- Internet Engineering Task Force. Another volunteer organization that meets regularly to discuss operational and technical problems.

Uses of Internet and the Web:

1. *Communication*: The internet and web have made communication faster and easier than ever before. We can now send emails, chat online, make video calls, and use social media platforms to connect with people all over the world.
2. *Information sharing*: The web has made it possible to access vast amounts of information on any topic from anywhere in the world. We can read news articles, watch videos, listen to podcasts, and access online libraries and databases.
3. *Online shopping*: The internet and web have revolutionized the way we shop. We can now browse and purchase products online, from clothes and groceries to electronics and furniture.
4. *Entertainment*: The internet and web provide a wealth of entertainment options, from streaming movies and TV shows to playing online games and listening to music.
5. *Education*: The web has made it possible to access educational resources from anywhere in the world. We can take online courses, access e-books and digital libraries, and connect with educators and other learners through online communities.
6. *Business*: The internet and web have transformed the way businesses operate. Companies can now use e-commerce platforms to sell products and services, collaborate with remote workers, and access global markets.
7. *Research*: The internet and web have made it easier for researchers to access and share information. We can now access scientific journals and databases, collaborate with other researchers online, and conduct surveys and experiments through online platforms.

Issues in Internet and the Web :

1. *Privacy and security*: The internet and web are vulnerable to various security threats, such as hacking, identity theft, and phishing attacks. These threats can compromise our personal information, such as login credentials, financial information, and personal data.

2. *Cyberbullying*: The anonymity of the internet and web can lead to cyberbullying, where individuals are harassed or threatened online. Cyberbullying can have severe consequences, including depression, anxiety, and suicide.
3. *Online* addiction: The internet and web can be addictive, and individuals can spend hours browsing social media or playing online games, leading to neglect of other important aspects of their lives.
4. *Disinformation*: The internet and web are filled with inaccurate or false information, which can lead to misinformation, propaganda, and conspiracy theories.
5. *Digital divide*: Access to the internet and web is not universal, and many individuals, particularly those in low-income areas or rural communities, lack access to reliable and high-speed internet.
6. *Online censorship*: Some governments or organizations may censor or restrict access to certain websites or information, limiting freedom of speech and expression.
7. *Environmental impact*: The internet and web consume a significant amount of energy, contributing to carbon emissions and climate change.

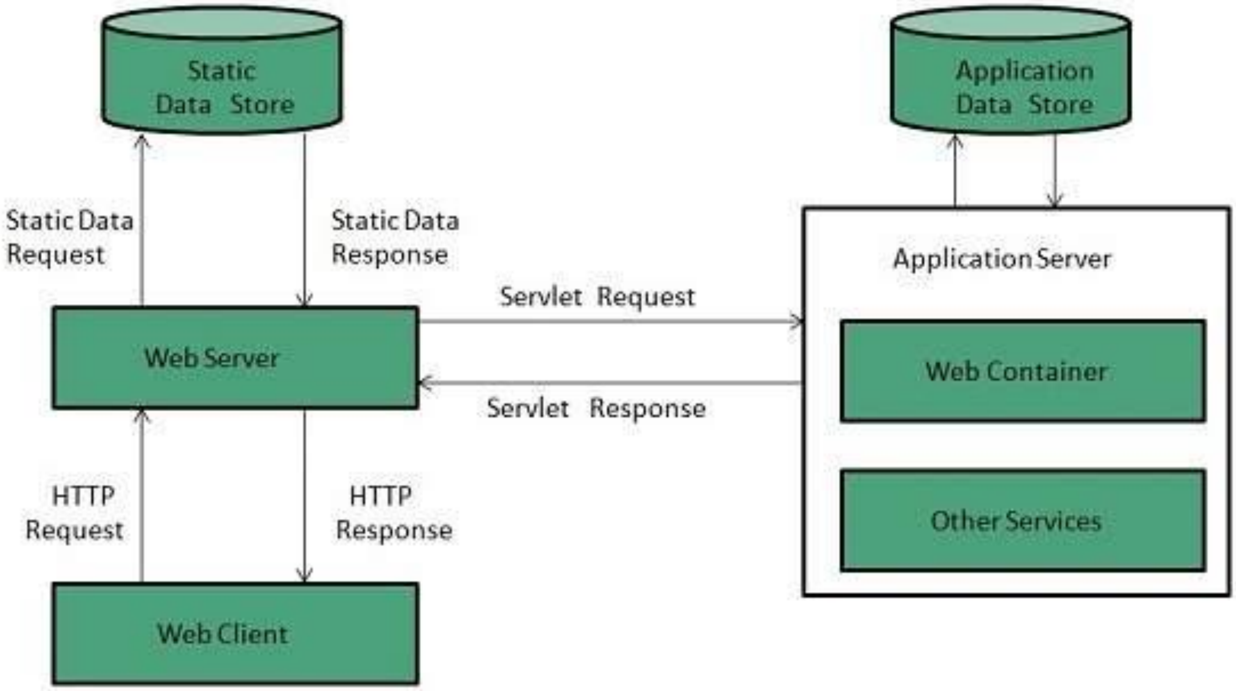
1.8 Web Server

Web server is a computer where the web content is stored. Basically web server is used to host the web sites but there exists other web servers also such as gaming, storage, FTP, email etc.

1.8.1 Web Server Working

Web server respond to the client request in either of the following two ways:

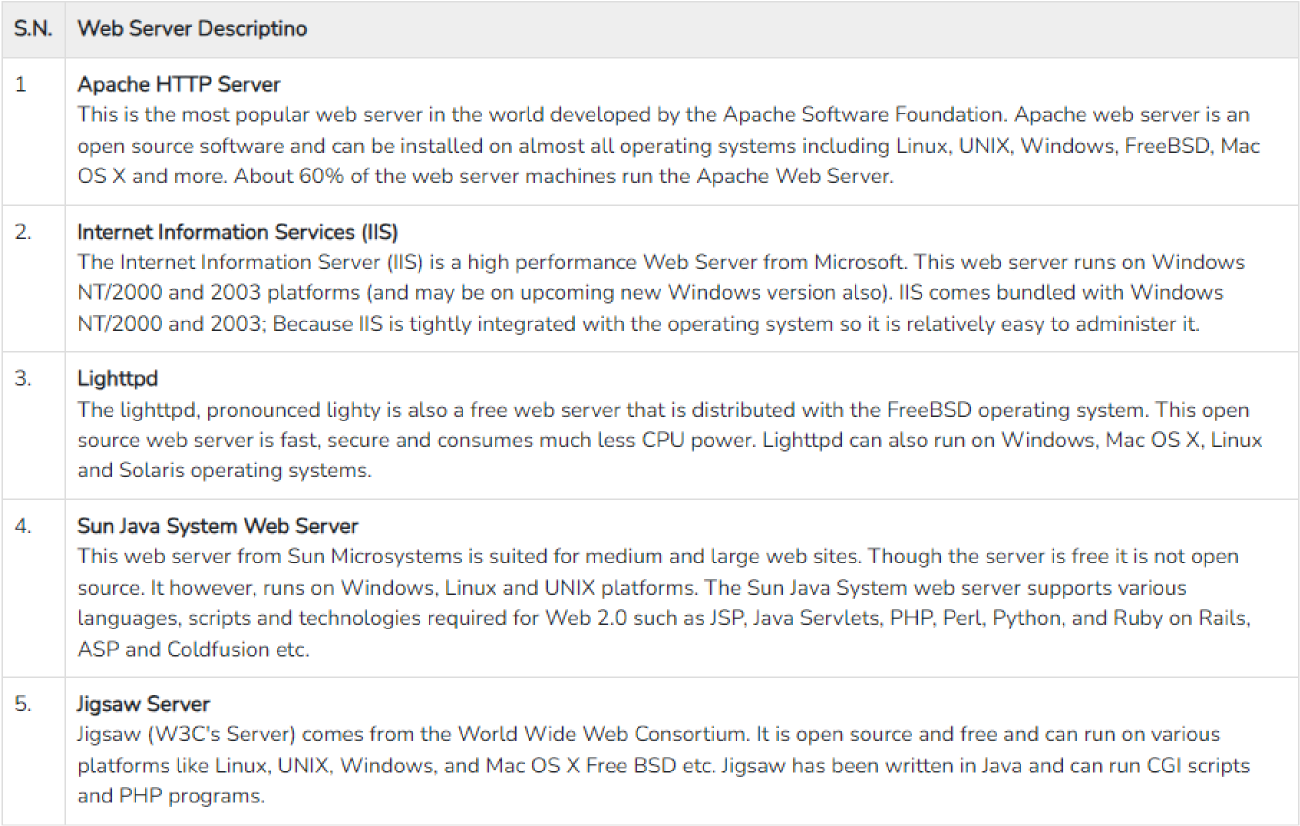
- Sending the file to the client associated with the requested URL.
- Generating response by invoking a script and communicating with database



Key Points

- When client sends request for a web page, the web server search for the requested page if requested page is found then it will send it to client with an HTTP response.
- If the requested web page is not found, web server will the send an **HTTP response:Error 404 Not found.**
- If client has requested for some other resources then the web server will contact to the application server and data store to construct the HTTP response.

Following table describes the most leading web servers available today:



S.N.	Web Server Descriptino
1	**Apache HTTP Server** This is the most popular web server in the world developed by the Apache Software Foundation. Apache web server is an open source software and can be installed on almost all operating systems including Linux, UNIX, Windows, FreeBSD, Mac OS X and more. About 60% of the web server machines run the Apache Web Server.
2.	**Internet Information Services (IIS)** The Internet Information Server (IIS) is a high performance Web Server from Microsoft. This web server runs on Windows NT/2000 and 2003 platforms (and may be on upcoming new Windows version also). IIS comes bundled with Windows NT/2000 and 2003; Because IIS is tightly integrated with the operating system so it is relatively easy to administer it.
3.	**Lighttpd** The lighttpd, pronounced lighty is also a free web server that is distributed with the FreeBSD operating system. This open source web server is fast, secure and consumes much less CPU power. Lighttpd can also run on Windows, Mac OS X, Linux and Solaris operating systems.
4.	**Sun Java System Web Server** This web server from Sun Microsystems is suited for medium and large web sites. Though the server is free it is not open source. It however, runs on Windows, Linux and UNIX platforms. The Sun Java System web server supports various languages, scripts and technologies required for Web 2.0 such as JSP, Java Servlets, PHP, Perl, Python, and Ruby on Rails, ASP and Coldfusion etc.
5.	**Jigsaw Server** Jigsaw (W3C's Server) comes from the World Wide Web Consortium. It is open source and free and can run on various platforms like Linux, UNIX, Windows, and Mac OS X Free BSD etc. Jigsaw has been written in Java and can run CGI scripts and PHP programs.

CHAPTER 2 – INTRODUCTION TO HTML

2.1 HTML

HTML (Hyper Text Markup Language) is a special language that helps us create and structure web pages on the internet. A website is simply a collection of multiple web pages, and HTML is an important element in creating any web page. One can think of it as the building blocks for web pages. It is basically a Language that has its own syntax and rules, which must be followed for any purposeful website creation.

Imagine you want to make a digital invite for Christmas with the occasion, date, and venue. You want the occasion to be in a bigger font and the date and venue in a smaller font. Now, you can use HTML to write a set of instructions or rules to tell your computer how to display the invite.

We can use different tags in HTML to define different parts of the web page. You put tags around your content to give it meaning, like special labels. For example, for the occasion heading, you use the heading tag: <h1>. For the date and venue, you use the <h2> (smaller heading) and <p> (para) tags.

2.1.1 HTML 1.0

One of the first versions of HTML, HTML 1.0, was released in 1993, laying down the foundations for building web pages. Some of its primary characteristics were:

- Structuring Elements: HTML 1.0 introduced elements that made it possible to add structure to the content on web pages. These included headings, lists, paragraphs, and images.
- Comparatively Simpler: HTML 1.0 was very simple in comparison to later versions of HTML. It didn’t have styling options or the ability to control how content would display in a web browser. Additionally, HTML 1.0 lacked any support for tables.
- Font Support: Although HTML 1.0 introduced fonts, it was limited. In other words, there was minimal scope for changing the text style or size.

Nonetheless, HTML 1.0 initiated the field of web development, and its features kept advancing with every updated version.

2.1.2 HTML 2.0

The HTML 2.0 version was released in 1995 and had considerable improvements from the previous version. Some of them were:

- Standardization of HTML: This version made HTML into a standard by establishing common rules and regulations that all web browsers had to follow.

- Forms: Most importantly, HTML 2.0 introduced the concept of forms, which allowed users to input data on web pages. However, the forms were still basic and only contained text boxes and buttons.
- Tables: HTML 2.0 introduced the <table> tag for creating tabular data on web pages, contributing to better organization of data.

Formation of the W3C: During HTML 2.0, browsers started making their own tags, because of which there was inconsistency across different browsers. To resolve the issue and establish standardization, the W3C (World Wide Web Consortium) was formed. This helped web browsers to render tags in a consistent manner.

2.1.3 HTML 3.2

HTML 3.2 was the next major successor to HTML 2.0 and was developed in 1997. The updated features included in it are:

- Upgraded Form Elements: HTML 3.2 brought better ways to create interactive forms on websites. Developers could make forms that were more interactive and dynamic for users.
- CSS Support: Another important feature included in HTML 3.2 was support for CSS (Cascading Style Sheets). It helped designers improve the look of web pages by styling and customizing HTML elements.
- Enhanced Image Features: Handling images became easier with HTML 3.2. It allowed for better control over image size, alignment, and text descriptions.
- Extended Character Set: HTML 3.2 also expanded the available characters for web pages. It included special symbols and international characters for a more diverse presentation of the content.

2.1.4 HTML 4.01

HTML 4.01, released in 1999, brought several advancements to the HTML language. Here are some of the updated features:

- CSS Linking: Previously, one had to place CSS on each page to apply the styles. However, with 4.01, CSS files could be linked and included in each HTML page using the <link> tag. This helped maintain consistent styles across web pages without repeating CSS code.
- New Tags: HTML 4.01 also introduced some new tags like “<fieldset>”, “<header>”, “<footer>”, and “<legend>”. These tags consequently enhanced the presentability of the content.
- Table Enhancement: In addition, HTML 4.01 made tables more powerful. We could use attributes like ‘colspan’ and ‘rowspan’ to make cells in a table span across multiple columns or rows. This made it easier to create more complicated and interesting tables.

2.1.5 XHTML 1.0

XHTML 1.0 stands for Extensible HyperText Markup Language 1.0 and was released in 2000.

Strict Standards and Compatibility: XHTML 1.0 is similar to HTML but has a stricter version with more stringent rules for elements, attributes, and syntax. Due to such strict criteria, a common standard was created for web pages. This reduced the scope for incompatibility between browsers.

Compatibility with XML: XHTML 1.0 offered compatibility with XML tools. It meant that XML parsing libraries and transformation tools, commonly used for working with XML documents, could also be utilized with XHTML documents.

Future Adaptability: Furthermore, XHTML 1.0 documents were easily adaptable to any future versions of HTML or XML without any significant changes.

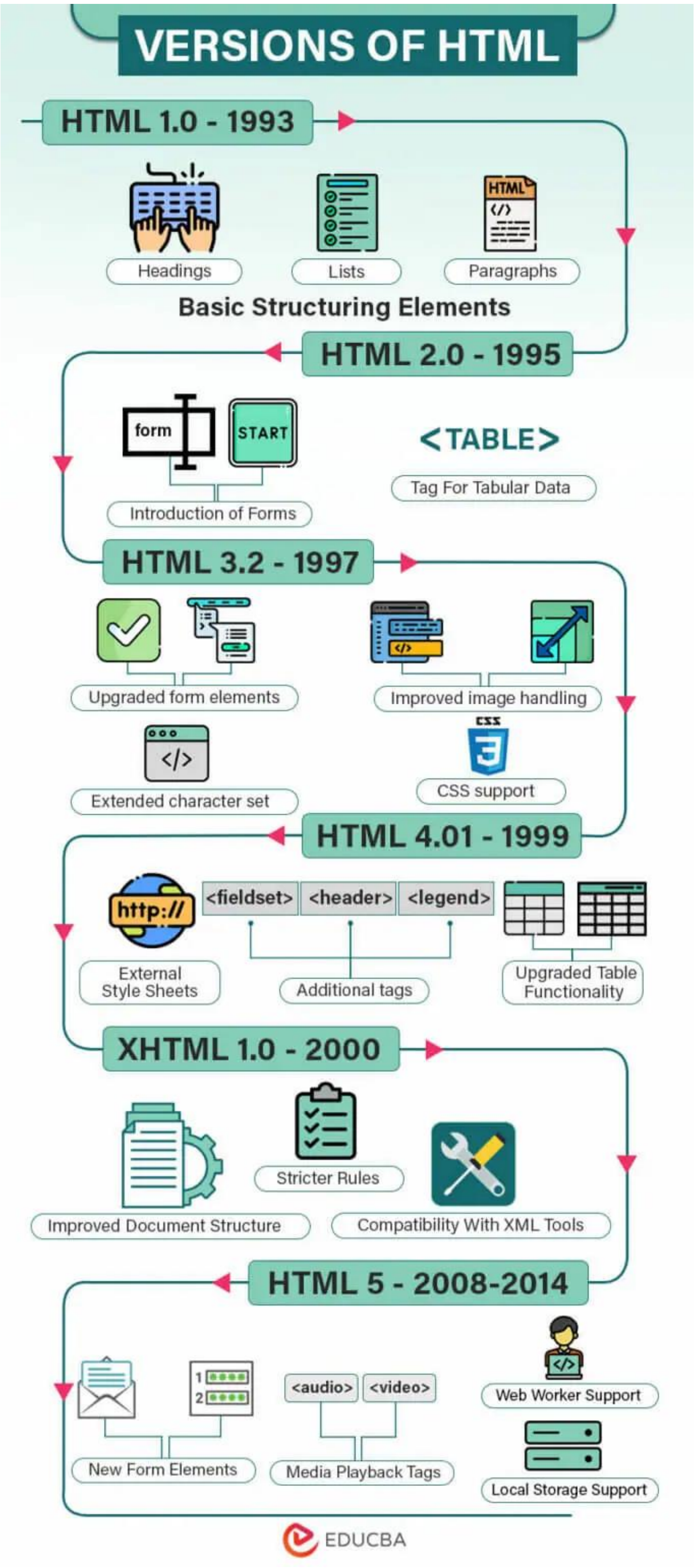
2.1.6 HTML5

WHATWG released the initial public draft of HTML5 in 2008, but it officially became a W3C recommendation on October 28, 2014. The version brought extensive support for new HTML tags. Furthermore, HTML5 provided support for new form elements like input elements of different types and geolocations support tags, etc.

Here are some key features and tags added in HTML5:

- New Form Elements: One important addition was the <input type="email"> tag, which confirms whether the user input is a valid email address. Likewise, another form element was the <input type="password"> tag, which was designed to capture passwords securely. The browser displayed special symbols as user input in the password field, thereby protecting the password from being revealed.
- Audio Tag: HTML5 introduced the <audio> tag, allowing developers to embed audio content directly into web pages. This tag enabled the seamless integration of audio clips and allowed playing audio directly on the webpage.
- Semantic Tags: Semantic tags, also known as structural tags, provide organization and structure to HTML pages. These tags provided a clearer hierarchy and meaning to different sections of a webpage. A few of the semantic tags introduced in HTML5 include <figcaption>, <header>, <footer>, etc. These tags also helped enhance the accessibility and search engine optimization of the webpage.
- Section Tag: The <section> tag defines a distinct section within an HTML page. This helps in organizing and delineating content, thereby providing logical divisions within the webpage.

2.2 HTML Timeline

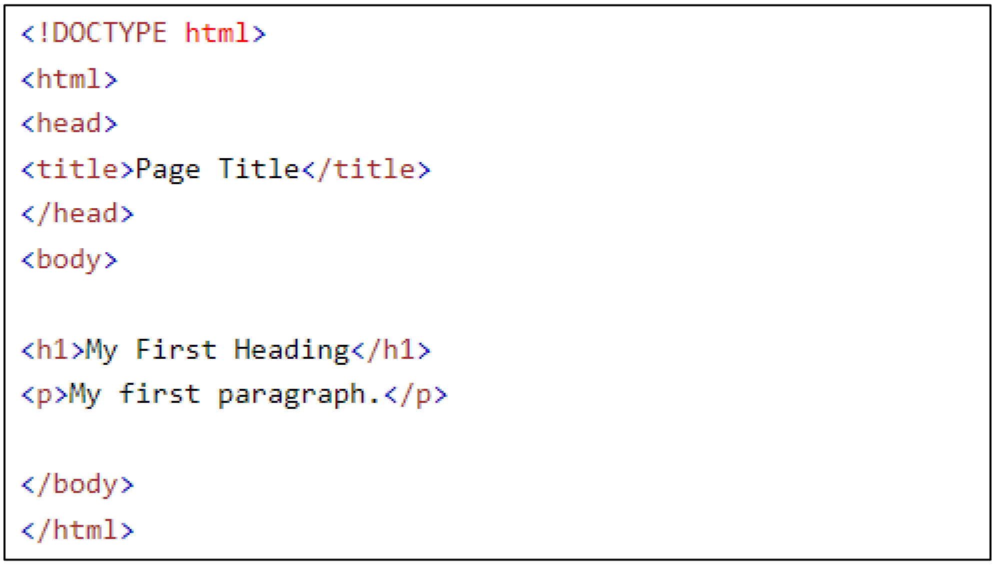


2.3 Introduction to HTML & Tags

HTML is the standard markup language for creating Web pages.

- HTML stands for Hyper Text Markup Language
- HTML is the standard markup language for creating Web pages
- HTML describes the structure of a Web page
- HTML consists of a series of elements
- HTML elements tell the browser how to display the content
- HTML elements label pieces of content such as "this is a heading", "this is a paragraph", "this is a link", etc.

Example Code:



```html
<!DOCTYPE html>
<html>
<head>
<title>Page Title</title>
</head>
<body>

<h1>My First Heading</h1>
<p>My first paragraph.</p>

</body>
</html>
```

Example Explained

- The <!DOCTYPE html> declaration defines that this document is an HTML5 document
- The <html> element is the root element of an HTML page
- The <head> element contains meta information about the HTML page
- The <title> element specifies a title for the HTML page (which is shown in the browser's title bar or in the page's tab)
- The <body> element defines the document's body, and is a container for all the visible contents, such as headings, paragraphs, images, hyperlinks, tables, lists, etc.
- The <h1> element defines a large heading
- The <p> element defines a paragraph

2.3.1 What is an HTML Element?

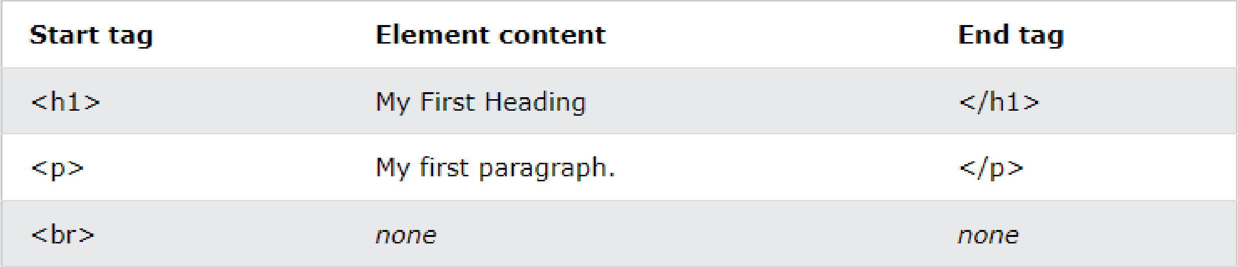
An HTML element is defined by a start tag, some content, and an end tag:

<tagname> Content goes here... </tagname>

The HTML **element** is everything from the start tag to the end tag:

<h1>My First Heading</h1>

<p>My first paragraph.</p>



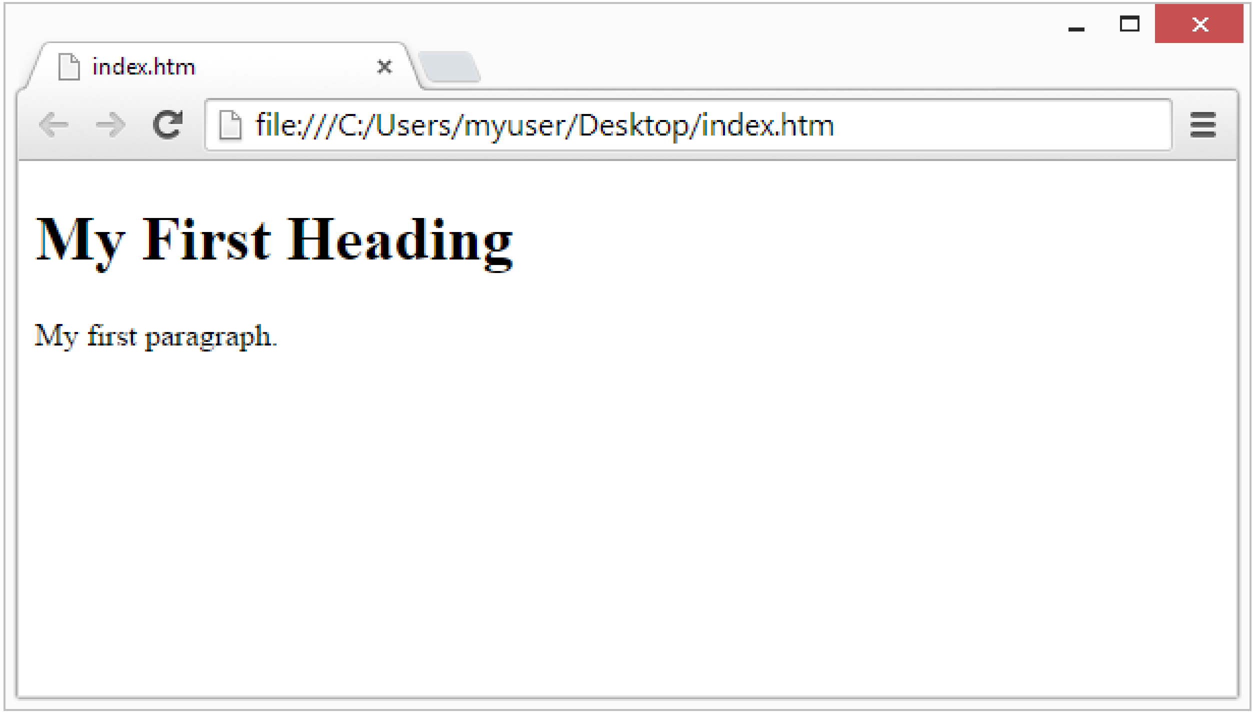
Start tag	Element content	End tag
<h1>	My First Heading	</h1>
<p>	My first paragraph.	</p>
<br>	*none*	*none*

Note: Some HTML elements have no content (like the <br> element). These elements are called empty elements. Empty elements do not have an end tag!

Web Browsers

The purpose of a web browser (Chrome, Edge, Firefox, Safari) is to read HTML documents and display them correctly.

A browser does not display the HTML tags, but uses them to determine how to display the document:

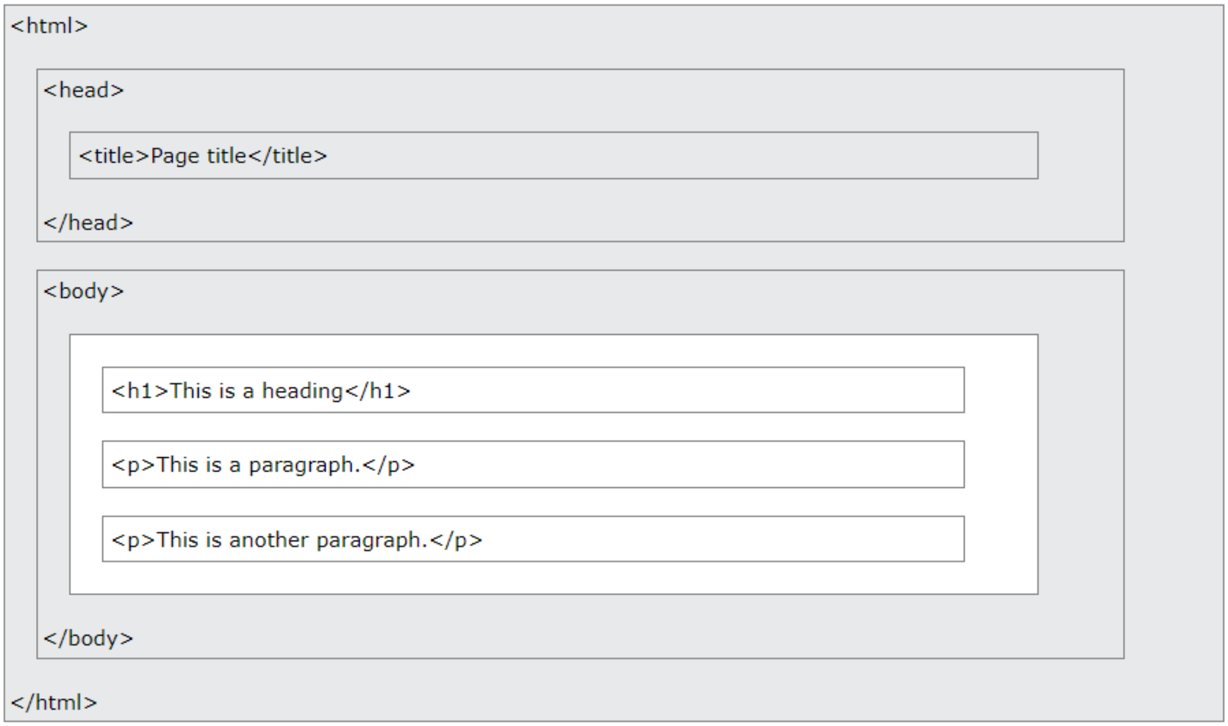


My First Heading

My first paragraph.

HTML Page Structure

Below is a visualization of an HTML page structure:

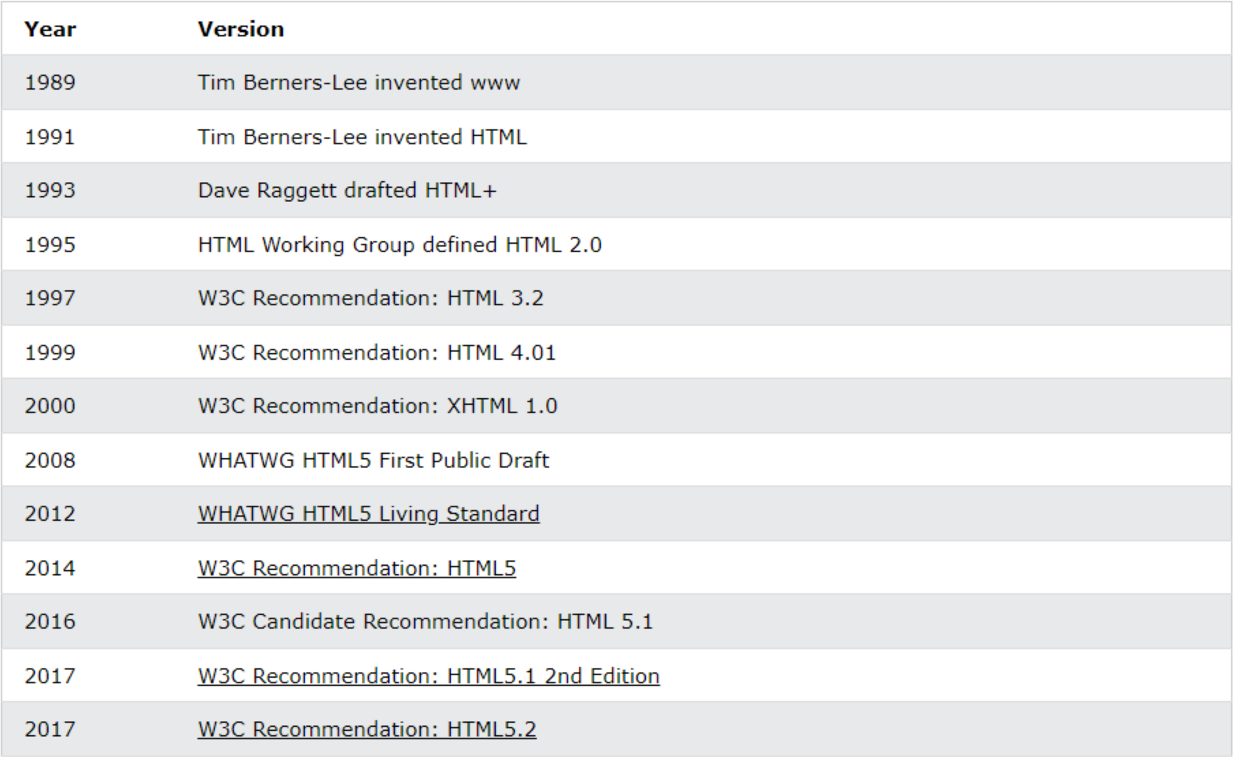


```
<html>
  <head>
    <title>Page title</title>
  </head>
  <body>
      <h1>This is a heading</h1>
      <p>This is a paragraph.</p>
      <p>This is another paragraph.</p>
  </body>
</html>
```

Note: The content inside the <body> section will be displayed in a browser. The content inside the <title> element will be shown in the browser's title bar or in the page's tab.

HTML History

Since the early days of the World Wide Web, there have been many versions of HTML:



Year	Version
1989	Tim Berners-Lee invented www
1991	Tim Berners-Lee invented HTML
1993	Dave Raggett drafted HTML+
1995	HTML Working Group defined HTML 2.0
1997	W3C Recommendation: HTML 3.2
1999	W3C Recommendation: HTML 4.01
2000	W3C Recommendation: XHTML 1.0
2008	WHATWG HTML5 First Public Draft
2012	WHATWG HTML5 Living Standard
2014	W3C Recommendation: HTML5
2016	W3C Candidate Recommendation: HTML 5.1
2017	W3C Recommendation: HTML5.1 2nd Edition
2017	W3C Recommendation: HTML5.2

2.3.2 HTML Editors

Web pages can be created and modified by using professional HTML editors.

However, for learning HTML we recommend a simple text editor like Notepad (PC) or TextEdit (Mac).

We believe that using a simple text editor is a good way to learn HTML.

Follow the steps below to create your first web page with Notepad or TextEdit.

Step 1: Open Notepad (PC)

Windows 8 or later:

- Open the **Start Screen** (the window symbol at the bottom left on your screen). Type **Notepad**.

Windows 7 or earlier:

- Open **Start** > **Programs > Accessories > Notepad**

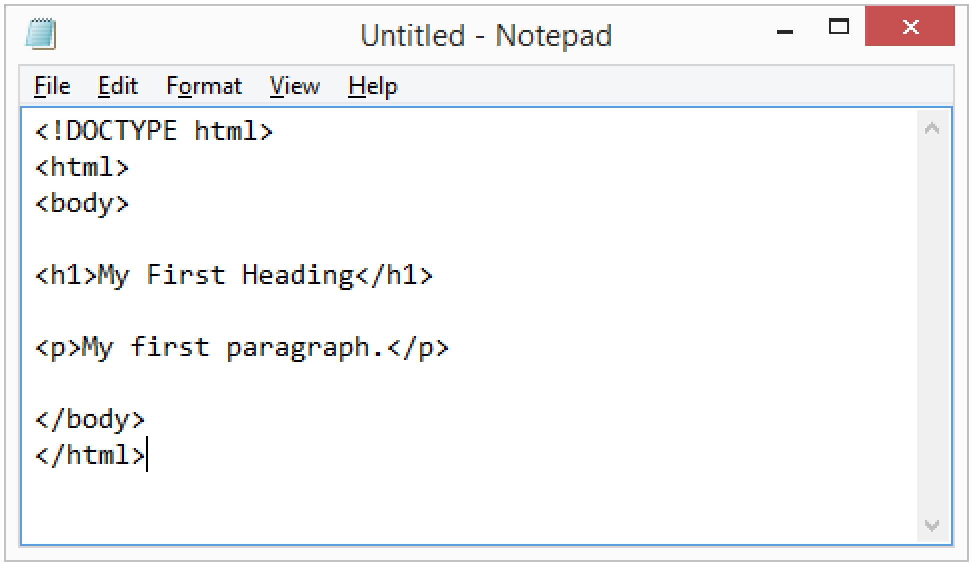
Step 2: Write Some HTML

Write or copy the following HTML code into Notepad:

```
<!DOCTYPE html>
<html>
```

```
<body>

<h1>My First Heading</h1>

<p>My first paragraph.</p>

</body>
</html>
```

Something like this

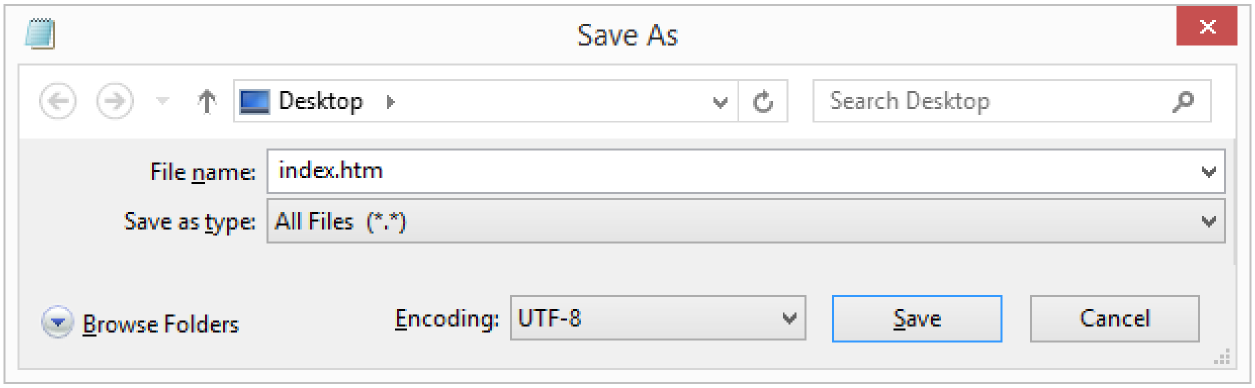


```
<!DOCTYPE html>
<html>
<body>

<h1>My First Heading</h1>

<p>My first paragraph.</p>

</body>
</html>
```

Step 3: Save the HTML Page

Save the file on your computer. Select File > Save as in the Notepad menu.

Name the file "index.htm" and set the encoding to UTF-8 (which is the preferred encoding for HTML files).



File name: index.htm

Save as type: All Files (*.*)

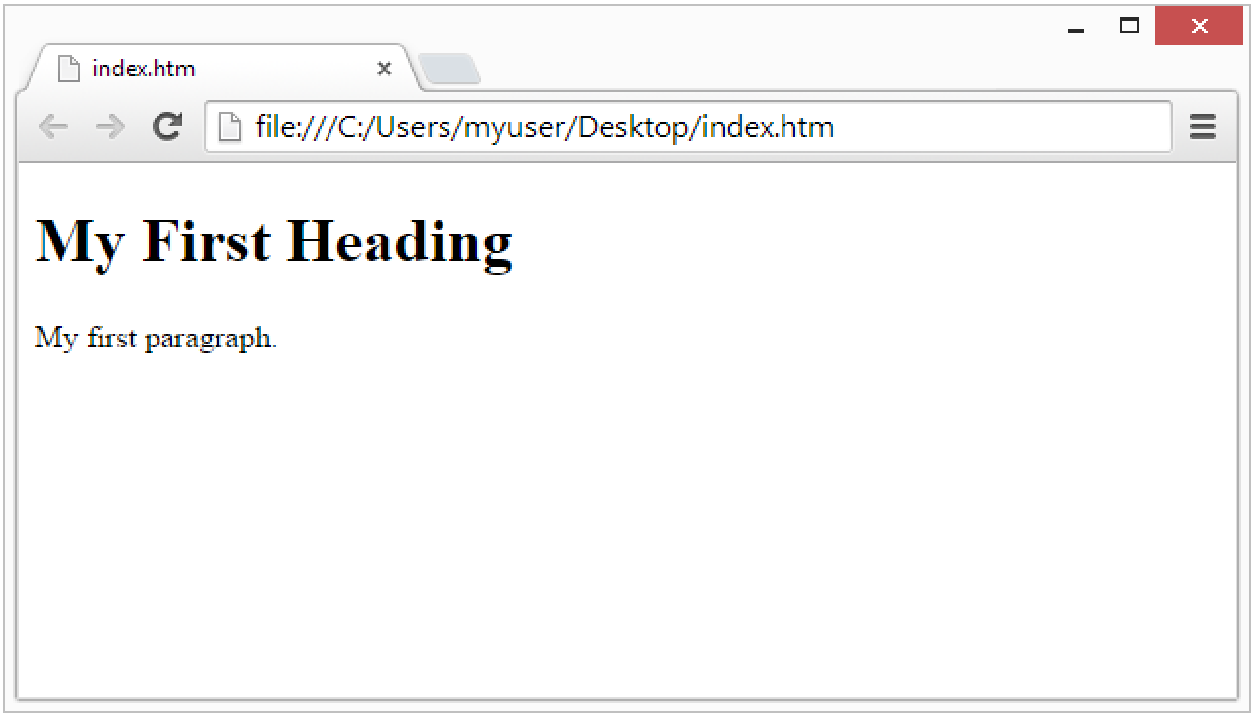
Encoding: UTF-8

Tip: You can use either .htm or .html as file extension. There is no difference; it is up to you.

STEP 4: VIEW THE HTML PAGE IN YOUR BROWSER

Open the saved HTML file in your favorite browser (double click on the file, or right-click - and choose "Open with").

The result will look much like this:



My First Heading

My first paragraph.

2.4 HTML Basics

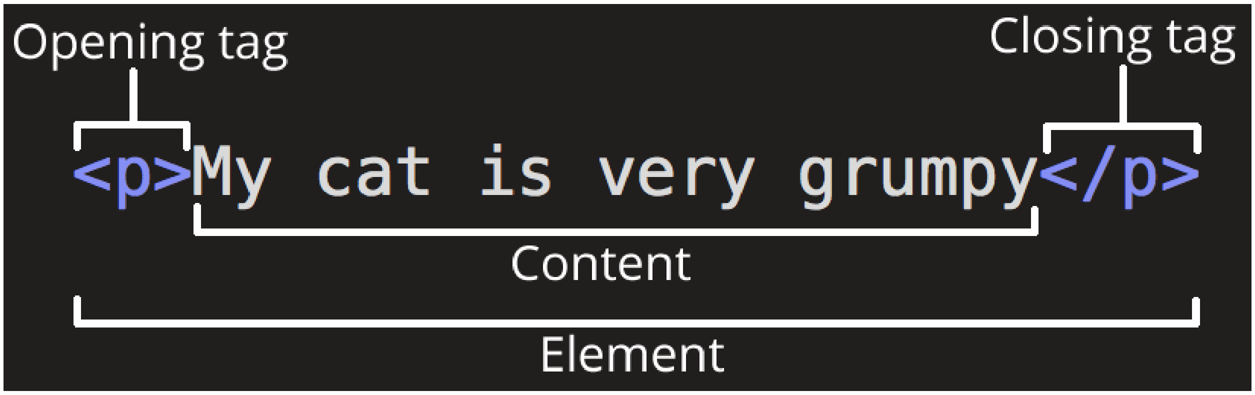
HTML (**H**yper**T**ext **M**arkup **L**anguage) is the code that is used to structure a web page and its content. For example, content could be structured within a set of paragraphs, a list of bulleted points, or using images and data tables. As the title suggests, this article will give you a basic understanding of HTML and its functions.

2.4.1 Anatomy of an HTML element

HTML is made up of a series of hierarchical elements. Most elements have an opening tag and a closing tag, with some content sandwiched in between. If you're used to marking up chapters, headings and subheadings, this concept should be easy to grasp.

If this sounds alien to you, just think of HTML as a tree with the elements as its branches.

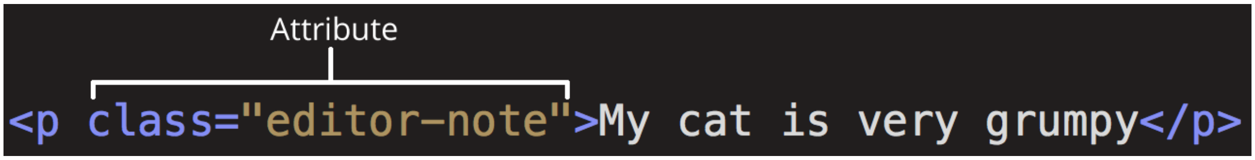
Every tag has its associated closing tag except some special tags which don't have any closing tags.



```
<p>My cat is very grumpy</p>
```

1. **The opening tag:** This consists of the name of the element (in this case, p), wrapped in opening and closing **angle brackets**. This states where the element begins or starts to take effect — in this case where the paragraph begins.
2. **The closing tag:** This is the same as the opening tag, except that it includes a *forward slash* before the element name. This states where the element ends — in this case where the paragraph ends. Failing to add a closing tag is one of the standard beginner errors and can lead to strange results.
3. **The content:** This is the content of the element, which in this case, is just text.
4. **The element:** The opening tag, the closing tag, and the content together comprise the element.

Elements can also have attributes that look like the following:



```
<p class="editor-note">My cat is very grumpy</p>
```

Attributes contain extra information about the element that you don't want to appear in the actual content. Here, class is the attribute name and **editor-note** is the attribute value. The class attribute allows you to give the element a non-unique identifier that can be used to target it (and any other elements with the same class value) with style information and other things. Some attributes have no value, such as **required**.

Attributes that set a value always have:

1. A space between it and the element name (or the previous attribute, if the element already has one or more attributes).
2. The attribute name followed by an equal sign.
3. The attribute value wrapped by opening and closing quotation marks.

Note: Simple attribute values that don't contain ASCII whitespace (or any of the characters " ' ` = < >) can remain unquoted, but it is recommended that you quote all attribute values, as it makes the code more consistent and understandable.

Nesting elements

You can put elements inside other elements too — this is called **nesting**. If we wanted to state that our cat is **very** grumpy, we could wrap the word "very" in a <strong> element, which means that the word is to be strongly emphasized:

```
<p>My cat is <strong>very</strong> grumpy.</p>
```

You do however need to make sure that your elements are properly nested. In the example above, we opened the <p> element first, then the <strong> element; therefore, we have to close the <strong> element first, then the <p> element. The following is incorrect:

```
<p>My cat is <strong>very grumpy.</p></strong>
```

The elements have to open and close correctly so that they are clearly inside or outside one another. If they overlap as shown above, then your web browser will try to make the best guess at what you were trying to say, which can lead to unexpected results. So don't do it!

2.4.2 Void elements

Some elements have no content and are called **void elements**. Take the <img> element that we already have in our HTML page:

```
<img src="images/firefox-icon.png" alt="My test image" />
```

This contains two attributes, but there is no closing </img> tag and no inner content. This is because an image element doesn't wrap content to affect it. Its purpose is to embed an image in the HTML page in the place it appears.

Anatomy of an HTML document

That wraps up the basics of individual HTML elements, but they aren't handy on their own. Now we'll look at how individual elements are combined to form an entire HTML page.

```
<!DOCTYPE html>
```

```
<html lang="en-US">
  <head>
    <meta charset="utf-8" />
    <meta name="viewport" content="width=device-width" />
    <title>My test page</title>
  </head>
  <body>
    <img src="images/firefox-icon.png" alt="My test image" />
  </body>
</html>
```

- Here, we have the following: <!DOCTYPE html> — doctype. It is a required preamble. In the mists of time, when HTML was young (around 1991/92), doctypes were meant to act as links to a set of rules that the HTML page had to follow to be considered good HTML, which could mean automatic error checking and other useful things. However, these days, they don't do much and are basically just needed to make sure your document behaves correctly. That's all you need to know for now.
- <html></html> — the <html> element. This element wraps all the content on the entire page and is sometimes known as the root element. It also includes the lang attribute, setting the primary language of the document.
- <head></head> — the <head> element. This element acts as a container for all the stuff you want to include on the HTML page that *isn't* the content you are showing to your page's viewers. This includes things like keywords and a page description that you want to appear in search results, CSS to style our content, character set declarations, and more.
- <meta charset="utf-8"> — This element sets the character set your document should use to UTF-8 which includes most characters from the vast majority of written languages. Essentially, it can now handle any textual content you might put on it. There is no reason not to set this, and it can help avoid some problems later on.
- <meta name="viewport" content="width=device-width"> — This viewport element ensures the page renders at the width of viewport, preventing mobile browsers from rendering pages wider than the viewport and then shrinking them down.
- <title></title> — the <title> element. This sets the title of your page, which is the title that appears in the browser tab the page is loaded in. It is also used to describe the page when you bookmark/favorite it.
- <body></body> — the <body> element. This contains *all* the content that you want to show to web users when they visit your page, whether that's text, images, videos, games, playable audio tracks, or whatever else.

2.4.3 HTML Elements Tags

The <!DOCTYPE> Declaration

- The <!DOCTYPE> declaration represents the document type, and helps browsers to display web pages correctly.
- It must only appear once, at the top of the page (before any HTML tags).
- The <!DOCTYPE> declaration is not case sensitive.
- The <!DOCTYPE> declaration for HTML5 is:

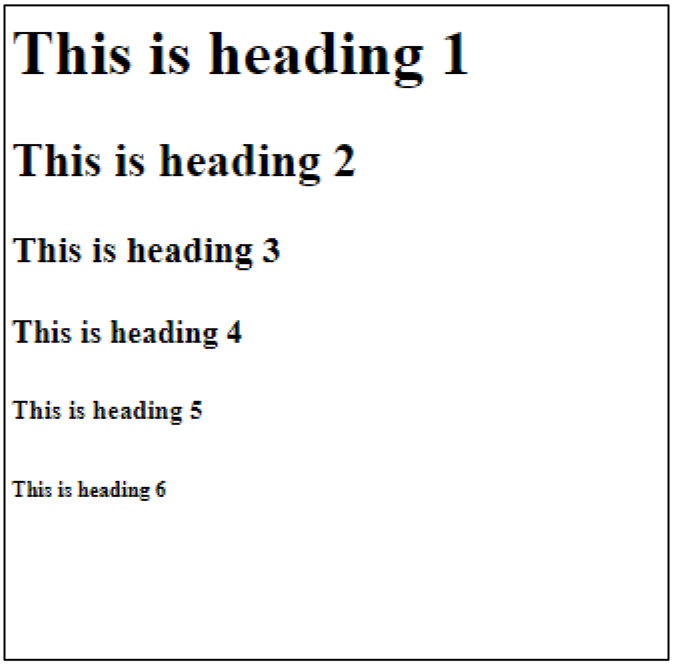
```
<!DOCTYPE html>
```

HTML Headings

- HTML headings are defined with the <h1> to <h6> tags.
- <h1> defines the most important heading. <h6> defines the least important heading:

```
<h1>This is heading 1</h1>
<h2>This is heading 2</h2>
<h3>This is heading 3</h3>
```

Output:



This is heading 1

This is heading 2

This is heading 3

This is heading 4

This is heading 5

This is heading 6

Headings Are Important

- Search engines use the headings to index the structure and content of your web pages.
- Users often skim a page by its headings. It is important to use headings to show the document structure.
- <h1> headings should be used for main headings, followed by <h2> headings, then the less important <h3>, and so on.

HTML Paragraphs

As explained above, <p> elements are for containing paragraphs of text; you'll use these frequently when marking up regular text content.

```
<p>This is a paragraph.</p>
<p>This is another paragraph.</p>
```

HTML Horizontal Rules

- The <hr> tag defines a thematic break in an HTML page, and is most often displayed as a horizontal rule.
- The <hr> element is used to separate content (or define a change) in an HTML page:

```
<h1>This is heading 1</h1>
<p>This is some text.</p>
<hr>
<h2>This is heading 2</h2>
<p>This is some other text.</p>
<hr>
```

The <hr> tag is an empty tag, which means that it has no end tag.

HTML Line Breaks

- The HTML <br> element defines a line break.
- Use <br> if you want a line break (a new line) without starting a new paragraph:

```
<p>This is<br>a paragraph<br>with line breaks.</p>
```

HTML Styles

- The HTML style attribute is used to add styles to an element, such as color, font, size, and more.
- Setting the style of an HTML element, can be done with the style attribute.
- The HTML style attribute has the following syntax:

```
   <tagname style="property:value;">
```

The ***property*** is a CSS property. The ***value*** is a CSS value.

HTML Text Formatting:

HTML contains several elements for defining text with a special meaning.

Formatting elements were designed to display special types of text:

- <b> - Bold text
- <strong> - Important text
- <i> - Italic text
- <em> - Emphasized text
- <mark> - Marked text
- <small> - Smaller text
- <del> - Deleted text
- <ins> - Inserted text
- <sub> - Subscript text

- <sup> - Superscript text

HTML <b> and <strong> Elements

The HTML <b> element defines bold text, without any extra importance.

```
<b>This text is bold</b>
```

The HTML <strong> element defines text with strong importance. The content inside is typically displayed in bold.

```
<strong>This text is important!</strong>
```

HTML <i> and <em> Elements

The HTML <i> element defines a part of text in an alternate voice or mood. The content inside is typically displayed in italic.

Tip: The <i> tag is often used to indicate a technical term, a phrase from another language, a thought, a ship name, etc.

```
<i>This text is italic</i>
```

The HTML <em> element defines emphasized text. The content inside is typically displayed in italic.

Tip: A screen reader will pronounce the words in <em> with an emphasis, using verbal stress.

```
<em>This text is emphasized</em>
```

HTML <small> Element

The HTML <small> element defines smaller text:

```
<small>This is some smaller text.</small>
```

HTML <mark> Element

The HTML <mark> element defines text that should be marked or highlighted:

```
<p>Do not forget to buy <mark>milk</mark> today.</p>
```

HTML <sub> Element

The HTML <sub> element defines subscript text. Subscript text appears half a character below the normal line, and is sometimes rendered in a smaller font. Subscript text can be used for chemical formulas, like H_2O:

```
<p>This is <sub>subscripted</sub> text.</p>
```

HTML <sup> Element

The HTML <sup> element defines superscript text. Superscript text appears half a character above the normal line, and is sometimes rendered in a smaller font. Superscript text can be used for footnotes, like WWW[1]:

```
<p>This is <sup>superscripted</sup> text.</p>
```

HTML <blockquote> for Quotations

The HTML <blockquote> element defines a section that is quoted from another source.
Browsers usually indent <blockquote> elements.

```
<p>Here is a quote from WWF's website:</p>
<blockquote cite="http://www.worldwildlife.org/who/index.html">
For 60 years, WWF has worked to help people and nature thrive. As the
world's leading conservation organization, WWF works in nearly 100
countries. At every level, we collaborate with people around the world
to develop and deliver innovative solutions that protect communities,
wildlife, and the places in which they live.
</blockquote>
```

HTML <abbr> for Abbreviations

The HTML <abbr> tag defines an abbreviation or an acronym, like "HTML", "CSS", "Mr.", "Dr.", "ASAP", "ATM".

Marking abbreviations can give useful information to browsers, translation systems and search-engines.

Tip: Use the global title attribute to show the description for the abbreviation/acronym when you mouse over the element**.**

```
<p>The <abbr title="World Health Organization">WHO</abbr> was founded in
1948.</p>
```

HTML <address> for Contact Information

The HTML <address> tag defines the contact information for the author/owner of a document or an article.

The contact information can be an email address, URL, physical address, phone number, social media handle, etc.

The text in the <address> element usually renders in *italic,* and browsers will always add a line break before and after the <address> element.

```
<address>
Written by John Doe.<br>
Visit us at:<br>
Example.com<br>
Box 564, Disneyland<br>
USA
</address>
```

HTML <cite> for Work Title

The HTML <cite> tag defines the title of a creative work (e.g. a book, a poem, a song, a movie, a painting, a sculpture, etc.).

Note: A person's name is not the title of a work.

The text in the <cite> element usually renders in *italic*.

```
<p><cite>The Scream</cite> by Edvard Munch. Painted in 1893.</p>
```

HTML Links

- HTML links are hyperlinks.
- You can click on a link and jump to another document.
- When you move the mouse over a link, the mouse arrow will turn into a little hand.
- The HTML <a> tag defines a hyperlink. It has the following syntax:

```
<a href="url">link text</a>
```

- The most important attribute of the <a> element is the href attribute, which indicates the link's destination.
- The *link text* is the part that will be visible to the reader.
- Clicking on the link text, will send the reader to the specified URL address.

```
<a href="https://www.iba-suk.edu.pk/">Visit IBA Sukkur!</a>
```

By default, links will appear as follows in all browsers:

- An unvisited link is underlined and blue

- A visited link is underlined and purple
- An active link is underlined and red

HTML Links - The target Attribute

By default, the linked page will be displayed in the current browser window. To change this, you must specify another target for the link.

The target attribute specifies where to open the linked document.

The target attribute can have one of the following values:

- _self - Default. Opens the document in the same window/tab as it was clicked
- _blank - Opens the document in a new window or tab
- _parent - Opens the document in the parent frame
- _top - Opens the document in the full body of the window

```
    <a href="     https://www.iba-suk.edu.pk/">target="_blank">Visit     IB
    Sukkur!</a>
```

Absolute URLs vs. Relative URLs

Both examples above are using an **absolute URL** (a full web address) in the href attribute.

A local link (a link to a page within the same website) is specified with a **relative URL** (without the "https://www" part):

```
<h2>Absolute URLs</h2>
<p><a href="https://www.w3.org/">W3C</a></p>
<p><a href="https://www.google.com/">Google</a></p>

<h2>Relative URLs</h2>
<p><a href="html_images.asp">HTML Images</a></p>
<p><a href="/css/default.asp">CSS Tutorial</a></p>
```

HTML Links - Use an Image as a Link

To use an image as a link, just put the <img> tag inside the <a> tag:

```
<a href="default.asp">
<img src="smiley.gif" alt="HTML
```

```
tutorial" style="width:42px;height:42px;">
</a>
```

Button as a Link

To use an HTML button as a link, you have to add some JavaScript code.

JavaScript allows you to specify what happens at certain events, such as a click of a button:

```
<button onclick="document.location='default.asp'">HTML Tutorial</button>
```

Link Titles

The title attribute specifies extra information about an element. The information is most often shown as a tooltip text when the mouse moves over the element.

```
<a href="https://www.iba-suk.edu.pk" title="Go to Sukkur IBA Portal">Visit
our Sukkur IBA Portal</a>
```

HTML Images

Images can improve the design and the appearance of a web page.

The HTML <img> tag is used to embed an image in a web page.

Images are not technically inserted into a web page; images are linked to web pages. The <img> tag creates a holding space for the referenced image.

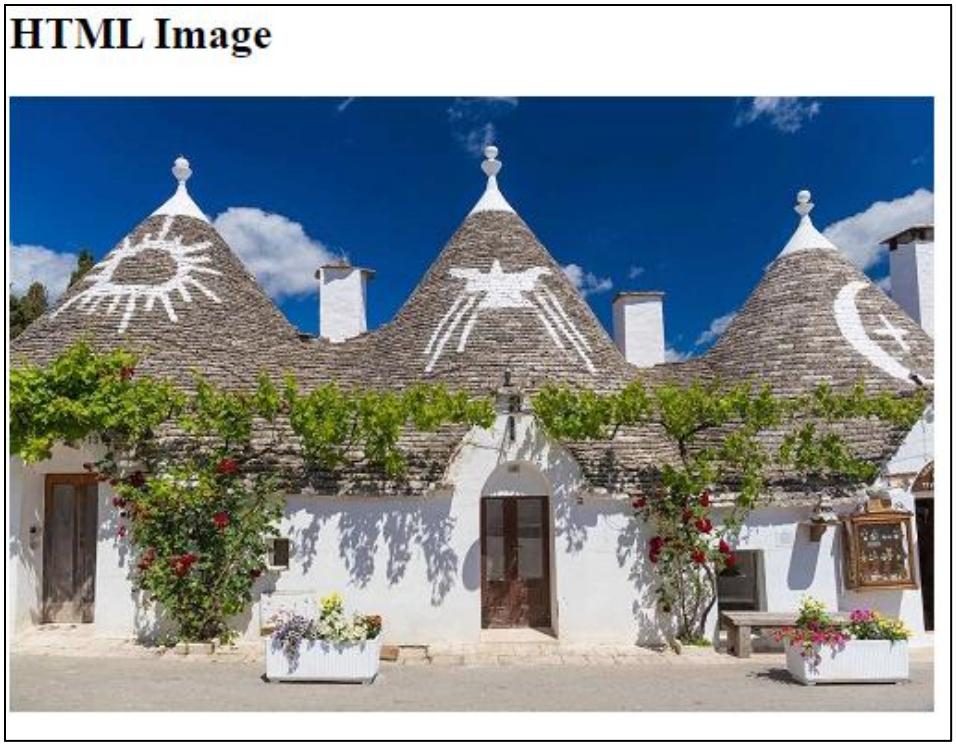
The <img> tag is empty, it contains attributes only, and does not have a closing tag.

The <img> tag has two required attributes:

- src - Specifies the path to the image
- alt - Specifies an alternate text for the image

```
<img src="pic_trulli.jpg" alt="Italian Trulli">
```

Output



HTML Image

The src Attribute

The required src attribute specifies the path (URL) to the image.

Note: When a web page loads, it is the browser, at that moment, that gets the image from a web server and inserts it into the page. Therefore, make sure that the image actually stays in the same spot in relation to the web page, otherwise your visitors will get a broken link icon. The broken link icon and the alt text are shown if the browser cannot find the image.

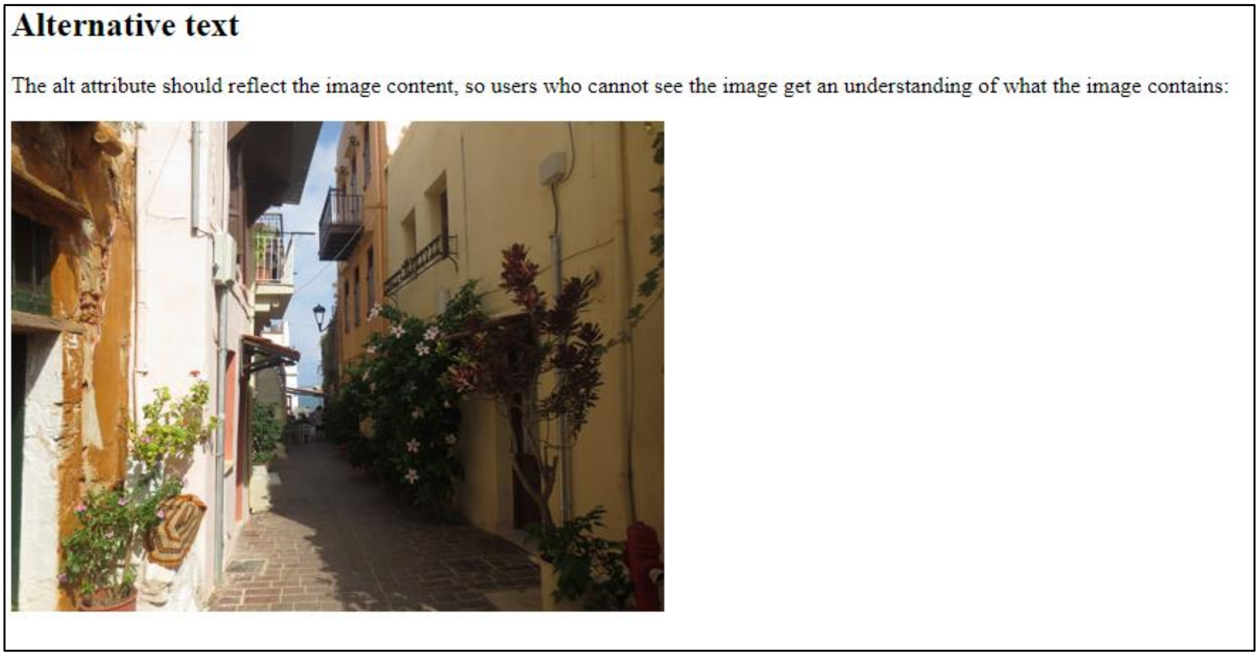
The alt Attribute

The required alt attribute provides an alternate text for an image, if the user for some reason cannot view it (because of slow connection, an error in the src attribute, or if the user uses a screen reader).

The value of the alt attribute should describe the image:

```
<img src="img_chania.jpg" alt="Flowers in Chania">
```

Output



Alternative text

The alt attribute should reflect the image content, so users who cannot see the image get an understanding of what the image contains:

Image Size - Width and Height

```
<img src="img_girl.jpg" alt="Girl in a jacket" width="500" height="600">
```

Output

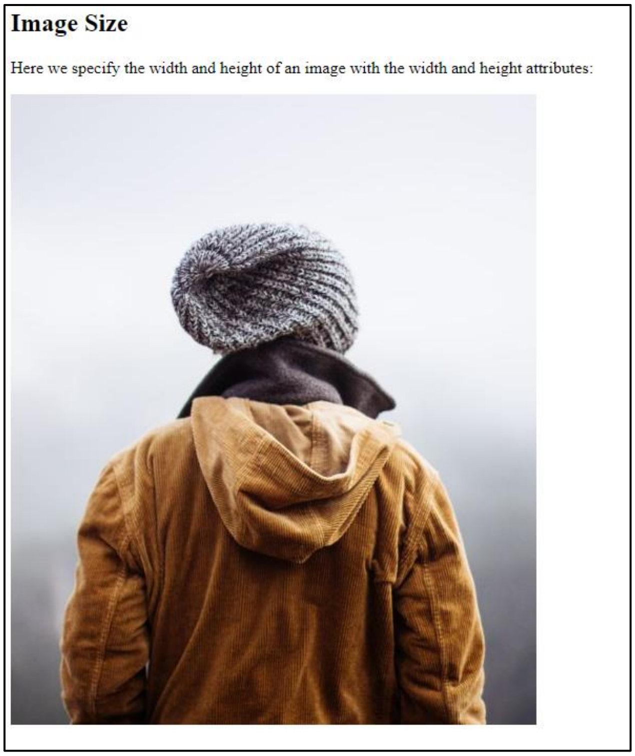
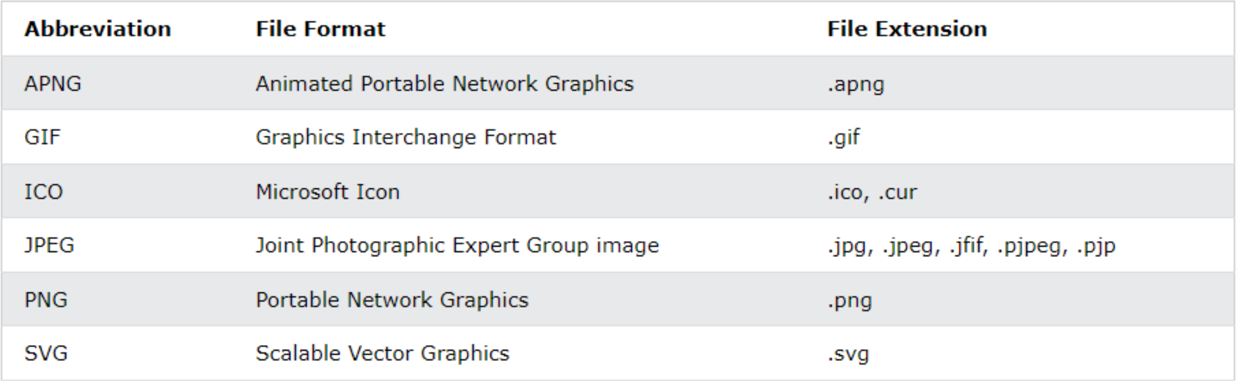


Image Size

Here we specify the width and height of an image with the width and height attributes:

Common Image Formats

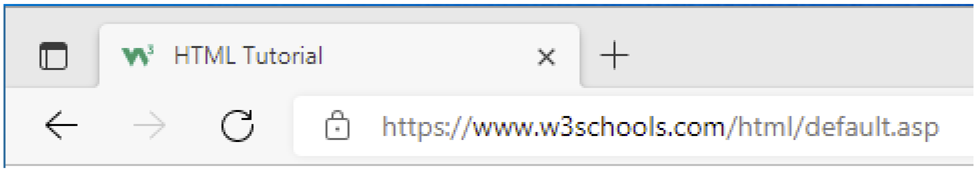
Here are the most common image file types, which are supported in all browsers (Chrome, Edge, Firefox, Safari, Opera):



Abbreviation	File Format	File Extension
APNG	Animated Portable Network Graphics	.apng
GIF	Graphics Interchange Format	.gif
ICO	Microsoft Icon	.ico, .cur
JPEG	Joint Photographic Expert Group image	.jpg, .jpeg, .jfif, .pjpeg, .pjp
PNG	Portable Network Graphics	.png
SVG	Scalable Vector Graphics	.svg

HTML Favicon

- A favicon is a small image displayed next to the page title in the browser tab.
- You can use any image you like as your favicon. You can also create your own favicon on sites like https://www.favicon.cc.



To add a favicon to your website, either save your favicon image to the root directory of your webserver, or create a folder in the root directory called images, and save your favicon image in this folder. A common name for a favicon image is "favicon.ico".

Next, add a <link> element to your "index.html" file, after the <title> element, like this:

```
<!DOCTYPE html>
<html>
<head>
  <title>My Page Title</title>
  <link rel="icon" type="image/x-icon" href="/images/favicon.ico">
</head>
<body>

<h1>This is a Heading</h1>
<p>This is a paragraph.</p>
```

```
</body>
</html>
```

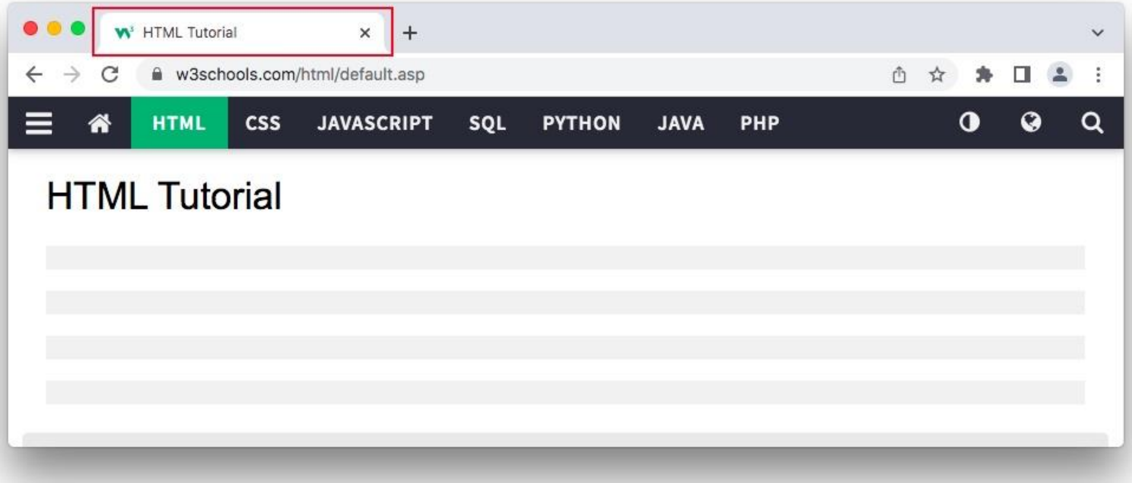
HTML Page Title

Every web page should have a page title to describe the meaning of the page.

The `<title>` element adds a title to your page:

```
<!DOCTYPE html>
<html>
<head>
  <title>HTML Tutorial</title>
</head>
<body>

The content of the document......

</body>
</html>
```

Output



The title should describe the content and the meaning of the page.

The page title is very important for search engine optimization (SEO). The text is used by search engine algorithms to decide the order when listing pages in search results.

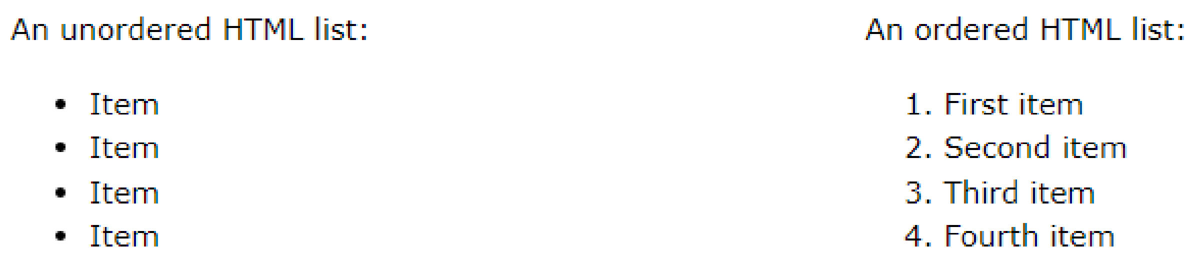
The <title> element:

- defines a title in the browser toolbar
- provides a title for the page when it is added to favorites
- displays a title for the page in search engine-results

So, try to make the title as accurate and meaningful as possible!

HTML Lists

HTML lists allow web developers to group a set of related items in lists.



An unordered HTML list:

- Item
- Item
- Item
- Item

An ordered HTML list:

1. First item
2. Second item
3. Third item
4. Fourth item

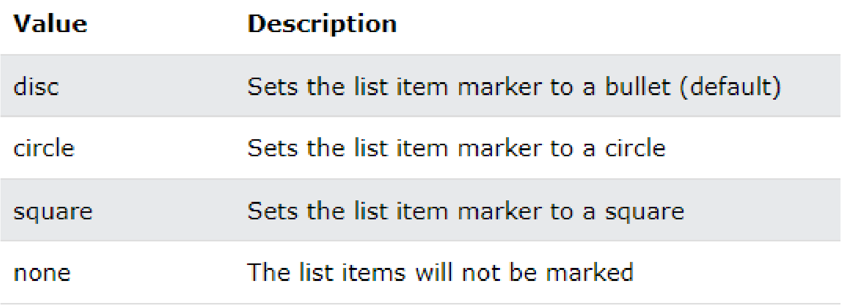
Unordered HTML List

An unordered list starts with the <ul> tag. Each list item starts with the <li> tag.

The list items will be marked with bullets (small black circles) by default:

```
<ul>
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ul>
```

The CSS list-style-type property is used to define the style of the list item marker. It can have one of the following values:



Value	Description
disc	Sets the list item marker to a bullet (default)
circle	Sets the list item marker to a circle
square	Sets the list item marker to a square
none	The list items will not be marked

```
<ul style="list-style-type:disc;">
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ul>
```

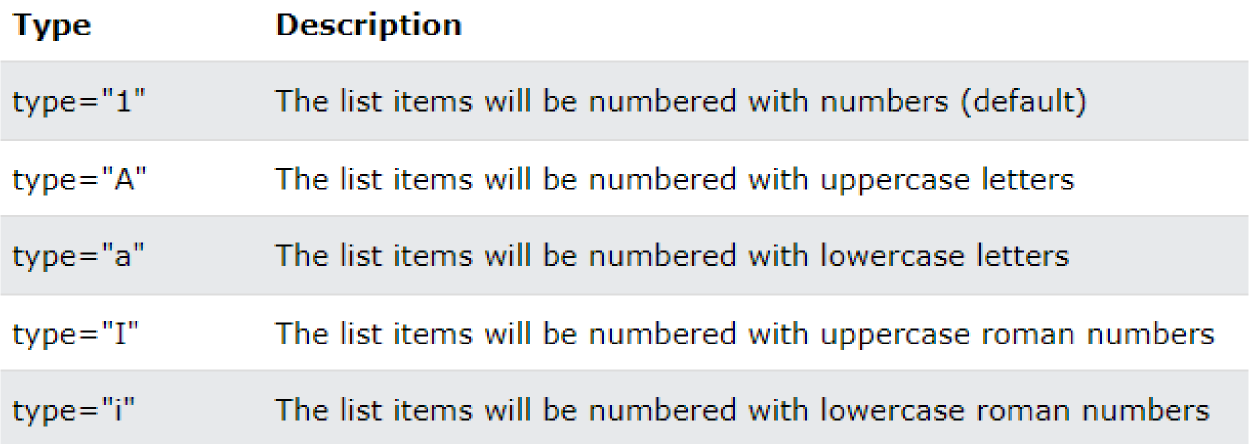
Ordered HTML List

An ordered list starts with the `<ol>` tag. Each list item starts with the `<li>` tag.

The list items will be marked with numbers by default:

```
<ol>
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ol>
```

The `type` attribute of the `<ol>` tag, defines the type of the list item marker:



Type	Description
type="1"	The list items will be numbered with numbers (default)
type="A"	The list items will be numbered with uppercase letters
type="a"	The list items will be numbered with lowercase letters
type="I"	The list items will be numbered with uppercase roman numbers
type="i"	The list items will be numbered with lowercase roman numbers

```
<ol type="A">
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ol>
```

HTML Description Lists

HTML also supports description lists.

A description list is a list of terms, with a description of each term.

The <dl> tag defines the description list, the <dt> tag defines the term (name), and the <dd> tag describes each term:

```
<dl>
  <dt>Coffee</dt>
  <dd>- black hot drink</dd>
  <dt>Milk</dt>
  <dd>- white cold drink</dd>
</dl>
```

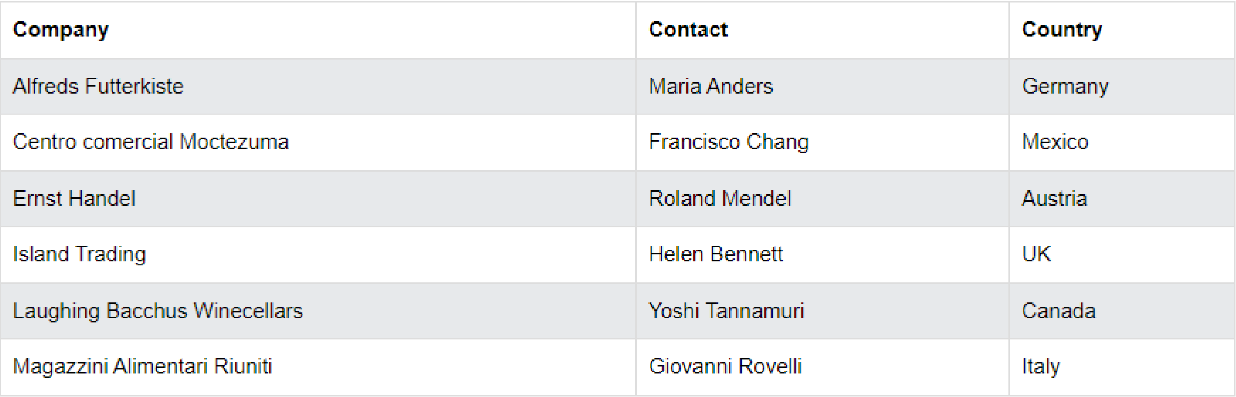
Nested HTML Lists

Lists can be nested (list inside list):

```
<ul>
  <li>Coffee</li>
  <li>Tea
    <ul>
      <li>Black tea</li>
      <li>Green tea</li>
    </ul>
  </li>
  <li>Milk</li>
</ul>
```

HTML Tables

HTML tables allow web developers to arrange data into rows and columns.



Company	Contact	Country
Alfreds Futterkiste	Maria Anders	Germany
Centro comercial Moctezuma	Francisco Chang	Mexico
Ernst Handel	Roland Mendel	Austria
Island Trading	Helen Bennett	UK
Laughing Bacchus Winecellars	Yoshi Tannamuri	Canada
Magazzini Alimentari Riuniti	Giovanni Rovelli	Italy

A table in HTML consists of table cells inside rows and columns.

```
<table>
  <tr>
    <th>Company</th>
    <th>Contact</th>
    <th>Country</th>
  </tr>
  <tr>
    <td>Alfreds Futterkiste</td>
    <td>Maria Anders</td>
    <td>Germany</td>
  </tr>
  <tr>
    <td>Centro comercial Moctezuma</td>
    <td>Francisco Chang</td>
    <td>Mexico</td>
  </tr>
</table>
```

Table Cells

Each table cell is defined by a `<td>` and a `</td>` tag.
Everything between `<td>` and `</td>` are the content of the table cell.

```
<table>
  <tr>
    <td>Emil</td>
    <td>Tobias</td>
    <td>Linus</td>
  </tr>
</table>
```

Table Rows

Each table row starts with a `<tr>` and ends with a `</tr>` tag.

```
<table>
  <tr>
    <td>Emil</td>
    <td>Tobias</td>
    <td>Linus</td>
  </tr>
  <tr>
    <td>16</td>
    <td>14</td>
    <td>10</td>
  </tr>
</table>
```

We can have as many rows as you like in a table; just make sure that the number of cells are the same in each row.

Table Headers

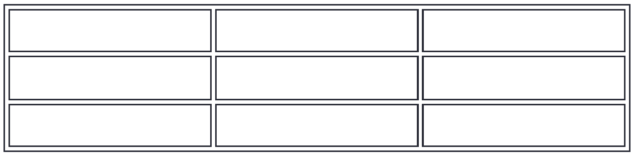
Sometimes you want your cells to be table header cells. In those cases use the `<th>` tag instead of the `<td>` tag:

```
<table>
  <tr>
    <th>Person 1</th>
    <th>Person 2</th>
    <th>Person 3</th>
  </tr>
  <tr>
    <td>Emil</td>
    <td>Tobias</td>
    <td>Linus</td>
  </tr>
  <tr>
    <td>16</td>
    <td>14</td>
    <td>10</td>
  </tr>
</table>
```

Table Borders

HTML tables can have borders of different styles and shapes.

When you add a border to a table, you also add borders around each table cell:



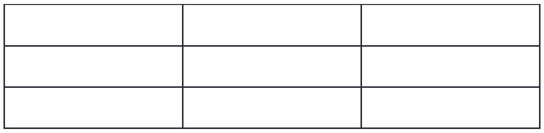
To add a border, use the CSS `border` property on `table`, `th`, and `td` elements:

```
table, th, td {
  border: 1px solid black;
}
```

Collapsed Table Borders

To avoid having double borders like in the example above, set the CSS `border-collapse` property to `collapse`.

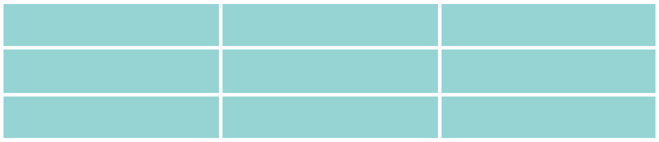
This will make the borders collapse into a single border:



```
table, th, td {
  border: 1px solid black;
  border-collapse: collapse;
}
```

Style Table Borders

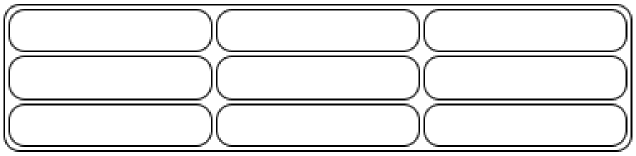
If you set a background color of each cell, and give the border a white color (the same as the document background), you get the impression of an invisible border:



```
table, th, td {
  border: 1px solid white;
  border-collapse: collapse;
}
th, td {
  background-color: #96D4D4;
}
```

Round Table Borders

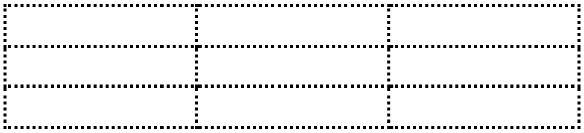
With the `border-radius` property, the borders get rounded corners:



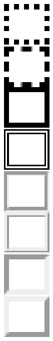
```
table, th, td {
  border: 1px solid black;
  border-radius: 10px;
}
```

Dotted Table Borders

With the `border-style` property, you can set the appearance of the border.



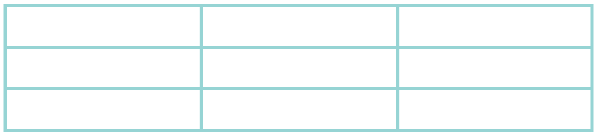
The following values are allowed:



- `dotted`
- `dashed`
- `solid`
- `double`
- `groove`
- `ridge`
- `inset`
- `outset`
- `none`
- `hidden`

Border Color

With the `border-color` property, you can set the color of the border.



```
 th, td {
  border-color: #96D4D4;
}
```

HTML Table Colspan & Rowspan

HTML tables can have cells that span over multiple rows and/or columns.

NAME		

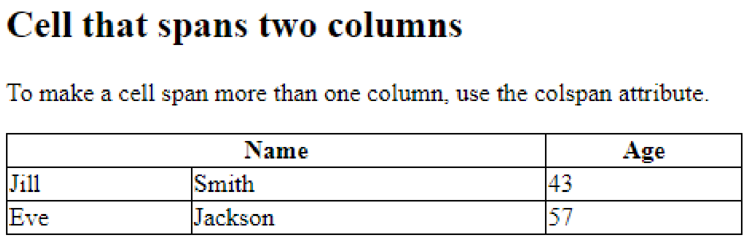
APRIL		

2022		
FIESTA		

HTML Table - Colspan

To make a cell span over multiple columns, use the colspan attribute:

```
<table>
  <tr>
    <th colspan="2">Name</th>
    <th>Age</th>
  </tr>
  <tr>
    <td>Jill</td>
    <td>Smith</td>
    <td>43</td>
  </tr>
  <tr>
    <td>Eve</td>
    <td>Jackson</td>
    <td>57</td>
  </tr>
</table>
```



Cell that spans two columns

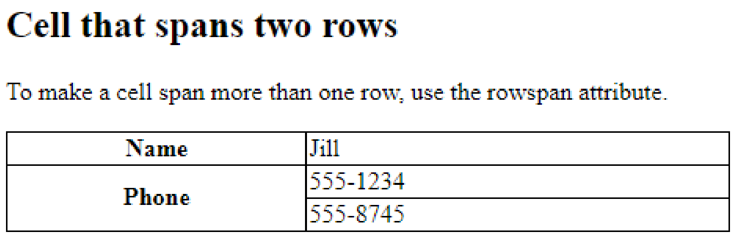
To make a cell span more than one column, use the colspan attribute.

Name		Age
Jill	Smith	43
Eve	Jackson	57

HTML Table - Rowspan

To make a cell span over multiple rows, use the rowspan attribute:

```
<table>
  <tr>
    <th>Name</th>
    <td>Jill</td>
  </tr>
  <tr>
```



Cell that spans two rows

To make a cell span more than one row, use the rowspan attribute.

Name	Jill
Phone	555-1234
	555-8745

```
    <th rowspan="2">Phone</th>
    <td>555-1234</td>
  </tr>
  <tr>
    <td>555-8745</td>
</tr>
</table>
```

HTML - The Head Element

The HTML `<head>` element is a container for the following elements: `<title>`, `<style>`, `<meta>`, `<link>`, `<script>`, and `<base>`.

The `<head>` element is a container for metadata (data about data) and is placed between the `<html>` tag and the `<body>` tag.

HTML metadata is data about the HTML document. Metadata is not displayed.

Metadata typically define the document title, character set, styles, scripts, and other meta information.

The following elements can go inside the `<head>` element:

- <title> (required in every HTML document)
- <style>
- <base>
- <link>
- <meta>
- <script>
- <noscript>

The HTML <meta> Element

The `<meta>` element is typically used to specify the character set, page description, keywords, author of the document, and viewport settings.

The metadata will not be displayed on the page, but is used by browsers (how to display content or reload page), by search engines (keywords), and other web services.

```
<head>
  <meta charset="UTF-8">
  <meta name="description" content="Free Web tutorials">
  <meta name="keywords" content="HTML, CSS, JavaScript">
  <meta name="author" content="John Doe">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
</head>
```

Setting The Viewport

The viewport is the user's visible area of a web page. It varies with the device - it will be smaller on a mobile phone than on a computer screen.

You should include the following `<meta>` element in all your web pages:

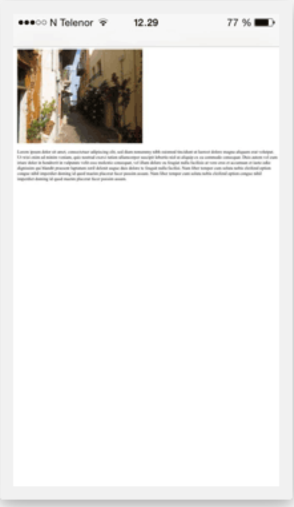
```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```

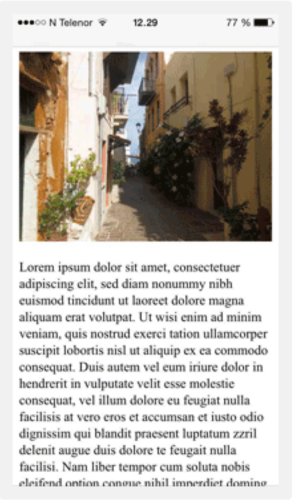
This gives the browser instructions on how to control the page's dimensions and scaling.

The `width=device-width` part sets the width of the page to follow the screen-width of the device (which will vary depending on the device).

The `initial-scale=1.0` part sets the initial zoom level when the page is first loaded by the browser.

Here is an example of a web page *without* the viewport meta tag, and the same web page *with* the viewport meta tag:





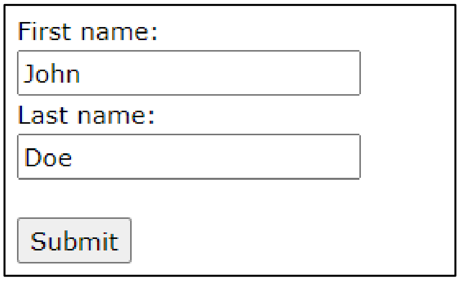
Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat. Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat. Duis autem vel eum iriure dolor in hendrerit in vulputate velit esse molestie consequat, vel illum dolore eu feugiat nulla facilisis at vero eros et accumsan et iusto odio dignissim qui blandit praesent luptatum zzril delenit augue duis dolore te feugait nulla facilisi. Nam liber tempor cum soluta nobis eleifend option congue nihil imperdiet doming

Without the viewport meta tag

With the viewport meta tag

HTML Forms

An HTML form is used to collect user input. The user input is most often sent to a server for processing.



First name:

John

Last name:

Doe

HTML Form Attributes

The `<form>` HTML element represents a document section containing interactive controls for submitting information.

The Action Attribute

The `action` attribute defines the action to be performed when the form is submitted.

Usually, the form data is sent to a file on the server when the user clicks on the submit button.

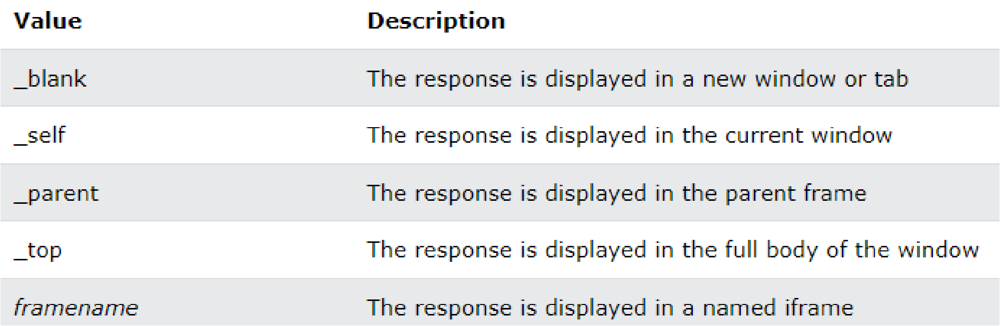
In the example below, the form data is sent to a file called "action_page.php". This file contains a server-side script that handles the form data:

```
<form action="/action_page.php">
  <label for="fname">First name:</label><br>
  <input type="text" id="fname" name="fname" value="John"><br>
  <label for="lname">Last name:</label><br>
  <input type="text" id="lname" name="lname" value="Doe"><br><br>
  <input type="submit" value="Submit">
</form>
```

The Target Attribute

The `target` attribute specifies where to display the response that is received after submitting the form.

The target attribute can have one of the following values:



Value	Description
_blank	The response is displayed in a new window or tab
_self	The response is displayed in the current window
_parent	The response is displayed in the parent frame
_top	The response is displayed in the full body of the window
framename	The response is displayed in a named iframe

```
<form action="/action_page.php" target="_blank">
```

The Method Attribute

The method attribute specifies the HTTP method to be used when submitting the form data.

The form-data can be sent as URL variables (with method="get") or as HTTP post transaction (with method="post").

The default HTTP method when submitting form data is GET.

```
<form action="/action_page.php" method="get">
```

Notes on GET:

- Appends the form data to the URL, in name/value pairs
- NEVER use GET to send sensitive data! (the submitted form data is visible in the URL!)
- The length of a URL is limited (2048 characters)
- Useful for form submissions where a user wants to bookmark the result
- GET is good for non-secure data, like query strings in Google

Notes on POST:

- Appends the form data inside the body of the HTTP request (the submitted form data is not shown in the URL)
- POST has no size limitations, and can be used to send large amounts of data.
- Form submissions with POST cannot be bookmarked

The HTML <form> element is used to create an HTML form for user input:

```
<form>
.
form elements
.
</form>
```

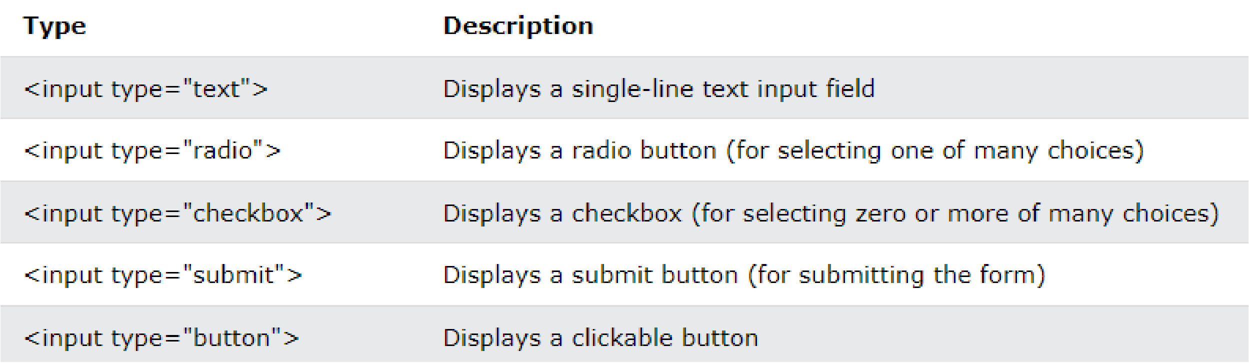
The `<form>` element is a container for different types of input elements, such as: text fields, checkboxes, radio buttons, submit buttons, etc.

The <input> Element

The HTML `<input>` element is the most used form element.

An `<input>` element can be displayed in many ways, depending on the `type` attribute.

Here are some examples:

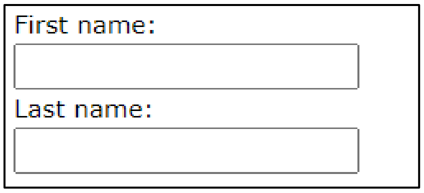


Type	Description
<input type="text">	Displays a single-line text input field
<input type="radio">	Displays a radio button (for selecting one of many choices)
<input type="checkbox">	Displays a checkbox (for selecting zero or more of many choices)
<input type="submit">	Displays a submit button (for submitting the form)
<input type="button">	Displays a clickable button

The `<input type="text">` defines a single-line input field for text input.

```
<form>
  <label for="fname">First name:</label><br>
  <input type="text" id="fname" name="fname"><br>
  <label for="lname">Last name:</label><br>
  <input type="text" id="lname" name="lname">
</form>
```

Output



First name:

□

Last name:

□

Note: The form itself is not visible. Also note that the default width of an input field is 20 characters.

The <label> Element

Notice the use of the <label> element in the example above.

The <label> tag defines a label for many form elements.

The <label> element is useful for screen-reader users, because the screen-reader will read out loud the label when the user focuses on the input element.

The <label> element also helps users who have difficulty clicking on very small regions (such as radio buttons or checkboxes) - because when the user clicks the text within the <label> element, it toggles the radio button/checkbox.

The for attribute of the <label> tag should be equal to the id attribute of the <input> element to bind them together.

Radio Buttons

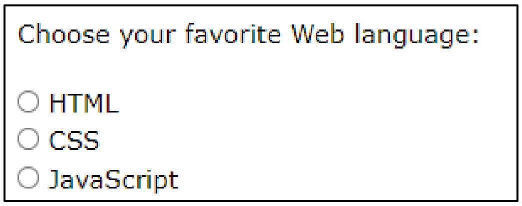
The <input type="radio"> defines a radio button.

Radio buttons let a user select ONE of a limited number of choices.

```
<p>Choose your favorite Web language:</p>

<form>
  <input type="radio" id="html" name="fav_language" value="HTML">
  <label for="html">HTML</label><br>
  <input type="radio" id="css" name="fav_language" value="CSS">
  <label for="css">CSS</label><br>
  <input type="radio" id="javascript" name="fav_language" value="JavaScr
ipt">
```

```
  <label for="javascript">JavaScript</label>
</form>
```

Output



Choose your favorite Web language:

○ HTML
○ CSS
○ JavaScript

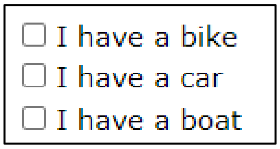
Checkboxes

The `<input type="checkbox">` defines a **checkbox**.

Checkboxes let a user select ZERO or MORE options of a limited number of choices.

```
<form>
  <input type="checkbox" id="vehicle1" name="vehicle1" value="Bike">
  <label for="vehicle1"> I have a bike</label><br>
  <input type="checkbox" id="vehicle2" name="vehicle2" value="Car">
  <label for="vehicle2"> I have a car</label><br>
  <input type="checkbox" id="vehicle3" name="vehicle3" value="Boat">
  <label for="vehicle3"> I have a boat</label>
</form>
```

Output



□ I have a bike
□ I have a car
□ I have a boat

The Submit Button

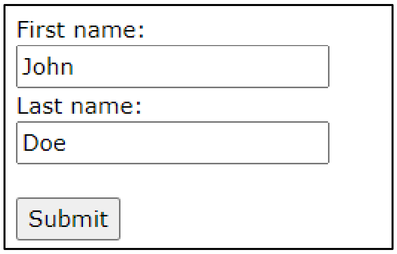
The `<input type="submit">` defines a button for submitting the form data to a form-handler.

The form-handler is typically a file on the server with a script for processing input data.

The form-handler is specified in the form's action attribute.

```
<form action="/action_page.php">
  <label for="fname">First name:</label><br>
  <input type="text" id="fname" name="fname" value="John"><br>
  <label for="lname">Last name:</label><br>
  <input type="text" id="lname" name="lname" value="Doe"><br><br>
  <input type="submit" value="Submit">
</form>
```

Output



First name:

John

Last name:

Doe

HTML Semantic Elements

A semantic element clearly describes its meaning to both the browser and the developer.

Examples of **non-semantic** elements: `<div>` and `<span>` - Tells nothing about its content.

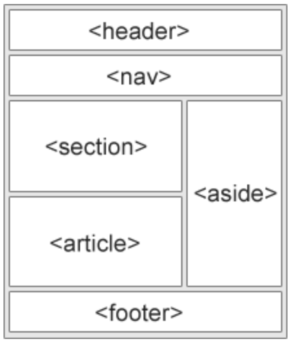
Examples of **semantic** elements: `<form>`, `<table>`, and `<article>` - Clearly defines its content.

Semantic Elements in HTML

Many web sites contain HTML code like: <div id="nav"> <div class="header"> <div id="footer"> to indicate navigation, header, and footer.

In HTML there are some semantic elements that can be used to define different parts of a web page:

- <article>
- <aside>
- <details>
- <figcaption>
- <figure>
- <footer>



- <header>
- <main>
- <mark>
- <nav>
- <section>
- <summary>
- <time>

HTML <section> Element

The <section> element defines a section in a document.

According to W3C's HTML documentation: "A section is a thematic grouping of content, typically with a heading."

Examples of where a <section> element can be used:

- Chapters
- Introduction
- News items
- Contact information

A web page could normally be split into sections for introduction, content, and contact information.

```
<section>
<h1>WWF</h1>
<p>The World Wide Fund for Nature (WWF) is an international organization
working on issues regarding the conservation, research and restoration
of the environment, formerly named the World Wildlife Fund. WWF was
founded in 1961.</p>
</section>

<section>
<h1>WWF's Panda symbol</h1>
<p>The Panda has become the symbol of WWF. The well-known panda logo of
WWF originated from a panda named Chi Chi that was transferred from the
Beijing Zoo to the London Zoo in the same year of the establishment of
WWF.</p>
</section>
```

HTML <article> Element

The <article> element specifies independent, self-contained content.

An article should make sense on its own, and it should be possible to distribute it independently from the rest of the web site.

Examples of where the <article> element can be used:

- Forum posts
- Blog posts
- User comments
- Product cards
- Newspaper articles

```
<article>
<h2>Google Chrome</h2>
<p>Google Chrome is a web browser developed by Google, released in 2008.
Chrome is the world's most popular web browser today!</p>
</article>

<article>
<h2>Mozilla Firefox</h2>
<p>Mozilla Firefox is an open-source web browser developed by Mozilla.
Firefox has been the second most popular web browser since January,
2018.</p>
</article>

<article>
<h2>Microsoft Edge</h2>
<p>Microsoft Edge is a web browser developed by Microsoft, released in
2015. Microsoft Edge replaced Internet Explorer.</p>
</article>
```

HTML <header> Element

The <header> element represents a container for introductory content or a set of navigational links.

A <header> element typically contains:

- one or more heading elements (<h1> - <h6>)
- logo or icon
- authorship information

```
<article>
  <header>
    <h1>What Does WWF Do?</h1>
    <p>WWF's mission:</p>
  </header>
  <p>WWF's mission is to stop the degradation of our planet's natural
environment,
  and build a future in which humans live in harmony with nature.</p>
</article>
```

Note: You can have several <header> elements in one HTML document. However, <header> cannot be placed within a <footer>, <address> or another <header> element.

HTML <footer> Element

The <footer> element defines a footer for a document or section.

A <footer> element typically contains:

- authorship information
- copyright information
- contact information
- sitemap
- back to top links
- related documents

You can have several <footer> elements in one document.

```
<footer>
  <p>Author: Hege Refsnes</p>
  <p><a href="mailto:hege@example.com">hege@example.com</a></p>
</footer>
```

HTML <nav> Element

The <nav> element defines a set of navigation links.

```
<nav>
  <a href="/html/">HTML</a> |
  <a href="/css/">CSS</a> |
  <a href="/js/">JavaScript</a> |
  <a href="/jquery/">jQuery</a>
</nav>
```

HTML <figure> and <figcaption> Elements

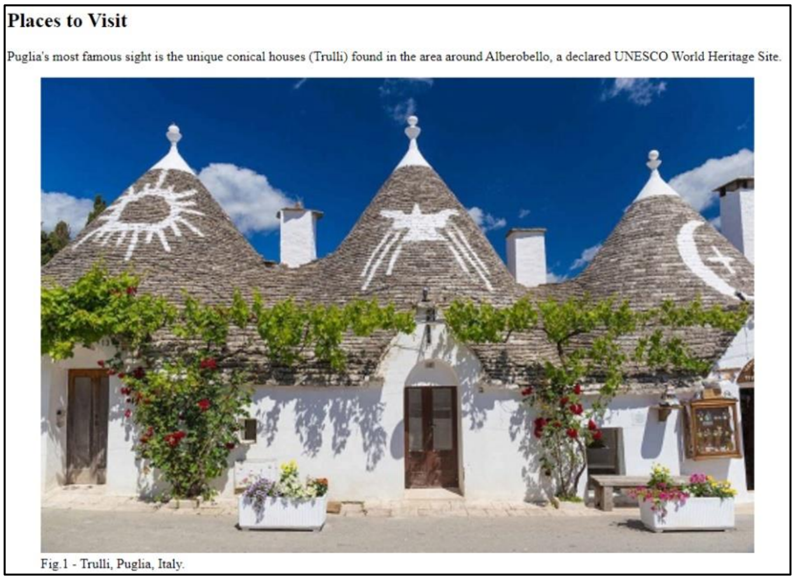
The <figure> tag specifies self-contained content, like illustrations, diagrams, photos, code listings, etc.

The <figcaption> tag defines a caption for a <figure> element. The <figcaption> element can be placed as the first or as the last child of a <figure> element.

The <img> element defines the actual image/illustration.

```
<figure>
  <img src="pic_trulli.jpg" alt="Trulli">
  <figcaption>Fig1. - Trulli, Puglia, Italy.</figcaption>
</figure>
```

Output



Places to Visit

Puglia's most famous sight is the unique conical houses (Trulli) found in the area around Alberobello, a declared UNESCO World Heritage Site.

Fig.1 - Trulli, Puglia, Italy.

What is form validation?

Go to any popular site with a registration form, and you will notice that they provide feedback when you don't enter your data in the format they are expecting. You'll get messages such as:

- "This field is required" (You can't leave this field blank).
- "Please enter your phone number in the format xxx-xxxx" (A specific data format is required for it to be considered valid).
- "Please enter a valid email address" (the data you entered is not in the right format).
- "Your password needs to be between 8 and 30 characters long and contain one uppercase letter, one symbol, and a number." (A very specific data format is required for your data).

This is called form validation. When you enter data, the browser and/or the web server will check to see that the data is in the correct format and within the constraints set by the application. Validation done in the browser is called client-side validation, while validation done on the server is called server-side validation. In this chapter we are focusing on client-side validation.

If the information is correctly formatted, the application allows the data to be submitted to the server and (usually) saved in a database; if the information isn't correctly formatted, it gives the user an error message explaining what needs to be corrected, and lets them try again.

We want to make filling out web forms as easy as possible. So why do we insist on validating our forms? There are three main reasons:

- **We want to get the right data**, in the right format. Our applications won't work properly if our users' data is stored in the wrong format, is incorrect, or is omitted altogether.
- **We want to protect our users' data**. Forcing our users to enter secure passwords makes it easier to protect their account information.
- **We want to protect ourselves**. There are many ways that malicious users can misuse unprotected forms to damage the application.

Different types of client-side validation

There are two different types of client-side validation that you'll encounter on the web:

- **Built-in form validation** uses HTML form validation features, which we've discussed in many places throughout this module. This validation generally doesn't require much JavaScript. Built-in form validation has better performance than JavaScript, but it is not as customizable as JavaScript validation.

- **JavaScript** validation is coded using JavaScript. This validation is completely customizable, but you need to create it all (or use a library).

Using built-in form validation

One of the most significant features of modern form controls is the ability to validate most user data without relying on JavaScript. This is done by using validation attributes on form elements. We've seen many of these earlier in the course, but to recap:

- required: Specifies whether a form field needs to be filled in before the form can be submitted.
- minlength and maxlength: Specifies the minimum and maximum length of textual data (strings).
- min and max: Specifies the minimum and maximum values of numerical input types.
- type: Specifies whether the data needs to be a number, an email address, or some other specific preset type.
- pattern: Specifies a regular expression that defines a pattern the entered data needs to follow.

Examples:

```
<form action="/action_page.php">
  <label for="username">Username:</label>
  <input type="text" id="username" name="username" required>
  <input type="submit">
</form>
```

HTML <div> Tag

The `<div>` tag defines a division or a section in an HTML document.

The `<div>` tag is used as a container for HTML elements - which is then styled with CSS or manipulated with JavaScript.

The `<div>` tag is easily styled by using the class or id attribute.

Any sort of content can be put inside the `<div>` tag!

Note: By default, browsers always place a line break before and after the <div> element.

```
<div class="myDiv">
  <h2>This is a heading in a div element</h2>
  <p>This is some text in a div element.</p>
</div>
```

HTML <span> Tag

The <span> tag is an inline container used to mark up a part of a text, or a part of a document.

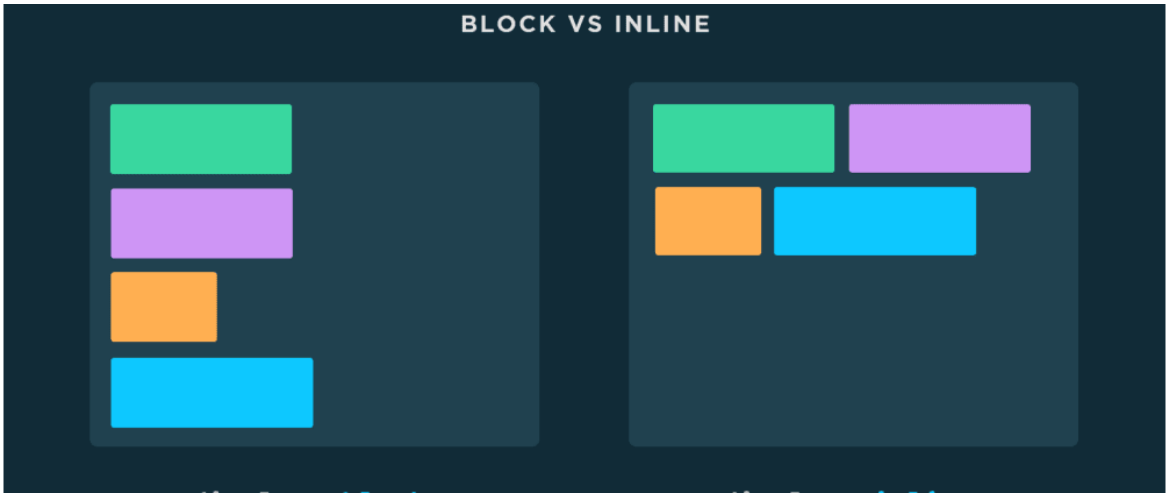
The <span> tag is easily styled by CSS or manipulated with JavaScript using the class or id attribute.

The <span> tag is much like the <div> element, but <div> is a block-level element and <span> is an inline element.

```
<p>My mother has <span style="color:blue">blue</span> eyes.</p>
```

Block vs Inline

- Block element always starts with new line and it takes complete available width.(stretches out to the left and right as far as it can).
- An inline element is the opposite of the block-level element.
- As a name itself indicates Inline element does not start with a new line it will embed with existing line.
- inline element takes the width only as much as necessary.



HTML Audio

The HTML `<audio>` element is used to play an audio file on a web page.

To play an audio file in HTML, use the `<audio>` element:

```
<audio controls>
  <source src="horse.ogg" type="audio/ogg">
  <source src="horse.mp3" type="audio/mpeg">
Your browser does not support the audio element.
</audio>
```

HTML Audio - How It Works

The `controls` attribute adds audio controls, like play, pause, and volume.

The `<source>` element allows you to specify alternative audio files which the browser may choose from. The browser will use the first recognized format.

The text between the `<audio>` and `</audio>` tags will only be displayed in browsers that do not support the `<audio>` element.

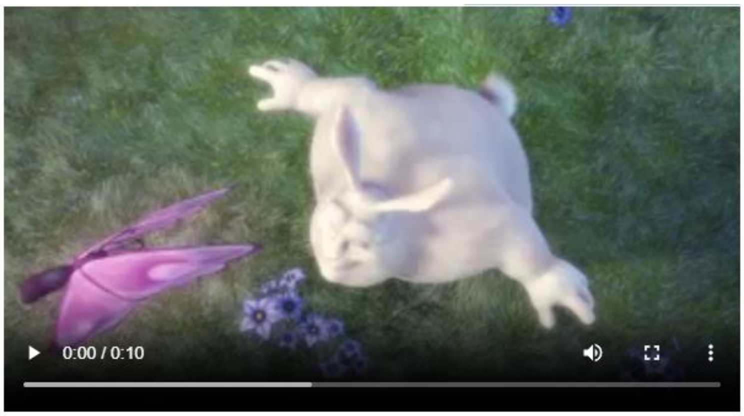
HTML <audio> Autoplay

To start an audio file automatically, use the `autoplay` attribute:

```
<audio controls autoplay>
  <source src="horse.ogg" type="audio/ogg">
  <source src="horse.mp3" type="audio/mpeg">
Your browser does not support the audio element.
</audio>
```

HTML Video

The HTML <video> element is used to show a video on a web page.



The controls attribute adds video controls, like play, pause, and volume.

It is a good idea to always include width and height attributes. If height and width are not set, the page might flicker while the video loads.

The <source> element allows you to specify alternative video files which the browser may choose from. The browser will use the first recognized format.

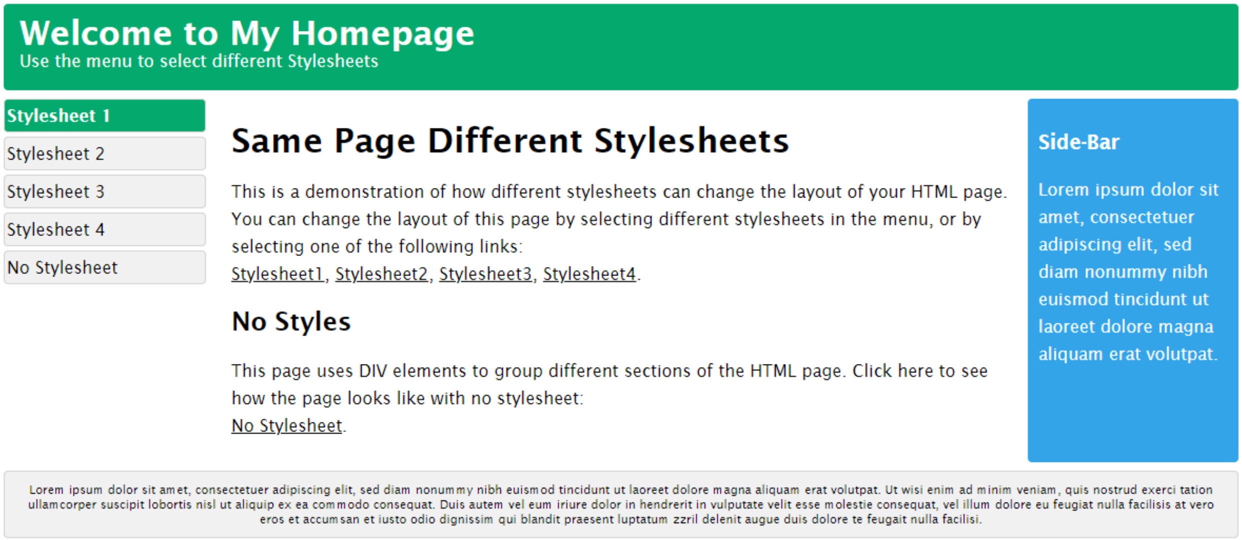
The text between the <video> and </video> tags will only be displayed in browsers that do not support the <video> element.

```
<video width="320" height="240" controls>
  <source src="movie.mp4" type="video/mp4">
  <source src="movie.ogg" type="video/ogg">
Your browser does not support the video tag.
</video>
```

CHAPTER 3 – INTRODUCTION TO CSS

3.1 CSS

CSS (Cascading Style Sheets) is used to style and lay out web pages — for example, to alter the font, color, size, and spacing of your content, split it into multiple columns, or add animations and other decorative features. This module provides a gentle beginning to your path towards CSS mastery with the basics of how it works, what the syntax looks like, and how you can start using it to add styling to HTML.



Same Page Different Stylesheets

This is a demonstration of how different stylesheets can change the layout of your HTML page. You can change the layout of this page by selecting different stylesheets in the menu, or by selecting one of the following links:
Stylesheet1, Stylesheet2, Stylesheet3, Stylesheet4.

No Styles

This page uses DIV elements to group different sections of the HTML page. Click here to see how the page looks like with no stylesheet:
No Stylesheet.

Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat.

Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat. Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat. Duis autem vel eum iriure dolor in hendrerit in vulputate velit esse molestie consequat, vel illum dolore eu feugiat nulla facilisis at vero eros et accumsan et iusto odio dignissim qui blandit praesent luptatum zzril delenit augue duis dolore te feugait nulla facilisi.

CSS is designed to make style sheets for the web. It is independent of HTML and can be used with any XML-based markup language. Now let’s try to break the acronym:

- Cascading: Falling of Styles
- Style: Adding designs/Styling our HTML tags
- Sheets: Writing our style in different documents

CSS History

- 1994: First Proposed by Hakon Wium Lie on 10th October
- 1996: CSS was published on 17th November with influencer Bert Bos
- Later he became co-author of CSS
- 1996: CSS became official with CSS was published in December
- 1997: Created CSS level 2 on 4th November
- 1998: Published on 12th May

CSS Example

```
body {
  background-color: lightblue;
}

h1 {
  color: white;
  text-align: center;
}

p {
  font-family: verdana;
  font-size: 20px;
}
```

CSS Solved a Big Problem

HTML was NEVER intended to contain tags for formatting a web page!

HTML was created to describe the content of a web page, like:

<h1>This is a heading</h1>

<p>This is a paragraph.</p>

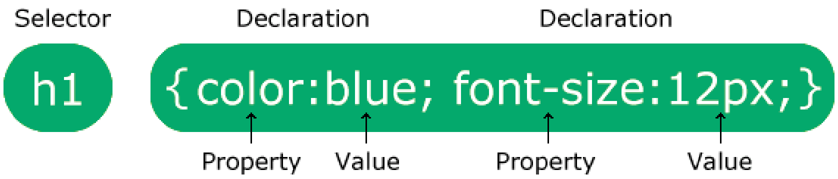
When tags like <font>, and color attributes were added to the HTML 3.2 specification, it started a nightmare for web developers. Development of large websites, where fonts and color information were added to every single page, became a long and expensive process.

To solve this problem, the World Wide Web Consortium (W3C) created CSS.

CSS removed the style formatting from the HTML page!

CSS Syntax

A CSS rule consists of a selector and a declaration block.



The selector points to the HTML element you want to style.

The declaration block contains one or more declarations separated by semicolons.

Each declaration includes a CSS property name and a value, separated by a colon.

Multiple CSS declarations are separated with semicolons, and declaration blocks are surrounded by curly braces.

Adding CSS in HTML

There are 3 ways to write CSS in our HTML file.

- Inline CSS
- Internal CSS
- External CSS

Inline CSS

- Before CSS this was the only way to apply styles
- Not an efficient way to write as it has a lot of redundancy
- Self-contained
- Uniquely applied on each element
- The idea of separation of concerns was lost

```
<h3 style=” color:red”> Have a great day </h3>
<p  style =” color: green”> I did this , I did that </p>
```

Internal CSS

- With the help of style tag, we can apply styles within the HTML file
- Redundancy is removed
- But the idea of separation of concerns still lost
- Uniquely applied on a single document

```
< style>
        h1{
              color:red;
           }
 </style>
<h3> Have a great day </h3>
```

External CSS

- With the help of <link> tag in the head tag, we can apply styles
- Reference is added
- File saved with .css extension
- Redundancy is removed
- The idea of separation of concerns is maintained
- Uniquely applied to each document

```
<!DOCTYPE html>
<html>
<head>
<link rel="stylesheet" href="mystyle.css">
</head>
<body>

<h1>This is a heading</h1>
<p>This is a paragraph.</p>

</body>
</html>
```

```
/*mystyle.css*/

body {
  background-color: grey;
}

h1 {
  color: navy;
  margin-left: 20px;
}
```

What style will be used when there is more than one style specified for an HTML element?

All the styles in a page will "cascade" into a new "virtual" style sheet by the following rules, where number one has the highest priority:

1. Inline style (inside an HTML element)
2. External and internal style sheets (in the head section)
3. Browser default

3.1.1 CSS Selectors

CSS selectors are used to "find" (or select) the HTML elements you want to style.

We can divide CSS selectors into five categories:

- Simple selectors (select elements based on name, id, class)
- Combinator selectors (select elements based on a specific relationship between them)
- Pseudo-class selectors (select elements based on a certain state)
- Pseudo-elements selectors (select and style a part of an element)
- Attribute selectors (select elements based on an attribute or attribute value)

The CSS element Selector

The element selector selects HTML elements based on the element name.

```css
p {
  text-align: center;
  color: red;
}
```

The CSS id Selector

The id selector uses the id attribute of an HTML element to select a specific element.

The id of an element is unique within a page, so the id selector is used to select one unique element!

To select an element with a specific id, write a hash (#) character, followed by the id of the element.

```css
#para1 {
  text-align: center;
  color: red;
}
```

The CSS class Selector

The class selector selects HTML elements with a specific class attribute.

To select elements with a specific class, write a period (.) character, followed by the class name.

```
.center {
  text-align: center;
  color: red;
}
```

The CSS Universal Selector

The universal selector (*) selects all HTML elements on the page.

```
* {
  text-align: center;
  color: blue;
}
```

The CSS Grouping Selector

The grouping selector selects all the HTML elements with the same style definitions.

Look at the following CSS code (the h1, h2, and p elements have the same style definitions):

It will be better to group the selectors, to minimize the code.

To group selectors, separate each selector with a comma.

```
h1 {
  text-align: center;
  color: red;
}

h2 {
  text-align: center;
  color: red;
}

p {
  text-align: center;
  color: red;
}
```

```
h1, h2, p {
  text-align: center;
  color: red;
}
```

CSS Combinators

A CSS selector can contain more than one simple selector. Between the simple selectors, we can include a combinator.

There are four different combinators in CSS:

- descendant selector (space)
- child selector (>)
- adjacent sibling selector (+)
- general sibling selector (~)

Descendant Selector

The descendant selector matches all elements that are descendants of a specified element.

The following example selects all <p> elements inside <div> elements:

```css
div p {
  background-color: yellow;
}
```

Child Selector (>)

The child selector selects all elements that are the children of a specified element.

The following example selects all <p> elements that are children of a <div> element:

```css
div > p {
  background-color: yellow;
}
```

Adjacent Sibling Selector (+)

The adjacent sibling selector is used to select an element that is directly after another specific element.

Sibling elements must have the same parent element, and "adjacent" means "immediately following".

The following example selects the first <p> element that are placed immediately after <div> elements:

```
div + p {
  background-color: yellow;
}
```

General Sibling Selector (~)

The general sibling selector selects all elements that are next siblings of a specified element.

The following example selects all <p> elements that are next siblings of <div> elements:

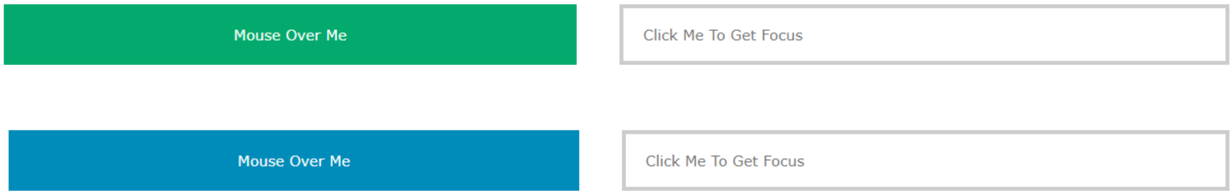
```
div ~ p {
  background-color: yellow;
}
```

CSS Pseudo-classes

A pseudo-class is used to define a special state of an element.

For example, it can be used to:

- Style an element when a user mouses over it
- Style visited and unvisited links differently
- Style an element when it gets focus



```
selector:pseudo-class {
  property: value;
}
```

Anchor Pseudo-classes

Links can be displayed in different ways:

```
/* unvisited link */
a:link {
  color: #FF0000;
}

/* visited link */
a:visited {
  color: #00FF00;
}

/* mouse over link */
a:hover {
  color: #FF00FF;
}

/* selected link */
a:active {
  color: #0000FF;
}
```

Pseudo-classes and HTML Classes

Pseudo-classes can be combined with HTML classes:

When you hover over the link in the example, it will change color:

```
div:hover {
  background-color: blue;
}
```

All pseudoclasses

Selector	Example	Example description
:active	a:active	Selects the active link

:checked	input:checked	Selects every checked <input> element
:disabled	input:disabled	Selects every disabled <input> element
:empty	p:empty	Selects every <p> element that has no children
:enabled	input:enabled	Selects every enabled <input> element
:first-child	p:first-child	Selects every <p> elements that is the first child of its parent
:first-of-type	p:first-of-type	Selects every <p> element that is the first <p> element of its parent
:focus	input:focus	Selects the <input> element that has focus
:hover	a:hover	Selects links on mouse over
:in-range	input:in-range	Selects <input> elements with a value within a specified range
:invalid	input:invalid	Selects all <input> elements with an invalid value
:lang(*language*)	p:lang(it)	Selects every <p> element with a lang attribute value starting with "it"
:last-child	p:last-child	Selects every <p> elements that is the last child of its parent
:last-of-type	p:last-of-type	Selects every <p> element that is the last <p> element of its parent
:link	a:link	Selects all unvisited links
:not(selector)	:not(p)	Selects every element that is not a <p> element
:nth-child(n)	p:nth-child(2)	Selects every <p> element that is the second child of its parent
:nth-last-child(n)	p:nth-last-child(2)	Selects every <p> element that is the second child of its parent, counting from the last child
:nth-last-of-type(n)	p:nth-last-of-type(2)	Selects every <p> element that is the second <p> element of its parent, counting from the last child
:nth-of-type(n)	p:nth-of-type(2)	Selects every <p> element that is the second <p> element of its parent
:only-of-type	p:only-of-type	Selects every <p> element that is the only <p> element of its parent

:only-child	p:only-child	Selects every <p> element that is the only child of its parent
:optional	input:optional	Selects <input> elements with no "required" attribute
:out-of-range	input:out-of-range	Selects <input> elements with a value outside a specified range
:read-only	input:read-only	Selects <input> elements with a "readonly" attribute specified
:read-write	input:read-write	Selects <input> elements with no "readonly" attribute
:required	input:required	Selects <input> elements with a "required" attribute specified
:root	root	Selects the document's root element
:target	#news:target	Selects the current active #news element (clicked on a URL containing that anchor name)
:valid	input:valid	Selects all <input> elements with a valid value
:visited	a:visited	Selects all visited links

Pseudo-Elements

A CSS pseudo-element is used to style specified parts of an element.

For example, it can be used to:

- Style the first letter, or line, of an element
- Insert content before, or after, the content of an element

Syntax

The syntax of pseudo-elements:

```
selector::pseudo-element {
  property: value;
}
```

The ::first-line Pseudo-element

The `::first-line` pseudo-element is used to add a special style to the first line of a text.

The following example formats the first line of the text in all <p> elements:

```
p::first-line {
  color: #ff0000;
  font-variant: small-caps;
}
```

The following properties apply to the ::first-line pseudo-element:

- font properties
- color properties
- background properties
- word-spacing
- letter-spacing
- text-decoration
- vertical-align
- text-transform
- line-height
- clear

Notice the double colon notation - ::first-line versus :first-line

The double colon replaced the single-colon notation for pseudo-elements in CSS3. This was an attempt from W3C to distinguish between pseudo-classes and pseudo-elements.

The single-colon syntax was used for both pseudo-classes and pseudo-elements in CSS2 and CSS1. For backward compatibility, the single-colon syntax is acceptable for CSS2 and CSS1 pseudo-elements.

The ::first-letter Pseudo-element

The ::first-letter pseudo-element is used to add a special style to the first letter of a text.

The following example formats the first letter of the text in all <p> elements:

```
p::first-letter {
  color: #ff0000;
  font-size: xx-large;
}
```

Note: The ::first-letter pseudo-element can only be applied to block-level elements.

The following properties apply to the ::first-letter pseudo- element:

- font properties
- color properties
- background properties
- margin properties
- padding properties
- border properties
- text-decoration
- vertical-align (only if "float" is "none")
- text-transform
- line-height
- float
- clear

CSS - The ::before Pseudo-element

The ::before pseudo-element can be used to insert some content before the content of an element.

The following example inserts an image before the content of each <h1> element:

```
h1::before {
  content: url(smiley.gif);
}
```

CSS - The ::after Pseudo-element

The ::after pseudo-element can be used to insert some content after the content of an element.

The following example inserts an image after the content of each <h1> element:

```
h1::after {
  content: url(smiley.gif);
}
```

CSS - The ::marker Pseudo-element

The ::marker pseudo-element selects the markers of list items.

The following example styles the markers of list items:

```
::marker {
  color: red;
  font-size: 23px;
}
```

CSS - The ::selection Pseudo-element

The `::selection` pseudo-element matches the portion of an element that is selected by a user.

The following CSS properties can be applied to `::selection`: `color`, `background`, `cursor`, and `outline`.

The following example makes the selected text red on a yellow background:

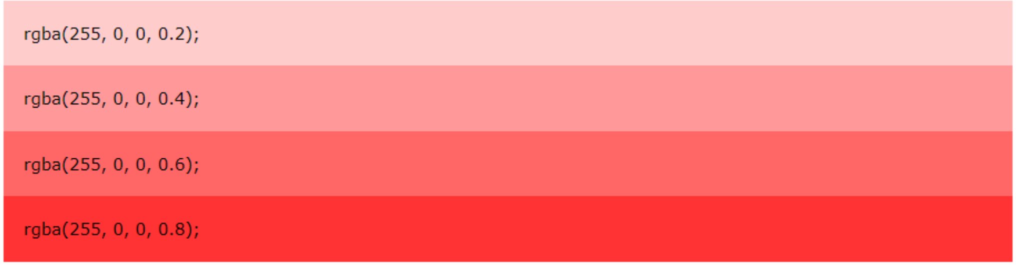
```
::selection {
  color: red;
  background: yellow;
}
```

All PseudoElements

Selector	Example	Example description
::after	p::after	Insert something after the content of each <p> element
::before	p::before	Insert something before the content of each <p> element
::first-letter	p::first-letter	Selects the first letter of each <p> element
::first-line	p::first-line	Selects the first line of each <p> element
::marker	::marker	Selects the markers of list items
::selection	p::selection	Selects the portion of an element that is selected by a user

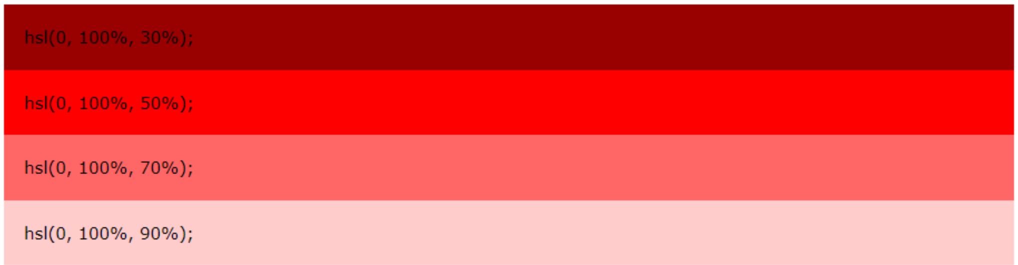
CSS Colors

- Three widely used techniques are as follows
    - RGBA



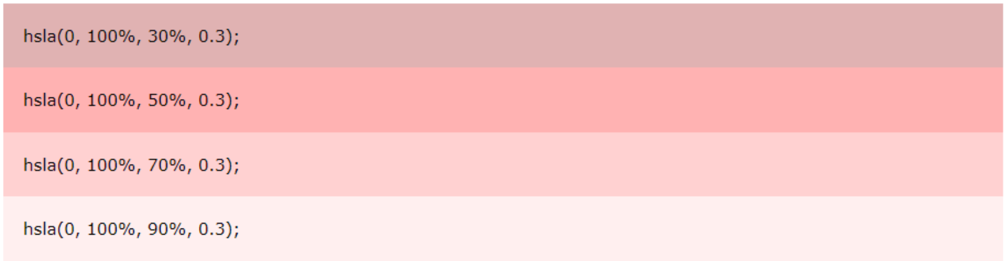
```
rgba(255, 0, 0, 0.2);
rgba(255, 0, 0, 0.4);
rgba(255, 0, 0, 0.6);
rgba(255, 0, 0, 0.8);
```

        - This starts with RGB and takes 3 parameter
        - 3 parameter basically corresponds to red, green and blue
        - The value of each parameter may vary from 0 to 255.
        - Eg: RGB(255,0,0); means color red
    - HEX



```
hsl(0, 100%, 30%);
hsl(0, 100%, 50%);
hsl(0, 100%, 70%);
hsl(0, 100%, 90%);
```

        - Hue is a degree on the color wheel (from 0 to 360):
        - 0 (or 360) is red
        - 120 is green
        - 240 is blue
        - Saturation is a percentage value: 100% is the full color.
        - Lightness is also a percentage; 0% is dark (black) and 100% is white.
    - HSL



```
hsla(0, 100%, 30%, 0.3);
hsla(0, 100%, 50%, 0.3);
hsla(0, 100%, 70%, 0.3);
hsla(0, 100%, 90%, 0.3);
```

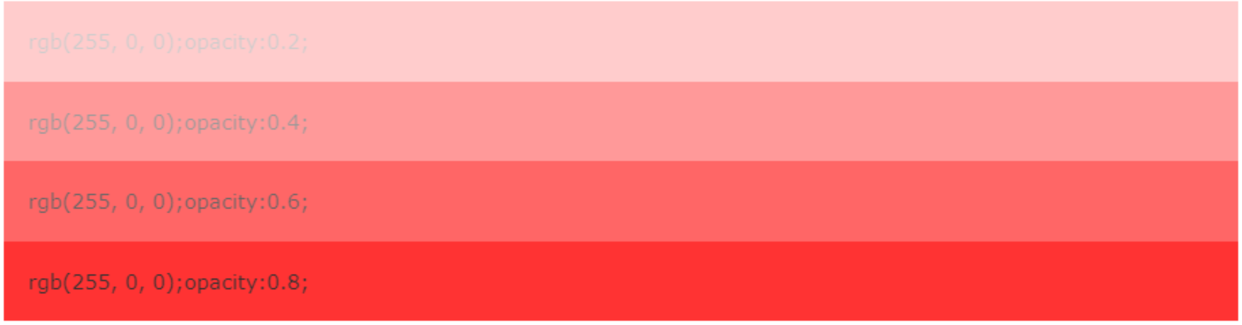
        - HSLA color values are an extension of HSL color values with an alpha channel - which specifies the opacity for a color.

- An HSLA color value is specified with: hsla(hue, saturation, lightness, alpha), where the alpha parameter defines the opacity. The alpha parameter is a number between 0.0 (fully transparent) and 1.0 (fully opaque).

CSS Opacity

The CSS `opacity` property sets the opacity for the whole element (both background color and text will be opaque/transparent).

The `opacity` property value must be a number between 0.0 (fully transparent) and 1.0 (fully opaque).



rgb(255, 0, 0);opacity:0.2;

rgb(255, 0, 0);opacity:0.4;

rgb(255, 0, 0);opacity:0.6;

rgb(255, 0, 0);opacity:0.8;

Notice that the text above will also be transparent/opaque!

The following example shows different elements with opacity:

```
#p1 {background-color:rgb(255,0,0);opacity:0.6;}  /* red with opacity */
#p2 {background-color:rgb(0,255,0);opacity:0.6;}  /* green with opacity
*/
#p3 {background-color:rgb(0,0,255);opacity:0.6;}  /* blue with opacity
*/
```

CSS Multiple Backgrounds

CSS allows you to add multiple background images for an element, through the `background-image` property.

The different background images are separated by commas, and the images are stacked on top of each other, where the first image is closest to the viewer.

The following example has two background images, the first image is a flower (aligned to the bottom and right) and the second image is a paper background (aligned to the top-left corner):

```
#example1 {
  background-image: url(img_flwr.gif), url(paper.gif);
  background-position: right bottom, left top;
  background-repeat: no-repeat, repeat;
}
```

Multiple background images can be specified using either the individual background properties (as above) or the `background` shorthand property.

The following example uses the `background` shorthand property (same result as example above):

```
#example1 {
  background: url(img_flwr.gif) right bottom no-repeat, url(paper.gif)
left top repeat;
}
```

CSS Background Size

The CSS `background-size` property allows you to specify the size of background images.

The size can be specified in lengths, percentages, or by using one of the two keywords: contain or cover.

The following example resizes a background image to much smaller than the original image (using pixels):

```
#div1 {
  background: url(img_flower.jpg);
  background-size: 100px 80px;
  background-repeat: no-repeat;
}
```

The two other possible values for `background-size` are `contain` and `cover`.

The `contain` keyword scales the background image to be as large as possible (but both its width and its height must fit inside the content area). As such, depending on the proportions of the background image and the background positioning area, there may be some areas of the background which are not covered by the background image.

The `cover` keyword scales the background image so that the content area is completely covered by the background image (both its width and height are equal to or exceed the content area). As such, some parts of the background image may not be visible in the background positioning area.

The following example illustrates the use of `contain` and `cover`:

```
#div1 {
  background: url(img_flower.jpg);
  background-size: contain;
  background-repeat: no-repeat;
}

#div2 {
  background: url(img_flower.jpg);
  background-size: cover;
  background-repeat: no-repeat;
}
```

Define Sizes of Multiple Background Images

The `background-size` property also accepts multiple values for background size (using a comma-separated list), when working with multiple backgrounds.

The following example has three background images specified, with different background-size value for each image:

```
#example1 {
  background: url(img_tree.gif) left top no-repeat, url(img_flwr.gif)
right bottom no-repeat, url(paper.gif) left top repeat;
  background-size: 50px, 130px, auto;
}
```

Full Size Background Image

Now we want to have a background image on a website that covers the entire browser window at all times.

The requirements are as follows:

- Fill the entire page with the image (no white space)

- Scale image as needed
- Center image on page
- Do not cause scrollbars

The following example shows how to do it; Use the <html> element (the <html> element is always at least the height of the browser window). Then set a fixed and centered background on it. Then adjust its size with the background-size property:

```
html {
  background: url(img_man.jpg) no-repeat center fixed;
  background-size: cover;
}
```

Hero Image

You could also use different background properties on a <div> to create a hero image (a large image with text), and place it where you want.

```
.hero-image {
  background: url(img_man.jpg) no-repeat center;
  background-size: cover;
  height: 500px;
  position: relative;
}
```

CSS background-origin Property

The CSS `background-origin` property specifies where the background image is positioned.

The property takes three different values:

- border-box - the background image starts from the upper left corner of the border
- padding-box - (default) the background image starts from the upper left corner of the padding edge
- content-box - the background image starts from the upper left corner of the content

The following example illustrates the `background-origin` property:

```
#example1 {
  border: 10px solid black;
  padding: 35px;
  background: url(img_flwr.gif);
  background-repeat: no-repeat;
  background-origin: content-box;
}
```

CSS background-clip Property

The CSS `background-clip` property specifies the painting area of the background.

The property takes three different values:

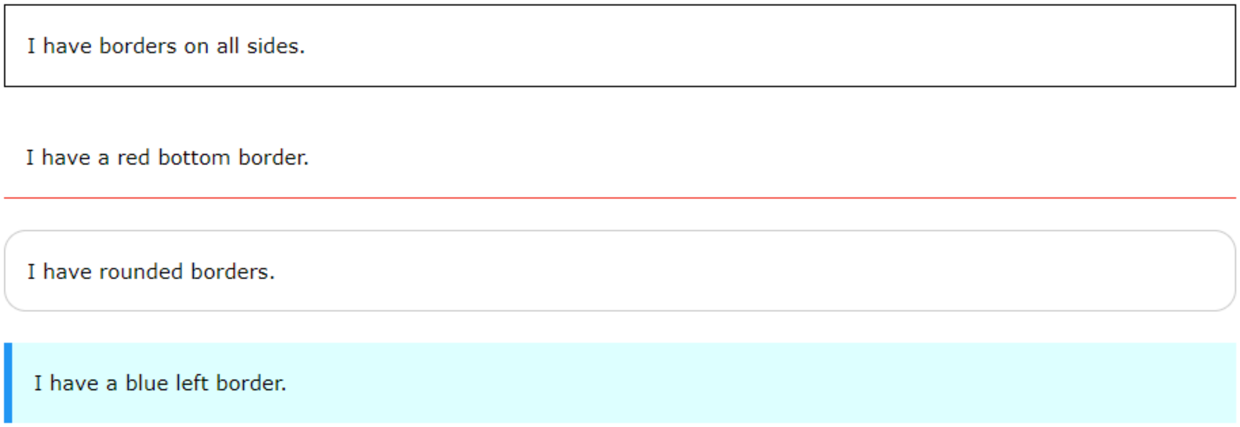
- border-box - (default) the background is painted to the outside edge of the border
- padding-box - the background is painted to the outside edge of the padding
- content-box - the background is painted within the content box

The following example illustrates the `background-clip` property:

```
#example1 {
  border: 10px dotted black;
  padding: 35px;
  background: yellow;
  background-clip: content-box;
}
```

CSS Borders

The CSS border properties allow you to specify the style, width, and color of an element's border.



I have borders on all sides.

I have a red bottom border.

I have rounded borders.

I have a blue left border.

CSS Border Style

The border-style property specifies what kind of border to display.

The following values are allowed:

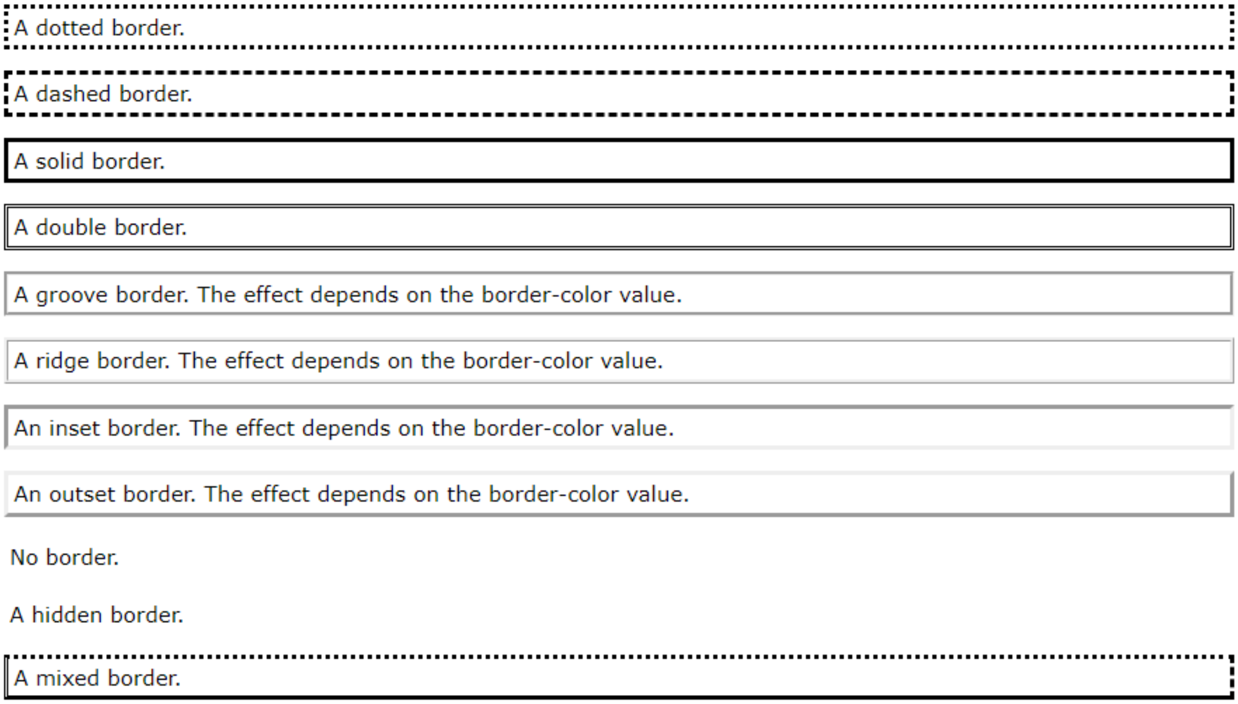
- dotted - Defines a dotted border
- dashed - Defines a dashed border
- solid - Defines a solid border
- double - Defines a double border
- groove - Defines a 3D grooved border. The effect depends on the border-color value
- ridge - Defines a 3D ridged border. The effect depends on the border-color value
- inset - Defines a 3D inset border. The effect depends on the border-color value
- outset - Defines a 3D outset border. The effect depends on the border-color value
- none - Defines no border
- hidden - Defines a hidden border

The border-style property can have from one to four values (for the top border, right border, bottom border, and the left border).

```
p.dotted {border-style: dotted;}
p.dashed {border-style: dashed;}
p.solid {border-style: solid;}
p.double {border-style: double;}
p.groove {border-style: groove;}
p.ridge {border-style: ridge;}
p.inset {border-style: inset;}
p.outset {border-style: outset;}
p.none {border-style: none;}
```

```
p.hidden {border-style: hidden;}
p.mix {border-style: dotted dashed solid double;}
```

Output



A dotted border.

A dashed border.

A solid border.

A double border.

A groove border. The effect depends on the border-color value.

A ridge border. The effect depends on the border-color value.

An inset border. The effect depends on the border-color value.

An outset border. The effect depends on the border-color value.

No border.

A hidden border.

A mixed border.

CSS Border - Shorthand Property

Like you saw in the previous page, there are many properties to consider when dealing with borders.

To shorten the code, it is also possible to specify all the individual border properties in one property.

The `border` property is a shorthand property for the following individual border properties:

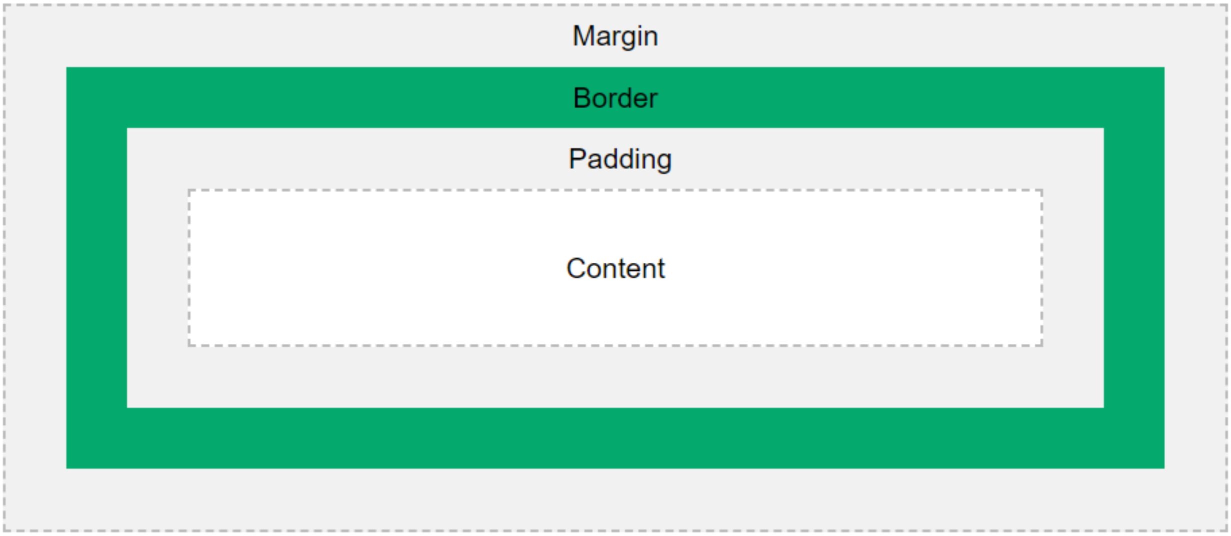
- `border-width`
- `border-style` (required)
- `border-color`

```
p {
  border: 5px solid red;
}
```

CSS Box Model

In CSS, the term "box model" is used when talking about design and layout.

The CSS box model is essentially a box that wraps around every HTML element. It consists of: margins, borders, padding, and the actual content. The image below illustrates the box model:

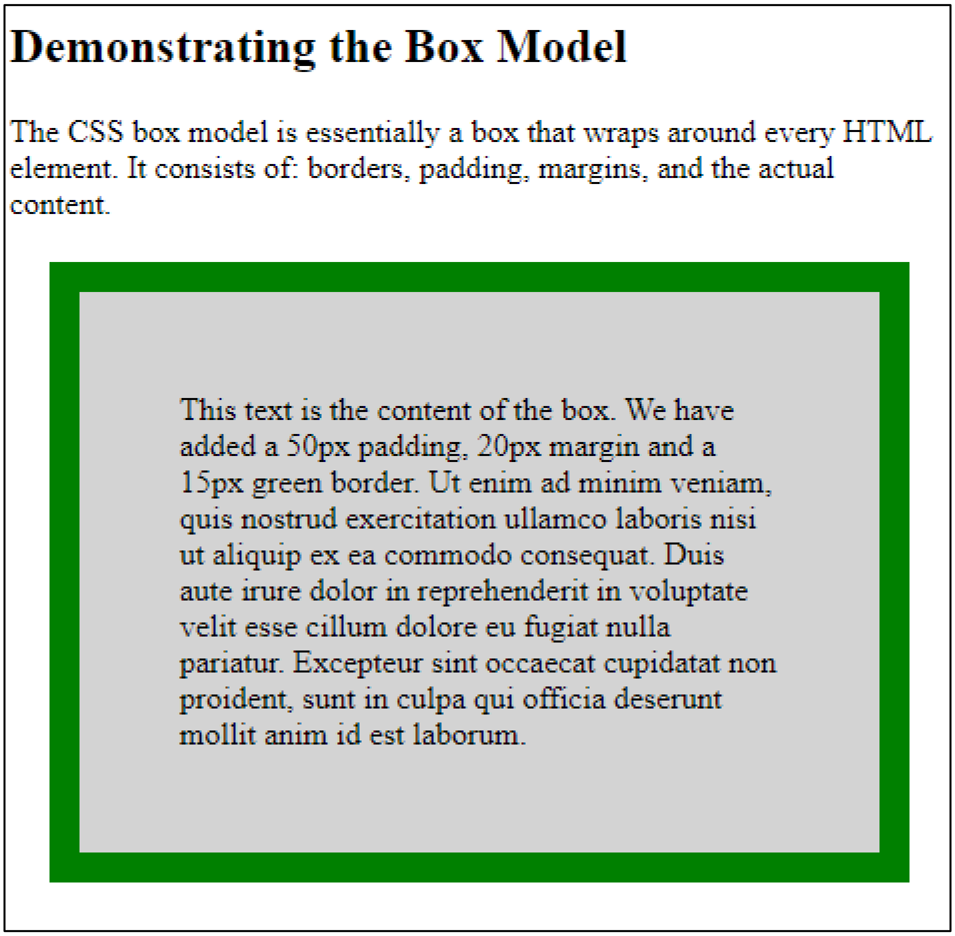


Explanation of the different parts:

- **Content** - The content of the box, where text and images appear
- **Padding** - Clears an area around the content. The padding is transparent
- **Border** - A border that goes around the padding and content
- **Margin** - Clears an area outside the border. The margin is transparent

```css
div {
  width: 300px;
  border: 15px solid green;
  padding: 50px;
  margin: 20px;
}
```

Output



Demonstrating the Box Model

The CSS box model is essentially a box that wraps around every HTML element. It consists of: borders, padding, margins, and the actual content.

This text is the content of the box. We have added a 50px padding, 20px margin and a 15px green border. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Width and Height of an Element

In order to set the width and height of an element correctly in all browsers, you need to know how the box model works.

```
div {
  width: 320px;
  padding: 10px;
  border: 5px solid gray;
  margin: 0;
}
```

Here is the calculation:

320px (width)
+ 20px (left + right padding)
+ 10px (left + right border)

+ 0px (left + right margin)
= 350px

The total width of an element should be calculated like this:

Total element width = width + left padding + right padding + left border + right border + left margin + right margin

The total height of an element should be calculated like this:

Total element height = height + top padding + bottom padding + top border + bottom border + top margin + bottom margin

CSS Margins

The CSS `margin` properties are used to create space around elements, outside of any defined borders.

With CSS, you have full control over the margins. There are properties for setting the margin for each side of an element (top, right, bottom, and left).

Margin - Individual Sides

CSS has properties for specifying the margin for each side of an element:

- `margin-top`
- `margin-right`
- `margin-bottom`
- `margin-left`

All the margin properties can have the following values:

- auto - the browser calculates the margin
- *length* - specifies a margin in px, pt, cm, etc.
- *%* - specifies a margin in % of the width of the containing element
- inherit - specifies that the margin should be inherited from the parent element

Tip: Negative values are allowed.

```
p {
  margin-top: 100px;
  margin-bottom: 100px;
  margin-right: 150px;
  margin-left: 80px;
}
```

Margin - Shorthand Property

To shorten the code, it is possible to specify all the margin properties in one property.

The `margin` property is a shorthand property for the following individual margin properties:

- `margin-top`
- `margin-right`
- `margin-bottom`
- `margin-left`

So, here is how it works:

If the `margin` property has four values:

- **margin: 25px 50px 75px 100px;**
    - top margin is 25px
    - right margin is 50px
    - bottom margin is 75px
    - left margin is 100px

If the `margin` property has three values:

- **margin: 25px 50px 75px;**
    - top margin is 25px
    - right and left margins are 50px
    - bottom margin is 75px

If the `margin` property has two values:

- **margin: 25px 50px;**
    - top and bottom margins are 25px
    - right and left margins are 50px

If the `margin` property has one value:

- **margin: 25px;**
    - all four margins are 25px

The auto Value

You can set the margin property to `auto` to horizontally center the element within its container.

The element will then take up the specified width, and the remaining space will be split equally between the left and right margins.

```
div {
  width: 300px;
  margin: auto;
  border: 1px solid red;
}
```

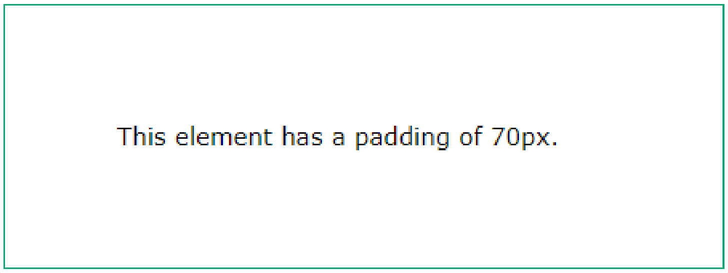
The inherit Value

This example lets the left margin of the <p class="ex1"> element be inherited from the parent element (<div>):

```
div {
  border: 1px solid red;
  margin-left: 100px;
}

p.ex1 {
  margin-left: inherit;
}
```

CSS Padding

Padding is used to create space around an element's content, inside of any defined borders.



This element has a padding of 70px.

The CSS `padding` properties are used to generate space around an element's content, inside of any defined borders.

With CSS, you have full control over the padding. There are properties for setting the padding for each side of an element (top, right, bottom, and left).

Padding - Individual Sides

CSS has properties for specifying the padding for each side of an element:

- `padding-top`
- `padding-right`
- `padding-bottom`
- `padding-left`

All the padding properties can have the following values:

- *length* - specifies a padding in px, pt, cm, etc.
- *%* - specifies a padding in % of the width of the containing element
- inherit - specifies that the padding should be inherited from the parent element

Note: Negative values are not allowed.

Padding - Shorthand Property

To shorten the code, it is possible to specify all the padding properties in one property.

The `padding` property is a shorthand property for the following individual padding properties:

- `padding-top`
- `padding-right`
- `padding-bottom`
- `padding-left`

So, here is how it works:

If the `padding` property has four values:

- **padding: 25px 50px 75px 100px;**
  - top padding is 25px
  - right padding is 50px
  - bottom padding is 75px
  - left padding is 100px

If the `padding` property has three values:

- **padding: 25px 50px 75px;**
  - top padding is 25px
  - right and left paddings are 50px
  - bottom padding is 75px

If the `padding` property has two values:

- **padding: 25px 50px;**
  - top and bottom paddings are 25px
  - right and left paddings are 50px

If the `padding` property has one value:

- **padding: 25px;**
  - all four paddings are 25px

Padding and Element Width

The CSS `width` property specifies the width of the element's content area. The content area is the portion inside the padding, border, and margin of an element (the box model).

So, if an element has a specified width, the padding added to that element will be added to the total width of the element. This is often an undesirable result.

div {
  width: 300px;
  padding: 25px;
}
	Here, the <div> element is given a width of 300px. However, the actual width of the <div> element will be 350px (300px + 25px of left padding + 25px of right padding):

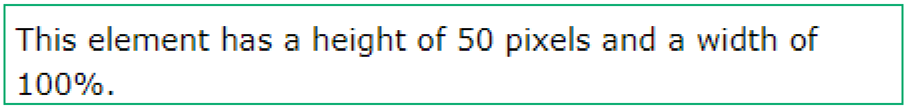
To keep the width at 300px, no matter the amount of padding, you can use the `box-sizing` property. This causes the element to maintain its actual width; if you increase the padding, the available content space will decrease.

div {
  width: 300px;
  padding: 25px;
  box-sizing: border-box;
}
	Use the box-sizing property to keep the width at 300px, no matter the amount of padding.

CSS Height, Width and Max-width

The CSS `height` and `width` properties are used to set the height and width of an element.

The CSS `max-width` property is used to set the maximum width of an element.



This element has a height of 50 pixels and a width of 100%.

CSS Setting height and width

The `height` and `width` properties are used to set the height and width of an element.

The height and width properties do not include padding, borders, or margins. It sets the height/width of the area inside the padding, border, and margin of the element.

CSS height and width Values

The `height` and `width` properties may have the following values:

- `auto` - This is default. The browser calculates the height and width
- `length` - Defines the height/width in px, cm, etc.
- `%` - Defines the height/width in percent of the containing block
- `initial` - Sets the height/width to its default value
- `inherit` - The height/width will be inherited from its parent value

```
div {
  height: 200px;
  width: 50%;
  background-color: powderblue;
}
```

Setting max-width

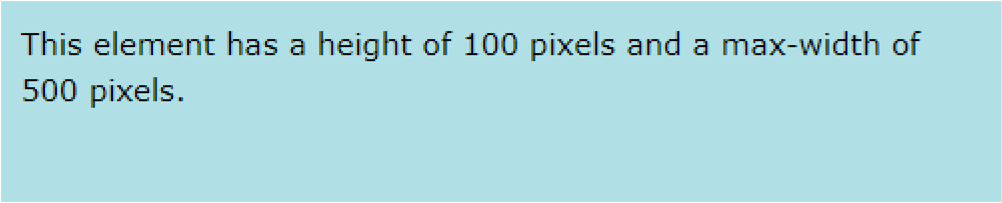
The `max-width` property is used to set the maximum width of an element.

The `max-width` can be specified in *length values*, like px, cm, etc., or in percent (%) of the containing block, or set to none (this is default. Means that there is no maximum width).

The problem with the <div> above occurs when the browser window is smaller than the width of the element (500px). The browser then adds a horizontal scrollbar to the page.

Using max-width instead, in this situation, will improve the browser's handling of small windows.

Tip: Drag the browser window to smaller than 500px wide, to see the difference between the two divs!



This element has a height of 100 pixels and a max-width of 500 pixels.

Note: If you for some reason use both the width property and the max-width property on the same element, and the value of the width property is larger than the max-width property; the max-width property will be used (and the width property will be ignored).

```
div {
  max-width: 500px;
  height: 100px;
  background-color: powderblue;
}
```

The position Property

The position property specifies the type of positioning method used for an element.

There are five different position values:

- static
- relative
- fixed
- absolute
- sticky

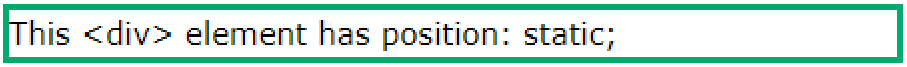
Elements are then positioned using the top, bottom, left, and right properties. However, these properties will not work unless the position property is set first. They also work differently depending on the position value.

position: static;

HTML elements are positioned static by default.

Static positioned elements are not affected by the top, bottom, left, and right properties.

An element with `position: static;` is not positioned in any special way; it is always positioned according to the normal flow of the page:



This <div> element has position: static;

Here is the CSS that is used:

```
div.static {
  position: static;
  border: 3px solid #73AD21;
}
```

position: relative;

An element with `position: relative;` is positioned relative to its normal position.

Setting the top, right, bottom, and left properties of a relatively-positioned element will cause it to be adjusted away from its normal position. Other content will not be adjusted to fit into any gap left by the element.

```
div.relative {
  position: relative;
  left: 30px;
  border: 3px solid #73AD21;
}
```

position: fixed;

An element with `position: fixed;` is positioned relative to the viewport, which means it always stays in the same place even if the page is scrolled. The top, right, bottom, and left properties are used to position the element.

A fixed element does not leave a gap in the page where it would normally have been located.

Notice the fixed element in the lower-right corner of the page. Here is the CSS that is used:

```
div.fixed {
  position: fixed;
  bottom: 0;
  right: 0;
  width: 300px;
  border: 3px solid #73AD21;
}
```

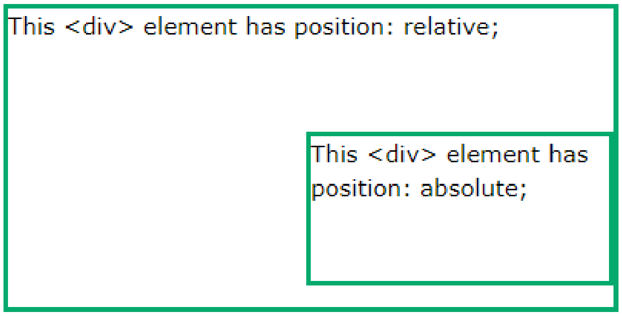
position: absolute;

An element with `position: absolute;` is positioned relative to the nearest positioned ancestor (instead of positioned relative to the viewport, like fixed).

However; if an absolute positioned element has no positioned ancestors, it uses the document body, and moves along with page scrolling.

Note: Absolute positioned elements are removed from the normal flow, and can overlap elements.

Here is a simple example:



This <div> element has position: relative;

This <div> element has position: absolute;

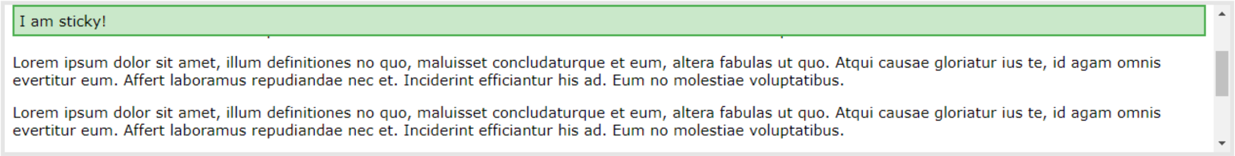
```
div.relative {
  position: relative;
  width: 400px;
  height: 200px;
  border: 3px solid #73AD21;
}
```

```
div.absolute {
  position: absolute;
  top: 80px;
  right: 0;
  width: 200px;
  height: 100px;
  border: 3px solid #73AD21;
}
```

position: sticky;

An element with `position: sticky;` is positioned based on the user's scroll position.

A sticky element toggles between `relative` and `fixed`, depending on the scroll position. It is positioned relative until a given offset position is met in the viewport - then it "sticks" in place (like position:fixed).



I am sticky!

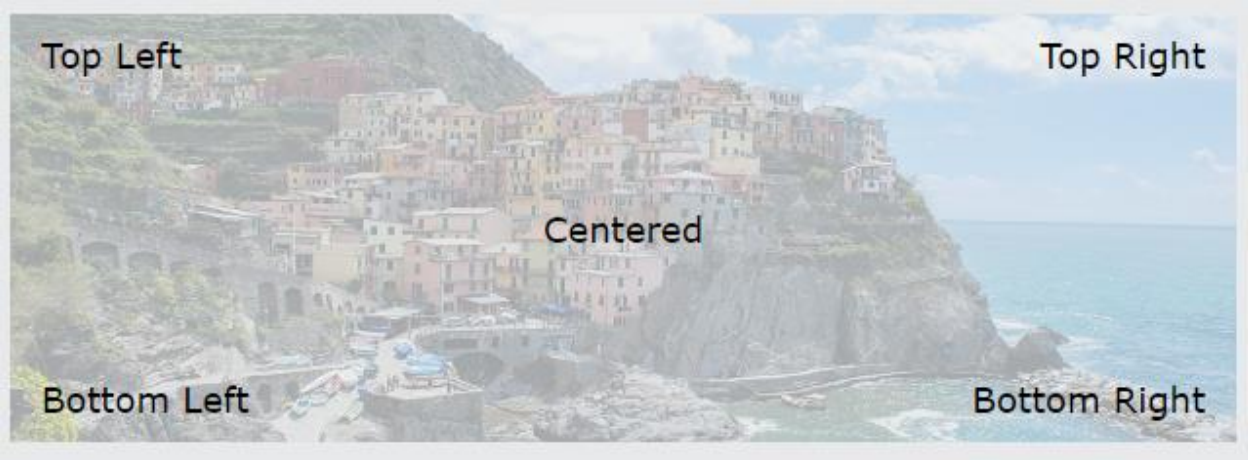
Lorem ipsum dolor sit amet, illum definitiones no quo, maluisset concludaturque et eum, altera fabulas ut quo. Atqui causae gloriatur ius te, id agam omnis evertitur eum. Affert laboramus repudiandae nec et. Inciderint efficiantur his ad. Eum no molestiae voluptatibus.

Lorem ipsum dolor sit amet, illum definitiones no quo, maluisset concludaturque et eum, altera fabulas ut quo. Atqui causae gloriatur ius te, id agam omnis evertitur eum. Affert laboramus repudiandae nec et. Inciderint efficiantur his ad. Eum no molestiae voluptatibus.

```
div.sticky {
  position: -webkit-sticky; /* Safari */
  position: sticky;
  top: 0;
  background-color: green;
  border: 2px solid #4CAF50;
}
```

Positioning Text In an Image

How to position text over an image:



CSS Layout and Clear

The CSS `float` property specifies how an element should float.

The CSS `clear` property specifies what elements can float beside the cleared element and on which side.

The float Property

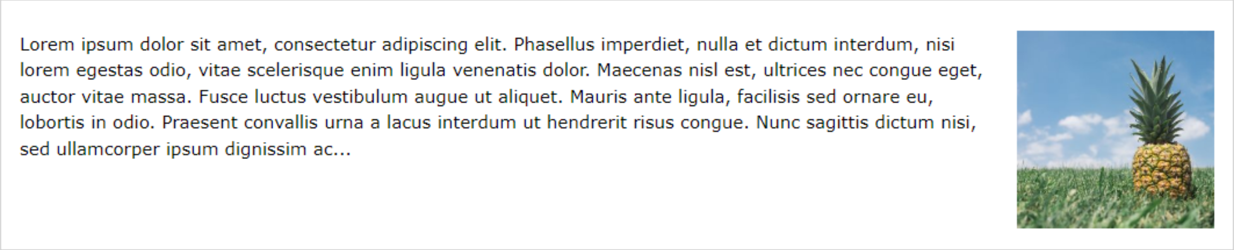
The `float` property is used for positioning and formatting content e.g. let an image float left to the text in a container.

The `float` property can have one of the following values:

- `left` - The element floats to the left of its container
- `right` - The element floats to the right of its container
- `none` - The element does not float (will be displayed just where it occurs in the text). This is default
- `inherit` - The element inherits the float value of its parent

Example - float: right;

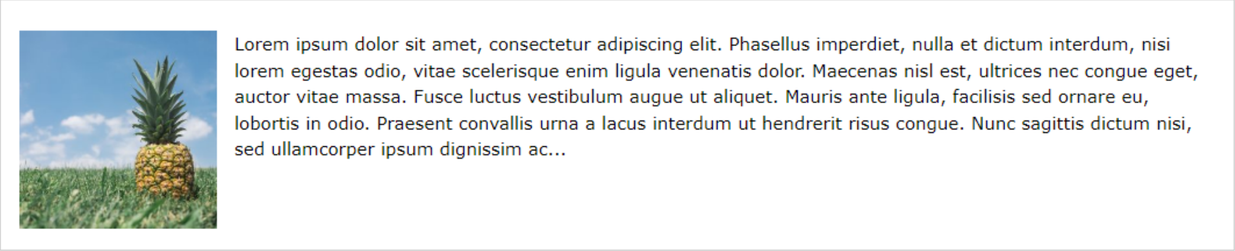
The following example specifies that an image should float to the **right** in a text:



Lorem ipsum dolor sit amet, consectetur adipiscing elit. Phasellus imperdiet, nulla et dictum interdum, nisi lorem egestas odio, vitae scelerisque enim ligula venenatis dolor. Maecenas nisl est, ultrices nec congue eget, auctor vitae massa. Fusce luctus vestibulum augue ut aliquet. Mauris ante ligula, facilisis sed ornare eu, lobortis in odio. Praesent convallis urna a lacus interdum ut hendrerit risus congue. Nunc sagittis dictum nisi, sed ullamcorper ipsum dignissim ac...

Example - float: left;

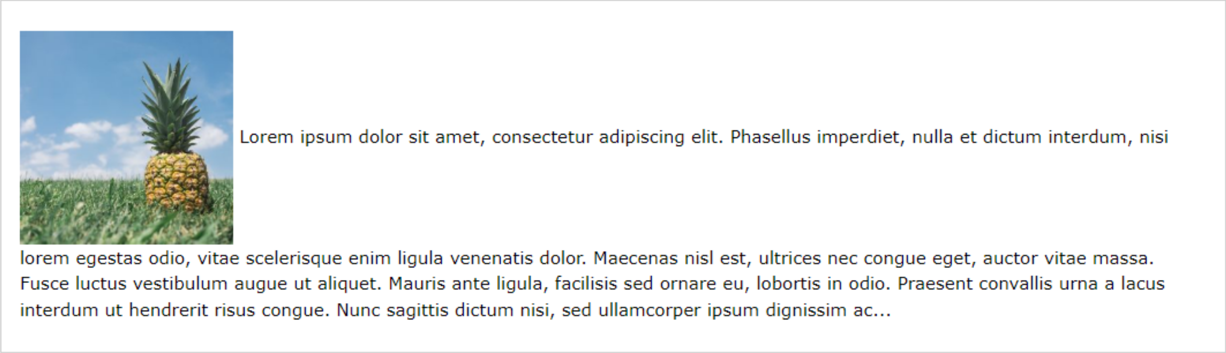
The following example specifies that an image should float to the **left** in a text:



Lorem ipsum dolor sit amet, consectetur adipiscing elit. Phasellus imperdiet, nulla et dictum interdum, nisi lorem egestas odio, vitae scelerisque enim ligula venenatis dolor. Maecenas nisl est, ultrices nec congue eget, auctor vitae massa. Fusce luctus vestibulum augue ut aliquet. Mauris ante ligula, facilisis sed ornare eu, lobortis in odio. Praesent convallis urna a lacus interdum ut hendrerit risus congue. Nunc sagittis dictum nisi, sed ullamcorper ipsum dignissim ac...

Example - No float

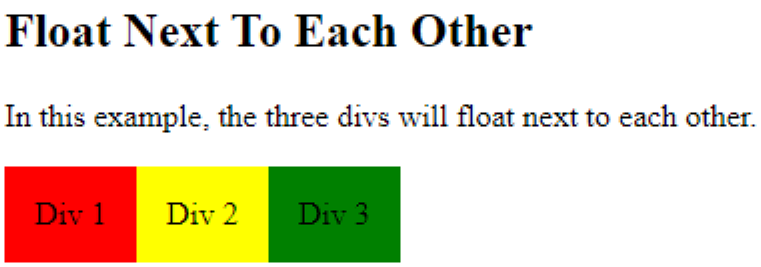
In the following example the image will be displayed just where it occurs in the text (float: none;):



Lorem ipsum dolor sit amet, consectetur adipiscing elit. Phasellus imperdiet, nulla et dictum interdum, nisi lorem egestas odio, vitae scelerisque enim ligula venenatis dolor. Maecenas nisl est, ultrices nec congue eget, auctor vitae massa. Fusce luctus vestibulum augue ut aliquet. Mauris ante ligula, facilisis sed ornare eu, lobortis in odio. Praesent convallis urna a lacus interdum ut hendrerit risus congue. Nunc sagittis dictum nisi, sed ullamcorper ipsum dignissim ac...

Example - Float Next To Each Other

Normally div elements will be displayed on top of each other. However, if we use `float: left` we can let elements float next to each other:



Float Next To Each Other

In this example, the three divs will float next to each other.

Code

```
div {
  float: left;
  padding: 15px;
}

.div1 {
  background: red;
}

.div2 {
  background: yellow;
}

.div3 {
  background: green;
}
```

CSS Specificity

If there are two or more CSS rules that point to the same element, the selector with the highest specificity value will "win", and its style declaration will be applied to that HTML element.

Think of specificity as a score/rank that determines which style declaration is ultimately applied to an element.

Look at the following examples:

```
<html>
<head>
  <style>
    p {color: red;}
  </style>
</head>
<body>

<p>Hello World!</p>
```

In this example, we have used the "p" element as selector, and specified a red color for this element. The text will be red:

```
</body>
</html>
```

Now, look at example 2:

```
<html>
<head>
  <style>
    .test {color: green;}
    p {color: red;}
  </style>
</head>
<body>

<p class="test">Hello World!</p>

</body>
</html>
```

In this example, we have added a class selector (named "test"), and specified a green color for this class. The text will now be green (even though we have specified a red color for the element selector "p"). This is because the class selector is given higher priority:

Now, look at example 3:

```
<html>
<head>
  <style>
    #demo {color: blue;}
    .test {color: green;}
    p {color: red;}
  </style>
</head>
<body>

<p id="demo" class="test">Hello
World!</p>

</body>
</html>
```

In this example, we have added the id selector (named "demo"). The text will now be blue, because the id selector is given higher priority:

Now, look at example 4:

```
<html>
<head>
  <style>
    #demo {color: blue;}
    .test {color: green;}
    p {color: red;}
  </style>
</head>
<body>

<p id="demo" class="test" style="color:
pink;">Hello World!</p>

</body>
</html>
```

In this example, we have added an inline style for the "p" element. The text will now be pink, because the inline style is given the highest priority:

Specificity Hierarchy

Every CSS selector has its place in the specificity hierarchy.

There are four categories which define the specificity level of a selector:

1. **Inline styles** - Example: <h1 style="color: pink;">
2. **IDs** - Example: #navbar
3. **Classes, pseudo-classes, attribute selectors** - Example: .test, :hover, [href]
4. **Elements and pseudo-elements** - Example: h1, ::before

How to Calculate Specificity?

Memorize how to calculate specificity!

Start at 0, add 100 for each ID value, add 10 for each class value (or pseudo-class or attribute selector), add 1 for each element selector or pseudo-element.

Note: Inline style gets a specificity value of 1000, and is always given the highest priority!

Note 2: There is one exception to this rule: if you use the !important rule, it will even override inline styles!

The table below shows some examples on how to calculate specificity values:

Selector	**Specificity Value**	**Calculation**
p	1	1
p.test	11	1 + 10
p#demo	101	1 + 100
<p style="color: pink;">	1000	1000
#demo	100	100
.test	10	10
p.test1.test2	21	1 + 10 + 10
#navbar p#demo	201	100 + 1 + 100
*	0	0 (the universal selector is ignored)

The selector with the highest specificity value will win and take effect!

3.1.2. Responsive Web Design

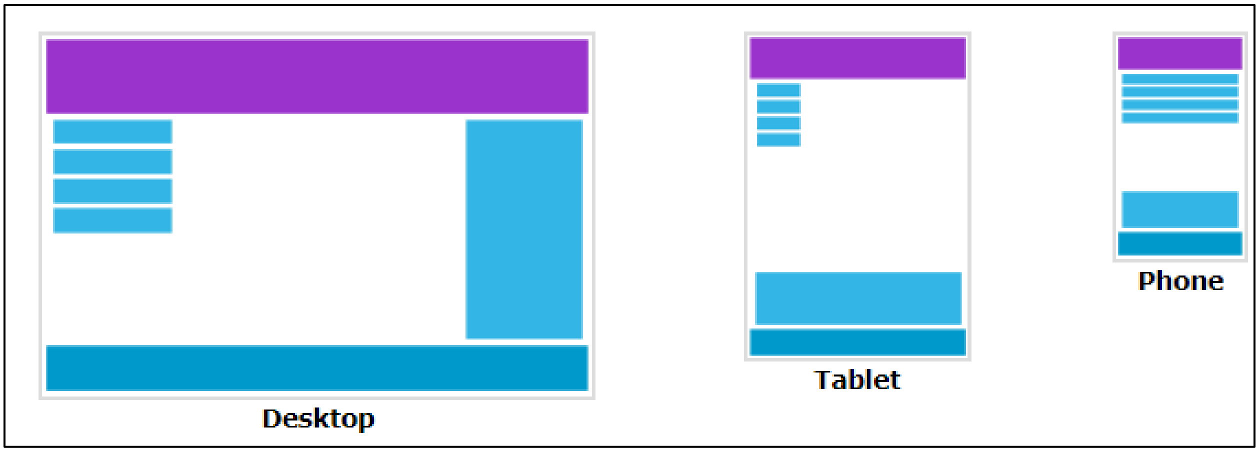
Responsive web design makes your web page look good on all devices.

Responsive web design uses only HTML and CSS.

Responsive web design is not a program or a JavaScript.

Web pages can be viewed using many different devices: desktops, tablets, and phones. Your web page should look good, and be easy to use, regardless of the device.

Web pages should not leave out information to fit smaller devices, but rather adapt its content to fit any device:



It is called responsive web design when you use CSS and HTML to resize, hide, shrink, enlarge, or move the content to make it look good on any screen.

Responsive Web Design - The Viewport

The viewport is the user's visible area of a web page.

The viewport varies with the device, and will be smaller on a mobile phone than on a computer screen.

Before tablets and mobile phones, web pages were designed only for computer screens, and it was common for web pages to have a static design and a fixed size.

Then, when we started surfing the internet using tablets and mobile phones, fixed size web pages were too large to fit the viewport. To fix this, browsers on those devices scaled down the entire web page to fit the screen.

This was not perfect!! But a quick fix.

HTML5 introduced a method to let web designers take control over the viewport, through the `<meta>` tag.

You should include the following `<meta>` viewport element in all your web pages:

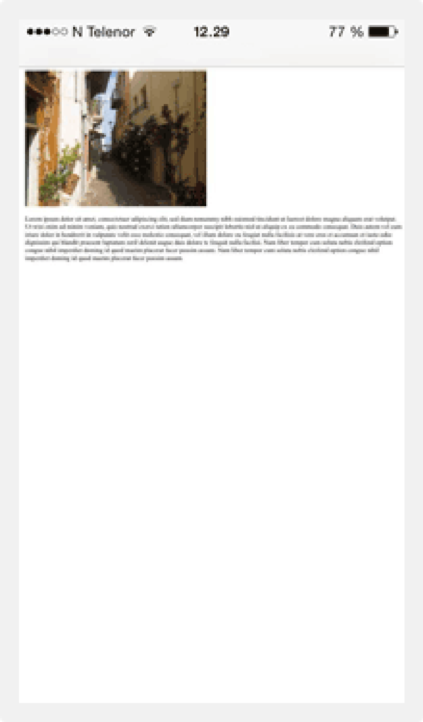
```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```

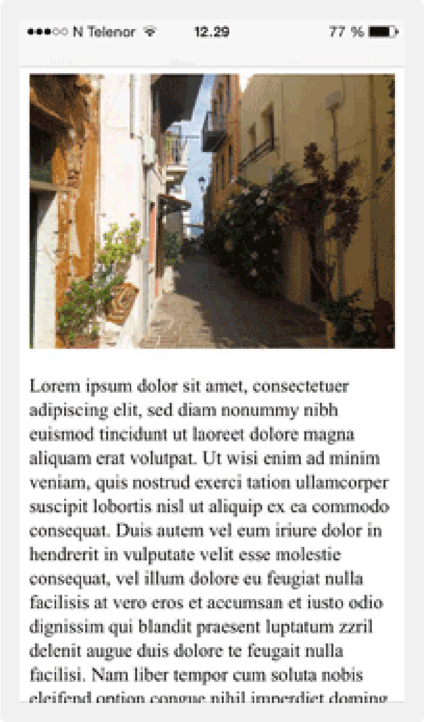
This gives the browser instructions on how to control the page's dimensions and scaling.

The `width=device-width` part sets the width of the page to follow the screen-width of the device (which will vary depending on the device).

The `initial-scale=1.0` part sets the initial zoom level when the page is first loaded by the browser.

Here is an example of a web page *without* the viewport meta tag, and the same web page *with* the viewport meta tag:





Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat. Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat. Duis autem vel eum iriure dolor in hendrerit in vulputate velit esse molestie consequat, vel illum dolore eu feugiat nulla facilisis at vero eros et accumsan et iusto odio dignissim qui blandit praesent luptatum zzril delenit augue duis dolore te feugait nulla facilisi. Nam liber tempor cum soluta nobis eleifend option congue nihil imperdiet doming

Without the viewport meta tag

With the viewport meta tag

Size Content to The Viewport

Users are used to scroll websites vertically on both desktop and mobile devices - but not horizontally!

So, if the user is forced to scroll horizontally, or zoom out, to see the whole web page it results in a poor user experience.

Some additional rules to follow:

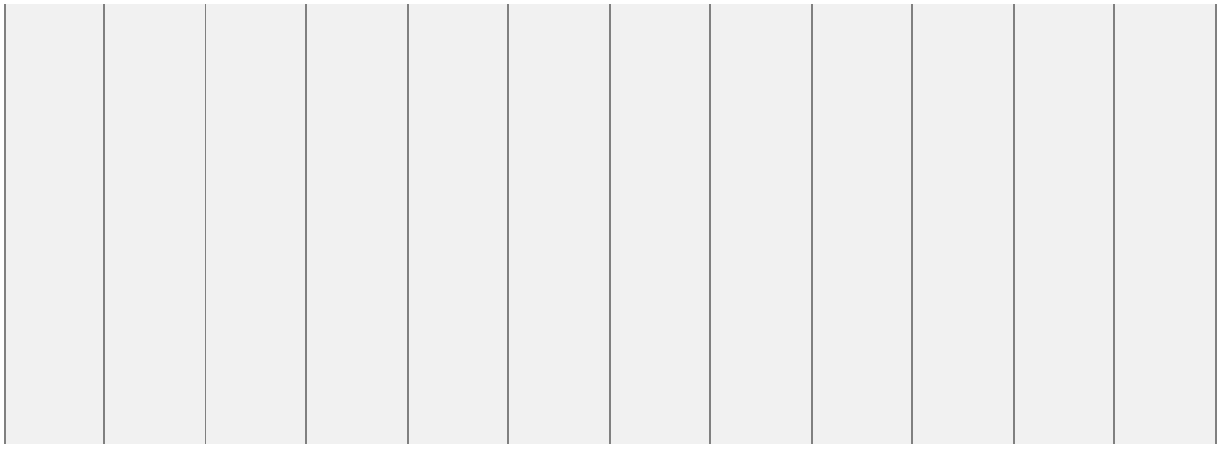
1. Do NOT use large fixed width elements - For example, if an image is displayed at a width wider than the viewport it can cause the viewport to scroll horizontally. Remember to adjust this content to fit within the width of the viewport.

2. Do NOT let the content rely on a particular viewport width to render well - Since screen dimensions and width in CSS pixels vary widely between devices, content should not rely on a particular viewport width to render well.

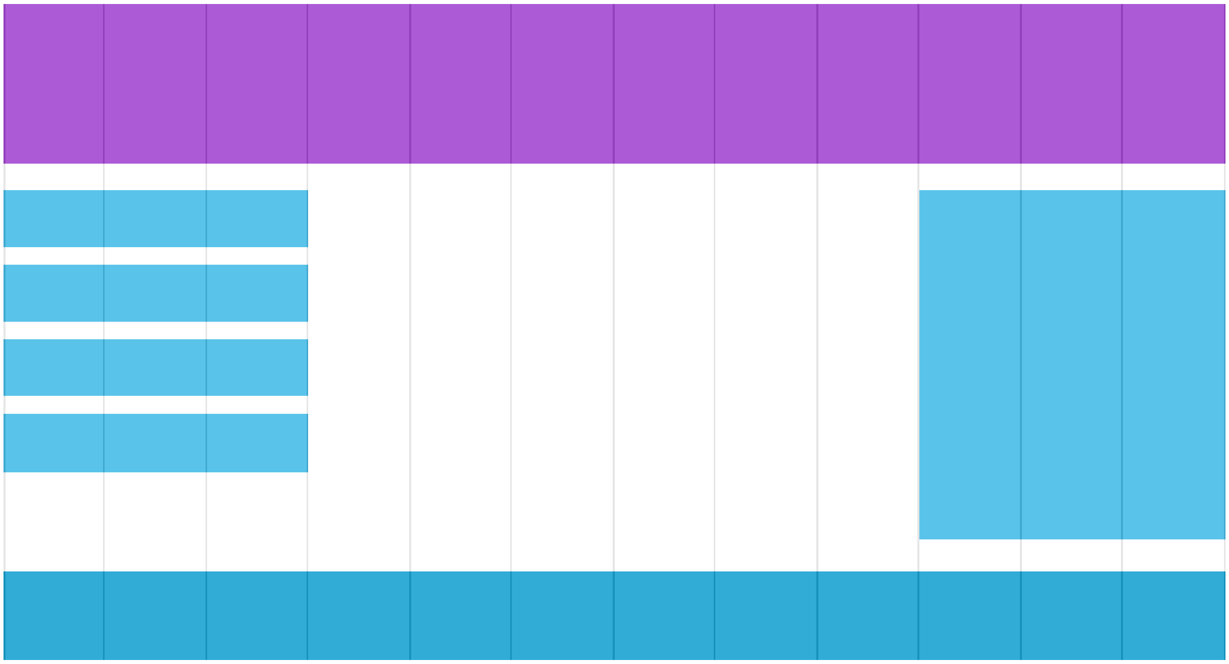
3. Use CSS media queries to apply different styling for small and large screens - Setting large absolute CSS widths for page elements will cause the element to be too wide for the viewport on a smaller device. Instead, consider using relative width values, such as width: 100%. Also, be careful of using large absolute positioning values. It may cause the element to fall outside the viewport on small devices.

Responsive Web Design - Grid-View

Many web pages are based on a grid-view, which means that the page is divided into columns:



Using a grid-view is very helpful when designing web pages. It makes it easier to place elements on the page.



A responsive grid-view often has 12 columns, and has a total width of 100%, and will shrink and expand as you resize the browser window.

Building a Responsive Grid-View

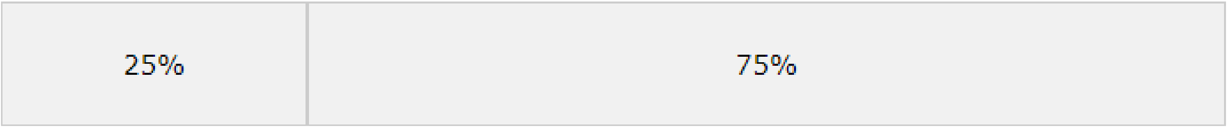
Lets start building a responsive grid-view.

First ensure that all HTML elements have the `box-sizing` property set to `border-box`. This makes sure that the padding and border are included in the total width and height of the elements.

Add the following code in your CSS

```
* {
  box-sizing: border-box;
}
```

The following example shows a simple responsive web page, with two columns:



```
.menu {
  width: 25%;
  float: left;
}
.main {
  width: 75%;
  float: left;
}
```

The example above is fine if the web page only contains two columns.

However, we want to use a responsive grid-view with 12 columns, to have more control over the web page.

First we must calculate the percentage for one column: 100% / 12 columns = 8.33%.

Then we make one class for each of the 12 columns, `class="col-"` and a number defining how many columns the section should span:

```
.col-1 {width: 8.33%;}
.col-2 {width: 16.66%;}
.col-3 {width: 25%;}
.col-4 {width: 33.33%;}
.col-5 {width: 41.66%;}
.col-6 {width: 50%;}
.col-7 {width: 58.33%;}
.col-8 {width: 66.66%;}
.col-9 {width: 75%;}
.col-10 {width: 83.33%;}
.col-11 {width: 91.66%;}
.col-12 {width: 100%;}
```

All these columns should be floating to the left, and have a padding of 15px:

```
[class*="col-"] {
  float: left;
  padding: 15px;
  border: 1px solid red;
}
```

Each row should be wrapped in a `<div>`. The number of columns inside a row should always add up to 12:

```
<div class="row">
  <div class="col-3">...</div> <!-- 25% -->
  <div class="col-9">...</div> <!-- 75% -->
</div>
```

The columns inside a row are all floating to the left, and are therefore taken out of the flow of the page, and other elements will be placed as if the columns do not exist. To prevent this, we will add a style that clears the flow:

```
.row::after {
  content: "";
  clear: both;
  display: table;
}
```

We also want to add some styles and colors to make it look better:

```
html {
  font-family: "Lucida Sans", sans-serif;
}

.header {
  background-color: #9933cc;
  color: #ffffff;
  padding: 15px;
}

.menu ul {
  list-style-type: none;
  margin: 0;
  padding: 0;
}

.menu li {
  padding: 8px;
  margin-bottom: 7px;
  background-color :#33b5e5;
  color: #ffffff;
  box-shadow: 0 1px 3px rgba(0,0,0,0.12), 0 1px 2px rgba(0,0,0,0.24);
}

.menu li:hover {
  background-color: #0099cc;
}
```

Responsive Web Design - Media Queries

Media query is a CSS technique introduced in CSS3.

It uses the `@media` rule to include a block of CSS properties only if a certain condition is true.

```
@media only screen and (max-width: 600px) {
  body {
    background-color: lightblue;
  }
}
```

Add a Breakpoint

Earlier in this tutorial we made a web page with rows and columns, and it was responsive, but it did not look good on a small screen.

Media queries can help with that. We can add a breakpoint where certain parts of the design will behave differently on each side of the breakpoint.

```
/*    For    desktop:    */
.col-1 {width: 8.33%;}
.col-2 {width: 16.66%;}
.col-3 {width: 25%;}
.col-4 {width: 33.33%;}
.col-5 {width: 41.66%;}
.col-6 {width: 50%;}
.col-7 {width: 58.33%;}
.col-8 {width: 66.66%;}
.col-9 {width: 75%;}
.col-10 {width: 83.33%;}
.col-11 {width: 91.66%;}
.col-12 {width: 100%;}

@media  only  screen  and
(max-width:        768px) {
  /* For mobile phones: */
         [class*="col-"] {
              width: 100%;
  }
}
```

When the screen (browser window) gets smaller than 768px, each column should have a width of 100%:

Always Design for Mobile First

Mobile First means designing for mobile before designing for desktop or any other device (This will make the page display faster on smaller devices).

This means that we must make some changes in our CSS.

Instead of changing styles when the width gets *smaller* than 768px, we should change the design when the width gets *larger* than 768px. This will make our design Mobile First:

```
/*              For              mobile              phones:              */
[class*="col-"] {
                                                                  width: 100%;
}
```

```
@media only screen and (min-width: 768px) {
  /* For desktop: */
  .col-1 {width: 8.33%;}
  .col-2 {width: 16.66%;}
  .col-3 {width: 25%;}
  .col-4 {width: 33.33%;}
  .col-5 {width: 41.66%;}
  .col-6 {width: 50%;}
  .col-7 {width: 58.33%;}
  .col-8 {width: 66.66%;}
  .col-9 {width: 75%;}
  .col-10 {width: 83.33%;}
  .col-11 {width: 91.66%;}
  .col-12 {width: 100%;}
}
```

Another Breakpoint

You can add as many breakpoints as you like.

We will also insert a breakpoint between tablets and mobile phones.

```
/* For mobile phones: */
[class*="col-"] {
  width: 100%;
}

@media only screen and (min-width: 600px) {
  /* For tablets: */
  .col-s-1 {width: 8.33%;}
  .col-s-2 {width: 16.66%;}
  .col-s-3 {width: 25%;}
  .col-s-4 {width: 33.33%;}
  .col-s-5 {width: 41.66%;}
  .col-s-6 {width: 50%;}
  .col-s-7 {width: 58.33%;}
  .col-s-8 {width: 66.66%;}
  .col-s-9 {width: 75%;}
  .col-s-10 {width: 83.33%;}
  .col-s-11 {width: 91.66%;}
  .col-s-12 {width: 100%;}
}
```

Note that the two sets of classes are almost identical, the only difference is the name (`col-` and `col-s-`):

```
@media only screen and (min-width:
768px) {
  /*      For      desktop:      */
             .col-1 {width: 8.33%;}
            .col-2 {width: 16.66%;}
               .col-3 {width: 25%;}
            .col-4 {width: 33.33%;}
            .col-5 {width: 41.66%;}
               .col-6 {width: 50%;}
            .col-7 {width: 58.33%;}
            .col-8 {width: 66.66%;}
               .col-9 {width: 75%;}
           .col-10 {width: 83.33%;}
           .col-11 {width: 91.66%;}
             .col-12 {width: 100%;}
}
```

Typical Device Breakpoints

There are tons of screens and devices with different heights and widths, so it is hard to create an exact breakpoint for each device. To keep things simple you could target five groups:

```
/* Extra small devices (phones, 600px and down) */
@media only screen and (max-width: 600px) {...}

/* Small devices (portrait tablets and large phones, 600px and up) */
@media only screen and (min-width: 600px) {...}

/* Medium devices (landscape tablets, 768px and up) */
@media only screen and (min-width: 768px) {...}

/* Large devices (laptops/desktops, 992px and up) */
@media only screen and (min-width: 992px) {...}

/* Extra large devices (large laptops and desktops, 1200px and up) */
@media only screen and (min-width: 1200px) {...}
```

Orientation: Portrait / Landscape

Media queries can also be used to change layout of a page depending on the orientation of the browser.

You can have a set of CSS properties that will only apply when the browser window is wider than its height, a so called "Landscape" orientation:

```
@media only screen and (orientation: landscape) {
  body {
    background-color: lightblue;
  }
}
```

Hide Elements With Media Queries

Another common use of media queries, is to hide elements on different screen sizes:

```
/* If the screen size is 600px wide or less, hide the element */
@media only screen and (max-width: 600px) {
  div.example {
    display: none;
  }
}
```

Change Font Size With Media Queries

You can also use media queries to change the font size of an element on different screen sizes:

```
/* If the screen size is 601px or more, set the font-size of <div> to
80px */
@media only screen and (min-width: 601px) {
  div.example {
    font-size: 80px;
  }
}

/* If the screen size is 600px or less, set the font-size of <div> to
30px */
@media only screen and (max-width: 600px) {
  div.example {
    font-size: 30px;
  }
```

CSS Flexbox

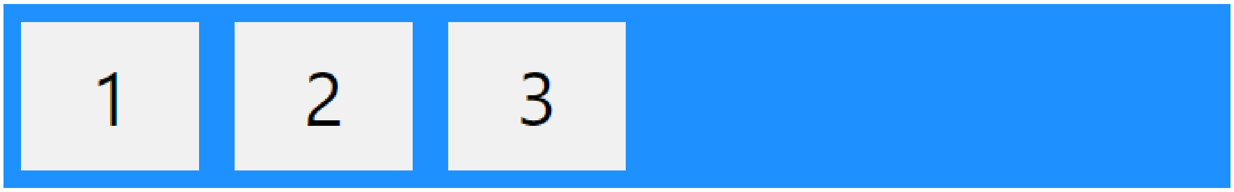
Before the Flexbox Layout module, there were four layout modes:

- Block, for sections in a webpage
- Inline, for text
- Table, for two-dimensional table data
- Positioned, for explicit position of an element

The Flexible Box Layout Module, makes it easier to design flexible responsive layout structure without using float or positioning.

Flexbox Elements

To start using the Flexbox model, you need to first define a flex container.



```
<div class="flex-container">
  <div>1</div>
  <div>2</div>
  <div>3</div>
</div>
```

The flex container becomes flexible by setting the `display` property to `flex`:

```
.flex-container {
  display: flex;
}
```

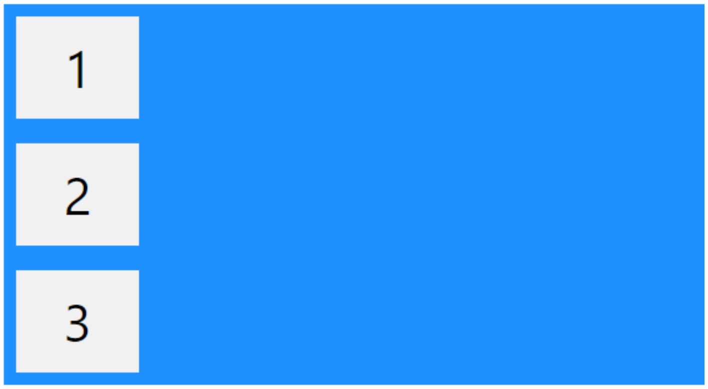
The flex container properties are:

- `flex-direction`

- flex-wrap
- flex-flow
- justify-content
- align-items
- align-content

The flex-direction Property

The flex-direction property defines in which direction the container wants to stack the flex items.



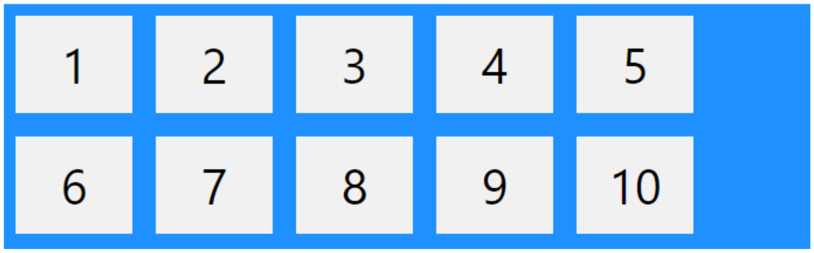
.flex-container {
  display: flex;
  flex-direction: column;
}
	.flex-container {
	  display: flex;
	  flex-direction: column-reverse;
	}

.flex-container {
  display: flex;
  flex-direction: row;
}
	.flex-container {
	  display: flex;
	  flex-direction: row-reverse;
	}

The flex-wrap Property

The flex-wrap property specifies whether the flex items should wrap or not.

The examples below have 12 flex items, to better demonstrate the flex-wrap property.



.flex-container {
  display: flex;
  flex-wrap: wrap;
}
	.flex-container {
	  display: flex;
	  flex-wrap: nowrap;
	}

.flex-container {
  display: flex;
  flex-wrap: wrap-reverse;
}	

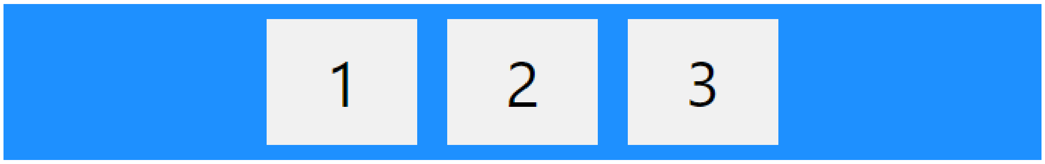
The flex-flow Property

The `flex-flow` property is a shorthand property for setting both the `flex-direction` and `flex-wrap` properties.

```
.flex-container {
  display: flex;
  flex-flow: row wrap;
}
```

The justify-content Property

The `justify-content` property is used to align the flex items:



.flex-container {
  display: flex;
  justify-content: flex-start;
}
	The `flex-start` value aligns the flex items at the beginning of the container (this is default):

.flex-container {
  display: flex;
  justify-content: flex-end;
}
	The `flex-end` value aligns the flex items at the end of the container:

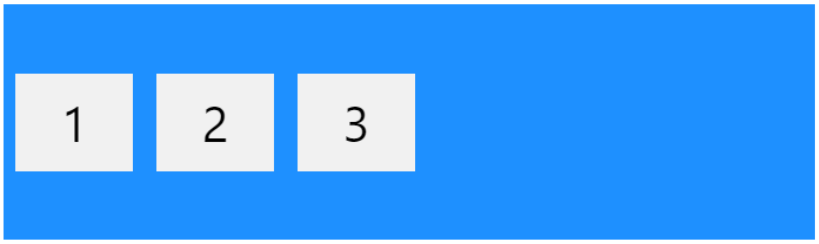
.flex-container {
  display: flex;
  justify-content: center;
}
	The `center` value aligns the flex items at the center of the container:

`.flex-container {`
`  display: flex;`
`  justify-content: space-around;`
`}`
	The `space-around` value displays the flex items with space before, between, and after the lines:

`.flex-container {`
`  display: flex;`
`  justify-content: space-between;`
`}`
	The `space-between` value displays the flex items with space between the lines:

The align-items Property

The `align-items` property is used to align the flex items.



In these examples we use a 200 pixels high container, to better demonstrate the `align-items` property.

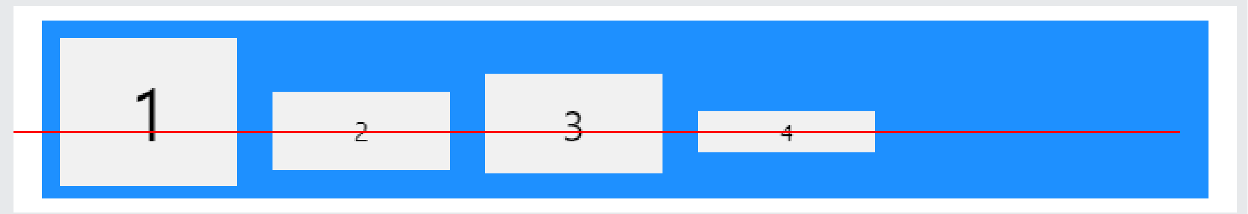
`.flex-container {`
`  display: flex;`
`  height: 200px;`
`  align-items: center;`
`}`
	The `center` value aligns the flex items in the middle of the container:

`.flex-container {`
`  display: flex;`
`  height: 200px;`
`  align-items: flex-start;`
`}`
	The `flex-start` value aligns the flex items at the top of the container:

`.flex-container {`
`  display: flex;`
`  height: 200px;`
`  align-items: flex-end;`
`}`
	The `flex-end` value aligns the flex items at the bottom of the container:

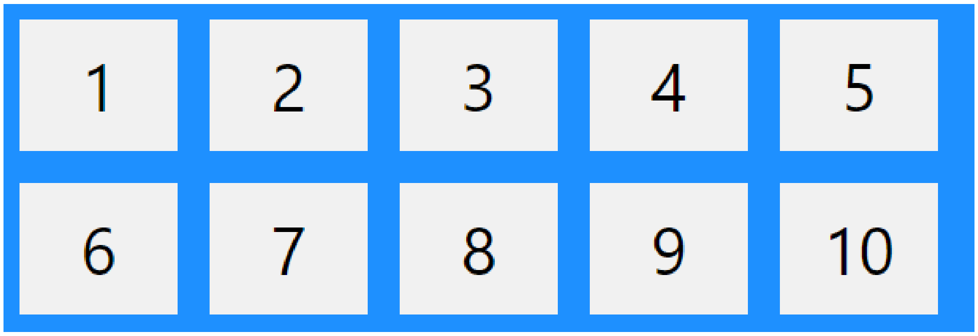
`.flex-container {`
`  display: flex;`
`  height: 200px;`
`  align-items: stretch;`
`}`
	The `stretch` value stretches the flex items to fill the container (this is default):

`.flex-container {`
`  display: flex;`
`  height: 200px;`
`  align-items: baseline;`
`}`
	The `baseline` value aligns the flex items such as their baselines aligns:



The align-content Property

The `align-content` property is used to align the flex lines.



In these examples we use a 600 pixels high container, with the `flex-wrap` property set to `wrap`, to better demonstrate the `align-content` property.

`.flex-container {`
`  display: flex;`
`  height: 600px;`
`  flex-wrap: wrap;`
`  align-content: space-between;`
`}`
	The `space-between` value displays the flex lines with equal space between them:

`.flex-container {`
`  display: flex;`
`  height: 600px;`
`  flex-wrap: wrap;`
`  align-content: space-around;`
`}`
	The `space-around` value displays the flex lines with space before, between, and after them:

`.flex-container {`
`  display: flex;`
`  height: 600px;`
`  flex-wrap: wrap;`
`  align-content: stretch;`
`}`
	The `stretch` value stretches the flex lines to take up the remaining space (this is default):

`.flex-container {`
`  display: flex;`
`  height: 600px;`
`  flex-wrap: wrap;`
`  align-content: center;`
`}`
	The `center` value displays the flex lines in the middle of the container:

`.flex-container {`
`  display: flex;`
`  height: 600px;`
`  flex-wrap: wrap;`
`  align-content: flex-start;`
`}`
	The `flex-start` value displays the flex lines at the start of the container:

`.flex-container {`
`  display: flex;`
`  height: 600px;`
`  flex-wrap: wrap;`
`  align-content: flex-end;`
`}`
	The `flex-end` value displays the flex lines at the end of the container:

The CSS Flexbox Container Properties

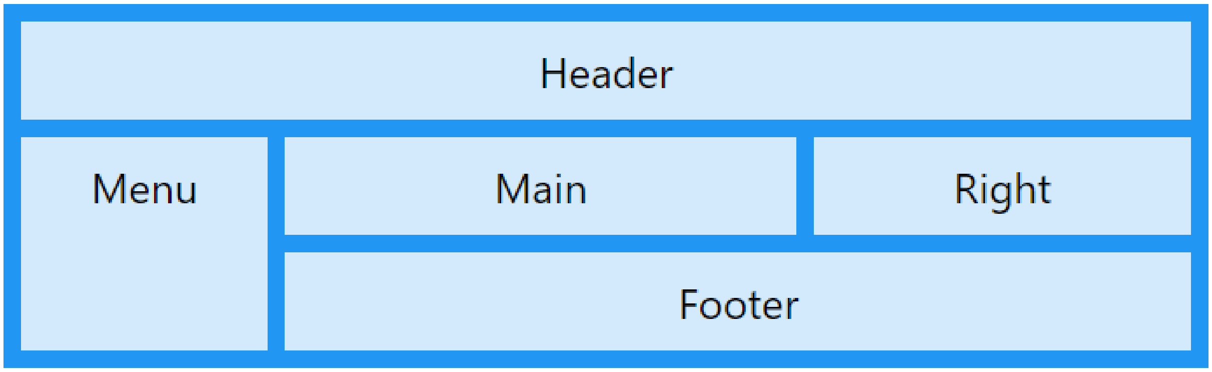
The following table lists all the CSS Flexbox Container properties:

Property	**Description**
align-content	Modifies the behavior of the flex-wrap property. It is similar to align-items, but instead of aligning flex items, it aligns flex lines
align-items	Vertically aligns the flex items when the items do not use all available space on the cross-axis
display	Specifies the type of box used for an HTML element

flex-direction	Specifies the direction of the flexible items inside a flex container
flex-flow	A shorthand property for flex-direction and flex-wrap
flex-wrap	Specifies whether the flex items should wrap or not, if there is not enough room for them on one flex line
justify-content	Horizontally aligns the flex items when the items do not use all available space on the main-axis

CSS Grid Layout Module

The CSS Grid Layout Module offers a grid-based layout system, with rows and columns, making it easier to design web pages without having to use floats and positioning.

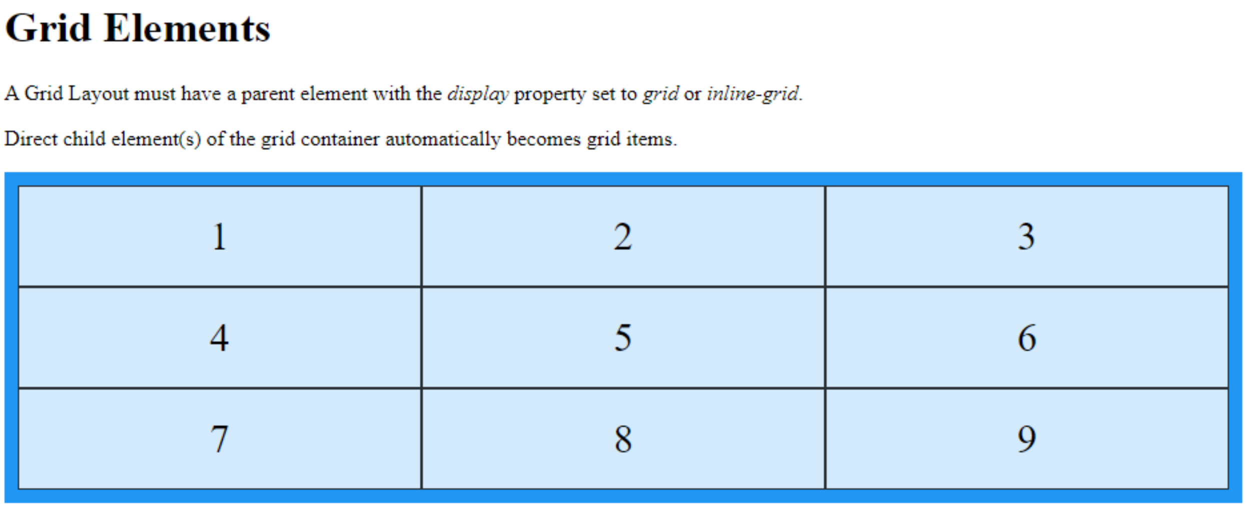


Display Property

An HTML element becomes a grid container when its `display` property is set to `grid` or `inline-grid`.

```
.grid-container {
  display: grid;
}
```

```
.grid-container {
  display: inline-grid;
}
```



Grid Elements

A Grid Layout must have a parent element with the *display* property set to *grid* or *inline-grid*.

Direct child element(s) of the grid container automatically becomes grid items.

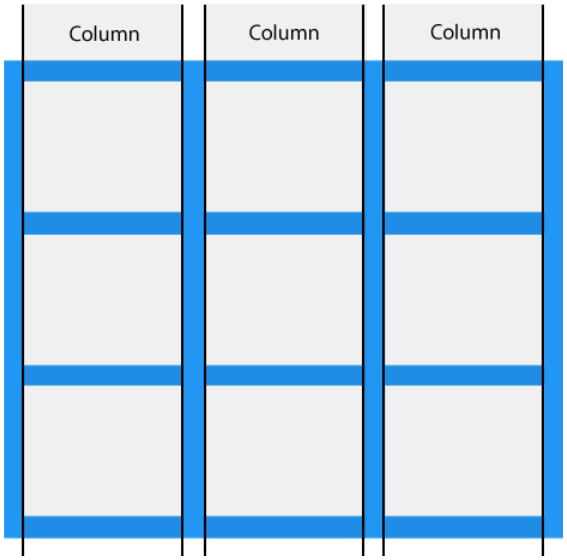
1	2	3
4	5	6
7	8	9

Code

```
<!DOCTYPE html>
<html>
<head>
<style>
.grid-container {
  display: grid;
  grid-template-columns: auto auto auto;
  background-color: #2196F3;
  padding: 10px;
}
.grid-item {
  background-color: rgba(255, 255, 255, 0.8);
  border: 1px solid rgba(0, 0, 0, 0.8);
  padding: 20px;
  font-size: 30px;
  text-align: center;
}
</style>
</head>
<body>

<h1>Grid Elements</h1>

<p>A Grid Layout must have a parent element with the <em>display</em> property set to
<em>grid</em> or <em>inline-grid</em>.</p>

<p>Direct child element(s) of the grid container automatically becomes grid items.</p>

<div class="grid-container">
  <div class="grid-item">1</div>
  <div class="grid-item">2</div>
  <div class="grid-item">3</div>
  <div class="grid-item">4</div>
  <div class="grid-item">5</div>
  <div class="grid-item">6</div>
  <div class="grid-item">7</div>
  <div class="grid-item">8</div>
  <div class="grid-item">9</div>
</div>

</body>
</html>
```

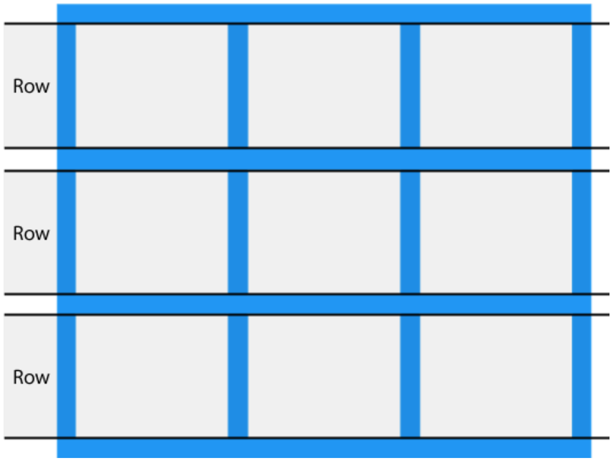
Grid Columns

The vertical lines of grid items are called *columns*.



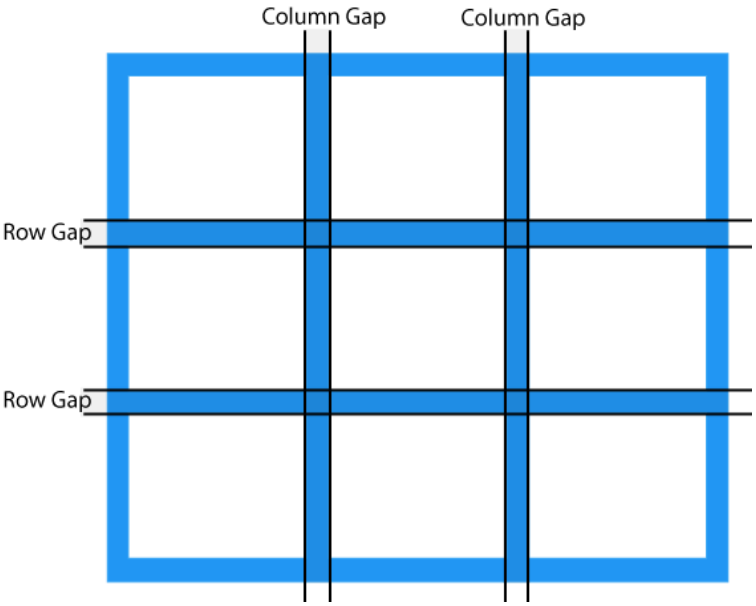
Grid Rows

The horizontal lines of grid items are called *rows*.



Grid Gaps

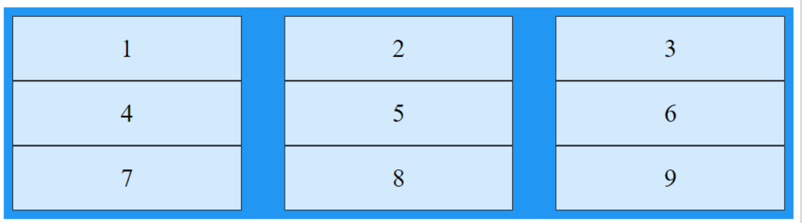
The spaces between each column/row are called *gaps*.



You can adjust the gap size by using one of the following properties:

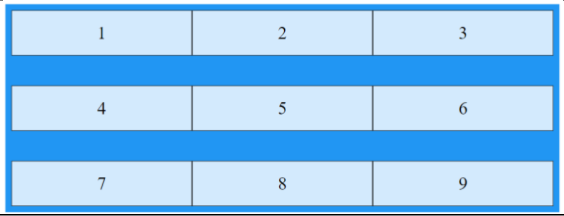
- `column-gap`
- `row-gap`
- `gap`

```
.grid-container {
  display: grid;
  column-gap: 50px;
}
```

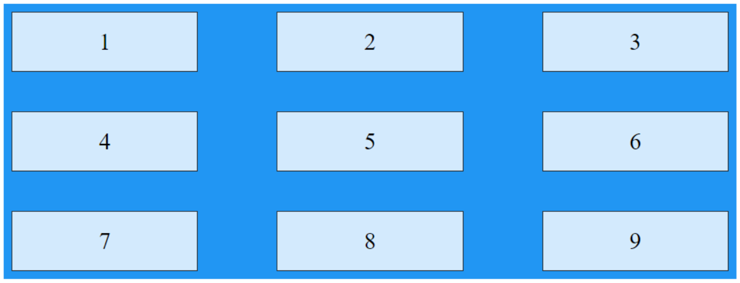


1 2 3
4 5 6
7 8 9

```
.grid-container {
  display: grid;
  row-gap: 50px;
}
```



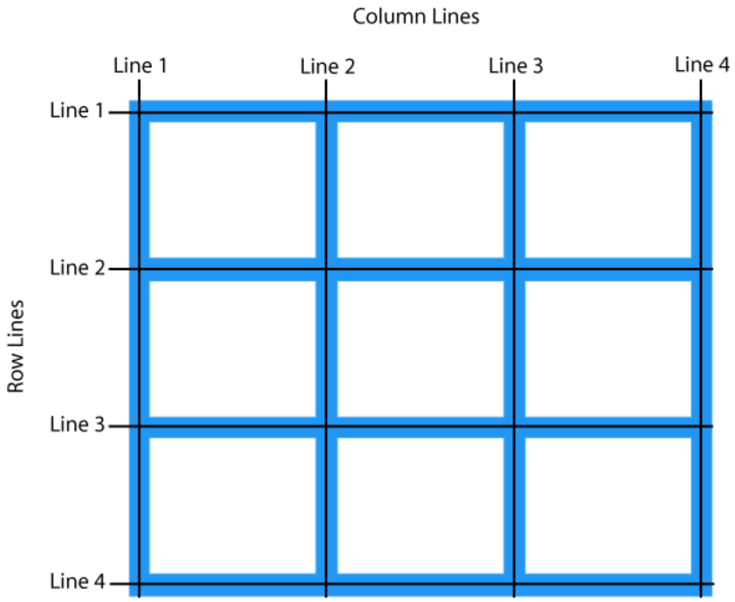
1 2 3
4 5 6
7 8 9

.grid-container {
  display: grid;
  gap: 50px 100px;
}
	

Grid Lines

The lines between columns are called *column lines*.

The lines between rows are called *row lines*.



.item1 {
  grid-column-start: 1;
  grid-column-end: 3;
}
	Place a grid item at column line 1, and let it end on column line 3:

.item1 {
  grid-row-start: 1;
  grid-row-end: 3;
}
	Place a grid item at row line 1, and let it end on row line 3:

CHAPTER 5 – INTRODUCTION TO BOOTSTRAP 5

5.1 Bootstrap

Bootstrap is a free and open-source CSS and JavaScript / jquery code library used to create dynamic website layouts and apps. Bootstrap is a famous front-end framework with a decent variety of pre-defined CSS codes.

Bootstrap 5 is the most recent and stable version of the Bootstrap framework. Bootstrap is responsive by default, and it takes a mobile-first approach. Bootstrap 5 is compatible with the most recent, stable releases of all major browsers and systems.

To create responsive web pages, Bootstrap employs a variety of classes. After months of redefining its features, Bootstrap 5 was officially published on June 16, 2020.

Bootstrap 5 is the most recent version of the most popular HTML, CSS, and JavaScript framework for constructing responsive, mobile-first websites, Bootstrap.

Responsive web design is the process of building websites that automatically change to look excellent on all platforms, from little phones to giant desktop computers.

Versions of Bootstrap 5

Bootstrap 5 (published in 2021) is the most recent version of Bootstrap (launched in 2013); it includes new components, a quicker style sheet, and improved responsiveness. Bootstrap 5 changes its logo with new version.

This Bootstrap tutorial series covers all of the framework's capabilities, beginning with the basics, such as the grid system, form styling methods, font styling mechanism, and strategies for styling user interface elements like tables, lists, photos, and so on.

We will also learn how to alter the ready-to-use Bootstrap components like list groups, input groups, button groups, pagination, labels cards, navbar, breadcrumbs, badges, progress bars, etc.

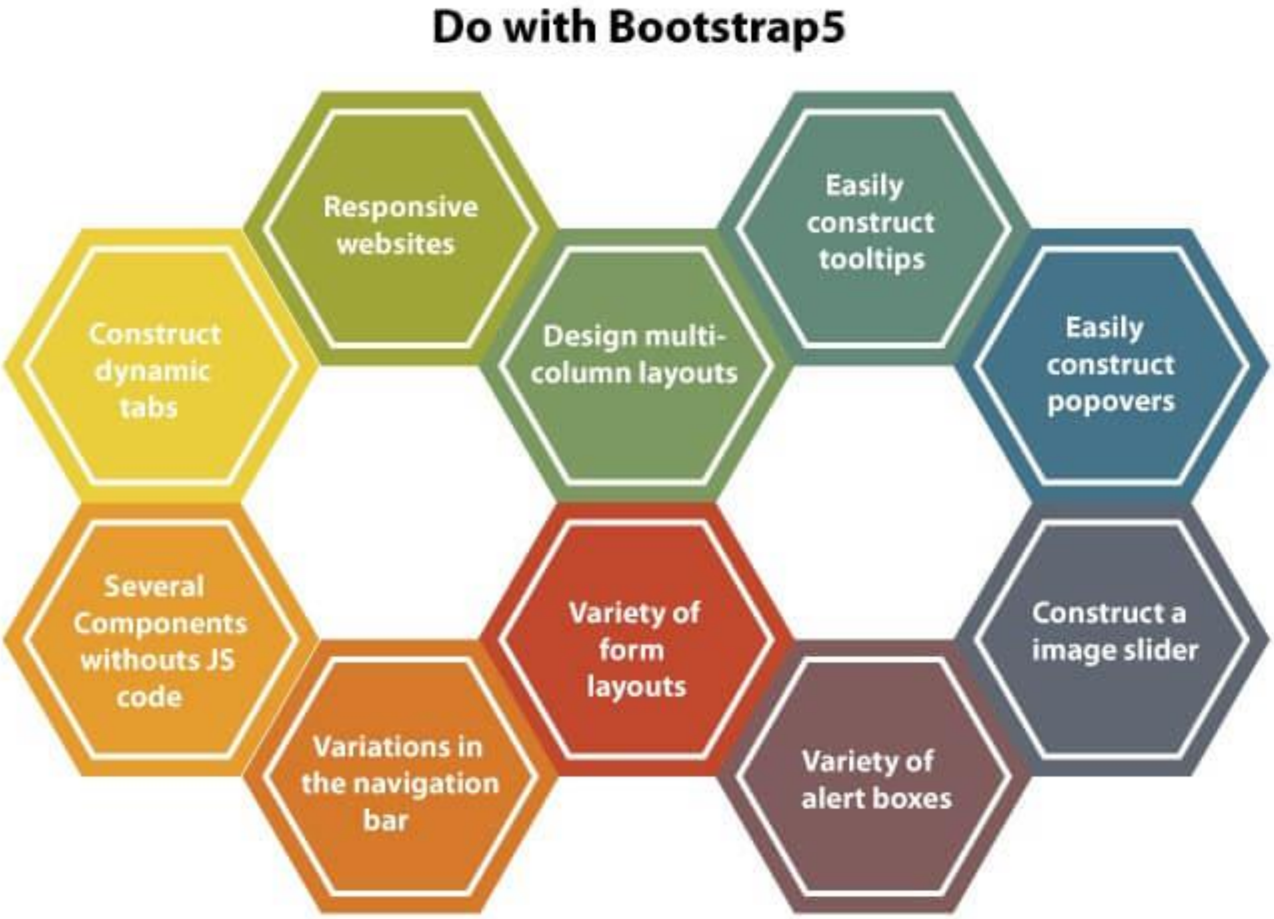
Finally, we will investigate some advanced Bootstrap capabilities, such as generating modal windows, dynamic tabs, tooltips, alerts, accordion menus, carousels, scrollspy, and so on, and how to tweak or enhance their present functionality using the available choices and methods.

Bootstrap 5 is compatible with Google Chrome, Firefox, Safari, Internet Explorer 10, etc.

What Can We Do With Bootstrap5?

There's a lot more we can do with Bootstrap.

- Responsive websites are simple to develop.
- With pre-defined classes, we can rapidly design multi-column layouts.
- We can rapidly construct a variety of form layouts.
- We can quickly construct several variations of the navigation bar.
- Components such as accordions, modals, and others can be simply created without writing any JS code.
- To manage a big content volume, we can easily construct dynamic tabs.
- To display hint text, easily construct tooltips and popovers.
- We may quickly construct a carousel or image slider to highlight our content.
- We can instantly construct a variety of alert boxes.


Do with Bootstrap5
Responsive websites
Easily construct tooltips
Construct dynamic tabs
Design multi-column layouts
Easily construct popovers
Several Components withouts JS code
Variety of form layouts
Construct a image slider
Variations in the navigation bar
Variety of alert boxes

Bootstrap 5 Example

Bootstrap 5 has a basic format for creating web pages. This web page provides fixed and responsive applications for all devices. We can see the following steps in the basic format of the example.

- **HTML File**: create html file in required IDE. This page creates web applications. For example, index.html.
- **Add the HTML 5 doctype**: Because Bootstrap version 5 relies on HTML elements and CSS attributes, you must include the HTML 5 doctype at the top of the page, along with the lang property and the right character set.

```
<!DOCTYPE html>
<html lang = "en">
<head>
<meta http-equiv = "Content-Type" content = "text/html; charset = windows-1252">
<title> Bootstrap 5 example title </title>
</head>
<body>
Write bootstrap 5 code!
</body>
</html>
```

- **Bootstrap is mobile-friendly**: Bootstrap 5 is designed to be mobile-responsive.

Bootstrap's core foundation includes mobile-first styling.

For appropriate rendering and touch zooming, you must include the following <meta> tag within the <head> element:

```
<meta name = "viewport" content = "width=device-width, initial-scale = 1">
```

The "width=device-width" section is used to set the width of the page to match the device's screen width (vary according to the device).

When the browser first loads the page, the initial-scale=1 component is utilized to set the initial zoom level.

- **Apply bootstrap 5 links**: the bootstrap 5 can download files from official websites or use online links. The following links use in the head section of the html page.

```
<link href = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel="stylesheet">
<script src = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js">
</script>
```

- **Use container**: the bootstrap 5 uses container or container-fluid class in the <div> tag. A container is used to enclose the contents of a website. The .container class provides a responsive or fixed-width container.

```
<div class = "container"> write code here... </div>
```

- **Use bootstrap 5 functions:** bootstrap 5 use row and column in the container to place functions. We can contains navbar, content, table, forms and other functions in container using grid system. We can use other classes and CSS styles on the web page.

Example

The following example shows the basic container, columns, and row of the bootstrap 5version. We can see the links and other related tags to create a basic web page.

Here, we use a container with light background with heading and paragraph elements. This container contains the content of the page.

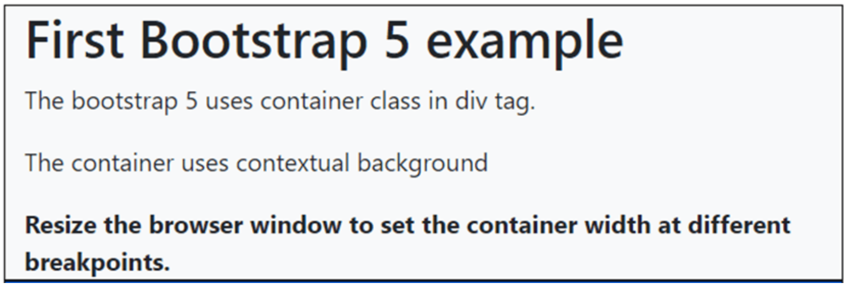
index.html

```
<!DOCTYPE html>
<html lang = "en">
<head>
<title> Bootstrap 5 Example </title>
<meta charset = "utf-8">
<meta name = "viewport" content = "width=device-width, initial-scale = 1">
<link href = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel="stylesheet">
<script src = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js">
</script>
```

```
</head>
<body>
<div class="container bg-light"style="border : 1px solid black;">
<h1> First Bootstrap 5 example </h1>
<p> The bootstrap 5 uses container class in div tag.</p>
<p> The container uses contextual background </p>
<b>
Resize the browser window to set the container width at different breakpoints.
</b>
</div>
</div>
</body>
</html>
```

Output:

The following image shows the output of the bootstrap 5 examples.



First Bootstrap 5 example

The bootstrap 5 uses container class in div tag.

The container uses contextual background

Resize the browser window to set the container width at different breakpoints.

Difference between Bootstrap4 and Bootstrap5 Framework

Bootstrap is a free and open-source toolkit for developing responsive web pages and online apps. HTML, CSS, and the JavaScript framework make it simple to create responsive, mobile-first websites. Its primary goal is to address the cross-browser compatibility issue. Bootstrap is at the heart of any website optimized for all screen sizes and looks great in all modern browsers.

Bootstrap 5 includes custom form controls. Bootstrap 4's form controls used whatever presets were available in each browser. The form controls in Bootstrap 5 will have a significantly more consistent appearance and feel across all browsers due to their custom design.

In bootstrap 5, some classes remove, which is available in the bootstrap 4 versions.

- a "row form"
- a "form inline"
- a "list inline"
- a "form inline"

In bootstrap 5, some classes are not available in the bootstrap 4 versions.

- gx-* : the horizontal/column gutters width
- gy-* :the vertical/row gutters width
- g-* :the horizontal & vertical gutter width
- rows-cols-auto

Comparison between Bootstrap 4 and Bootstrap 5

Features	Bootstrap 4 version	Bootstrap 5 version
Released date	Bootstrap 4 was released in January 2018, and it marked a significant advance over previous versions.	Bootstrap 5 was released in May 2021. This version came after multiple iterations of alpha and beta releases, and as a result, it

		had a slew of important additions and new features.
Links	<link rel = "stylesheet" href = "https://cdn.jsdelivr.net/npm/bootstrap@4.6.1/dist/css/bootstrap.min.css"> <script src = "https://cdn.jsdelivr.net/npm/bootstrap@4.6.1/dist/js/bootstrap.bundle.min.js"> </script>	<link href = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel = "stylesheet"> <script src = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js"> </script>
Grid System	Bootstrap 4 has five levels: xs, sm, md, lg, and xl.	Bootstrap 5 has six levels: xs, sm, md, lg, xl, and xxl.
Jquery plug-in	Bootstrap 4 includes jquery and all associated plug-ins.	Jquery has been deleted and replaced with vanilla JS and several functional plug-ins in Bootstrap 5.
Color	Bootstrap 4 provides a few colors for a web page.	Extra colors insert into the looks, and the card's color palette was upgraded. There are a variety of colors to pick from the Bootstrap 5 framework.
Internet explorer	Bootstrap 4 is compatible with both Internet Explorer 10 and 11.	Bootstrap 5 does not support Internet Explorer 10 and 11.
Elements of Form	Radio buttons and checkboxes appear differently in different operating systems and browsers. The form takes advantage of whatever basic browser features are available.	The appearance of form elements will not change while using a different operating system or browser. The forms alternate and add form controls; they are not browser-dependent.

API for Utilities	We are unable to alter utilities in Bootstrap 4.	Bootstrap 5 provided us the ability to tweak and develop our utility.
Utilize a gutter.	Bootstrap 4 uses glutter in a px font size.	Use .g* with font size set to rem in bootstrap 5.
Vertical Groups	Columns can be positioned relative to one another.	Columns cannot be positioned relative to one another.
Icons for Bootstrap	Bootstrap 4 lacks SVG icons; we must rely on font-awesome for icons.	Bootstrap 5 has its own set of SVG icons.
Jumbotron	Bootstrap 4 compatible with jumbotron.	It is not compatible with jumbotron.
deck of cards	The card deck is used to make an isset of cards with the same width and height.	The card deck class has been removed from bootstrap.
Navbar	We have the inline-block attribute, and the white dropdown will be the default for the dropdown-menu-dark class.	The inline-block property has been removed, and the dropdown-menu-dark class now has a black dropdown as the default.
Generator of Static Websites	Bootstrap 4 makes use of the Jekyll software.	Bootstrap 5 uses a fast static generator which is Hugo.
grid flexbox	It makes vertical designs easier to construct, and columns and rows can be simply implemented. The justify-center-	A more complex grid system is offered, and columns do not have relative locations.

	content classes use directly to align according to the requirement.	
CSS usability	Bootstrap 4 uses fewer CSS properties.	Bootstrap 4 uses more CSS properties.
offcanvas component	There is no off-canvas element.	The offcanvas component is now available.
RTL support	Languages do not enable RTL (right to the left) switching.	It enables RTL (right to the left) language switching.

Bootstrap 5 Containers

Bootstrap 5 Containers are a core Bootstrap building piece that contains and align your content within a specific device or viewport. The bootstrap 5 container contains responsive width with the required row and column for the advanced features and content. The container class works as fixed and responsive on the web page.

Bootstrap 5 to Create Containers

Containers are the most fundamental Bootstrap layout elements necessary when using the grid system. Containers are mostly used to envelop material with padding. In the case of a fixed-width layout, they are also used to center the content horizontally on the page.

Bootstrap 5 has three types of containers:

- **Container**: The ".container class" has a maximum width at each responsive breakpoint.
- **Container-fluid**: The ".container-fluid" class has a width of 100 percent at all breakpoints.
- **Responsive Container**: The ".container-breakpoint" has a width of 100 percent until the specified breakpoint.

Bootstrap 5 "container" class

The container class contains content, navbar, image slider, and other functions with a grid system. It is a fixed-width container for web pages. The container class provides spaces inside the body section. It's placed in the center of the body section of the html page.

Example

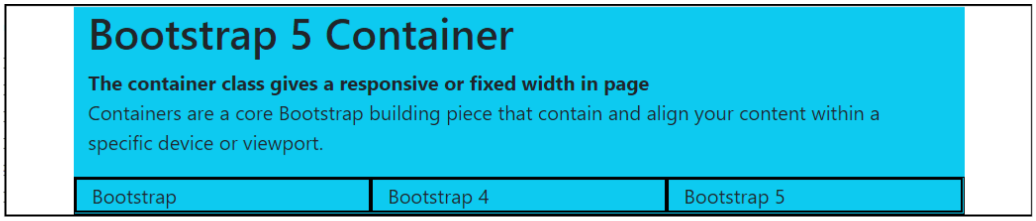
The following example shows container in body tag with content. Bootstrap 5 uses the "container" class with "bg-info" background color. We can add row, columns and other features in the container tag.

```
<!DOCTYPE html>
<html lang="en">
<head>
<title> Bootstrap 5 container Example </title>
<meta charset = "utf-8">
<meta name = "viewport" content = "width=device-width, initial-scale=1">
```

```
<link href = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel = "stylesheet">
<script src = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js">
</script>
<style>
body{
border:2px solid black;
}
</style>
</head>
<body>
<div class="container bg-info">
<h1> Bootstrap 5 Container </h1>
<b> The container class gives a responsive or fixed width in page </b>
<p> Containers are a core Bootstrap building piece that contain and align your content within a specific device or viewport. </p>
<div class = "row" style="border : 1px solid black;">
<div class = "col" style="border : 2px solid black;"> Bootstrap </div>
<div class = "col" style="border : 2px solid black;"> Bootstrap 4 </div>
<div class = "col" style="border : 2px solid black;"> Bootstrap 5 </div>
</div>
</div>
</body>
</html>
```

Output:

The following image shows the output of the container class example.



Bootstrap 5 Container

The container class gives a responsive or fixed width in page

Containers are a core Bootstrap building piece that contain and align your content within a specific device or viewport.

Bootstrap	Bootstrap 4	Bootstrap 5

Bootstrap 5 "container-fluid" class

The container-fluid class contains content, navbar, image slider, and other functions with a grid system. The container class acquires space inside of the body section. If we need to use all space of the web page, then use the container-fluid class. It is a full-width container for web pages.

Example

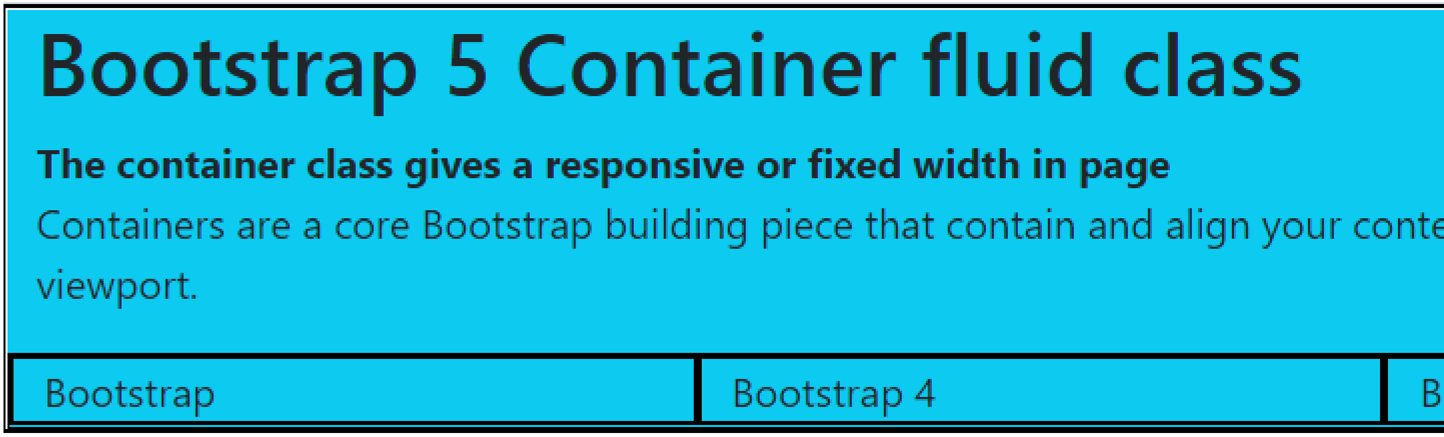
The following example shows container in body tag with content. Bootstrap 5 uses the "container-fluid" class with "bg-info" background color. We can add fixed or responsive row, columns and other features in the container-fluid element.

```
<!DOCTYPE html>
<html lang="en">
<head>
<title> Bootstrap 5 container fluid Example </title>
<meta charset = "utf-8">
<meta name = "viewport" content = "width=device-width, initial-scale=1">
<link href = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel = "stylesheet">
<script src = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js">
</script>
<style>
body{
border:2px solid black;
}
</style>
</head>
<body>
<div class="container-fluid bg-info">
<h1> Bootstrap 5 Container fluid class </h1>
<b> The container class gives a responsive or fixed width in page </b>
<p> Containers are a core Bootstrap building piece that contain and align your content within a specific device or viewport. </p>
<div class = "row" style="border : 1px solid black;">
```

```
<div class = "col" style="border : 2px solid black;"> Bootstrap </div>
<div class = "col" style="border : 2px solid black;"> Bootstrap 4 </div>
<div class = "col" style="border : 2px solid black;"> Bootstrap 5 </div>
</div>
</div>
</body>
</html>
```

Output:

The following image shows the output of the container class example.



Bootstrap 5 Container fluid class

The container class gives a responsive or fixed width in page

Containers are a core Bootstrap building piece that contain and align your conte

viewport.

Bootstrap

Bootstrap 4

B

Bootstrap 5 responsive container

The container-breakpoint class contains content and other functions with the grid system. The breakpoint places the size of the fixed-width for particular devices. The container class acquires space inside of the body section. The responsive container shows the container as per device width.

The following table shows breakpoint with width size.

Classes container-breakpoint	Extra Small < 576px	Small (sm) ≥ 576px	Medium(md) ≥ 768px	Large(lg) ≥992px	X-Large (xl) ≥1200px	XX-Large (xxl) ≥1400px

"container" class	100%	540px	720px	960px	1140px	1320px
"container-sm" class	100%	540px	720px	960px	1140px	1320px
"container-md" class	100%	100%	720px	960px	1140px	1320px
"container-lg" class	100%	100%	100%	960px	1140px	1320px
"container-xl" class	100%	100%	100%	100%	1140px	1320px
"container-xxl" class	100%	100%	100%	100%	100%	1320px

Example

The following example shows a responsive container in a body tag with content. Bootstrap 5 creates a responsive container using the "container-breakpoint" class. The default class is "container" for all web pages. If we need to use for small, medium, large, extra large and double extra large then use "container-sm", "container-md", "container-lg", "container-xl" and "container-xxl" class respectively.

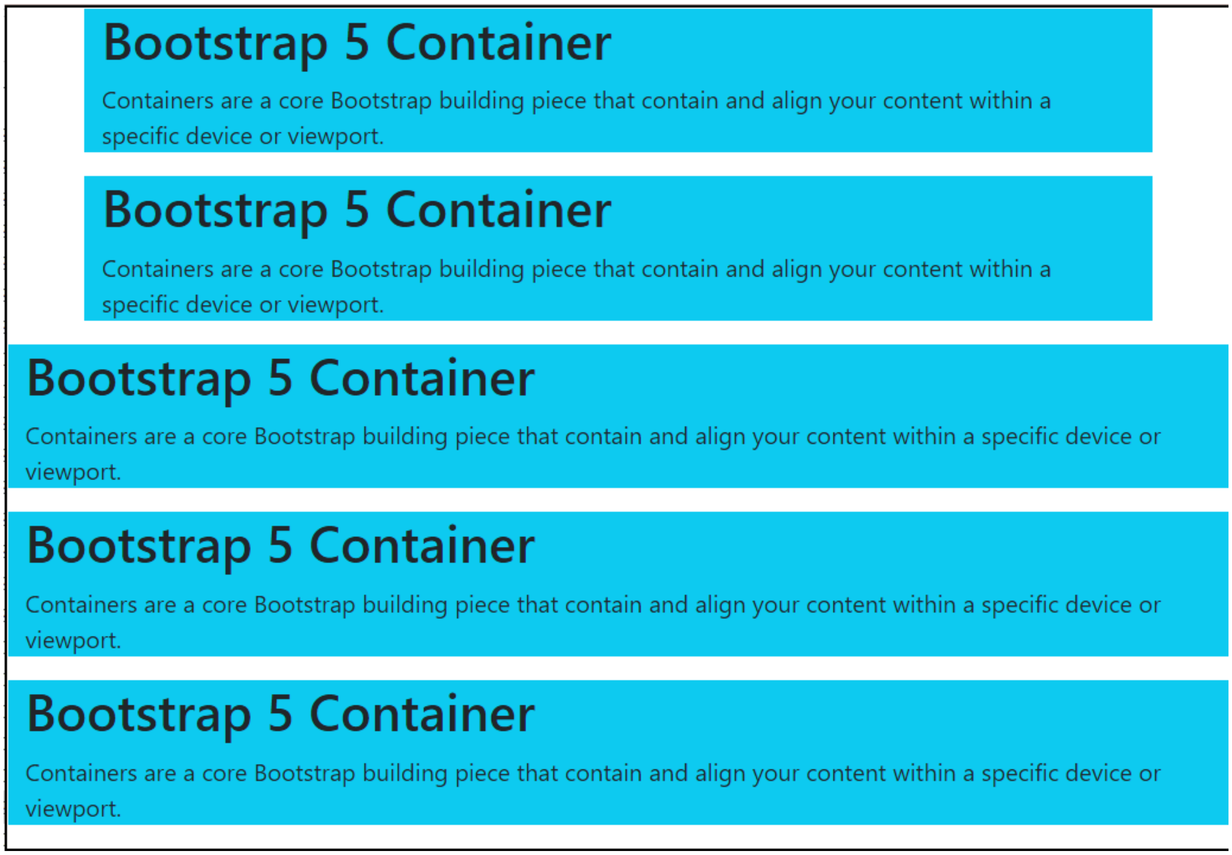
```
<!DOCTYPE html>
<html lang="en">
<head>
<title> Bootstrap 5 container Example </title>
<meta charset = "utf-8">
<meta name = "viewport" content = "width=device-width, initial-scale=1">
<link href = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel = "stylesheet">
```

```
<script src = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js">
</script>
<style>
body{
border:2px solid black;
}
</style>
</head>
<body>
<div class="container-sm bg-info">
<h1> Bootstrap 5 Container </h1>
<p> Containers are a core Bootstrap building piece that contain and align your content within a specific device or viewport. </p>
</div>
<div class="container-md bg-info">
<h1> Bootstrap 5 Container </h1>
<p> Containers are a core Bootstrap building piece that contain and align your content within a specific device or viewport. </p>
</div>
<div class="container-lg bg-info">
<h1> Bootstrap 5 Container </h1>
<p> Containers are a core Bootstrap building piece that contain and align your content within a specific device or viewport. </p>
</div>
<div class="container-xl bg-info">
<h1> Bootstrap 5 Container </h1>
<p> Containers are a core Bootstrap building piece that contain and align your content within a specific device or viewport. </p>
</div>
<div class="container-xxl bg-info">
<h1> Bootstrap 5 Container </h1>
<p> Containers are a core Bootstrap building piece that contain and align your content within a specific device or viewport. </p>
</div>
```

```
</body>
</html>
```

Output:

The following image shows the output of the container class example.



Bootstrap 5 Container

Containers are a core Bootstrap building piece that contain and align your content within a specific device or viewport.

Bootstrap 5 Container

Containers are a core Bootstrap building piece that contain and align your content within a specific device or viewport.

Bootstrap 5 Container

Containers are a core Bootstrap building piece that contain and align your content within a specific device or viewport.

Bootstrap 5 Container

Containers are a core Bootstrap building piece that contain and align your content within a specific device or viewport.

Bootstrap 5 Container

Containers are a core Bootstrap building piece that contain and align your content within a specific device or viewport.

Padding for Containers

By default, containers have left and right padding but no top or bottom padding. As a result, we frequently use spacing utilities such as extra padding and margins to improve their appearance. Container fluid does not have any padding on a page. We need to use padding with size and side. For example, the ".pb-size" class implies adding bottom padding with size.

Example

Bootstrap 5 uses the "py-5" class for padding the container. The "py" works for spacing the top and bottom sides simultaneously.

The container-fluid uses the "py" and "px" for vertical and horizontal padding with 5 sizes.

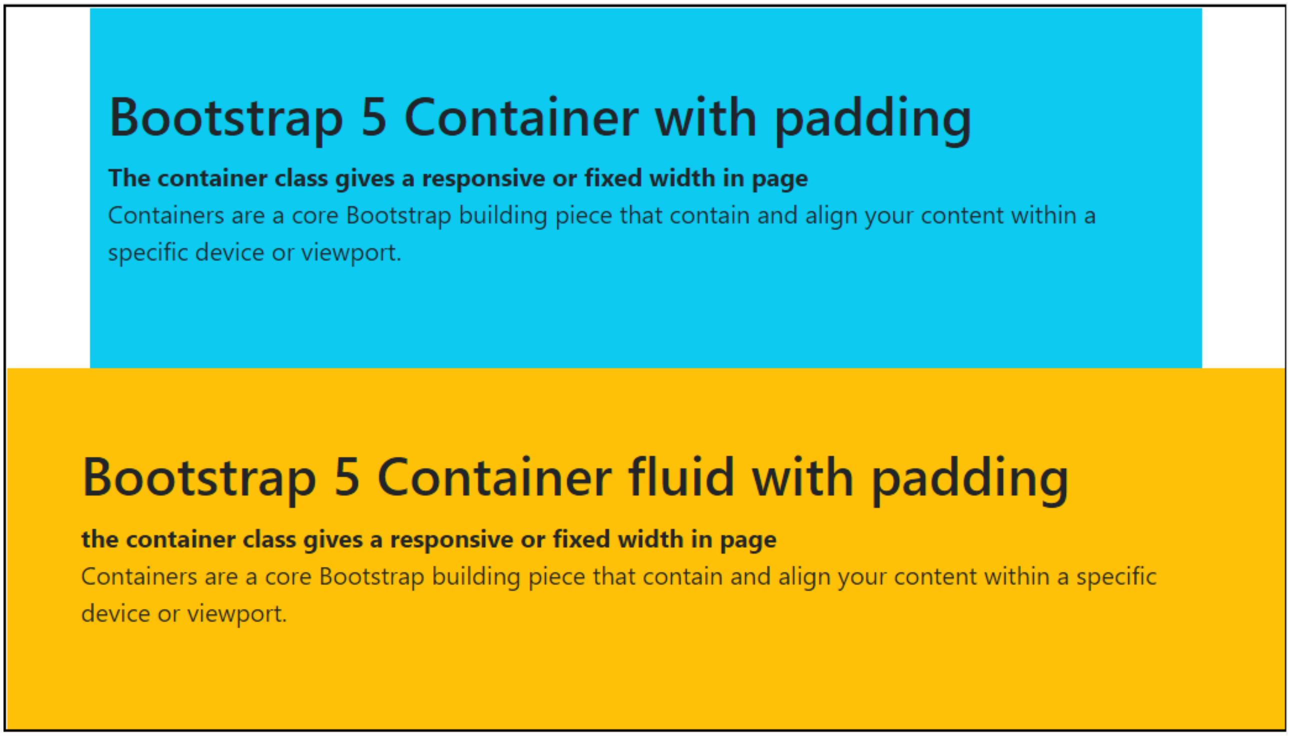
The following example shows responsive container in body tag with padding.

```
<!DOCTYPE html>
<html lang="en">
<head>
<title> Bootstrap 5 container Example </title>
<meta charset = "utf-8">
<meta name = "viewport" content = "width=device-width, initial-scale=1">
<link href = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel = "stylesheet">
<script src = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js">
</script>
<style>
body{
border:2px solid black;
}
</style>
</head>
<body>
<div class="container py-5 bg-info">
<h1> Bootstrap 5 Container with padding </h1>
<b>The container class gives a responsive or fixed width in page </b>
<p> Containers are a core Bootstrap building piece that contain and align your content within a specific device or viewport. </p>
</div>
<div class="container-fluid py-5 px-5 bg-warning">
<h1> Bootstrap 5 Container fluid with padding </h1>
<b>the container class gives a responsive or fixed width in page </b>
<p> Containers are a core Bootstrap building piece that contain and align your content within a specific device or viewport. </p>
</div>
</body>
```

```
</html>
```

Output:

The following image shows the output of the container class example.



Bootstrap 5 Container with padding

The container class gives a responsive or fixed width in page

Containers are a core Bootstrap building piece that contain and align your content within a specific device or viewport.

Bootstrap 5 Container fluid with padding

the container class gives a responsive or fixed width in page

Containers are a core Bootstrap building piece that contain and align your content within a specific device or viewport.

Border, Color, Margin for Containers

By default, containers have left and right spacing but no top or bottom padding. As a result, we frequently use spacing utilities such as extra padding and margins to improve their appearance. Container fluid does not have any space on the page. We need to use margin for any side. For example, the ".m(side)-size" class implies adding the required side's margin with size.

Example

The container uses "my-3" for horizontal margins such as the top and bottom side with the "3" size. The "border" class uses to create a border around the container, and the "text-white" class is for white color container content. The "bg-light" uses to display light grey background.

The "bg-success" applies on the second container to set the green background. This container uses the "5" size margin from all sides using the "m-5" class.

The container-fluid applies "bg-warning" with border and "m-5" margin.

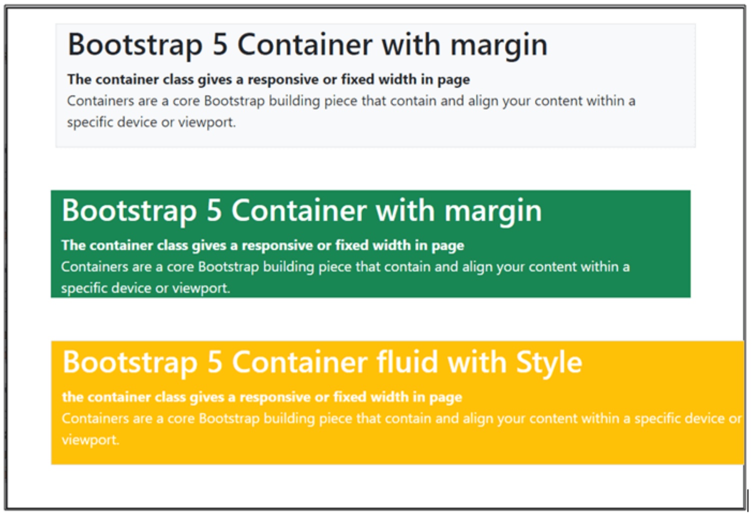
The following example shows responsive container in body tag with margin, border, and color.

```
<!DOCTYPE html>
<html lang="en">
<head>
<title> Bootstrap 5 container Example </title>
<meta charset = "utf-8">
<meta name = "viewport" content = "width=device-width, initial-scale=1">
<link href = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel = "stylesheet">
<script src = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js">
</script>
<style>
body{
border:2px solid black;
}
</style>
</head>
<body>
<div class="container my-3 bg-light border">
<h1> Bootstrap 5 Container with margin </h1>
<b>The container class gives a responsive or fixed width in page </b>
<p> Containers are a core Bootstrap building piece that contain and align your content within a specific device or viewport. </p>
</div>
<div class="container m-5 bg-success text-white">
<h1> Bootstrap 5 Container with margin </h1>
<b>The container class gives a responsive or fixed width in page </b>
<p> Containers are a core Bootstrap building piece that contain and align your content within a specific device or viewport. </p>
```

```
</div>
<div class="container-fluid m-5 bg-warning text-white border">
<h1> Bootstrap 5 Container fluid with Style </h1>
<b>the container class gives a responsive or fixed width in page </b>
<p> Containers are a core Bootstrap building piece that contain and align your content within a specific device or viewport. </p>
</div>
</body>
</html>
```

Output:

The following image shows the output of the container class example.



Bootstrap 5 Container with margin

The container class gives a responsive or fixed width in page

Containers are a core Bootstrap building piece that contain and align your content within a specific device or viewport.

Bootstrap 5 Container with margin

The container class gives a responsive or fixed width in page

Containers are a core Bootstrap building piece that contain and align your content within a specific device or viewport.

Bootstrap 5 Container fluid with Style

the container class gives a responsive or fixed width in page

Containers are a core Bootstrap building piece that contain and align your content within a specific device or viewport.

Bootstrap 5 Grid

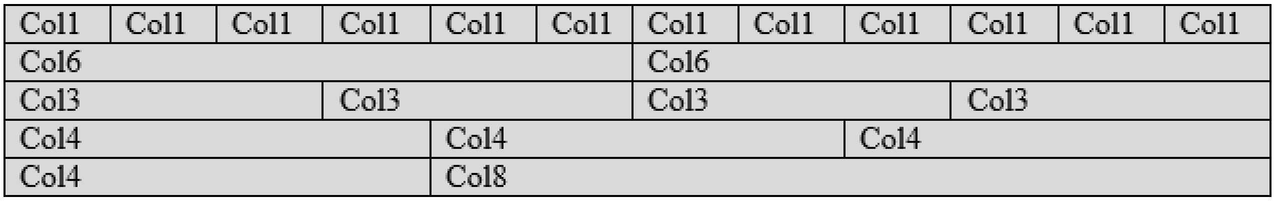
The Bootstrap 5 grid system is designed with flexbox and supports up to 12 columns across the page. If we don't want to use all 12 columns separately, you can combine them to make wider columns.

The grid system is responsive, and the columns will re-arrange themselves automatically based on screen size. It is not necessary to use all 12 available columns. Make sure the total is less than or equal to 12.

We can utilize Bootstrap's default grid classes to easily create layouts for various devices like mobile phones, tablets, laptops, and desktops.

For example, the .col-* classes can construct grid columns for especially small devices such as mobile phones in portrait mode, and the .col-sm-* classes for mobile phones in landscape mode.

The following table shows the format of the bootstrap 5 grid systems.



Col1	Col1	Col1	Col1	Col1	Col1	Col1	Col1	Col1	Col1	Col1	Col1
Col6						Col6					
Col3			Col3			Col3			Col3		
Col4				Col4				Col4			
Col4				Col8							

Default Grid System

The default grid system helps create columns using the "col" class for a single column in a row. We can use the "col" class twelve times to create twelve columns in one row.

Syntax of the Grid System

The default grid system in bootstrap 5 syntax shows below

```
<div class = "row">
 <div class = "col">
 </div>
 <div class = "col">
 </div>
```

```
  <div class = "col">
  </div>
  <div class = "col">
  </div>
</div>
```

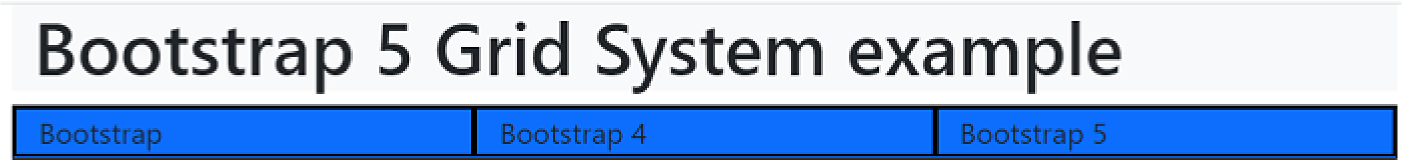
Examples

The given example shows default grid system in bootstrap 5 with classes and function. The default grid system uses "row" and "col" class as per requirement. Here, we uses three "col" classes for three column in one row.

```
<!DOCTYPE html>
<html lang = "en">
<head>
<title> Bootstrap 5 Example </title>
<meta charset = "utf-8">
<meta name = "viewport" content = "width=device-width, initial-scale = 1">
<link href = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel="stylesheet">
<script src = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js">
</script>
</head>
<body>
<div class="container bg-light">
 <h1> Bootstrap 5 Grid System example </h1>
</div>
<div class="container bg-primary">
<div class = "row" style="border : 1px solid black;">
<div class = "col" style="border : 2px solid black;"> Bootstrap </div>
<div class = "col" style="border : 2px solid black;"> Bootstrap 4 </div>
<div class = "col" style="border : 2px solid black;"> Bootstrap 5 </div>
</div>
</div>
</body>
```

```
</html>
```

Output:



Bootstrap 5 Grid System example

Bootstrap Bootstrap 4 Bootstrap 5

Grid System with Size

We can apply the size of the column to create the customized column. The grid system helps create columns with the required size using the "col-size" class for a single column in a row. For example, the "col-3" uses to create three column sizes in one column.

Syntax of the Grid System

The bootstrap 5 grid system with size syntax shows below.

```
<div class = "row">
  <div class = "col-size">
  </div>
  <div class = "col-size">
  </div>
  <div class = "col-size">
  </div>
  <div class = "col-size">
  </div>
</div>
```

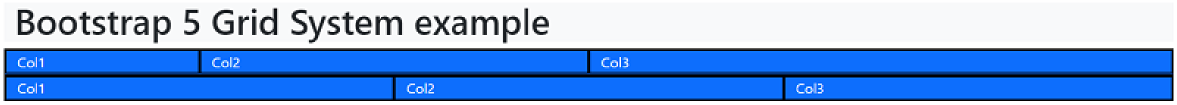
Examples

The given example shows default grid system in bootstrap 5 with classes and function. In this example, we can use column with sizes. First row uses uneven size of the column which total 12 column. Second row uses even size of the column which is 4 columns. We can use multiple column with equal to less than twelve column size.

```
<!DOCTYPE html>
<html lang = "en">
```

```
<head>
<title> Bootstrap 5 Example </title>
<meta charset = "utf-8">
<meta name = "viewport" content = "width=device-width, initial-scale = 1">
<link href = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel="stylesheet">
<script src = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js">
</script>
</head>
<body>
<div class="container bg-light">
 <h1> Bootstrap 5 Grid System example </h1>
</div>
<div class="container bg-primary text-white">
<div class = "row" style="border : 1px solid black;">
<div class = "col-2" style="border : 2px solid black;"> Col1 </div>
<div class = "col-4" style="border : 2px solid black;"> Col2 </div>
<div class = "col-6" style="border : 2px solid black;"> Col3 </div>
</div>
<div class = "row" style="border : 1px solid black;">
<div class = "col-4" style="border : 2px solid black;"> Col1 </div>
<div class = "col-4" style="border : 2px solid black;"> Col2 </div>
<div class = "col-4" style="border : 2px solid black;"> Col3 </div>
</div>
</div>
</body>
</html>
```

Output:



```
Bootstrap 5 Grid System example
Col1          Col2                  Col3
Col1                    Col2                    Col3
```

Responsive Grid Classes

The Bootstrap 5 classes use for a responsive grid system. The classes combine to create more dynamic and flexible layouts. The Bootstrap 5 adds an "xxl" breakpoint for a double extra-large device.

The Bootstrap 5 grid system has six classes:

Grid Class	Device	Device size
.col	Extra small devices	Screen width must be less than 576px
.col-sm	Small devices	Screen width must be equal to or greater than 576px
.col-md	Medium devices	Screen width must be equal to or greater than 768px
.col-lg	Large devices	Screen width must be equal to or greater than 992px
.col-xl	Extra-large devices	Screen width must be equal to or greater than 1200px
.col-xxl	Double extra (xxlarge) devices	Screen width must be equal to or greater than 1400px

Syntax of the Grid System

The bootstrap 5 grid system with size syntax shows below. We can use single or multiple grid class in one element.

```
<div class = "row">
  <div class = "col-sm|md|lg|xl|xxl-size">
  </div>
  <div class = "col-sm|md|lg|xl|xxl-size">
  </div>
  <div class = "col-sm|md|lg|xl|xxl-size">
  </div>
  </div>
```

Example

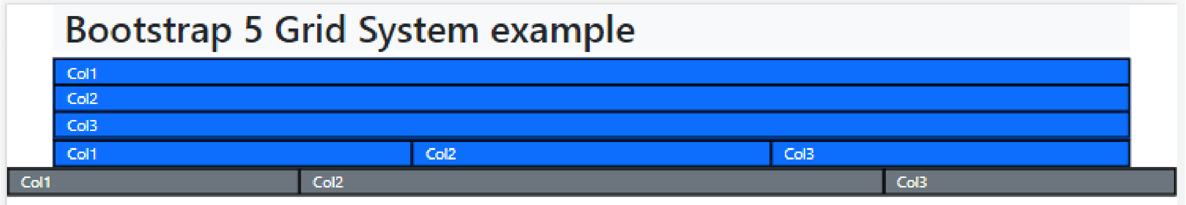
The given example shows default grid system in bootstrap 5 with classes and function. In this example, we use the "col-xxl" and "col-sm" classes in the container and container fluid element.

```
<!DOCTYPE html>
<html lang = "en">
<head>
<title> Bootstrap 5 Example </title>
<meta charset = "utf-8">
<meta name = "viewport" content = "width=device-width, initial-scale = 1">
<link href = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel="stylesheet">
<script src = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js">
</script>
</head>
<body>
<div class="container bg-light">
 <h1> Bootstrap 5 Grid System example </h1>
</div>
<div class="container bg-primary text-white">
<div class = "row" style="border : 1px solid black;">
<div class = "col-xxl-2" style="border : 2px solid black;"> Col1 </div>
<div class = "col-xxl-4" style="border : 2px solid black;"> Col2 </div>
<div class = "col-xxl-6" style="border : 2px solid black;"> Col3 </div>
</div>
<div class = "row" style="border : 1px solid black;">
<div class = "col-sm-4" style="border : 2px solid black;"> Col1 </div>
<div class = "col-sm-4" style="border : 2px solid black;"> Col2 </div>
<div class = "col-sm-4" style="border : 2px solid black;"> Col3 </div>
</div>
</div>
<div class="container-fluid bg-secondary text-white">
```

```
<div class = "row" style="border : 1px solid black;">
<div class = "col-sm-3" style="border : 2px solid black;"> Col1 </div>
<div class = "col-sm-6" style="border : 2px solid black;"> Col2 </div>
<div class = "col-sm-3" style="border : 2px solid black;"> Col3 </div>
</div>
</div>
</body>
</html>
```

Output:

The following output shows the grid system for small and large screen width devices. This grid class uses container and container fluid.



Bootstrap 5 Grid System example

Col1

Col2

Col3

Col1 Col2 Col3

Col1 Col2 Col3

Example with grid class

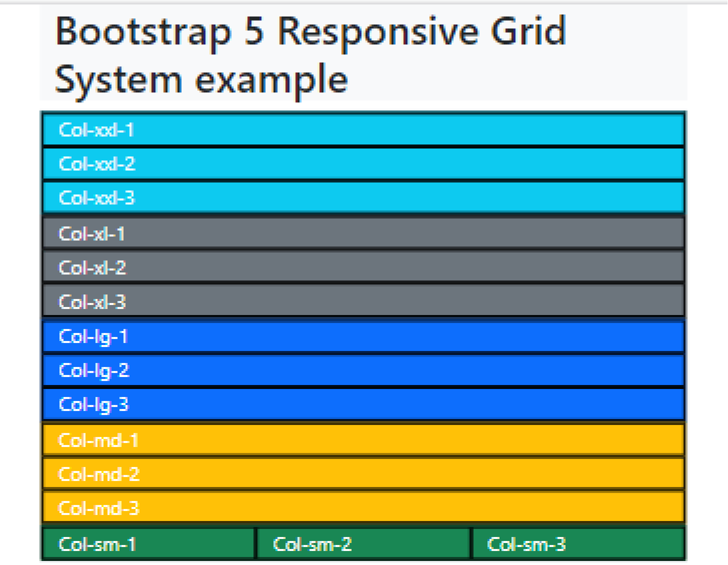
The given example shows default grid system in bootstrap 5 with classes and function. If we need to use for small, medium, large, extra large and double extra large screen width then use "col-sm", "col-md", "col-lg", "col-xl" and "col-xxl" class respectively.

```
<!DOCTYPE html>
<html lang = "en">
<head>
<title> Bootstrap 5 Example </title>
<meta charset = "utf-8">
<meta name = "viewport" content = "width=device-width, initial-scale = 1">
<link href = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel="stylesheet">
<script src = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js">
</script>
```

```
</head>
<body>
<div class="container bg-light">
 <h1> Bootstrap 5 Responsive Grid System example </h1>
</div>
<div class="container text-white border">
<div class = "row bg-info" style="border : 1px solid black;">
<div class = "col-xxl-4" style="border : 2px solid black;"> Col-xxl-1 </div>
<div class = "col-xxl-4" style="border : 2px solid black;"> Col-xxl-2 </div>
<div class = "col-xxl-4" style="border : 2px solid black;"> Col-xxl-3 </div>
</div>
<div class = "row bg-secondary" style="border : 1px solid black;">
<div class = "col-xl-4" style="border : 2px solid black;"> Col-xl-1 </div>
<div class = "col-xl-4" style="border : 2px solid black;"> Col-xl-2 </div>
<div class = "col-xl-4" style="border : 2px solid black;"> Col-xl-3 </div>
</div>
<div class = "row bg-primary" style="border : 1px solid black;">
<div class = "col-lg-4" style="border : 2px solid black;"> Col-lg-1 </div>
<div class = "col-lg-4" style="border : 2px solid black;"> Col-lg-2 </div>
<div class = "col-lg-4" style="border : 2px solid black;"> Col-lg-3 </div>
</div>
<div class = "row bg-warning" style="border : 1px solid black;">
<div class = "col-md-4" style="border : 2px solid black;"> Col-md-1 </div>
<div class = "col-md-4" style="border : 2px solid black;"> Col-md-2 </div>
<div class = "col-md-4" style="border : 2px solid black;"> Col-md-3 </div>
</div>
<div class = "row bg-success" style="border : 1px solid black;">
<div class = "col-sm-4" style="border : 2px solid black;"> Col-sm-1 </div>
<div class = "col-sm-4" style="border : 2px solid black;"> Col-sm-2 </div>
<div class = "col-sm-4" style="border : 2px solid black;"> Col-sm-3 </div>
</div>
</div>
</body>
</html>
```

Output

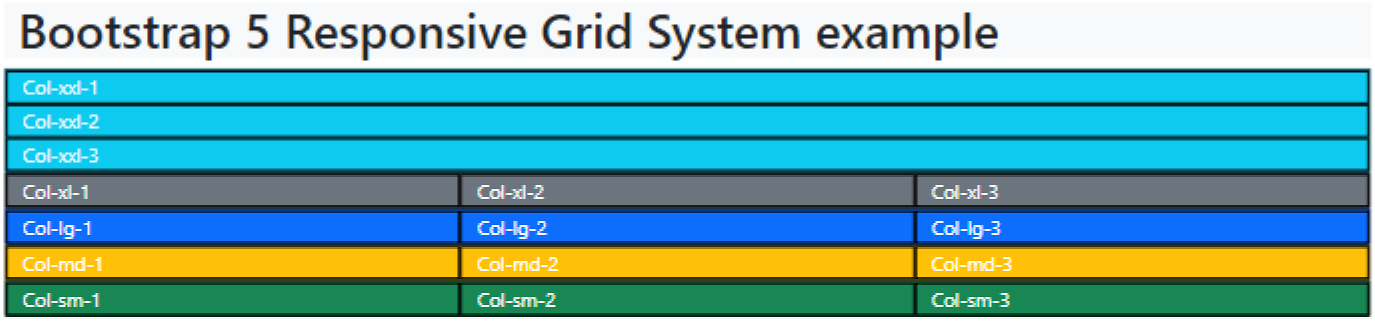
The following output shows the grid system for a small screen width device.



Bootstrap 5 Responsive Grid System example

Col-xxl-1
Col-xxl-2
Col-xxl-3
Col-xl-1
Col-xl-2
Col-xl-3
Col-lg-1
Col-lg-2
Col-lg-3
Col-md-1
Col-md-2
Col-md-3
Col-sm-1 Col-sm-2 Col-sm-3

The following output shows the grid system for a double extra-large screen width device.



Bootstrap 5 Responsive Grid System example

Col-xxl-1
Col-xxl-2
Col-xxl-3
Col-xl-1 Col-xl-2 Col-xl-3
Col-lg-1 Col-lg-2 Col-lg-3
Col-md-1 Col-md-2 Col-md-3
Col-sm-1 Col-sm-2 Col-sm-3

Stacked-to-horizontal

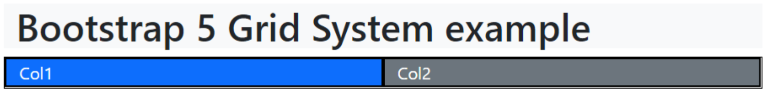
Let's build a simple grid system that starts stacked on extra-small devices and becomes horizontal on larger devices.

Example

The following example demonstrates a simple "stacked-to-horizontal" two-column layout. It will result in a 50 percent or 50 percent split on all displays except exceptionally small screens, which will stack (100 percent). The example given shows the default grid system in bootstrap 5 with classes and functions.

```
<!DOCTYPE html>
<html lang = "en">
<head>
<title> Bootstrap 5 Example </title>
<meta charset = "utf-8">
<meta name = "viewport" content = "width=device-width, initial-scale = 1">
<link href = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel="stylesheet">
<script src = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js">
</script>
</head>
<body>
<div class = "container bg-light">
 <h1> Bootstrap 5 Grid System example </h1>
</div>
<div class = "container text-white">
<div class = "row" style = "border: 1px solid black;">
<div class = "col-sm-6 bg-primary" style = "border: 2px solid black;"> Col1 </div>
<div class = "col-sm-6 bg-secondary" style = "border: 2px solid black;"> Col2 </div>
</div>
</div>
</body>
</html>
```

Output



Bootstrap 5 Grid System example

Col1 Col2

Auto-layout Grid System

Bootstrap will detect the number of columns and ensure that each column has the same width. To establish equal-width columns for all devices with Bootstrap 5, remove the number from .col-size-* and only use the .col-size class on a given number of col components. Bootstrap 5 columns should be responsive using the size classes.

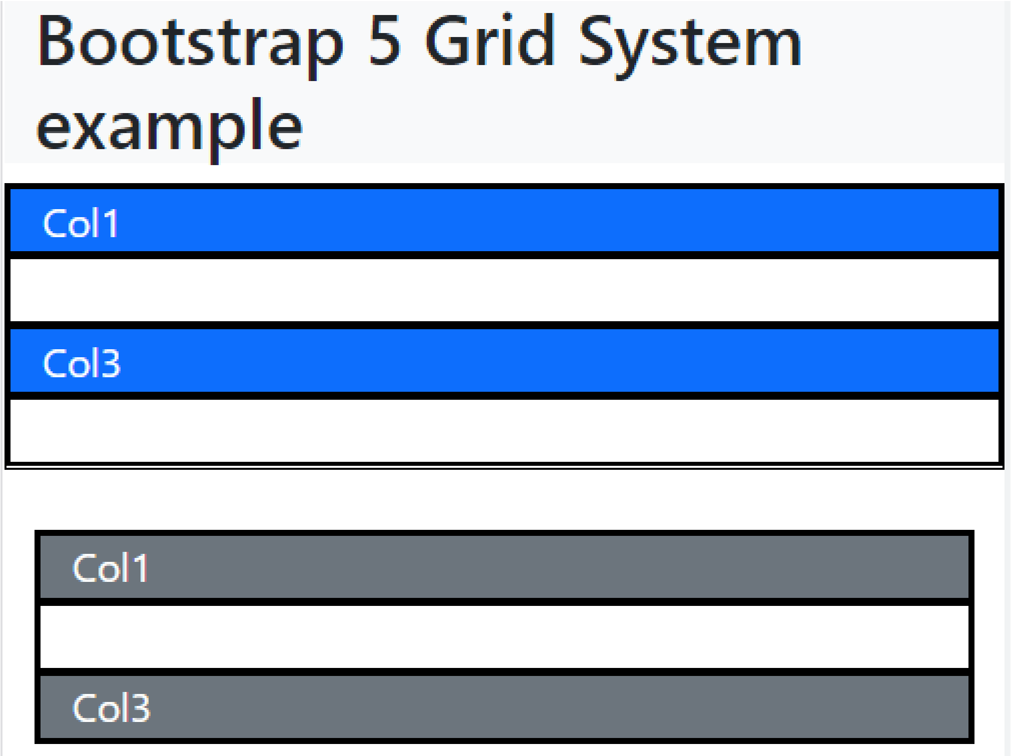
Example

The given example shows auto layout grid system in bootstrap 5 with classes and function. The "col-sm" class uses on the container and container-fluid of the web page. The "col-sm" class uses for four time to create four columns.

```
<!DOCTYPE html>
<html lang = "en">
<head>
<title> Bootstrap 5 Example </title>
<meta charset = "utf-8">
<meta name = "viewport" content = "width=device-width, initial-scale = 1">
<link href = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel="stylesheet">
<script src = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js">
</script>
</head>
<body>
<div class="container bg-light">
 <h1> Bootstrap 5 Grid System example </h1>
</div>
<div class="container text-white">
<div class = "row" style="border : 1px solid black;">
<div class = "col-sm bg-primary" style = "border: 2px solid black;"> Col1 </div>
<div class = "col-sm" style = "border: 2px solid black;"> Col2 </div>
<div class = "col-sm bg-primary" style = "border: 2px solid black;"> Col3 </div>
<div class = "col-sm" style = "border: 2px solid black;"> Col4 </div>
</div>
<div class="container-fluid text-white">
<div class = "row" style = "border : 1px solid black;">
<div class = "col-sm bg-secondary" style = "border : 2px solid black;"> Col1 </div>
<div class = "col-sm" style = "border : 2px solid black;"> Col2 </div>
<div class = "col-sm bg-secondary" style = "border : 2px solid black;"> Col3 </div>
</div>
</div>
```

```
</body>
</html>
```

Output:



Bootstrap 5 Grid System example

Col1

Col3

Col1

Col3

Bootstrap 5 forms

Bootstrap5 forms use to create a function for user interaction. The form element collects and updates information from the user using fields such as checkboxes, radio buttons, or text fields. We can insert and save users' data in the database.

The form controls in Bootstrap 5 provide classes to form designs. Use these classes to enable browser and device-specific rendering. Bootstrap 5 version uses a suitable type property on all inputs to take advantage of current input controls such as email verification, number selection, password, text, etc.

Bootstrap 5 form types

Bootstrap 5 has a form type that allows displaying the form in the right format alongside other functions. There are two kinds of form layouts or types in Bootstrap 5:

- Bootstrap 5 Stacked Forms
- Bootstrap 5 Inline Form

Bootstrap 5 Stacked forms

The bootstrap 5 stacked form generates a full-width form with mandatory input fields such as a text field, an email, a password, and a submit button. To maintain form margins on a web page, use a form wrapper element with the ".form-control" class.

We can use the "form-label" class for the element heading. This form layout is used to make a registration form, a login form, and a contact form.

Syntax

The bootstrap 5 stacked forms syntax shows in below.

```
<form>
<div>
<input type = "input" class = "form-control" placeholder = "name">
</div>
<div>
<input type = "email" class = "form-control" placeholder = "email">
</div>
```

```
</form>
```

Example

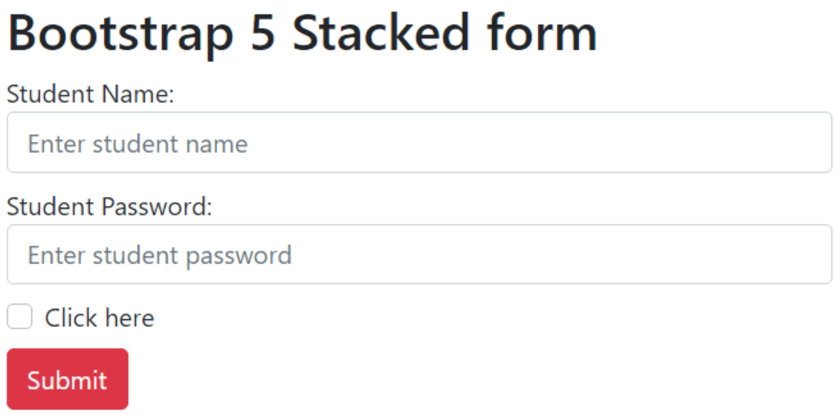
The following example shows us to create stacked form using bootstrap 5 classes. Here, we can use "form-control" class in input element. Each input element needs individual "div" tag.

```
<!DOCTYPE html>
<html lang = "en">
<head>
<title> Bootstrap 5 form Example </title>
<meta charset = "utf-8">
<meta name = "viewport" content = "width=device-width, initial-scale = 1">
<link href = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel = "stylesheet">
<script src = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js"></script>
</head>
<body>
<div class = "container mt-3">
<h1> Bootstrap 5 Stacked form</h1>
<form action = " ">
<div class = "my-2">
<label for = "uname"> Student Name: </label>
<input type = "text" class = "form-control" id = "uname" placeholder = "Enter student name" name = "sname">
</div>
<div class="my-2">
<label for = "pswd"> Student Password: </label>
<input type = "password" class = "form-control" id = "pswd" placeholder = "Enter student password" name="pswd">
</div>
<div class="form-check mb-2">
<label class = "form-check-label" id = "fcl">
```

```
<input class = "form-check-input" type = "checkbox" name = "click" id = "ad"> Click here
</label>
</div>
<button type = "submit" class = "btn btn-danger"> Submit </button>
</form>
</div>
</body>
</html>
```

Output

The output shows stacked form using bootstrap 5 class and element in the below image.



Bootstrap 5 Stacked form

Student Name:

Enter student name

Student Password:

Enter student password

☐ Click here

Submit

Bootstrap 5 Inline Form

An inline form's elements are in a single line and left-aligned. It only applies to forms within viewports with a minimum width of 576px. If screen width is less than 576 px, an inline form displays horizontally by default. The form screen width is greater than the small size screen.

Syntax

The bootstrap 5 stacked forms syntax shows in below. The "form-control" class uses in input element. This element places in the different columns.

```
<form>
<div class = "col">
<input type = "input" class = "form-control" placeholder = "name">
```

```
</div>
<div class = "col">
<input type = "email" class = "form-control" placeholder = "email">
</div>
</div>
</form>
```

Example

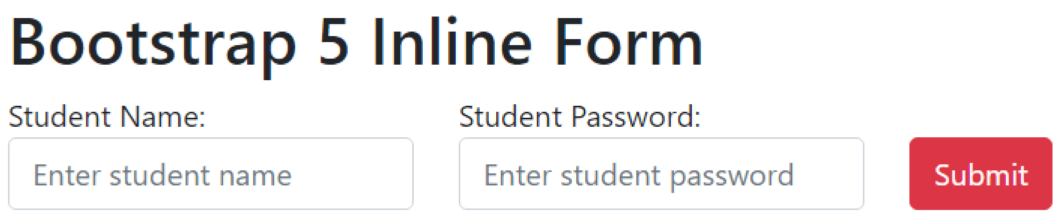
The following example shows us to create stacked form using bootstrap 5 classes. Here, we can use "form-control" class in input element. Each input element needs individual "div" tag.

```
<!DOCTYPE html>
<html lang = "en">
<head>
<title> Bootstrap 5 form Example </title>
<meta charset = "utf-8">
<meta name = "viewport" content="width=device-width, initial-scale=1">
<link href = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel = "stylesheet">
<script src = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js">
</script>
</head>
<body>
<div class = "container mt-3">
<h1> Bootstrap 5 Inline Form</h1>
<form>
<div class = "row">
<div class = "col">
<label for = "uname"> Student Name: </label>
<input type = "text" class = "form-control" id = "uname" placeholder = "Enter student name" name = "sname">
</div>
<div class = "col">
```

```
<label for = "pswd"> Student Password: </label>
<input type = "password" class = "form-control" id = "pswd" placeholder = "Enter student password" name = "pswd">
</div>
<div class = "col mt-4">
<button type = "submit" class = "btn btn-danger"> Submit </button>
</div>
</div>
</form>
</div>
</body>
</html>
```

Output:

The output shows inline form using bootstrap 5 class and element in the below image.



Bootstrap 5 Inline Form

Bootstrap 5 Form control size

Bootstrap 5 version uses a form control class with the required size. We can apply large, medium, and small sizes of the input elements. The form control size uses as per design and user interface. The "form-control" class uses the default size or medium size of the element. The "form-control-lg" and "form-control-sm" classes apply for elements' large and small sizes, respectively.

Syntax

The bootstrap 5 large size form control's syntax shows in below.

```
<input type = "input" class = "form-control-lg" placeholder = "name">
```

The bootstrap 5 medium or default size form control's syntax shows in below.

```
<input type = "input" class = "form-control" placeholder = "name">
```

The bootstrap 5 small size form control's syntax shows in below.

```
<input type = "input" class = "form-control-sm" placeholder = "name">
```

Example

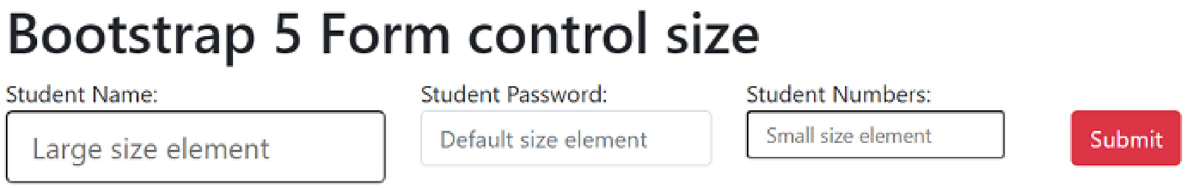
The following example shows us to create a stacked form using bootstrap 5 classes. We can use the "form-control" class in the input element. Each input element places an individual "div" tag with form control size.

- The first element uses a large form control class, the "form-control-lg" class.
- The second element uses the default size form control class, the "form-control" class.
- The third element uses a small size form control class, the "form-control-sm" class.

```
<!DOCTYPE html>
<html lang = "en">
<head>
<title> Bootstrap 5 form Example </title>
<meta charset = "utf-8">
<meta name = "viewport" content = "width = device-width, initial-scale=1">
<link href = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel = "stylesheet">
<script src  = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js">
</script>
</head>
<body>
<div class = "container mt-3">
<h1> Bootstrap 5 Form control size </h1>
<form>
<div class = "row">
<div class = "col">
```

```
<label for = "uname"> Student Name: </label>
<input type = "text" class = "form-control-lg" id = "uname" placeholder = "Large size element" name = "sname">
</div>
<div class = "col">
<label for = "pswd"> Student Password: </label>
<input type = "password" class = "form-control" id = "pswd" placeholder = "Default size element" name = "pswd">
</div>
<div class = "col">
<label for = " nmbr "> Student Numbers: </label>
<input type = "nmbr" class = "form-control-sm" id = "nmbr" placeholder = " Large size element " name = "nmbr">
</div>
<div class = "col mt-4">
<button type = "submit" class = "btn btn-danger"> Submit </button>
</div>
</div>
</form>
</div>
</body>
</html>
```

Output:



Bootstrap 5 Form control size

Student Name:

Large size element

Student Password:

Default size element

Student Numbers:

Small size element

Bootstrap 5 Form Textarea

Bootstrap 5 version uses a textarea element with a form-control class. The textarea element applies the row with the required size. It uses for user interaction, such as written comments and messages in a form.

Syntax

The bootstrap 5 form textarea syntax shows in below.

```
<textarea class = "form-control" rows = "3" id = "text1" name = "text1">
</textarea>
```

Example

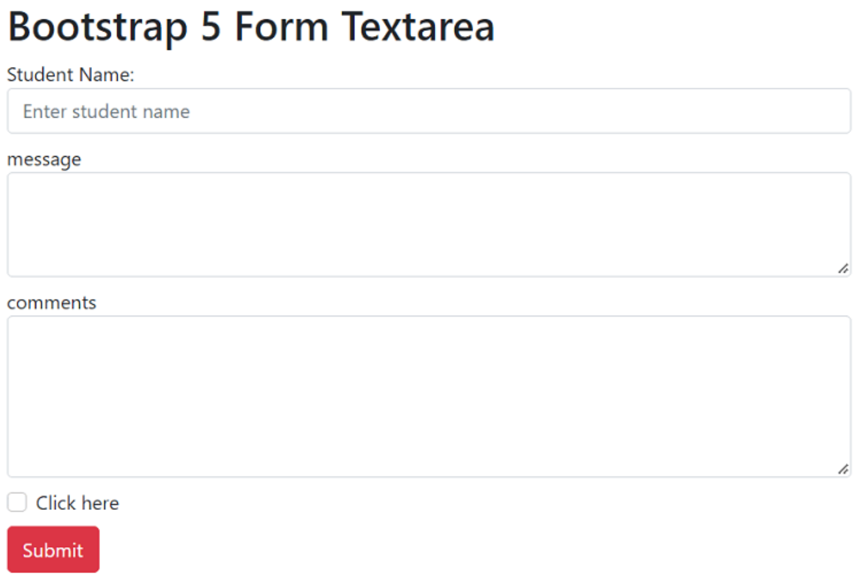
The following example shows us to create textarea in the form using bootstrap 5 classes. Here, we can use "form-control" class in input element with <textarea> tag. Each input element needs individual "div" tag. The "message" input uses <textarea> tag with the "3" rows. The "comment" input uses <textarea> tag with the "5" rows.

```
<!DOCTYPE html>
<html lang = "en">
<head>
<title> Bootstrap 5 form Example </title>
<meta charset = "utf-8">
<meta name = "viewport" content = "width=device-width, initial-scale = 1">
<link href = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel = "stylesheet">
<script src = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js"></script>
</head>
<body>
<div class = "container mt-3">
<h1> Bootstrap 5 Form Textarea </h1>
<form action = " ">
<div class = "my-2">
<label for = "uname"> Student Name: </label>
<input type = "text" class = "form-control" id = "sname" placeholder = "Enter student name" name = "sname">
</div>
<div class="my-2">
<label for = " text1"> message </label>
```

```
<textarea class = "form-control" rows = "3" id = "text1" name = "text1">
</textarea>
</div>
<div class="my-2">
<label for = " text2"> comments </label>
<textarea class = "form-control" rows = "5" id = "text2" name = "text2">
</textarea>
</div>
<div class="form-check mb-2">
<label class = "form-check-label" id = "fcl">
<input class = "form-check-input" type = "checkbox" name = "click" id = "ad"> Click here
</label>
</div>
<button type = "submit" class = "btn btn-danger"> Submit </button>
</form>
</div>
</body>
</html>
```

Output:

The output shows a textarea in a form using the bootstrap 5 class and element below the image.



Bootstrap 5 Form Textarea

Student Name:

Enter student name

message

comments

□ Click here

Bootstrap 5 Readonly and disabled element

Bootstrap 5 uses the "disabled" and "readonly" elements with the form-control class. The disabled keyword uses to show the data or disable a given function. This keyword uses in the input element to disable the input field. The "readonly function uses to read the input field data but does not provide writing access.

Syntax

The bootstrap 5 disabled in form syntax shows in below.

```
<input type = "input" class = "form-control" placeholder = "name" disabled>
```

The bootstrap 5 readonly in form syntax shows in below.

```
<input type = "input" class = "form-control" placeholder = "name" readonly>
```

Example

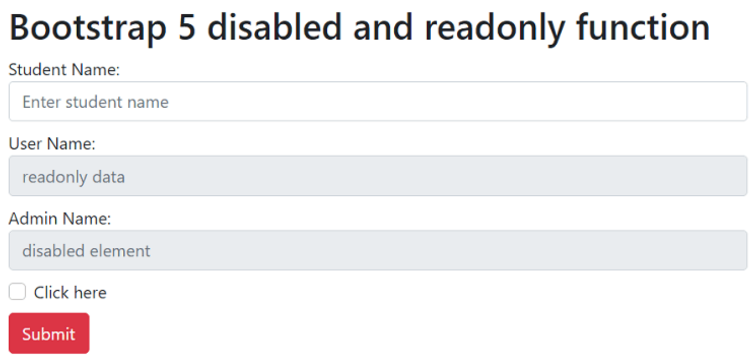
The following example shows us to create a "disable" input element in the form using bootstrap 5 classes and functions. Here, we can use the "form-control" class in the input element with <input> tag. Each input element uses required functions to enable and disables data.

- The first input element uses default function to write data.
- second input function provides reading access but not writing access.
- Third input element disables input function, we can read placeholders value.

```
<!DOCTYPE html>
<html lang = "en">
<head>
<title> Bootstrap 5 form Example </title>
<meta charset = "utf-8">
<meta name = "viewport" content = "width=device-width, initial-scale = 1">
<link href = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel = "stylesheet">
```

```
<script src = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js"></script>
</head>
<body>
<div class = "container mt-3">
<h1> Bootstrap 5 Form disabled and readonly function </h1>
<form action = " ">
<div class = "my-2">
<label for = "uname"> Student Name: </label>
<input type = "text" class = "form-control" id = "sname" placeholder = "Enter student name" name = "sname">
</div>
<div class = "my-2">
<label for = "uname"> User Name: </label>
<input type = "text" class = "form-control" id = "sname" placeholder = "readonly data" name = "sname" readonly>
</div>
<div class = "my-2">
<label for = "uname"> Admin Name: </label>
<input type = "text" class = "form-control" id = "sname" placeholder = "disabled element" name = "sname" disabled>
</div>
<div class="form-check mb-2">
<label class = "form-check-label" id = "fcl">
<input class = "form-check-input" type = "checkbox" name = "click" id = "ad"> Click here
</label>
</div>
<button type = "submit" class = "btn btn-danger"> Submit </button>
</form>
</div>
</body>
</html>
```

Output:



Bootstrap 5 disabled and readonly function

Student Name:

Enter student name

User Name:

readonly data

Admin Name:

disabled element

□ Click here

Bootstrap 5 plaintext form control

Bootstrap 5 version uses the "form-control-plaintext" class to display input elements in simple text. We can read placeholder data and write new information for user interaction. It provides for styling the form element only.

Syntax

The bootstrap 5 plain text with form element syntax shows in below.

```
<input type = "input" class = "form-control-plaintext" placeholder = "name">
```

Example

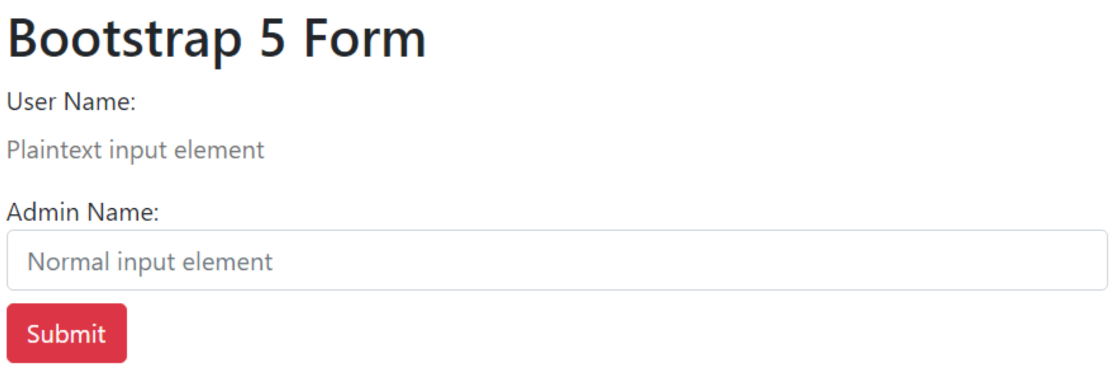
The following example shows us to create input elements with plain text and normal text style data.

- The first input element uses a plain text input element using the "form-control-plaintext" class.
- The second input element uses a normal input element using the "form-control" class.

```
<!DOCTYPE html>
<html lang = "en">
<head>
<title> Bootstrap 5 form Example </title>
<meta charset = "utf-8">
<meta name = "viewport" content = "width=device-width, initial-scale = 1">
<link href = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel = "stylesheet">
```

```
<script src = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js"></script>
</head>
<body>
<div class = "container mt-3">
<h1> Bootstrap 5 Form </h1>
<form action = " ">
<div class = "my-2">
<label for = "sname"> User Name: </label>
<input type = "text" class = "form-control-plaintext" id = "sname" placeholder = "Plaintext input element" name = "sname">
</div>
<div class = "my-2">
<label for = "aname"> Admin Name: </label>
<input type = "text" class = "form-control" id = "aname" placeholder = "Normal input element" name = "aname">
</div>
<button type = "submit" class = "btn btn-danger"> Submit </button>
</form>
</div>
</body>
</html>
```

Output:



Bootstrap 5 Form

User Name:

Plaintext input element

Admin Name:

Bootstrap 5 Select Menus

A bootstrap5 Select menu displays a set of alternatives or information in a form. We can choose one or more options for user interaction.

A select menu is a dropdown list created with the <select> element in Bootstrap 5. The <select> element is most commonly used to collect user input in a form. After the form is submitted, the name property must refer to the form data.

Syntax

The basic bootstrap 5 select menu syntax shows below. We can use <select> tag with the "form-select" class in the form. The <option> tag provides multiple values to select from form element.

```
<select class = "form-select">
  <option> value1 </option>
  <option> value2 </option>
</select>
```

The basic bootstrap 5 select menu of multiple values syntax shows below. The "multiple" keyword shows multiple values in single element.

```
<select multiple class = "form-select">
  <option> value1 </option>
  <option> value2 </option>
  <option> value3 </option>
  <option> value4 </option>
</select>
```

Example

The following example shows us to create a select menu in the form using bootstrap 5 classes.

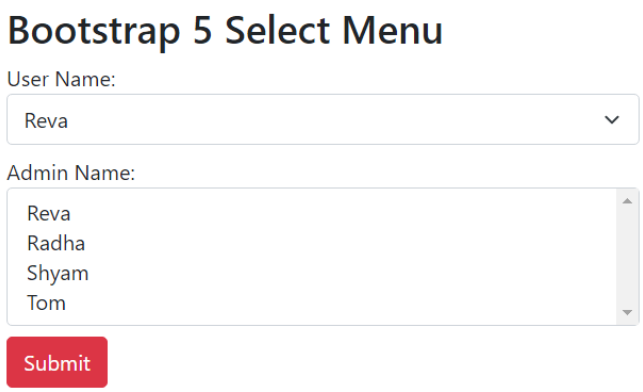
- In "user name" element, we can use "form-select" class in <select> tag. We can see the bootstrap version in the dropdown list for data selection.
- In "admin name" element, we use "multiple" keyword for display all values.

```
<!DOCTYPE html>
<html lang = "en">
<head>
<title> Bootstrap 5 select menus Example </title>
<meta charset = "utf-8">
<meta name = "viewport" content = "width=device-width, initial-scale = 1">
<link href = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel = "stylesheet">
<script src = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js"></script>
</head>
<body>
<div class = "container mt-3">
<h1> Bootstrap 5 Select Menu </h1>
<form action = " ">
<div class = "my-2">
<label for = "uname"> User Name: </label>
<select class = "form-select" id = "uname" name = "uname">
  <option> Reva </option>
  <option> Radha </option>
  <option> Shyam </option>
  <option> Tom </option>
</select>
</div>
<div class = "my-2">
<label for = "uname"> Admin Name: </label>
<select multiple class = "form-select" id = "uname" name = "uname">
    <option> Reva </option>
  <option> Radha </option>
  <option> Shyam </option>
  <option> Tom </option>
</select>
</div>
<button type = "submit" class = "btn btn-danger"> Submit </button>
```

```
</form>
</div>
</body>
</html>
```

Output:

The output shows a select menu using the bootstrap 5 class and element below the image.



Bootstrap 5 Select Menu

User Name: Reva

Admin Name:

Bootstrap 5 Select Menu Size

Bootstrap 5 version uses a "form-select" class with the required size. We can apply large, medium, and small sizes of the select tag with the select menu class. The "form-select" class uses the default size or medium size of the element. The "form-select-lg" and "form-select-sm" classes apply for elements' large and small sizes, respectively.

Syntax

The bootstrap 5 large size form select menu's syntax shows in below.

```
<select class = "form-select-lg">
<option> value1 </option>
</select>
```

The bootstrap 5 medium or default size form select menu's syntax shows in below.

```
<select class = "form-select">
<option> value1 </option>
</select>
```

The bootstrap 5 small size form select menu's syntax shows in below.

```
<select class = "form-select-sm">
<option> value1 </option>
</select>
```

Example

The following example shows us to create a select menu in form using bootstrap 5 classes. We can use the "form-select" class in the "select" element. Each value places an individual "option" tag.

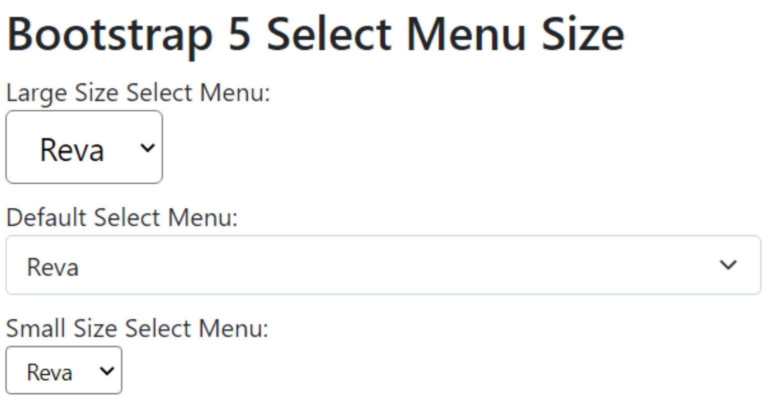
- The first element uses a large form control class, the "form-select-lg" class.
- The second element uses the default size form control class, the "form-select" class.
- The third element uses a small size form control class, the "form-select-sm" class.

```
<!DOCTYPE html>
<html lang = "en">
<head>
<title> Bootstrap 5 form Example </title>
<meta charset = "utf-8">
<meta name = "viewport" content = "width = device-width, initial-scale=1">
<link href = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel = "stylesheet">
<script src  = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js">
</script>
</head>
<body>
<div class = "container mt-3">
<h1> Bootstrap 5 Select Menu Size </h1>
<form>
<div class = "my-2">
<label for = "uname"> Large Size Select Menu: </label> <br>
<select class = "form-select-lg" id = "uname" name = "uname">
 <option> Reva </option>
```

```
  <option> Radha </option>
  <option> Shyam </option>
  <option> Tom </option>
</select>
</div>
<div class = "my-2">
<label for = "uname"> Default Select Menu </label>
<select class = "form-select" id = "uname" name = "uname">
  <option> Reva </option>
  <option> Radha </option>
  <option> Shyam </option>
  <option> Tom </option>
</select>
</div>
<div class = "my-2">
<label for = "uname"> Small Size Select Menu: </label> <br>
<select class = "form-select-sm" id = "uname" name = "uname">
  <option> Reva </option>
  <option> Radha </option>
  <option> Shyam </option>
  <option> Tom </option>
</select>
</div>
</form>
</div>
</body>
</html>
```

Output:

The output shows a select menu with size using bootstrap 5 class and element below image.



Bootstrap 5 Select Menu Size

Large Size Select Menu:

Reva

Default Select Menu:

Reva

Small Size Select Menu:

Reva

Bootstrap 5 disabled Select Menu

Bootstrap 5 uses the "disabled" element with the form-select class. The disabled keyword uses to show the data or disable a select menu function.

Syntax

The basic bootstrap 5 select menu syntax shows below. We can use <select> tag with the "form-select" class in the form. The <option> tag provides multiple values to select from form element.

```
<select class = "form-select" disabled>
  <option> value1 </option>
  <option> value2 </option>
</select>
```

Example

The following example shows us to create select menu in the form using bootstrap 5 classes.

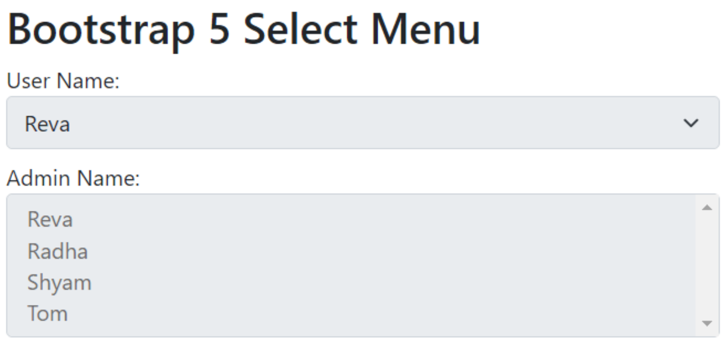
In "user name" and "admin name" elements, we can use "form-select" class in <select> tag. The select tag contains the "disabled" keyword for single or multiple data value.

```
<!DOCTYPE html>
<html lang = "en">
<head>
<title> Bootstrap 5 select menus Example </title>
```

```
<meta charset = "utf-8">
<meta name = "viewport" content = "width=device-width, initial-scale = 1">
<link href = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel = "stylesheet">
<script src = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js"></script>
</head>
<body>
<div class = "container mt-3">
<h1> Bootstrap 5 Select Menu </h1>
<form action = " ">
<div class = "my-2">
<label for = "uname"> User Name: </label>
<select class = "form-select" id = "uname" name = "uname" disabled>
 <option> Reva </option>
 <option> Radha </option>
 <option> Shyam </option>
 <option> Tom </option>
</select>
</div>
<div class = "my-2">
<label for = "uname"> Admin Name: </label>
<select multiple class = "form-select" id = "uname" name = "uname" disabled>
  <option> Reva </option>
 <option> Radha </option>
 <option> Shyam </option>
 <option> Tom </option>
</select>
</div>
</form>
</div>
</body>
</html>
```

Output:

The output shows a select menu using the bootstrap 5 class and element below the image.



Bootstrap 5 Select Menu

User Name:

Reva

Admin Name:

Reva
Radha
Shyam
Tom

Bootstrap 5 Checkboxes

Bootstrap 5 Checkboxes are used when there are lists of possibilities, and the user can choose from one or more than one selection. In other words, bootstrap 5 checkboxes in the list are independent of the others; thus, clicking one box does not uncheck the others.

Bootstrap 5 Checkboxes are used when you want the user to choose from a list of predefined alternatives.

Default Checkbox

Bootstrap 5 default checkboxes show unchecked and non-disabled in the form. Bootstrap 5 checkboxes are used in the form of multiple selections of the single input element. It is a user interaction function and independent from each other.

Syntax

The following syntax shows us to create bootstrap 5 checkbox in form.

```
<div class = "form-check">
  <input class = "form-check-input"
   type = "checkbox" id = "checkbox1" name="option1" value="good" >
  <label class="form-check-label"> First Option </label>
</div>
```

Example

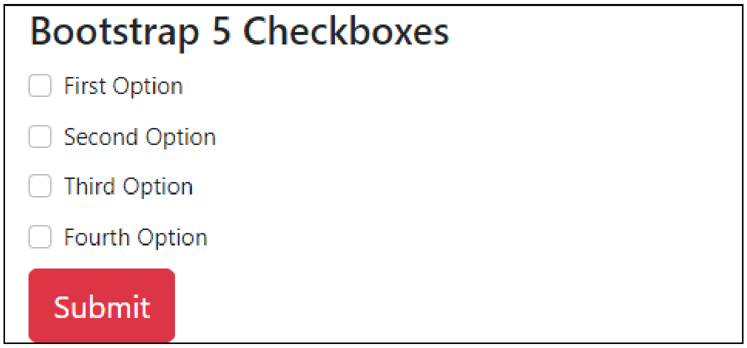
- Use a wrapper element with the class="form-check" to ensure suitable label and checkbox margins to style checkboxes.
- Then, add the ".form-check-label" and ".form-check-input" classes used to label elements and show checkboxes. It places inside of the ".form-check" class of the container.
- We can use the following example and its output to show bootstrap 5 checkboxes.

```
<!DOCTYPE html>
<html lang = "en">
<head>
<title> Bootstrap 5 form Example </title>
```

```
<meta charset = "utf-8">
<meta name = "viewport" content = "width=device-width, initial-scale = 1">
<link href = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel = "stylesheet">
<script src = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js">
</script>
</head>
<body>
<div class = "container mt-3 w-25" style="border:1px solid black;">
<h5> Bootstrap 5 Checkboxes </h5>
<form action = " " style="font-size:12px;">
<div class = "form-check">
  <input class = "form-check-input"
    type = "checkbox" id = "checkbox1" name = "option1" value = "good" >
  <label class = "form-check-label"> First Option </label>
</div>
<div class = "form-check">
  <input class = "form-check-input"
    type = "checkbox" id = "checkbox3" name = "option2" value = "great" >
  <label class = "form-check-label"> Second Option </label>
</div>
<div class = "form-check">
  <input class = "form-check-input"
    type = "checkbox" id = "checkbox3" name = "option3" value = "better" >
  <label class = "form-check-label"> Third Option </label>
</div>
<div class = "form-check">
  <input class = "form-check-input"
    type = "checkbox" id = "checkbox4" name = "option4" value = "best" >
  <label class = "form-check-label"> Fourth Option </label>
</div>
<button type = "submit" class = "btn btn-danger"> Submit </button>
</form>
</div>
```

```
</body>
</html>
```

Output



Bootstrap 5 Checkboxes

□ First Option

□ Second Option

□ Third Option

□ Fourth Option

Submit

Checked Checkbox

If we want the checkbox to be checked by default, use the checked property in the checkbox input element.

Syntax

The following syntax shows us to create bootstrap 5 checked checkbox by default in form.

```
div class = "form-check">
  <input class = "form-check-input"
    type = "checkbox" id = "checkbox1" name = "option1" value = "good" checked>
  <label class = "form-check-label"> First Option </label>
</div>
```

Example

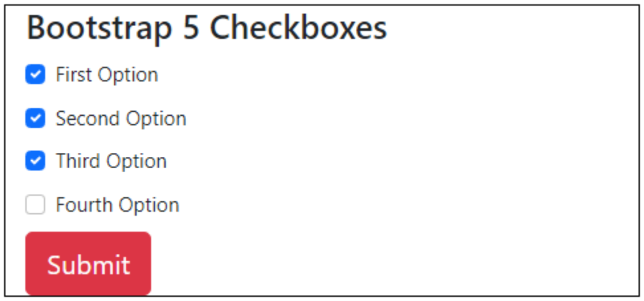
The first three checkboxes check by default using the "checked" keyword. The last checkbox shows unchecked by default.

```
<!DOCTYPE html>
<html lang = "en">
<head>
<title> Bootstrap 5 form Example </title>
<meta charset = "utf-8">
```

```
<meta name = "viewport" content = "width=device-width, initial-scale = 1">
<link href = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel = "stylesheet">
<script src = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js">
</script>
</head>
<body>
<div class = "container mt-3 w-25" style="border:1px solid black;">
<h5> Bootstrap 5 Checkboxes </h5>
<form action = " " style="font-size:12px;">
<div class = "form-check">
  <input class = "form-check-input"
    type = "checkbox" id = "checkbox1" name = "option1" value = "good" checked>
  <label class = "form-check-label"> First Option </label>
</div>
<div class = "form-check">
  <input class = "form-check-input"
    type = "checkbox" id = "checkbox3" name="option2" value="great" checked>
  <label class="form-check-label"> Second Option </label>
</div>
<div class = "form-check">
  <input class = "form-check-input"
    type = "checkbox" id = "checkbox3" name = "option3" value = "better" checked>
  <label class = "form-check-label"> Third Option </label>
</div>
<div class = "form-check">
  <input class = "form-check-input"
    type = "checkbox" id = "checkbox4" name = "option4" value = "best" >
  <label class = "form-check-label"> Fourth Option </label>
</div>
<button type = "submit" class = "btn btn-danger"> Submit </button>
</form>
</div>
</body>
```

```
</html>
```

Output



Bootstrap 5 Checkboxes

☑ First Option

☑ Second Option

☑ Third Option

☐ Fourth Option

Disabled Checkbox

If we want the checkbox to be disabled by default, use the disabled property in the checkbox input element.

Syntax

The following syntax shows us to create bootstrap 5 disabled checkbox in form.

```
<div class = "form-check">
  <input class = "form-check-input"
   type = "checkbox" id = "checkbox1" name = "option1" value = "good" disabled>
  <label class = "form-check-label"> First Option </label>
</div>
```

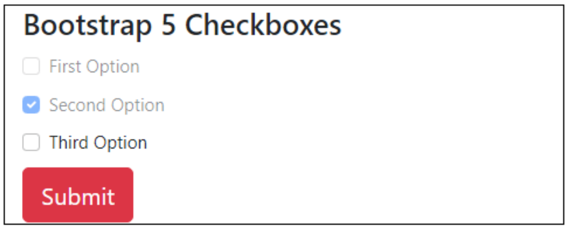
Example

The following example shows disabled, checked and unchecked checkboxes using bootstrap 5 elements. First checkbox uses disabled keyword. Second checkbox uses disabled and checked keyword. Third checkbox shows default checkbox function.

```
<!DOCTYPE html>
<html lang = "en">
<head>
<title> Bootstrap 5 form Example </title>
<meta charset = "utf-8">
```

```
<meta name = "viewport" content = "width=device-width, initial-scale = 1">
<link href = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel = "stylesheet">
<script src = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js">
</script>
</head>
<body>
<div class = "container mt-3 w-25" style="border:1px solid black;">
<h5> Bootstrap 5 Checkboxes </h5>
<form action = " " style="font-size:12px;">
<div class = "form-check">
 <input class = "form-check-input"
  type = "checkbox" id = "checkbox1" name = "option1" value = "good" disabled>
 <label class = "form-check-label"> First Option </label>
</div>
<div class = "form-check">
 <input class = "form-check-input"
  type = "checkbox" id = "checkbox3" name = "option2" value = "great" checked disabled>
 <label class = "form-check-label"> Second Option </label>
</div>
<div class = "form-check">
 <input class = "form-check-input"
  type = "checkbox" id = "checkbox3" name = "option3" value = "better" >
 <label class = "form-check-label"> Third Option </label>
</div>
<button type = "submit" class = "btn btn-danger"> Submit </button>
</form>
</div>
</body>
</html>
```

Output



Bootstrap 5 Checkboxes

□ First Option

☑ Second Option

□ Third Option

Bootstrap 5 Radio Button

Bootstrap 5 Radio buttons are used when there is a list of two or more mutually exclusive options, and the user must select exactly one option. In other words, clicking a non-selected radio button will deselect whatever other button was previously chosen in the list.

Bootstrap 5 Radio buttons are used when you wish to limit the user to only one option from a list of preset selections.

Default Radio button

Bootstrap 5 default radio button shows unchecked and non-disabled in the form. Bootstrap 5 radio button uses in the form of a single selection of the single input element. It is a user interaction function and is dependent on each other.

Syntax

The following syntax shows us to create bootstrap 5 radio button in form.

```
<div class="form-check">
<input type = "radio" class = "form-check-input" id = "btn4" name = "optradio" value = "good">
<label class = "form-check-label" for = "btn4"> Value 4 </label>
</div>
```

Example

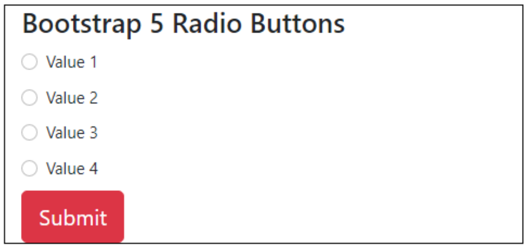
- To style the radio button, use a wrapper element with the class = "form-check" to ensure suitable label and checkbox margins.
- Then, add the ".form-check-label" and ".form-check-input" classes used to label elements and show the radio button. It places inside of the ".form-check" class of the container.
- We can use the following example and its output to show bootstrap 5 checkboxes.

```
<!DOCTYPE html>
<html lang = "en">
<head>
<title> Bootstrap 5 radio button Example </title>
<meta charset = "utf-8">
```

```
<meta name = "viewport" content = "width=device-width, initial-scale = 1">
<link href = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel = "stylesheet">
<script src = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js">
</script>
</head>
<body>
<div class = "container mt-3 w-25" style="border:1px solid black;">
<h5> Bootstrap 5 Radio Buttons </h5>
<form action = " " style="font-size:12px;">
<div class="form-check">
<input type = "radio" class = "form-check-input" id = "btn1" name = "optradio" value = "good">
<label class = "form-check-label" for = "btn1"> Value 1 </label>
</div>
<div class="form-check">
<input type = "radio" class = "form-check-input" id = "btn2" name = "optradio" value = "good">
<label class = "form-check-label" for = "btn2"> Value 2 </label>
</div>
<div class="form-check">
<input type = "radio" class = "form-check-input" id = "btn3" name = "optradio" value = "good">
<label class = "form-check-label" for = "btn3"> Value 3 </label>
</div>
<div class="form-check">
<input type = "radio" class = "form-check-input" id = "btn4" name = "optradio" value = "good">
<label class = "form-check-label" for = "btn4"> Value 4 </label>
</div>
<button type = "submit" class = "btn btn-danger"> Submit </button>
</form>
</div>
</body>
</html>
```

Output

The following image shows the default bootstrap 5 radio buttons in the form.


Bootstrap 5 Radio Buttons

○ Value 1

○ Value 2

○ Value 3

○ Value 4

Checked Radio button

The checked radio button selects the default value using bootstrap 5 elements and keywords. This radio button provides an on the button, but we can uncheck and select another button per requirement. We can check one radio button.

Syntax

```
<div class="form-check">
<input type = "radio" class = "form-check-input" id = "btn2" name = "optradio" value = "good" checked>
<label class = "form-check-label" for = "btn2"> Option </label>
</div>
```

Example

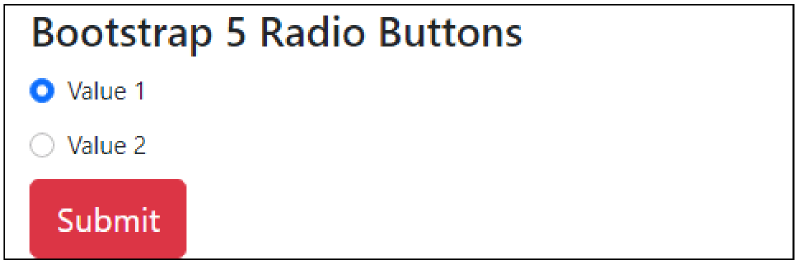
The following example shows one checked and other unchecked radio button. The first element is checked by default but we can select other option using radio button.

```
<!DOCTYPE html>
<html lang = "en">
<head>
<title> Bootstrap 5 radio button Example </title>
<meta charset = "utf-8">
<meta name = "viewport" content = "width=device-width, initial-scale = 1">
<link href = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel = "stylesheet">
<script src = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js">
</script>
</head>
<body>
```

```
<div class = "container mt-3 w-25" style="border:1px solid black;">
<h5> Bootstrap 5 Radio Buttons </h5>
<form action = " " style="font-size:12px;">
 <div class="form-check">
<input type = "radio" class = "form-check-input" id = "btn1" name = "optradio" value = "good" checked>
<label class = "form-check-label" for = "btn1"> Value 1 </label>
</div>
<div class = "form-check">
<input type = "radio" class = "form-check-input" id = "btn2" name = "optradio" value = "good">
<label class = "form-check-label" for = "btn2"> Value 2 </label>
</div>
<button type = "submit" class = "btn btn-danger"> Submit </button>
</form>
</div>
</body>
</html>
```

Output

The following image shows checked bootstrap 5 radio buttons in the form.



Bootstrap 5 Radio Buttons

● Value 1

○ Value 2

Submit

Example

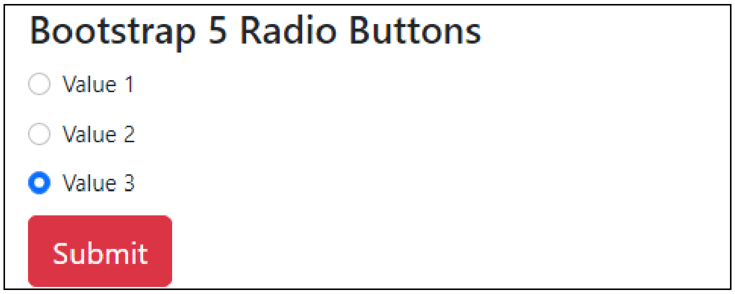
The following example shows two checked and one unchecked radio button. The first element is checked does not display as by default checked button. The last checked button shows as default checked button. we can select other option using radio button.

```
<!DOCTYPE html>
<html lang = "en">
```

```
<head>
<title> Bootstrap 5 radio button Example </title>
<meta charset = "utf-8">
<meta name = "viewport" content = "width=device-width, initial-scale = 1">
<link href = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel = "stylesheet">
<script src = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js">
</script>
</head>
<body>
<div class = "container mt-3 w-25" style="border:1px solid black;">
<h5> Bootstrap 5 Radio Buttons </h5>
<form action = " " style="font-size:12px;">
<div class = "form-check">
<input type = "radio" class = "form-check-input" id = "btn1" name = "optradio" value = "good" checked>
<label class = "form-check-label" for = "btn1"> Value 1 </label>
</div>
<div class = "form-check">
<input type = "radio" class = "form-check-input" id = "btn2" name = "optradio" value = "good">
<label class = "form-check-label" for = "btn2"> Value 2 </label>
</div>
<div class = "form-check">
<input type = "radio" class = "form-check-input" id = "btn3" name = "optradio" value = "good" checked>
<label class = "form-check-label" for = "btn3"> Value 3 </label>
</div>
<button type = "submit" class = "btn btn-danger"> Submit </button>
</form>
</div>
</body>
</html>
```

Output

The following image shows the default bootstrap 5 radio buttons in the form.



Bootstrap 5 Radio Buttons

○ Value 1

○ Value 2

● Value 3

Disabled Radio button

If we want the radio button to be disabled by default, use the disabled property in the bootstrap 5 radio button input element.

Syntax

The following syntax shows us to create bootstrap 5 disabled radio buttons by default in form.

```
<div class = "form-check">
<input type = "radio" class = "form-check-input" id = "btn2" name = "optradio" value = "good" disabled>
<label class = "form-check-label" for = "btn2"> Option </label>
</div>
```

Example

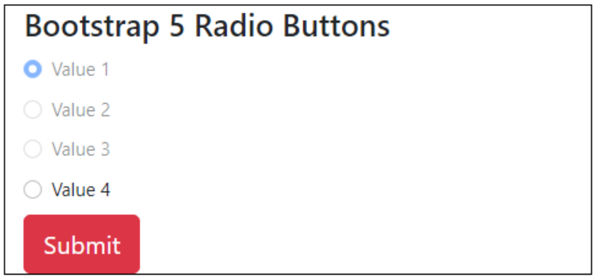
The following example shows two radio button disabled, one radio button enabled and one radio button checked. We can disabled multiple radio button in the form.

```
<!DOCTYPE html>
<html lang = "en">
<head>
<title> Bootstrap 5 radio button Example </title>
<meta charset = "utf-8">
```

```
<meta name = "viewport" content = "width=device-width, initial-scale = 1">
<link href = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel = "stylesheet">
<script src = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js">
</script>
</head>
<body>
<div class = "container mt-3 w-25" style="border:1px solid black;">
<h5> Bootstrap 5 Radio Buttons </h5>
<form action = " " style="font-size:12px;">
<div class = "form-check">
<input type = "radio" class = "form-check-input" id = "btn1" name = "optradio" value = "good" checked disabled>
<label class = "form-check-label" for = "btn1"> Value 1 </label>
</div>
<div class = "form-check">
<input type = "radio" class = "form-check-input" id = "btn2" name = "optradio" value = "good" disabled>
<label class = "form-check-label" for = "btn2"> Value 2 </label>
</div>
<div class = "form-check">
<input type = "radio" class = "form-check-input" id = "btn3" name = "optradio" value = "good" disabled>
<label class = "form-check-label" for = "btn3"> Value 3 </label>
</div>
<div class = "form-check">
<input type = "radio" class = "form-check-input" id = "btn4" name = "optradio" value = "good">
<label class = "form-check-label" for = "btn4"> Value 4 </label>
</div>
<button type = "submit" class = "btn btn-danger"> Submit </button>
</form>
</div>
</body>
</html>
```

Output

The following image shows the default bootstrap 5 radio buttons in the form.



Bootstrap 5 Radio Buttons

● Value 1

○ Value 2

○ Value 3

○ Value 4

Bootstrap 5 Range

Bootstrap 5 range uses custom inputs or values for cross-browser consistency and built-in customization.

The bootstrap 5 range input type allows us to ask the user for a number even if the user does not care or an unaware of the specific numeric value selected. Here are a few examples of how range inputs are typically used: Audio controls such as volume and balance and filter controls.

Bootstrap 5 technology uses classes to create custom range controls. The track or background and value are both styled to look the same in all browsers. We do not now support "filling" their track from the left or right of the thumb to visually indicate progress because only Firefox does.

Bootstrap 5 custom range

The ".form-range" class creates custom range controls with an input element and the "range" type. The custom range works as a simple and default function for the value.

Syntax

The bootstrap 5 custom range element shows syntax below.

```
<label for = "customranges" class = "form-label"> bootstrap 5 range </label>
<input type = "range" class = "form-range" id = "customranges">
```

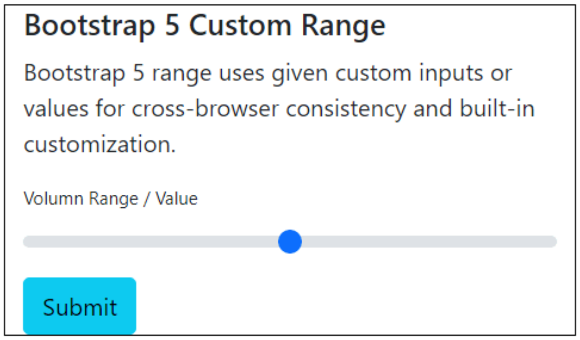
Example

The following example creates custom range using input tag with range type and the "form-range" class in the form.

```
<!DOCTYPE html>
<html lang = "en">
<head>
  <title> Bootstrap 5 Range Example </title>
  <meta charset = "utf-8">
  <meta name = "viewport" content = "width=device-width, initial-scale=1">
  <link href = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel="stylesheet">
  <script src = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js">
  </script>
</head>
<body>
<div class = "container mt-3 w-25" style = "border:1px solid black;">
<h5> Bootstrap 5 Custom Range </h5>
  <p> Bootstrap 5 range uses given custom inputs or values for cross-browser consistency and built-in customization. </p>
  <form action = "" style = "font-size:12px;">
    <label for = "customranges" class = "form-label"> Volumn Range / Value </label>
    <input type = "range" class = "form-range" id = "customranges">
    <button type = "submit" class = "btn btn-info mt-2"> Submit </button>
  </form>
</div>
</body>
</html>
```

Output

The below image shows the custom range using bootstrap 5 classes and elements.



Bootstrap 5 Custom Range

Bootstrap 5 range uses given custom inputs or values for cross-browser consistency and built-in customization.

Volumn Range / Value

Bootstrap 5 Disabled range

The ".form-range" class uses the "disabled" keyword to create disabled range controls. It requires the input element and the "range" type. The disabled range deactivates the function, and we do not use the function.

Syntax

The bootstrap 5 disabled range element shows syntax below.

```
<input type = "range" class = "form-range" id = "customranges" disabled>
```

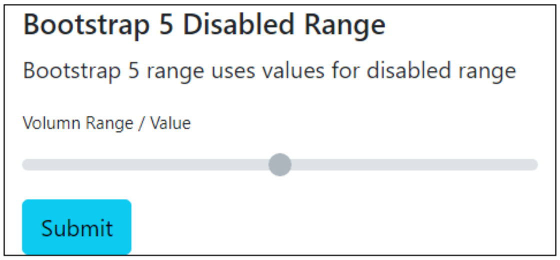
Example

The following example creates disabled range using the "disabled" keyword with range type and the "form-range" class in the form.

```
<!DOCTYPE html>
<html lang = "en">
<head>
  <title> Bootstrap 5 Range Example </title>
  <meta charset = "utf-8">
  <meta name = "viewport" content = "width=device-width, initial-scale=1">
  <link href = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel="stylesheet">
  <script src = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js">
  </script>
</head>
<body>
<div class = "container mt-1 w-25" style="border:1px solid black;">
  <h5> Bootstrap 5 Disabled Range </h5>
  <p> Bootstrap 5 range uses values for disabled range </p>
  <form action = "" style="font-size:12px;">
    <label for = "customranges" class = "form-label"> Volumn Range / Value </label>
    <input type = "range" class = "form-range" id = "customranges" disabled>
    <button type = "submit" class = "btn btn-info mt-2"> Submit </button>
  </form>
</div>
</body>
</html>
```

Output

The below image shows the custom range using bootstrap 5 classes and elements.



Bootstrap 5 Disabled Range

Bootstrap 5 range uses values for disabled range

Volumn Range / Value

Bootstrap 5 min and max range

The range function provides a minimum and maximum value. The minimum and maximum values are 0 and 100, respectively. It requires the input element and the "range" type. The ".form-range" class uses the "min" and "max" keywords with values to control the range.

Syntax

The bootstrap 5 disabled range element shows syntax below.

```
<label for = "customranges" class = "form-label"> bootstrap 5 range </label>
<input type = "range" class = "form-range" id = "customranges" min = "0" max = "5" >
```

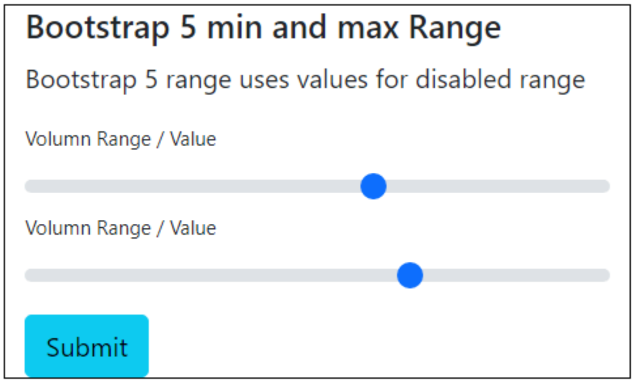
Example

The following example creates minimum and maximum value's range using the elements in the form.

```
 <!DOCTYPE html>
<html lang = "en">
<head>
  <title> Bootstrap 5 Range Example </title>
  <meta charset = "utf-8">
  <meta name = "viewport" content = "width=device-width, initial-scale=1">
  <link href = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel="stylesheet">
  <script src = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js">
  </script>
</head>
<body>
<div class = "container mt-1 w-25" style = "border:1px solid black;">
  <h5> Bootstrap 5 min and max Range </h5>
  <p> Bootstrap 5 range uses values for disabled range </p>
  <form action = "" style = "font-size:12px;">
    <label for = "customranges" class = "form-label"> Volumn Range / Value </label>
    <input type = "range" class = "form-range" id = "customranges" min = "0" max = "5" >
<label for = "customranges" class = "form-label"> Volumn Range / Value </label>
    <input type = "range" class = "form-range" id = "customranges" min = "0" max = "3" >
    <button type = "submit" class = "btn btn-info mt-2"> Submit </button>
  </form>
</div>
</body>
</html>
```

Output

The below image shows the minimum and maximum range using bootstrap 5 classes and elements.



Bootstrap 5 min and max Range

Bootstrap 5 range uses values for disabled range

Volumn Range / Value

Volumn Range / Value

Bootstrap 5 step range

A bootstrap 5 range function provides steps to control function with data. The interval between the range numbers is set to 1 by default. The "step" attribute allows us to adjust it. This function requires the input element and the "range" type. The ".form-range" class uses the "step" keyword with value to control the range.

Syntax

The bootstrap 5 step range element shows syntax below.

```
<label for = "customranges" class = "form-label"> bootstrap 5 range </label>
<input type = "range" class = "form-range" id = "customranges" step = "5" >
```

Example

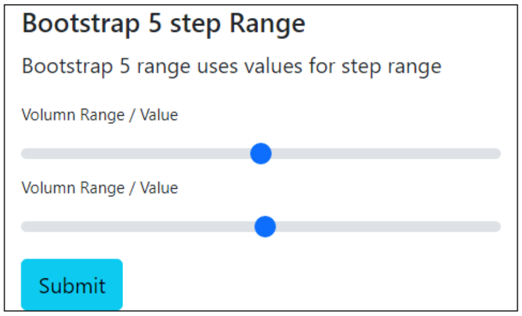
The following example creates step range using the "step" keyword with value and the "form-range" class in the form.

```
<!DOCTYPE html>
<html lang = "en">
<head>
  <title> Bootstrap 5 Range Example </title>
  <meta charset = "utf-8">
  <meta name = "viewport" content = "width=device-width, initial-scale=1">
  <link href = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel="stylesheet">
  <script src = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js">
  </script>
</head>
<body>
<div class = "container mt-1 w-25" style = "border:1px solid black;">
  <h5> Bootstrap 5 step Range </h5>
  <p> Bootstrap 5 range uses values for step range </p>
  <form action = "" style = "font-size:12px;">
    <label for = "customranges" class = "form-label"> Volumn Range / Value </label>
    <input type = "range" class = "form-range" id = "customranges"  step = "5" >
<label for = "customranges" class = "form-label"> Volumn Range / Value </label>
    <input type = "range" class = "form-range" id = "customranges" step = "3" >
    <button type = "submit" class = "btn btn-info mt-2"> Submit </button>
  </form>
```

```
</div>
</body>
</html>
```

Output

The below image shows the step range using bootstrap 5 classes and elements.



Bootstrap 5 step Range

Bootstrap 5 range uses values for step range

Volumn Range / Value

Volumn Range / Value

Bootstrap 5 Input Groups

The input-group class is a container for enhancing an input by placing a symbol, text, or button in front of or behind the input field. It helps to understand the form text using bootstrap 5 classes. To add the help text in front of the input, use the ".input-group" and add the form.

The input group uses a checkbox, radio button, dropdown list and another form element for form elements. This function uses the form element for user interaction. The bootstrap function creates an attractive and responsive input group in the web pages.

Basic input groups

The ".input-group" class is a container for enhancing an input by placing an icon, text, or button in front of or behind the input field as "help text." Use the ".input-group-text" class to style the provided help text.

Syntax

The following syntax shows us to create bootstrap 5 input groups.

```
<div class = "input-group">
    <span class = "input-group-text"> text or symbol </span>
    <input type = "text" class = "form-control" placeholder = "write element">
  </div>
```

Example

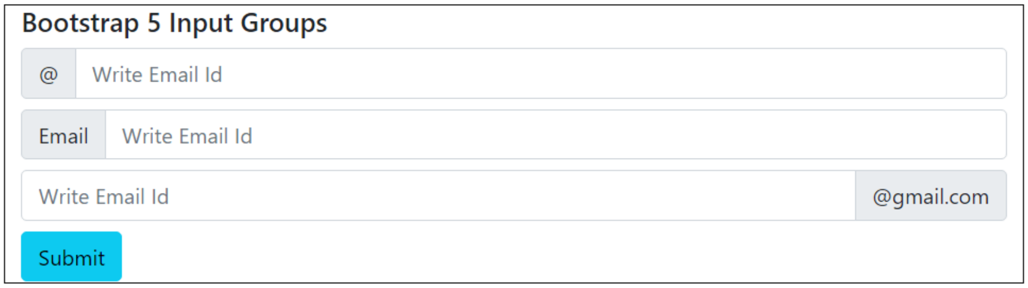
The following example shows input group values before and after input elements. We can write the "input-group-text" class with span tag for front and behind the input value elements. The "input-group" class contains value, symbol, and short text of the input elements.

```
<!DOCTYPE html>
<html lang = "en">
<head>
<title> Bootstrap 5 Input Groups Example </title>
<meta charset = "utf-8">
<meta name = "viewport" content = "width=device-width, initial-scale=1">
```

```
<link href = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel = "stylesheet">
<script src = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js">
</script>
</head>
<body>
<div class = "container mt-3 w-50" style = "border:1px solid black;">
<h5> Bootstrap 5 Input Groups </h5>
<form action = "" style = "font-size:12px;">
<div class = "input-group">
<span class = "input-group-text"> @ </span>
<input type = "text" class = "form-control" placeholder = "Write Email Id">
</div>
<div class = "input-group mt-2">
<span class = "input-group-text"> Email </span>
<input type = "text" class = "form-control" placeholder = "Write Email Id">
</div>
<div class = "input-group mt-2">
<input type = "text" class = "form-control" placeholder = "Write Email Id">
<span class = "input-group-text"> @gmail.com </span>
</div>
<button type = "submit" class = "btn btn-info mt-2"> Submit </button>
</form>
</div>
</body>
</html>
```

Output

The following image shows the input group before and after the input element.



Bootstrap 5 Input Groups

Input groups with size

The ".input-group" class is a container for enhancing an input with the required size. Use the ".input-group-sm" and the ".input-group-lg" classes to provide the input group's small and large sizes in the form.

Syntax

The following syntax shows us to create bootstrap 5 input groups for small size.

```
<div class = "input-group input-group-sm">
  <span class = "input-group-text"> text or symbol </span>
  <input type = "text" class = "form-control" placeholder = "write element">
 </div>
```

The following syntax shows us to create bootstrap 5 input groups for default size.

```
<div class = "input-group">
  <span class = "input-group-text"> text or symbol </span>
  write input element here!
 </div>
```

The following syntax shows us to create bootstrap 5 input groups for large size.

```
<div class = "input-group input-group-lg">
  <span class = "input-group-text"> text or symbol </span>
  write input element here!
 </div>
```

Example

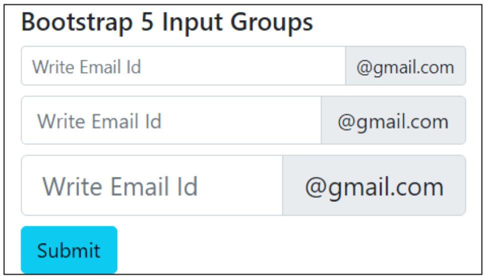
The following example shows input group after input elements with required size. We can write the "input-group-sm", "input-group", and "input-group-lg" classes for small, default, and large input group respectively.

```
<!DOCTYPE html>
<html lang = "en">
<head>
<title> Bootstrap 5 Input Groups Example </title>
<meta charset = "utf-8">
<meta name = "viewport" content = "width=device-width, initial-scale=1">
<link href = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel="stylesheet">
<script src = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js">
</script>
</head>
<body>
<div class = "container mt-2 w-50" style = "border:1px solid black;">
<h5> Bootstrap 5 Input Groups </h5>
<form action = "" style = "font-size:12px;">
<div class = "input-group mt-2 input-group-sm">
<input type = "text" class = "form-control" placeholder = "Write Email Id">
<span class = "input-group-text"> @gmail.com </span>
</div>
<div class = "input-group mt-2 input-group">
<input type = "text" class = "form-control" placeholder = "Write Email Id">
<span class = "input-group-text"> @gmail.com </span>
</div>
<div class = "input-group mt-2 input-group-lg">
<input type = "text" class = "form-control" placeholder = "Write Email Id">
<span class = "input-group-text"> @gmail.com </span>
</div>
<button type = "submit" class = "btn btn-info mt-2"> Submit </button>
</form>
```

```
</div>
</body>
</html>
```

Output

The following image shows the input group with the required size per the input element format.



Bootstrap 5 Input Groups

Multiple Input Groups and Elements

The ".input-group" class is a container containing the symbol and input element. The "input-group-text" class uses multiple times with input elements to display multiple input groups.

Syntax

The following syntax shows us to create multiple input groups with input elements.

```
<div class = "input-group">
   <span class = "input-group-text"> text </span>
   <span class = "input-group-text"> symbol </span>
   write input element here!
 </div>
```

The following syntax shows us to create input groups with multiple input elements.

```
<div class = "input-group input-group-sm">
   <span class = "input-group-text"> text or symbol </span>
   <input type = "text" class = "form-control" placeholder = "write first element">
```

```
<input type = "text" class = "form-control" placeholder = "write second element">
  </div>
```

Example

The following example shows form with multiple form elements and input group functions.

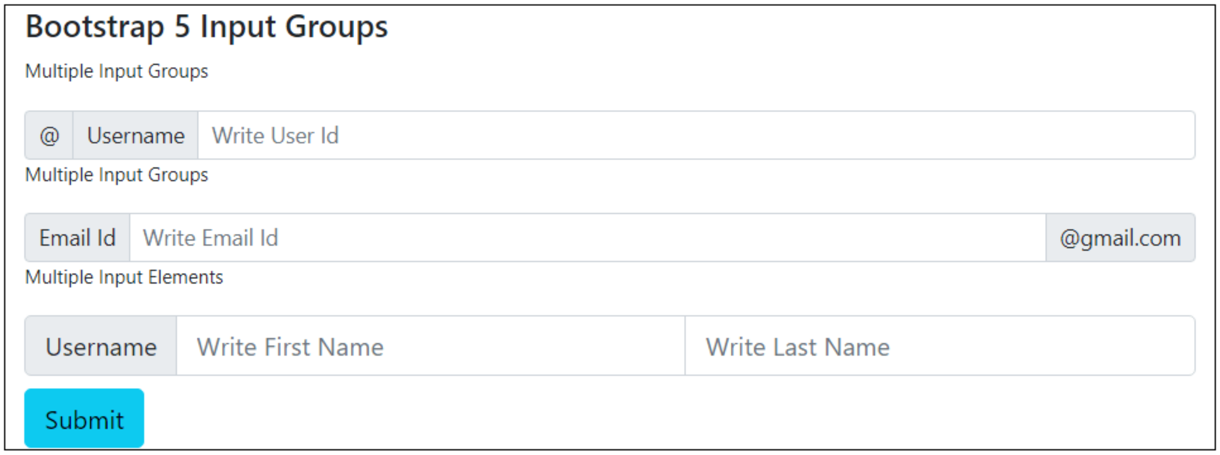
- First two elements use multiple input groups using the "span" tag and the "input-group-text" class.
- Last element uses with single input group and multiple form elements.

```
<!DOCTYPE html>
<html lang = "en">
<head>
<title> Bootstrap 5 Input Groups Example </title>
<meta charset = "utf-8">
<meta name = "viewport" content = "width=device-width, initial-scale=1">
<link href = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel="stylesheet">
<script src = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js">
</script>
</head>
<body>
<div class = "container mt-2 w-50" style = "border:1px solid black;">
<h5> Bootstrap 5 Input Groups </h5>
 <form action = "" style = "font-size:12px;">
<p> Multiple Input Groups </p>
<div class = "input-group mt-2 input-group-sm">
<span class = "input-group-text"> @ </span>
<span class = "input-group-text"> Username </span>
<input type = "text" class = "form-control" placeholder = "Write User Id">
</div>
<p> Multiple Input Groups </p>
<div class = "input-group mt-2 input-group-sm">
<span class = "input-group-text"> Email Id </span>
```

```
<input type = "text" class = "form-control" placeholder = "Write Email Id">
<span class = "input-group-text"> @gmail.com </span>
</div>
<p> Multiple Input Elements </p>
<div class = "input-group mt-2 input-group">
<span class = "input-group-text"> Username </span>
<input type = "text" class = "form-control" placeholder = "Write First Name">
<input type = "text" class = "form-control" placeholder = "Write Last Name">
</div>
<button type = "submit" class = "btn btn-info mt-2"> Submit </button>
</form>
</div>
</body>
</html>
```

Output

The following output image shows multiple input groups and input elements.



Input Groups radio buttons and checkbox

The ".input-group" class is a container containing the radio button and checkbox with the input element. The "input-group-text" class uses in the "div" tag. The "checkbox" and "radio" types uses inside of the "input-group-text" class.

Syntax

The following syntax shows us to create input groups with checkbox input elements.

```
<div class = "input-group">
  <div class = "input-group-text">
    <input type = "checkbox">
  </div>
    Write input element here!
 </div>
```

The following syntax shows us to create input groups with radio button.

```
<div class = "input-group">
  <div class = "input-group-text">
    <input type = "radio">
  </div>
    Write input element here!
 </div>
```

Example

The following example shows form with multiple form elements and input group functions.

- First two elements use checkbox symbol in input groups using the "span" tag and the "input-group-text" class.
- Last two elements use radio button function in input groups using the "span" tag and the "input-group-text" class.

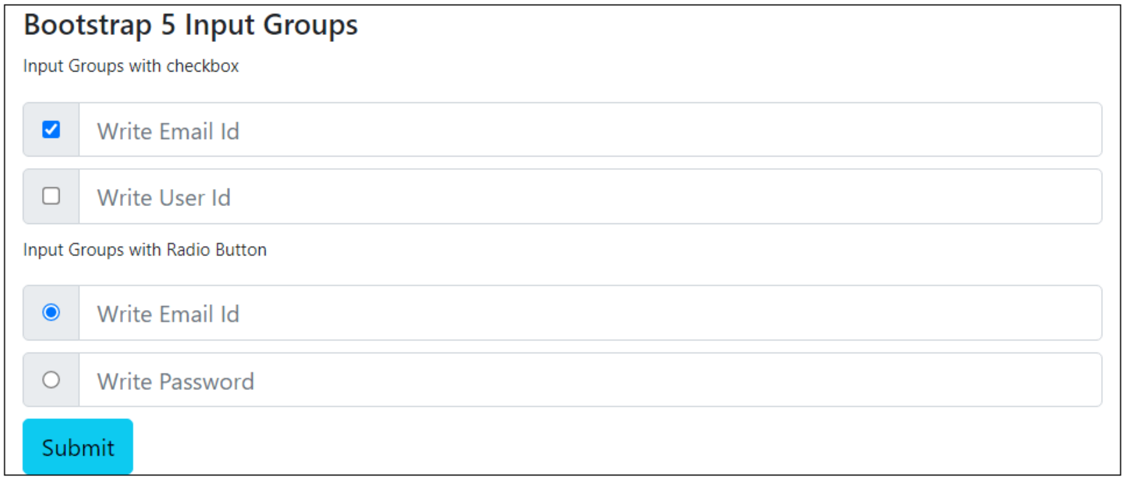
```
<!DOCTYPE html>
<html lang = "en">
<head>
<title> Bootstrap 5 Input Groups Example </title>
<meta charset = "utf-8">
```

```
<meta name = "viewport" content = "width=device-width, initial-scale=1">
<link href = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel="stylesheet">
<script src = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js">
</script>
</head>
<body>
<div class = "container mt-2 w-50" style = "border:1px solid black;">
<h5> Bootstrap 5 Input Groups </h5>
 <form action = "" style = "font-size:12px;">
<p> Input Groups with checkbox </p>
<div class = "input-group my-2 ">
    <div class = "input-group-text">
        <input type = "checkbox">
    </div>
        <input type = "text" class = "form-control" placeholder = "Write Email Id">
  </div>
<div class = "input-group mt-2 mb-2">
    <div class = "input-group-text">
        <input type = "checkbox">
    </div>
        <input type = "text" class = "form-control" placeholder = "Write User Id">
  </div>
<p> Input Groups with Radio Button </p>
<div class = "input-group mt-2 ">
    <div class = "input-group-text">
        <input type = "radio">
    </div>
        <input type = "text" class = "form-control" placeholder = "Write Email Id">
  </div>
<div class = "input-group mt-2 ">
    <div class = "input-group-text">
        <input type = "radio">
    </div>
```

```
        <input type = "text" class = "form-control" placeholder = "Write Password">
  </div>
<button type = "submit" class = "btn btn-info mt-2"> Submit </button>
</form>
</div>
</body>
</html>
```

Output

The following output image shows input groups with a checkbox and radio button for input elements.



Bootstrap 5 Input Groups

Input Groups with checkbox

☑ Write Email Id

☐ Write User Id

Input Groups with Radio Button

● Write Email Id

○ Write Password

Input Groups buttons

The ".input-group" class is a container for containing the button class as an input group with an input element. The "input-group-text" class does not use the symbol, value and text. The "btn btn-primary" class uses inside the "input-group" class.

Syntax

The following syntax shows us to create input groups with form buttons.

```
<div class = "input-group">
    <button class = "btn btn-primary" type = "button"> Input Button </button>
   <input type = "text" class = "form-control" placeholder = "write first element">
```

```
 </div>
```

Example

The following example shows form with multiple form elements and input group functions.

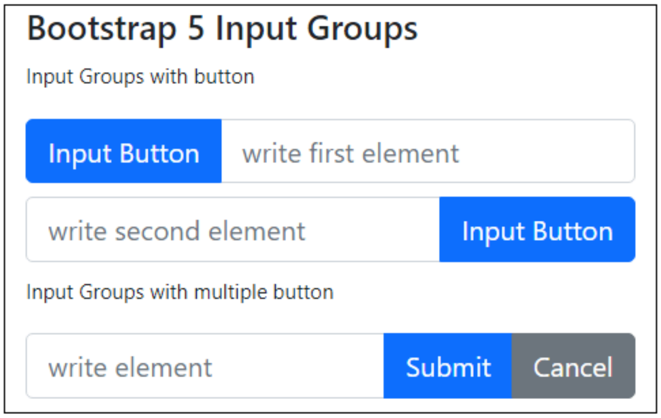
- First elements use primary button as an input groups in the "input-group" class.
- Second elements use primary button as a input groups in the "input-group" class after input element.
- Third elements use primary and secondary buttons as a input groups in the "input-group" class for multiple input groups.

```
<!DOCTYPE html>
<html lang = "en">
<head>
<title> Bootstrap 5 Input Groups Example </title>
<meta charset = "utf-8">
<meta name = "viewport" content = "width=device-width, initial-scale=1">
<link href = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel="stylesheet">
<script src = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js">
</script>
</head>
<body>
<div class = "container mt-2 w-25" style = "border:1px solid black;">
<h5> Bootstrap 5 Input Groups </h5>
 <form action = "" style = "font-size:12px;">
<p> Input Groups with button </p>
<div class = "input-group mt-2 mb-2">
   <button class = "btn btn-primary" type = "button"> Input Button </button>
  <input type = "text" class = "form-control" placeholder = "write first element">
 </div>
<div class = "input-group mt-2 mb-2">
  <input type = "text" class = "form-control" placeholder = "write second element">
```

```
   <button class = "btn btn-primary" type = "button"> Input Button </button>
  </div>
<p> Input Groups with multiple button </p>
<div class = "input-group mt-2 mb-2">
      <input type = "text" class = "form-control" placeholder = "write element">
<button class = "btn btn-primary" type = "button"> Submit </button>
<button class = "btn btn-secondary" type = "button"> Cancel </button>
  </div>
</form>
</div>
</body>
</html>
```

Output

The following output image shows input groups with multiple buttons for input elements.



Bootstrap 5 Input Groups

Input Groups with button

Input Groups with multiple button

Input Groups with dropdown

The ".input-group" class contains the dropdown list as an input group. The ".input-group" class is a container containing the dropdown list as an input group with an input element. The "dropdown-toggle" class uses for selecting data through the dropdown list. The "dropdown-item" class contains data or items in the dropdown function.

Syntax

The following syntax shows us to create dropdown list as an input group with form elements.

```
<div class = "input-group">
<button type = "button" class = "btn btn-success dropdown-toggle" data-bs-toggle = "dropdown">
Dropdown Link
</button>
<ul class = "dropdown-menu">
<li>
<a class = "dropdown-item" href = "www.google.com"> Data1 </a>
</li>
<li>
<a class = "dropdown-item" href = "#"> Data2 </a>
</li>
<li>
<a class = "dropdown-item" href = "#"> Data3 </a>
</li>
</ul>
<input type = "text" class = "form-control" placeholder = " Enter UserId">
</div>
</div>
```

Example

The following example shows form with multiple form elements and input group functions.

- First elements use dropdown list with the success button as an input group.
- Second elements use dropdown list with simple button after input element.
- Third elements use multiple dropdown lists as an input groups. Here, input groups uses warning and danger contextual buttons.

```
<!DOCTYPE html>
<html lang = "en">
<head>
<title> Bootstrap 5 Input Groups Example </title>
<meta charset = "utf-8">
```

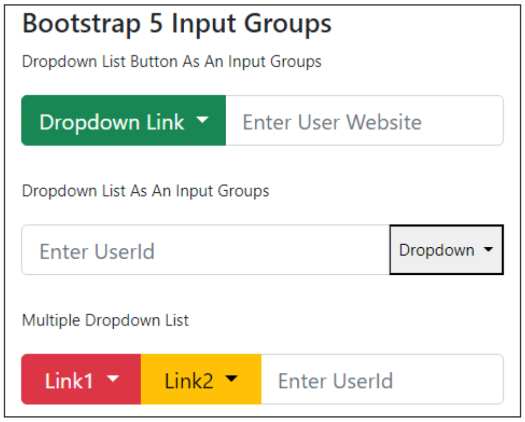
```
<meta name = "viewport" content = "width=device-width, initial-scale=1">
<link href = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel="stylesheet">
<script src = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js">
</script>
</head>
<body>
<div class = "container mt-2 w-25" style = "border:1px solid black;">
<h5> Bootstrap 5 Input Groups </h5>
 <form action = "" style = "font-size:12px;">
<p> Dropdown List Button As An Input Groups </p>
<div class = "input-group">
<button type = "button" class = "btn btn-success dropdown-toggle" data-bs-toggle = "dropdown">
Dropdown Link
</button>
<ul class = "dropdown-menu">
<li>
<a class = "dropdown-item" href = "www.google.com"> google </a>
</li>
<li>
<a class = "dropdown-item" href = "www.javatpoint.com"> JavaTpoint </a>
</li>
<li>
<a class = "dropdown-item" href = "www.gmail.com"> Gmail </a>
</li>
</ul>
<input type = "text" class = "form-control" placeholder = "Enter User Website">
</div>
<p class = "mt-4"> Dropdown List As An Input Groups </p>
<div class = "input-group">
<input type = "text" class = "form-control" placeholder = "Enter UserId">
<button type = "button" class = "dropdown-toggle" data-bs-toggle = "dropdown">
Dropdown
```

```
</button>
<ul class = "dropdown-menu">
<li>
<a class = "dropdown-item" href = "#"> Student </a>
</li>
<li>
<a class = "dropdown-item" href = "#"> Teacher </a>
</li>
<li>
<a class = "dropdown-item" href = "#"> Intern </a>
</li>
<li>
<a class = "dropdown-item" href = "#"> Employee </a>
</li>
</ul>
</div>
<p class = "mt-4"> Multiple Dropdown List </p>
<div class = "input-group mb-2">
<button type = "button" class = "btn btn-danger dropdown-toggle w-25" data-bs-toggle = "dropdown">
Link1
</button>
<ul class = "dropdown-menu">
<li>
<a class = "dropdown-item" href = "www.google.com"> google </a>
</li>
<li>
<a class = "dropdown-item" href = "www.javatpoint.com"> JavaTpoint </a>
</li>
</ul>
<button type = "button" class = " btn btn-warning dropdown-toggle w-25" data-bs-toggle = "dropdown">
Link2
</button>
```

```
<ul class = "dropdown-menu">
<li>
<a class = "dropdown-item" href = "#"> Teacher </a>
</li>
<li>
<a class = "dropdown-item" href = "#"> Employee </a>
</li>
</ul>
<input type = "text" class = "form-control" placeholder = "Enter UserId">
</div>
</form>
</div>
</body>
</html>
```

Output

The following output image shows input groups with a dropdown list for input elements.



Bootstrap 5 Input Groups

Dropdown List Button As An Input Groups

Dropdown List As An Input Groups

Multiple Dropdown List

Bootstrap 5 Floating Labels

The "form-floating" works to enable floating labels with text form fields in the Bootstrap 5 version. A bootstrap 5 floating label is a text label that shows at full font size within an input field. The label "floats" above the input field when interacting, allowing the user to enter a value.

A bootstrap 5 Floating labels indicate the required kind of input for a field. Except for full-width text fields, which employ the input's placeholder attribute, every Text Field and Select should have a label. Labels are always visible and aligned with the input line. They can be floating or resting. The label is a textual description or heading for the Text Field.

Default floating labels

Bootstrap 5 floating labels are also called animated labels. Enclose a pair of <input class = "form-control"> and <label> components in the ".form-floating" class to enable floating labels with Bootstrap's text field inputs. The placeholder uses for the CSS style in the floating label.

Syntax

The following syntax shows how to create default floating or animated label.

```
<div class = "form-floating">
<input type = "text" class = "form-control" id = "ui" placeholder = "Enter UserId" name = "ui">
<label for = "ui"> UserId </label>
</div>
```

Example

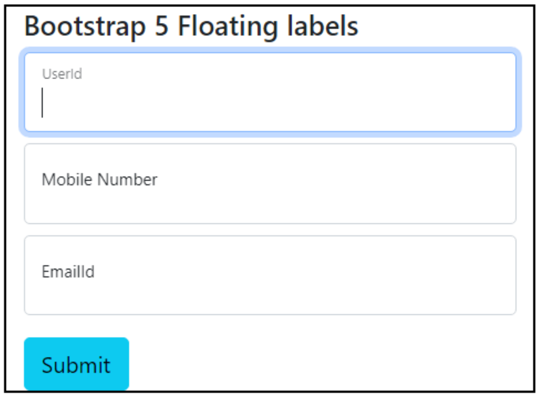
The following example uses the "form-floating" class in the <div> tag. The <input> and <label> tag uses for the insert data and display animated label respectively. Here, example shows the "text" type in the "input" tag but we can use email, number and other input types.

```
<!DOCTYPE html>
<html lang = "en">
<head>
<title> Bootstrap 5 Floating labels Example </title>
```

```
<meta charset = "utf-8">
<meta name = "viewport" content = "width=device-width, initial-scale=1">
<link href = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel="stylesheet">
<script src = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js">
</script>
</head>
<body>
<div class = "container mt-2 w-25" style = "border:2px solid black;">
<h5> Bootstrap 5 Floating labels </h5>
<form action = "" style = "font-size:12px;">
<div class = "form-floating mt-2 mb-2">
<input type = "text" class = "form-control" id = "ui" placeholder = "Enter UserId" name = "ui">
<label for = "ui"> UserId </label>
</div>
<div class = "form-floating mt-2 mb-2">
<input type = "text" class = "form-control" id = "mobile" placeholder = "Enter Mobile Number" name = "mobile">
<label for = "mobile"> Mobile Number </label>
</div>
<div class = "form-floating mt-2 mb-2">
<input type = "text" class = "form-control" id = "email" placeholder = "Enter EmailId" name = "ei">
<label for = "ei"> EmailId </label>
</div>
<button type = "submit" class = "btn btn-info mt-2"> Submit </button>
</form>
</div>
</body>
</html>
```

Output

The following image shows the bootstrap 5 floating labels output.



Bootstrap 5 Floating labels

UserId

Mobile Number

EmailId

Example

The following example uses the "form-floating" class in the <div> tag. The <input> tag is used to insert data into the database, and the <label> tag is used to display animated labels in the form.

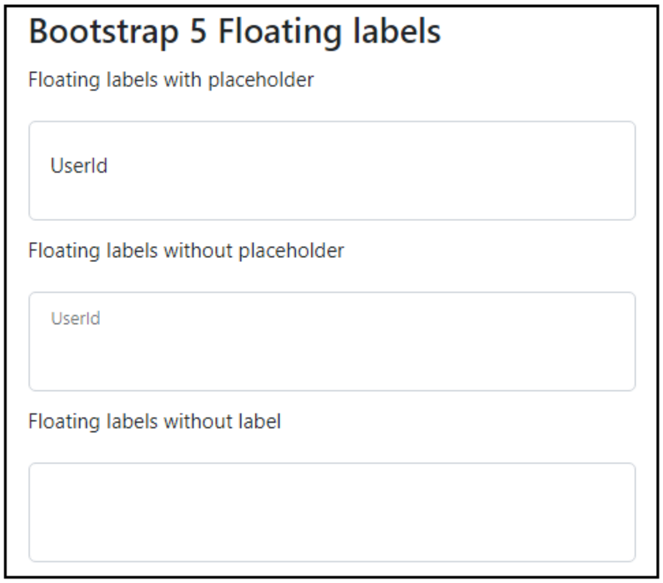
- The first floating label element uses the "placeholder" keyword and the <label> tag together in the form.
- The second floating label function does not use a placeholder keyword but uses a <label> tag.
- The third floating label function does not use the <label> tag. The label tag works like a placeholder in the input tag.

```
<!DOCTYPE html>
<html lang = "en">
<head>
<title> Bootstrap 5 Floating labels Example </title>
<meta charset = "utf-8">
<meta name = "viewport" content = "width=device-width, initial-scale=1">
<link href = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel="stylesheet">
<script src = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js">
</script>
</head>
<body>
<div class = "container mt-2 w-25" style = "border:2px solid black;">
<h5> Bootstrap 5 Floating labels </h5>
```

```
<form action = "" style = "font-size:12px;">
<p> Floating labels with placeholder </p>
<div class = "form-floating mt-2 mb-2">
<input type = "text" class = "form-control" id = "ui" placeholder = "Enter UserId" name = "ui">
<label for = "ui"> UserId </label>
</div>
<p> Floating labels without placeholder </p>
<div class = "form-floating mt-2 mb-2">
<input type = "text" class = "form-control" id = "ui" name = "ui">
<label for = "ui"> UserId </label>
</div>
<p> Floating labels without label </p>
<div class = "form-floating mt-2 mb-2">
<input type = "text" class = "form-control" id = "ui" placeholder = "Enter UserId" name = "ui">
</div>
</div>
</form>
</div>
</body>
</html>
```

Output

The following image shows the bootstrap 5 floating labels output.



Bootstrap 5 Floating labels

Floating labels with placeholder

UserId

Floating labels without placeholder

UserId

Floating labels without label

Textarea uses floating labels.

Bootstrap 5 floating labels are also called animated labels. The pair of <textarea class = "form-control"> and <label> components in the ".form-floating" class to enable floating labels with Bootstrap's text field inputs. The placeholder uses for the CSS style in the bootstrap5 floating label.

Syntax

The following syntax shows how to create floating label in the textarea function.

```
<div class = "form-floating">
<textarea class = "form-control" id = "ui" placeholder = "Enter require message" name = "ui">
</textarea>
<label for = "ui"> Message </label>
</div>
```

Example

The following example uses the "form-floating" class in the <div> tag. The <textarea> and <label> tag uses for the insert large size data and display animated label respectively.

Output

- The first floating label in the textarea uses the "placeholder" keyword with the <label> tag.

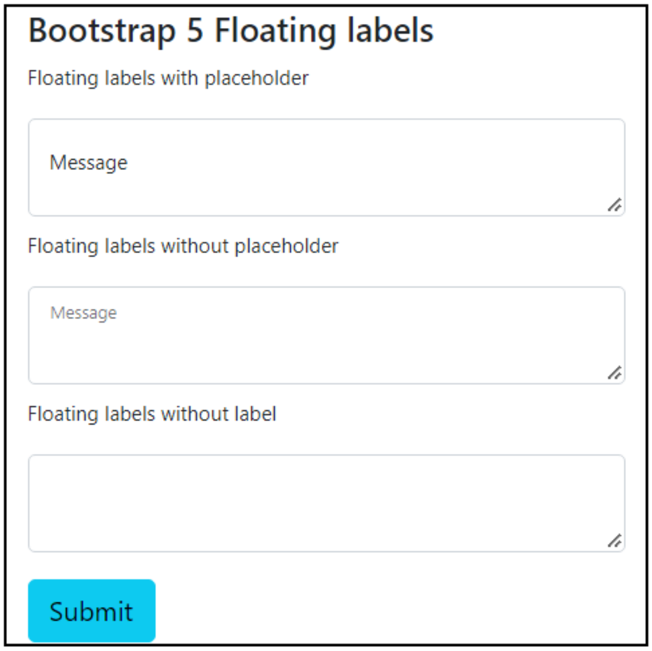
- The second floating label in the textarea does not use placeholder keyword.
- The third floating label in the textarea does not uses <label> tag.

```
<!DOCTYPE html>
<html lang = "en">
<head>
<title> Bootstrap 5 Floating labels Example </title>
<meta charset = "utf-8">
<meta name = "viewport" content = "width=device-width, initial-scale=1">
<link href = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel="stylesheet">
<script src = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js">
</script>
</head>
<body>
<div class = "container mt-2 w-25" style = "border:2px solid black;">
<h5> Bootstrap 5 Floating labels </h5>
<form action = "" style = "font-size:12px;">
<p> Floating labels with placeholder </p>
<div class = "form-floating mt-2 mb-2">
<textarea class = "form-control" id = "ui" placeholder = "Enter require message" name = "ui">
</textarea>
<label for = "ui"> Message </label>
</div>
<p> Floating labels without placeholder </p>
<div class = "form-floating mt-2 mb-2">
<textarea class = "form-control" id = "ui" name = "ui">
</textarea>
<label for = "ui"> Message </label>
</div>
<p> Floating labels without label </p>
<div class = "form-floating mt-2 mb-2">
<textarea class = "form-control" id = "ui" placeholder = "Enter require message" name = "ui">
</textarea>
```

```
</div>
<button type = "submit" class = "btn btn-info mt-2"> Submit </button>
</form>
</div>
</body>
</html>
```

Output

The following image shows the bootstrap 5 floating labels output.



Bootstrap 5 Floating labels

Floating labels with placeholder

Floating labels without placeholder

Floating labels without label

Select Menus use floating labels.

The <select class = "form-control"> and <option> tags use in the ".form-floating" class. This floating label class enables floating labels with Bootstrap's input elements. The <label> tags uses to display label in the select menu. the <select> tag shows input tab and <option> tag shows values for the input data.

Syntax

The following syntax shows how to create floating label in the select label function.

```
<div class = "form-floating">
  <select class = "form-select" id = "data1" name = "datalist">
    <option> value1 </option>
    <option> value2 </option>
    <option> value3 </option>
    <option> value4 </option>
  </select>
  <label for = "data1" class = "form-label"> Select values: </label>
</div>
```

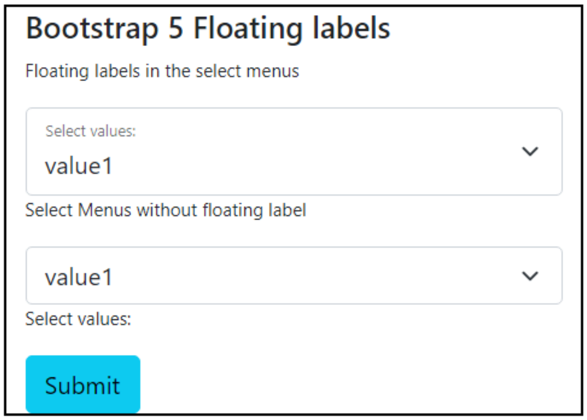
Example

The following example shows the "form-floating" class with select menus in the <div> tag. The <select>, <option> and <label> tag uses for the display input tab, select data from options data and display animated label respectively.

```
<!DOCTYPE html>
<html lang = "en">
<head>
<title> Bootstrap 5 Floating labels Example </title>
<meta charset = "utf-8">
<meta name = "viewport" content = "width=device-width, initial-scale=1">
<link href = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel="stylesheet">
<script src = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js">
</script>
</head>
<body>
<div class = "container mt-2 w-25" style = "border:2px solid black;">
<h5> Bootstrap 5 Floating labels </h5>
<form action = "" style = "font-size:12px;">
<p> Floating labels in the select menus </p>
<div class = "form-floating">
  <select class = "form-select" id = "data1" name = "datalist">
    <option> value1 </option>
    <option> value2 </option>
    <option> value3 </option>
    <option> value4 </option>
  </select>
  <label for = "data1" class = "form-label"> Select values: </label>
</div>
<p> Select Menus without floating label </p>
<div class = "">
  <select class = "form-select" id = "data1" name = "datalist">
    <option> value1 </option>
    <option> value2 </option>
    <option> value3 </option>
    <option> value4 </option>
  </select>
  <label for = "data1" class = "form-label"> Select values: </label>
</div>
<button type = "submit" class = "btn btn-info mt-2"> Submit </button>
</form>
</div>
</body>
</html>
```

Output

The following image shows the bootstrap 5 floating labels output.



Bootstrap 5 Floating labels

Floating labels in the select menus

Select values:

value1

Select Menus without floating label

value1

Select values:

Bootstrap 5 Colors

Bootstrap 5 provides multiple color classes for the web component. Bootstrap 5 colors convey a message about the function, element, and text. It provides a class to display text and background in the required color. For example, the red color gives the ":danger" class in the tag.

Bootstrap 5 Text color

Bootstrap 5 provides a color class for text to convey the particular meaning of the sentence. Bootstrap 5 uses color classes to display different colors are called contextual classes.

Syntax

The following table shows us text color and its class in detail.

Bootstrap 5 text color	Description
"text-primary"	This class shows content is important and shows in blue color.
"text-secondary"	This class shows content is secondary and shown in grey color.
"text-muted"	This class shows content is muted and shown in light grey color.
"text-light"	This class shows light grey color content (visible on dark background).
"text-dark"	This class shows in dark grey color content.
"text-info"	This class shows content that is informative and shown in sky blue color.
"text-warning"	This class conveys warning text and is shown in yellow color.
"text-success"	This class conveys success text and shows it in green color.
"text-danger"	This class conveys danger or alert text and is shown in red color.
"text-body"	This class shows in dark grey color content.
"text-white"	This class shows in white color content (visible on dark background).

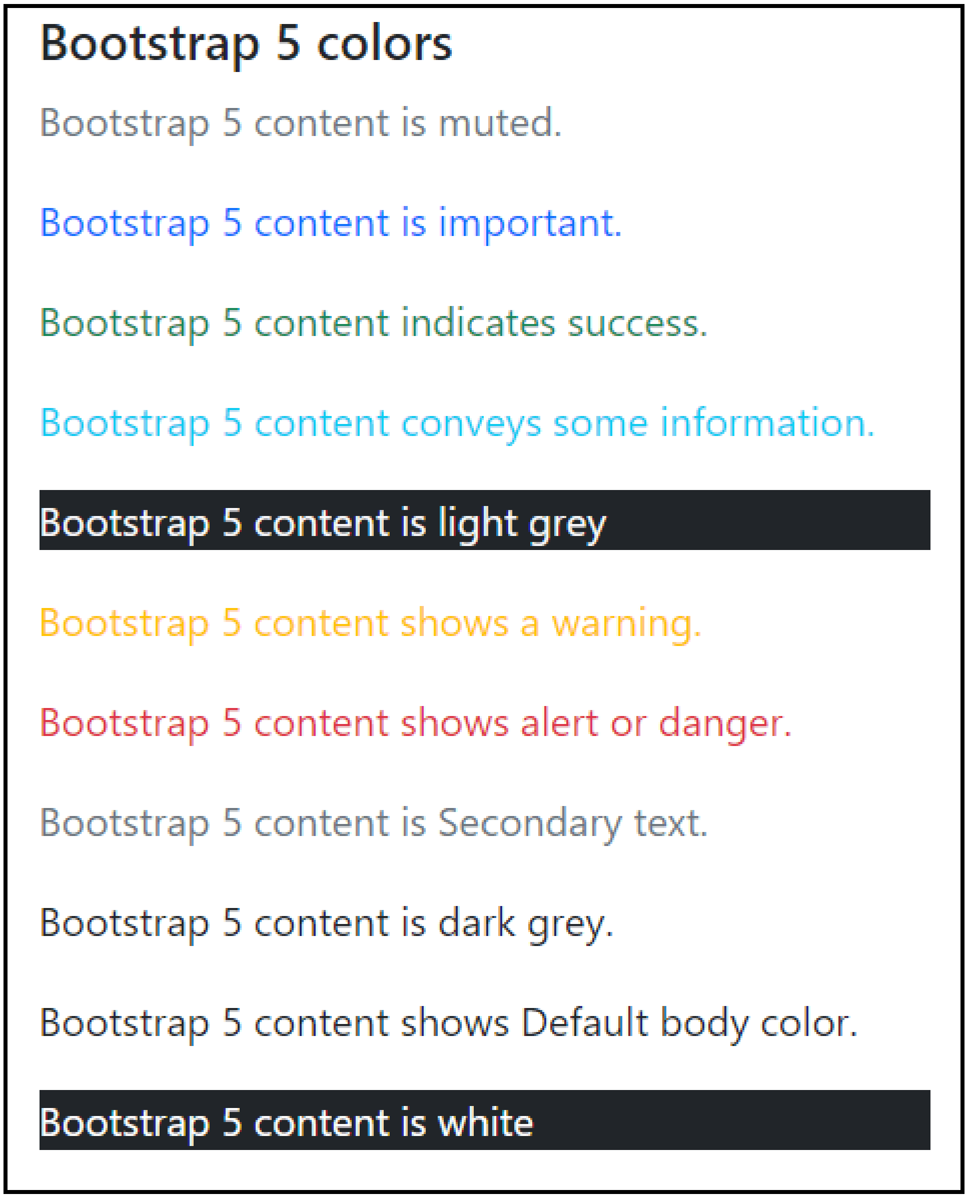
Example

The following example shows us text color and its class. Here, we can see all the bootstrap 5 color classes for the content.

```
<!DOCTYPE html>
<html lang = "en">
<head>
<title> Bootstrap 5 colors Example </title>
<meta charset = "utf-8">
<meta name = "viewport" content = "width=device-width, initial-scale=1">
<link href = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel="stylesheet">
<script src = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js">
</script>
</head>
<body>
<div class = "container mt-2 w-25" style = "border:2px solid black;">
<h5> Bootstrap 5 colors </h5>
  <p class = "text-muted"> Bootstrap 5 content is muted. </p>
  <p class = "text-primary"> Bootstrap 5 content is important. </p>
  <p class = "text-success"> Bootstrap 5 content indicates success. </p>
  <p class = "text-info"> Bootstrap 5 content conveys some information. </p>
  <p class = "text-light bg-dark"> Bootstrap 5 content is light grey </p>
  <p class = "text-warning"> Bootstrap 5 content shows a warning. </p>
  <p class = "text-danger"> Bootstrap 5 content shows alert or danger.</p>
  <p class = "text-secondary"> Bootstrap 5 content is Secondary text. </p>
  <p class = "text-dark"> Bootstrap 5 content is dark grey. </p>
  <p class = "text-body"> Bootstrap 5 content shows Default body color. </p>
  <p class = "text-white bg-dark"> Bootstrap 5 content is white </p>
</div>
</body>
</html>
```

Output

The output shows text or content color in the below image.



Bootstrap 5 colors

Bootstrap 5 content is muted.

Bootstrap 5 content is important.

Bootstrap 5 content indicates success.

Bootstrap 5 content conveys some information.

Bootstrap 5 content is light grey

Bootstrap 5 content shows a warning.

Bootstrap 5 content shows alert or danger.

Bootstrap 5 content is Secondary text.

Bootstrap 5 content is dark grey.

Bootstrap 5 content shows Default body color.

Bootstrap 5 content is white

Bootstrap 5 Background color

Bootstrap 5 provides a color class to the background of the element. These colors convey a particular meaning of the function or features. Bootstrap 5 uses background color classes to display different colors are called contextual classes.

Syntax

The following table shows us the background color and its class in detail.

Bootstrap 5 background color	Description
"bg-primary"	This class shows element is important and shows the background in blue color.
"bg-secondary"	This class shows element is secondary and shows a background in grey color.
"bg-muted"	This class shows element is muted and shown in light grey color.
"bg-light"	This class shows a background in light grey color (visible on dark background).
"bg-dark"	This class shows a background in dark grey color.
"bg-info"	This class shows content that is informative and shown in sky blue color.
"bg-warning"	This class conveys warning background of the function and shows it in yellow color.
"bg-success"	This class conveys success background of the function and shows it in green color.
"bg-danger"	This class conveys danger or alert text and is shown in red color.
"bg-body"	This class shows in dark grey color content.
"bg-white"	This class shows in white color content (visible on dark background).

Example

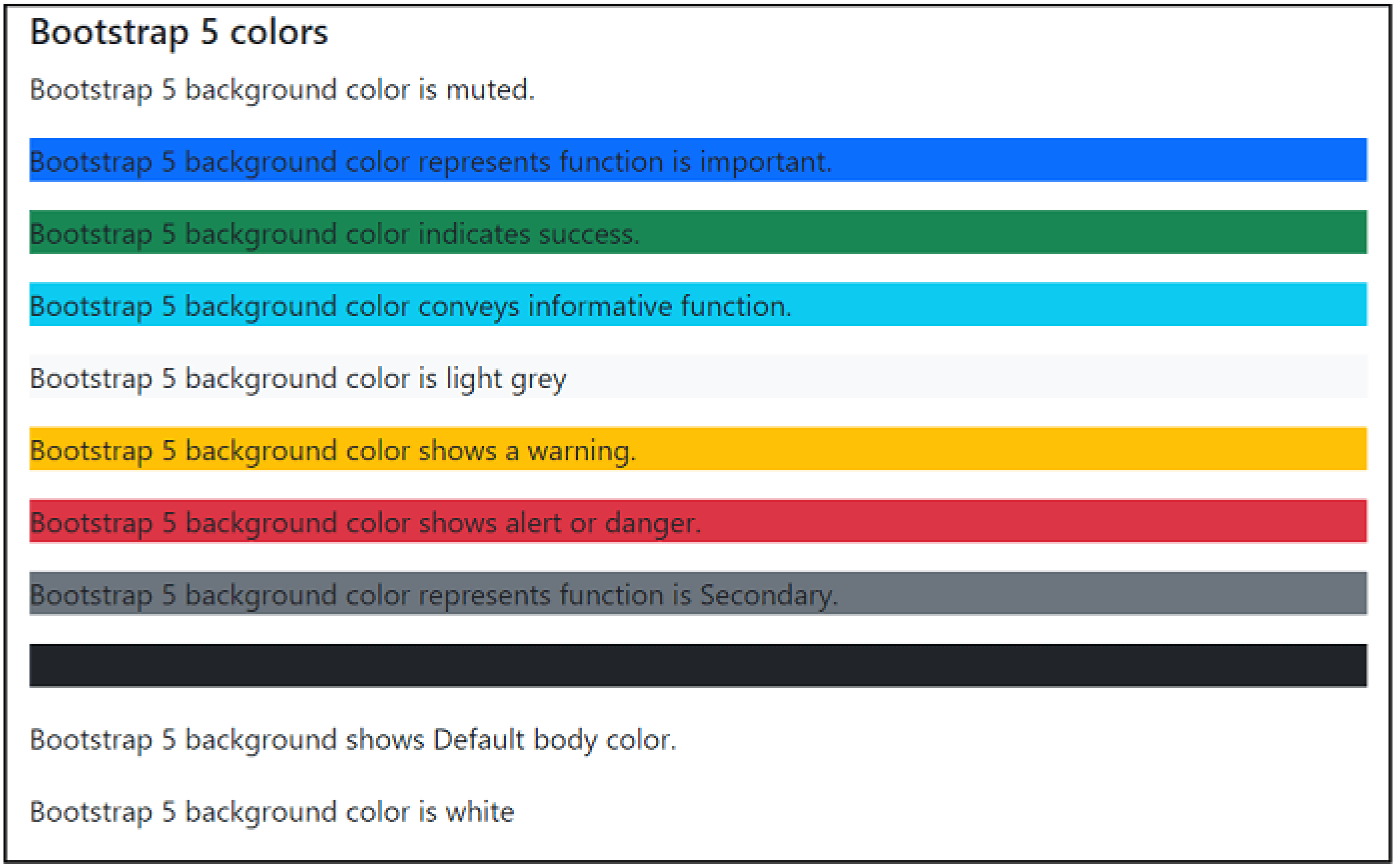
The following example shows us background color and its class. Here, we can see all the bootstrap 5 color classes for the background of the element.

```
<!DOCTYPE html>
<html lang = "en">
```

```
<head>
<title> Bootstrap 5 colors Example </title>
<meta charset = "utf-8">
<meta name = "viewport" content = "width=device-width, initial-scale=1">
<link href = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel="stylesheet">
<script src = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js">
</script>
</head>
<body>
<div class = "container mt-2 w-50" style = "border: 2px solid black;">
<h5> Bootstrap 5 colors </h5>
  <p class = "bg-muted"> Bootstrap 5 background color is muted. </p>
  <p class = "bg-primary"> Bootstrap 5 background color represents function is important. </p>
  <p class = "bg-success"> Bootstrap 5 background color indicates success. </p>
  <p class = "bg-info"> Bootstrap 5 background color conveys informative function. </p>
  <p class = "bg-light"> Bootstrap 5 background color is light grey </p>
  <p class = "bg-warning"> Bootstrap 5 background color shows a warning. </p>
  <p class = "bg-danger"> Bootstrap 5 background color shows alert or danger.</p>
  <p class = "bg-secondary"> Bootstrap 5 background color represents function is Secondary.    </p>
  <p class = "bg-dark"> Bootstrap 5 background color is dark grey. </p>
  <p class = "bg-body"> Bootstrap 5 background shows Default body color. </p>
  <p class = "bg-white"> Bootstrap 5 background color is white </p>
</div>
</body>
</html>
```

Output

The output shows the background color of the element in the below image.



Bootstrap 5 colors

Bootstrap 5 background color is muted.

Bootstrap 5 background color represents function is important.

Bootstrap 5 background color indicates success.

Bootstrap 5 background color conveys informative function.

Bootstrap 5 background color is light grey

Bootstrap 5 background color shows a warning.

Bootstrap 5 background color shows alert or danger.

Bootstrap 5 background color represents function is Secondary.

Bootstrap 5 background shows Default body color.

Bootstrap 5 background color is white

Bootstrap 5 colors example

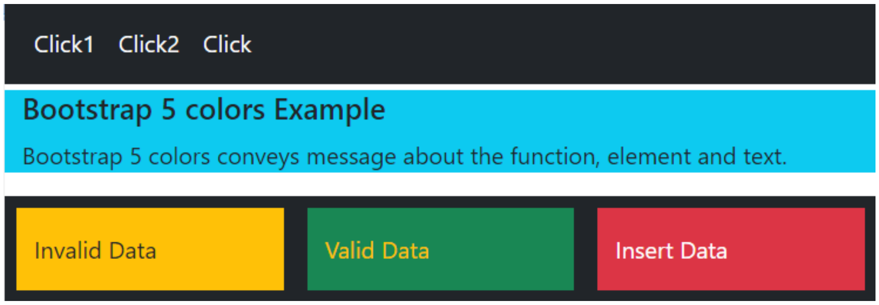
The following example uses bootstrap 5 background color in the web page. Here, we can use navigation bar, contrainer-fluid, and grid system in web page. This element displays with required background color and color text.

```
<!DOCTYPE html>
<html lang = "en">
<head>
<title> Bootstrap 5 colors Example </title>
<meta charset = "utf-8">
<meta name = "viewport" content="width=device-width, initial-scale=1">
<link href = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel = "stylesheet">
<script src = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js">
</script>
</head>
```

```html
<body>
<nav class = "navbar navbar-expand-sm bg-dark">
<div class = "container-fluid">
<ul class = "navbar-nav">
<li class = "nav-item">
<a class = "nav-link text-white" href = "#"> Click1 </a>
</li>
<li class = "nav-item">
<a class = "nav-link text-white" href = "#"> Click2 </a>
</li>
<li class = "nav-item">
<a class = "nav-link text-white" href = "#"> Click </a>
</li>
</ul>
</div>
</nav>
<div class = "container-fluid mt-1 bg-info">
<h5> Bootstrap 5 colors Example </h5>
<p> Bootstrap 5 colors conveys message about the function, element and text. </p>
</div>
<div class = "container-fluid mt-1">
<div class = "row bg-dark ">
<div class = "col bg-warning text-body m-2 py-3"> Invalid Data </div>
<div class = "col bg-success text-warning m-2 py-3"> Valid Data </div>
<div class = "col bg-danger text-white m-2 py-3"> Insert Data </div>
</div>
</div>
</body>
</html>
```

Output

The output shows the content and background color of the web page in the below image.



Bootstrap 5 colors Example

Bootstrap 5 colors conveys message about the function, element and text.

Bootstrap 5 table

Bootstrap 5 tables display data in a grid format similar to row and column layouts. Using Bootstrap 5 tables, you can dramatically enhance the appearance of basic HTML tables. It is a user interface function to interact with user and system information.

Basic table

Bootstrap 5 table requires a table tag with the "table" class to get the default table format. The table includes table head and table body data using bootstrap 5 tags.

Syntax

The following syntax shows default table format using bootstrap 5 classes.

```
<table class = "table">
<thead>
<tr>  <th> heading </th> </tr>
</thead>
<tbody>
<tr>  <td> data </td> </tr>
</tbody>
</table>
```

- The table header uses the <thead> (table header) and <th> (table heading data) with <tr> (table row) tag.
- The table body uses the <tbody> (table body) and <td> (table data) with <tr> (table row) tag.

Example

The following example uses four column for the table header and four rows for the table body. Here, we can use basic table format using bootstrap 5 table class.

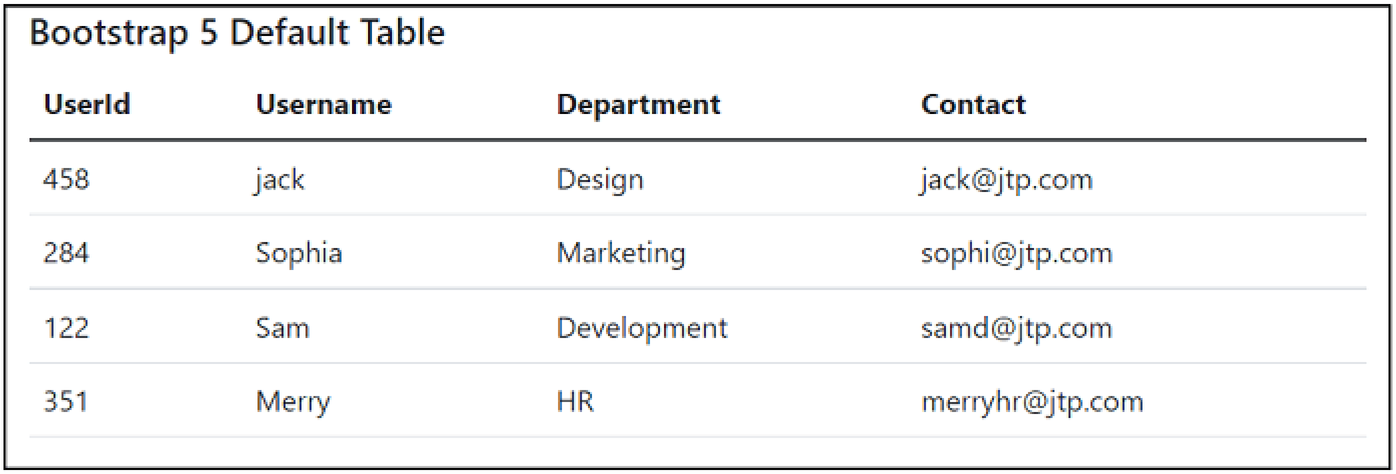
```
<!DOCTYPE html>
<html lang = "en">
<head>
<title> Bootstrap 5 table Example </title>
<meta charset = "utf-8">
```

```
<meta name = "viewport" content = "width=device-width, initial-scale=1">
<link href = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel="stylesheet">
<script src = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js">
</script>
</head>
<body>
<div class = "container mt-2 w-50" style = "border:2px solid black;">
<h5> Bootstrap 5 Default Table </h5>
<table class = "table">
<thead>
<tr>
<th> UserId </th>
<th> Username </th>
<th> Department </th>
<th> Contact </th>
</tr>
</thead>
<tbody>
<tr>
<td> 458 </td>
<td> jack </td>
<td> Design </td>
<td> jack@jtp.com </td>
</tr>
<tr>
<td> 284 </td>
<td> Sophia </td>
<td> Marketing </td>
<td> sophi@jtp.com </td>
</tr>
<tr>
<td> 122 </td>
<td> Sam </td>
```

```
<td> Development </td>
<td> samd@jtp.com </td>
</tr>
<tr>
<td> 351 </td>
<td> Merry </td>
<td> HR </td>
<td> merryhr@jtp.com </td>
</tr>
</tbody>
</table>
</div>
</body>
</html>
```

Output

The below output image shows the default table format of the given an example.



Bootstrap 5 Default Table

UserId	Username	Department	Contact
458	jack	Design	jack@jtp.com
284	Sophia	Marketing	sophi@jtp.com
122	Sam	Development	samd@jtp.com
351	Merry	HR	merryhr@jtp.com

Basic table classes

We can see some bootstrap 5 table classes in detail. The basic bootstrap 5 table requires a table tag with the class to get the basic table format. The border and hover effect is a basic requirement of the table.

- The "table" class creates the default table on the web page.

- The "table-hover" class uses to highlight table rows where the mouse arrow is pointed.
- The "table-bordered" class displays a border around the table (including rows and columns).
- The "table-striped" class displays the table in strip format. These table rows display in light grey and white color.

Syntax

The following syntax shows basic table format using bootstrap 5 classes.

```
<table class = "table table-hover table-bordered table-striped">
<thead>
<tr>  <th> heading </th> </tr>
</thead>
<tbody>
<tr>  <td> data </td> </tr>
</tbody>
</table>
```

- The table tag uses multiple basic classes simultaneously to display several styles.

Example

The following example uses four-column for the table header and four rows for the table body. Here, we can use basic table format using bootstrap 5 table classes. The table uses a hover effect, a striped style with a border.

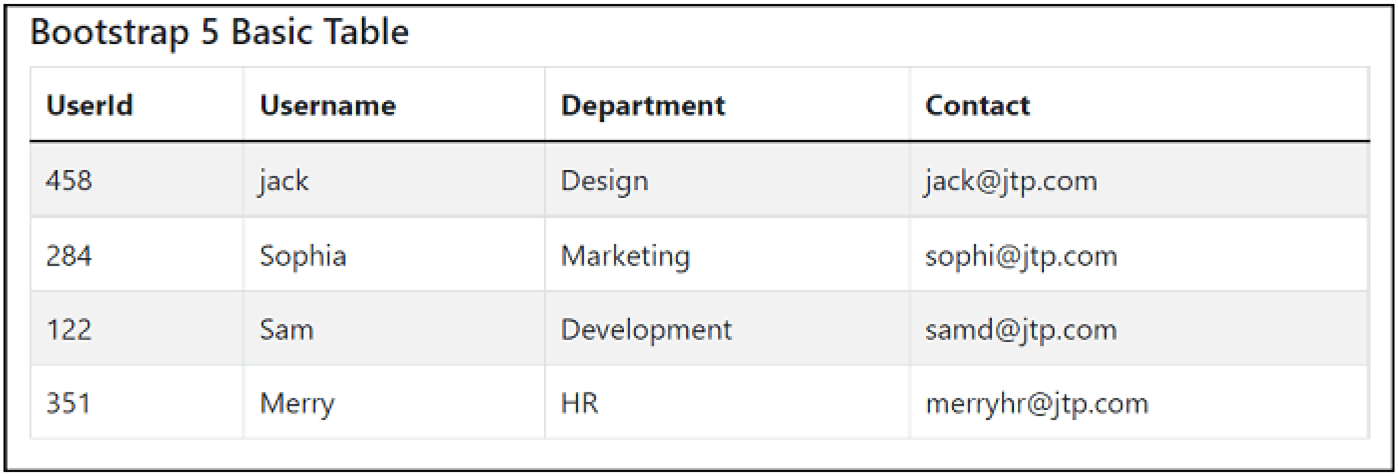
```
<!DOCTYPE html>
<html lang = "en">
<head>
<title> Bootstrap 5 table Example </title>
<meta charset = "utf-8">
<meta name = "viewport" content = "width=device-width, initial-scale=1">
<link href = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel="stylesheet">
<script src = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js">
</script>
```

```html
</head>
<body>
<div class = "container mt-2 w-50" style = "border:2px solid black;">
<h5> Bootstrap 5 Basic Table </h5>
<table class = "table table-hover table-bordered table-striped ">
<thead>
<tr>
<th> UserId </th>
<th> Username </th>
<th> Department </th>
<th> Contact </th>
</tr>
</thead>
<tbody>
<tr>
<td> 458 </td>
<td> jack </td>
<td> Design </td>
<td> jack@jtp.com </td>
</tr>
<tr>
<td> 284 </td>
<td> Sophia </td>
<td> Marketing </td>
<td> sophi@jtp.com </td>
</tr>
<tr>
<td> 122 </td>
<td> Sam </td>
<td> Development </td>
<td> samd@jtp.com </td>
</tr>
<tr>
<td> 351 </td>
```

```
<td> Merry </td>
<td> HR </td>
<td> merryhr@jtp.com </td>
</tr>
</tbody>
</table>
</div>
</body>
</html>
```

Output

The below output image shows the basic table format of the given an example.



Bootstrap 5 Basic Table

UserId	Username	Department	Contact
458	jack	Design	jack@jtp.com
284	Sophia	Marketing	sophi@jtp.com
122	Sam	Development	samd@jtp.com
351	Merry	HR	merryhr@jtp.com

Bootstrap 5 small table

Bootstrap 5 small table shows a small size of the columns than the default size of the table. The "table-sm" class uses with "table" class by default, but we can use other Bootstrap 5 table classes.

Syntax

The following syntax shows small size table using bootstrap 5 classes.

```
<table class = "table table-sm">
</table>
```

Example

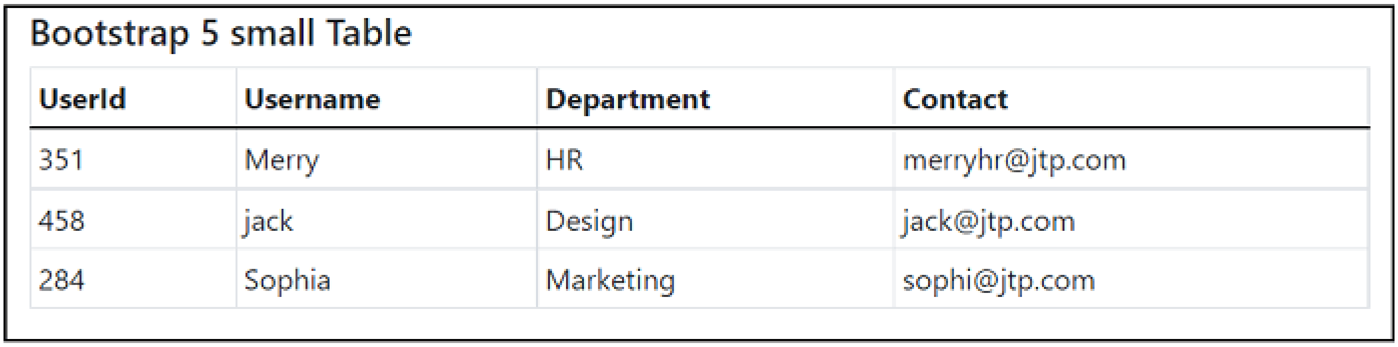
The following example uses four columns for the table header and three rows for the table body. Here, we can use small size table format using bootstrap 5 table class.

```
<!DOCTYPE html>
<html lang = "en">
<head>
<title> Bootstrap 5 table Example </title>
<meta charset = "utf-8">
<meta name = "viewport" content = "width=device-width, initial-scale=1">
<link href = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel="stylesheet">
<script src = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js">
</script>
</head>
<body>
<div class = "container mt-2 w-50" style = "border:2px solid black;">
<h5> Bootstrap 5 small Table </h5>
<table class = "table table-bordered table-sm">
<thead>
<tr>
<th> UserId </th>
<th> Username </th>
<th> Department </th>
<th> Contact </th>
</tr>
</thead>
<tbody>
<tr>
<td> 351 </td>
<td> Merry </td>
<td> HR </td>
<td> merryhr@jtp.com </td>
</tr>
```

```
<tr>
<td> 458 </td>
<td> jack </td>
<td> Design </td>
<td> jack@jtp.com </td>
</tr>
<tr>
<td> 284 </td>
<td> Sophia </td>
<td> Marketing </td>
<td> sophi@jtp.com </td>
</tr>
</tbody>
</table>
</div>
</body>
</html>
```

Output

The below output image shows the small size table giving an example.



Bootstrap 5 small Table

UserId	Username	Department	Contact
351	Merry	HR	merryhr@jtp.com
458	jack	Design	jack@jtp.com
284	Sophia	Marketing	sophi@jtp.com

Contextual table

Bootstrap 5 contextual tables provide several classes to show eye-catch tables. Bootstrap 5 table requires a table tag with the contextual table class to get a different color table format. This table color conveys some message or meaning of the table data.

Syntax

The following table shows us the table/table column's color and class in detail.

Bootstrap 5 table color	Description
"table-primary"	This class shows table is important and shown in blue color.
"table-secondary"	This class shows the table is secondary and shown in grey color.
"table-light"	This class shows in a light grey color table (visible on dark background).
"table-dark"	This class shows in a dark grey color table.
"table-info"	This class shows a table that is informative and shown in sky blue color.
"table-warning"	This class conveys a warning table shown in yellow color.
"table-success"	This class conveys a success table shown in green color.
"table-danger"	This class conveys a danger or alert table shown in red color.
"table-active"	This class shows in a dark grey color table.

Example

The following example uses four rows for the table header and the table body. Here, we can use different contextual color using bootstrap 5 table classes. Here, table uses contextual class in the <tr> tag to show table row in particular color.

```
<!DOCTYPE html>
<html lang = "en">
<head>
```

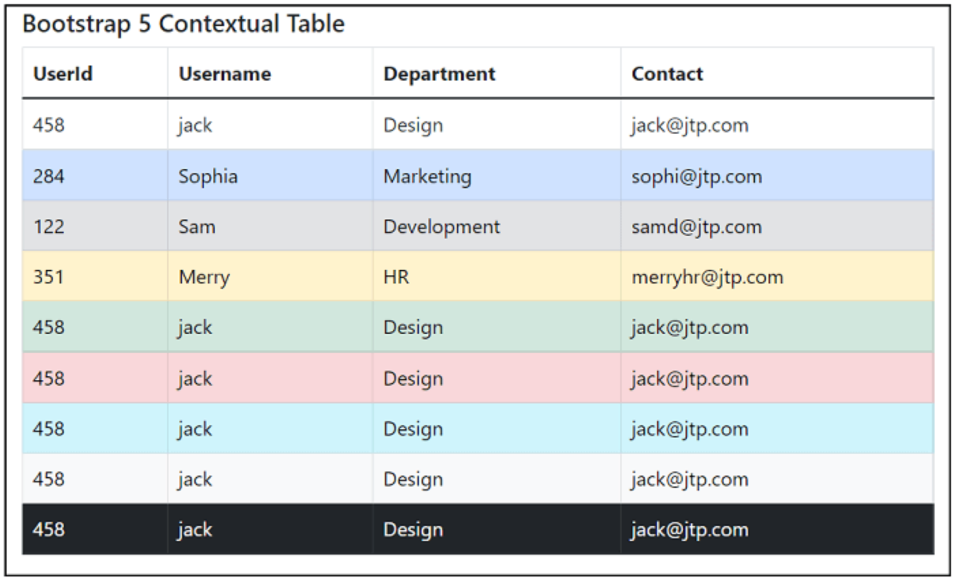
```
<title> Bootstrap 5 table Example </title>
<meta charset = "utf-8">
<meta name = "viewport" content = "width=device-width, initial-scale=1">
<link href = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel="stylesheet">
<script src = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js">
</script>
</head>
<body>
<div class = "container mt-2 w-50" style = "border:2px solid black;">
<h5> Bootstrap 5 Contextual Table </h5>
<table class = "table table-bordered">
<thead>
<tr>
<th> UserId </th>
<th> Username </th>
<th> Department </th>
<th> Contact </th>
</tr>
</thead>
<tbody>
<tr>
<td> 458 </td>
<td> jack </td>
<td> Design </td>
<td> jack@jtp.com </td>
</tr>
<tr class = "table-primary">
<td> 284 </td>
<td> Sophia </td>
<td> Marketing </td>
<td> sophi@jtp.com </td>
</tr>
<tr class = "table-secondary">
```

```
<td> 122 </td>
<td> Sam </td>
<td> Development </td>
<td> samd@jtp.com </td>
</tr>
<tr class = "table-warning">
<td> 351 </td>
<td> Merry </td>
<td> HR </td>
<td> merryhr@jtp.com </td>
</tr>
<tr class = "table-success">
<td> 458 </td>
<td> jack </td>
<td> Design </td>
<td> jack@jtp.com </td>
</tr>
<tr class = "table-danger">
<td> 458 </td>
<td> jack </td>
<td> Design </td>
<td> jack@jtp.com </td>
</tr>
<tr class = "table-info">
<td> 458 </td>
<td> jack </td>
<td> Design </td>
<td> jack@jtp.com </td>
</tr>
<tr class = "table-light">
<td> 458 </td>
<td> jack </td>
<td> Design </td>
<td> jack@jtp.com </td>
```

```
</tr>
<tr class = "table-dark">
<td> 458 </td>
<td> jack </td>
<td> Design </td>
<td> jack@jtp.com </td>
</tr>
</tbody>
</table>
</div>
</body>
</html>
```

Output

The below table image shows a contextual table of the given an example.



Bootstrap 5 Contextual Table

UserId	Username	Department	Contact
458	jack	Design	jack@jtp.com
284	Sophia	Marketing	sophi@jtp.com
122	Sam	Development	samd@jtp.com
351	Merry	HR	merryhr@jtp.com
458	jack	Design	jack@jtp.com
458	jack	Design	jack@jtp.com
458	jack	Design	jack@jtp.com
458	jack	Design	jack@jtp.com
458	jack	Design	jack@jtp.com

Bootstrap 5 dark table

Bootstrap 5 table shows the table's dark color (black or dark grey) and its data. The "table-dark" class uses with the "table" class by default, but we can use other Bootstrap 5 table classes. The bootstrap 5 tables can use the hover effect or striped table format with a border.

Syntax

The following syntax shows dark color table using bootstrap 5 classes.

```
<table class = "table table-dark">
</table>
```

- The "table-dark" class shows the black background color of the table.
- The table information is displayed in white color.

The following syntax shows dark color with a striped table using bootstrap 5 classes.

```
<table class = "table table-dark table-striped">
</table>
```

- The "table-dark" class shows the table's black and grey background color.
- The table information is displayed in white color of both background colors.

Example

The following example uses four columns for the table header and four rows for the table body. Here, we can use table format using bootstrap 5 table classes.

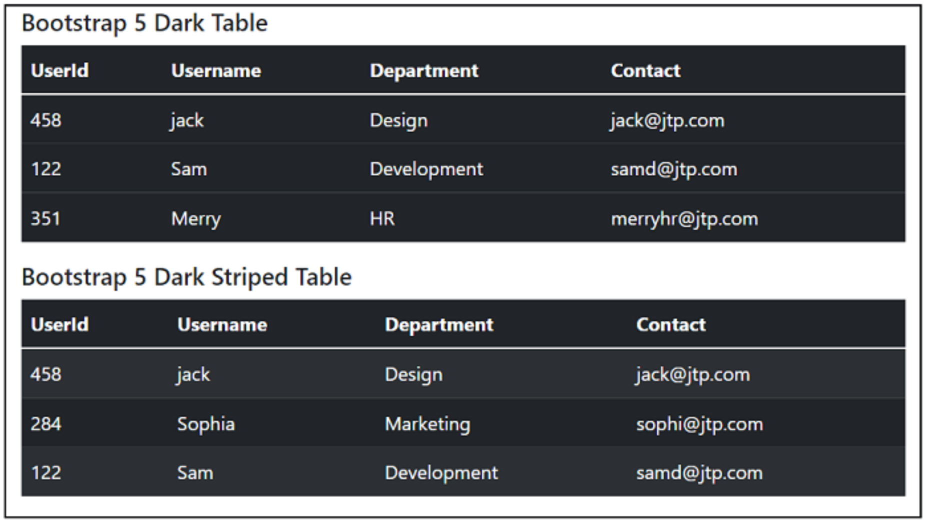
```
<!DOCTYPE html>
<html lang = "en">
<head>
<title> Bootstrap 5 table Example </title>
<meta charset = "utf-8">
<meta name = "viewport" content = "width=device-width, initial-scale=1">
<link href = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel="stylesheet">
<script src = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js">
</script>
</head>
<body>
<div class = "container mt-2 w-50" style = "border:2px solid black;">
<h5> Bootstrap 5 Dark Table </h5>
```

```
<table class = "table table-dark ">
<thead>
<tr>
<th> UserId </th>
<th> Username </th>
<th> Department </th>
<th> Contact </th>
</tr>
</thead>
<tbody>
<tr>
    <td> 458 </td>
    <td> jack </td>
    <td> Design </td>
    <td> jack@jtp.com </td>
   </tr>
   <tr>
    <td> 122 </td>
    <td> Sam </td>
    <td> Development </td>
    <td> samd@jtp.com </td>
   </tr>
   <tr>
    <td> 351 </td>
    <td> Merry </td>
    <td> HR </td>
    <td> merryhr@jtp.com </td>
   </tr>
</tbody>
</table>
<h5> Bootstrap 5 Dark Striped Table </h5>
<table class = "table table-dark table-striped">
<thead>
<tr>
```

```
<th> UserId </th>
<th> Username </th>
<th> Department </th>
<th> Contact </th>
</tr>
</thead>
<tbody>
<tr>
        <td> 458 </td>
        <td> jack </td>
        <td> Design </td>
        <td> jack@jtp.com </td>
      </tr>
       <tr>
        <td> 284 </td>
        <td> Sophia </td>
        <td> Marketing </td>
        <td> sophi@jtp.com </td>
      </tr>
       <tr>
        <td> 122 </td>
        <td> Sam </td>
        <td> Development </td>
        <td> samd@jtp.com </td>
      </tr>
       </tbody>
</table>
</div>
</body>
</html>
```

Output

The table below shows a dark color table and dark striped table output.



Bootstrap 5 Dark Table

UserId	Username	Department	Contact
458	jack	Design	jack@jtp.com
122	Sam	Development	samd@jtp.com
351	Merry	HR	merryhr@jtp.com

Bootstrap 5 Dark Striped Table

UserId	Username	Department	Contact
458	jack	Design	jack@jtp.com
284	Sophia	Marketing	sophi@jtp.com
122	Sam	Development	samd@jtp.com

Table Header color

Bootstrap 5 table shows the table's header in contextual color using classes. For example, the "table-dark" class displays a dark color header with white color text. The header color does not affect the table body.

Syntax

The following syntax shows dark header table using bootstrap 5 classes.

```
<table class = "table">
<thead class = "table-dark">
<tr> <th> heading </th> </tr>
</thead>
<tbody>
<tr> <td> data </td> </tr>
</tbody>
</table>
```

The following syntax shows yellow header table using bootstrap 5 classes.

```
<table class = "table">
<thead class = "table-warning">
</thead>
</table>
```

Example

The following example shows two tables with different header color. The "table-dark" and "table-warning" classes use in the <thead> tag.

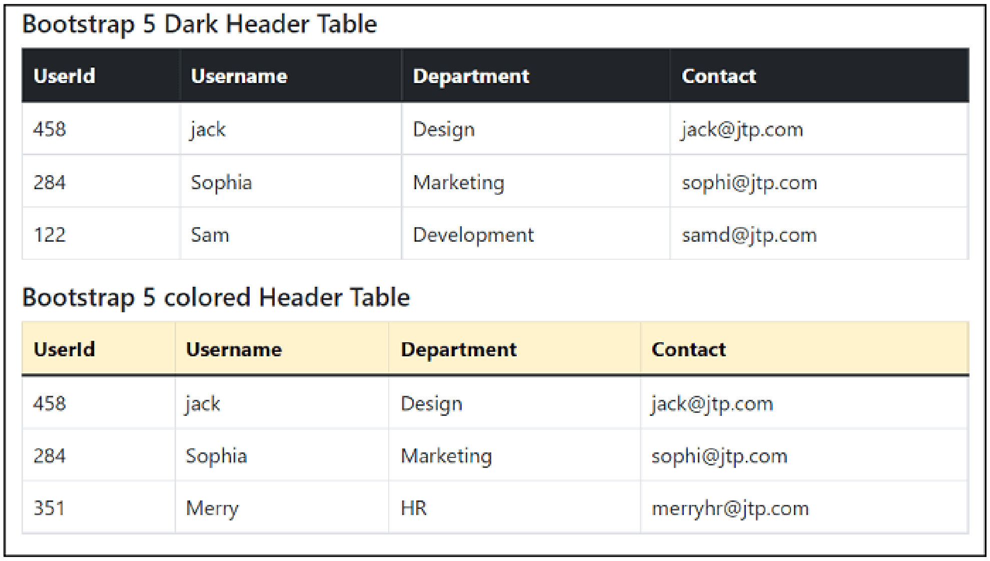
```
<!DOCTYPE html>
<html lang = "en">
<head>
<title> Bootstrap 5 table Example </title>
<meta charset = "utf-8">
<meta name = "viewport" content = "width=device-width, initial-scale=1">
<link href = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel="stylesheet">
<script src = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js">
</script>
</head>
<body>
<div class = "container mt-2 w-50" style = "border:2px solid black;">
<h5> Bootstrap 5 Dark Header Table </h5>
<table class = "table table-bordered ">
<thead class = "table-dark">
<tr>
<th> UserId </th>
<th> Username </th>
<th> Department </th>
<th> Contact </th>
</tr>
</thead>
<tbody>
<tr>
<td> 458 </td>
<td> jack </td>
<td> Design </td>
<td> jack@jtp.com </td>
</tr>
```

```
<tr>
<td> 284 </td>
<td> Sophia </td>
<td> Marketing </td>
<td> sophi@jtp.com </td>
</tr>
<tr>
<td> 122 </td>
<td> Sam </td>
<td> Development </td>
<td> samd@jtp.com </td>
</tr>
</tbody>
</table>
<h5> Bootstrap 5 colored Header Table </h5>
<table class = "table table-bordered">
<thead class = "table-warning">
<tr>
<th> UserId </th>
<th> Username </th>
<th> Department </th>
<th> Contact </th>
</tr>
</thead>
<tbody>
<tr>
<td> 458 </td>
<td> jack </td>
<td> Design </td>
<td> jack@jtp.com </td>
</tr>
<tr>
<td> 284 </td>
<td> Sophia </td>
```

```
<td> Marketing </td>
<td> sophi@jtp.com </td>
</tr>
<tr>
<td> 351 </td>
<td> Merry </td>
<td> HR </td>
<td> merryhr@jtp.com </td>
</tr>
</tbody>
</table>
</div>
</body>
</html>
```

Output

The below output image shows the table header color of the given an example.



Bootstrap 5 Dark Header Table

UserId	Username	Department	Contact
458	jack	Design	jack@jtp.com
284	Sophia	Marketing	sophi@jtp.com
122	Sam	Development	samd@jtp.com

Bootstrap 5 colored Header Table

UserId	Username	Department	Contact
458	jack	Design	jack@jtp.com
284	Sophia	Marketing	sophi@jtp.com
351	Merry	HR	merryhr@jtp.com

Bootstrap 5 borderless table

Bootstrap 5 table shows the borderless table and its information. The "table-borderless" class uses with "table" class by default, but we can use other basic table classes.

Syntax

The following syntax shows borderless table using bootstrap 5 classes.

```
<table class = "table table-borderless">
</table>
```

Example

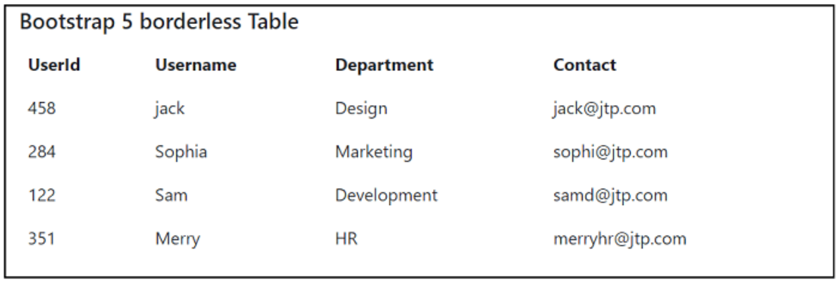
The following example uses four columns for the table header and three rows for the table body.

```
<!DOCTYPE html>
<html lang = "en">
<head>
<title> Bootstrap 5 table Example </title>
<meta charset = "utf-8">
<meta name = "viewport" content = "width=device-width, initial-scale=1">
<link href = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel="stylesheet">
<script src = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js">
</script>
</head>
<body>
<div class = "container mt-2 w-50" style = "border:2px solid black;">
<h5> Bootstrap 5 borderless Table </h5>
<table class = "table table-borderless">
<thead>
<tr>
<th> UserId </th>
<th> Username </th>
<th> Department </th>
<th> Contact </th>
</tr>
</thead>
<tbody>
```

```
<tr>
<td> 458 </td>
<td> jack </td>
<td> Design </td>
<td> jack@jtp.com </td>
</tr>
<tr>
<td> 284 </td>
<td> Sophia </td>
<td> Marketing </td>
<td> sophi@jtp.com </td>
</tr>
<tr>
<td> 122 </td>
<td> Sam </td>
<td> Development </td>
<td> samd@jtp.com </td>
</tr>
<tr>
<td> 351 </td>
<td> Merry </td>
<td> HR </td>
<td> merryhr@jtp.com </td>
</tr>
</tbody>
</table>
</div>
</body>
</html>
```

Output

The below output image shows the borderless table format of the given an example.



Bootstrap 5 borderless Table

UserId	Username	Department	Contact
458	jack	Design	jack@jtp.com
284	Sophia	Marketing	sophi@jtp.com
122	Sam	Development	samd@jtp.com
351	Merry	HR	merryhr@jtp.com

Responsive table

Bootstrap 5 table requires a table tag with the responsive table class to be placed in different device sizes. This table displays small, medium, large, extra-large, and double extra-large size devices. Bootstrap 5 uses the "table-responsive" class by default. The table becomes scrollable for small size devices.

Syntax

The following syntax shows responsive table using bootstrap 5 classes.

```
<div class = "table-responsive">
<table class = "table">
<thead>
<tr>
<th> heading </th>
</tr>
</thead>
<tbody>
<tr>  <td> data </td> </tr>
</tbody>
</table>
</div>
```

The following table shows us responsive table classes in detail.

Bootstrap 5 responsive table	Description

"table-responsive"	This bootstrap 5 table class is used for responsive tables by default. It works on all sizes of devices.
"table-responsive-sm"	This class is used to display responsive tables for small size devices. The devices width upto 576 px (greater than 576px).
"table-responsive-md"	This class is used to display responsive tables for medium size devices. The devices width upto 768 px (greater than 768px).
"table-responsive-lg"	This class uses to display responsive tables for large size devices. The devices width upto 992 px (greater than 992px).
"table-responsive-xl"	This class is used to display responsive tables for extra large size devices. The devices width upto 1200 px (greater than 1200px).
"table-responsive-xxl"	This class is used to display responsive tables for double extra-large size devices. The devices width upto 1400 px (greater than 1400px).

Example

The following example uses four rows for the table header and the table body. Here, we can use different contextual color using bootstrap 5 table classes. Here, table uses responsive class in the <div> tag to show scrollable table. The <table> tag uses in the <div> tag with the its information.

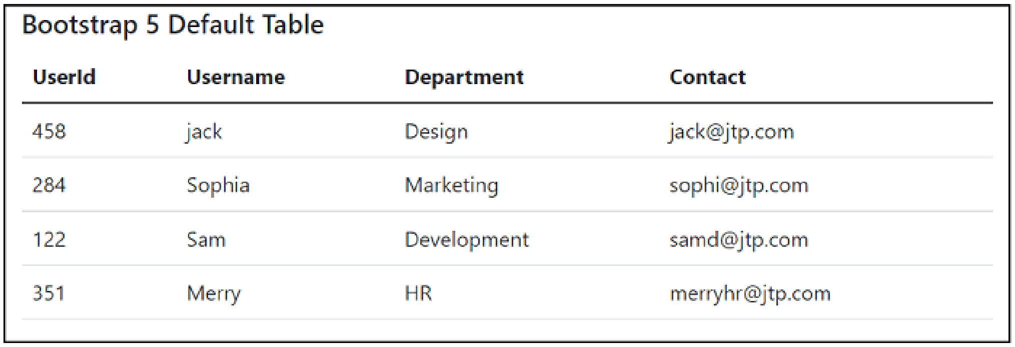
```
<!DOCTYPE html>
<html lang = "en">
<head>
<title> Bootstrap 5 table Example </title>
<meta charset = "utf-8">
<meta name = "viewport" content = "width=device-width, initial-scale=1">
<link href = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel="stylesheet">
<script src = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js">
</script>
```

```
</head>
<body>
<div class = "container mt-2 w-50" style = "border:2px solid black;">
<h5> Bootstrap 5 Contextual Table </h5>
<div class = "table-responsive">
<table class = "table table-bordered">
<thead class = "table-dark">
<tr>
<th> UserId </th>
<th> Username </th>
<th> Department </th>
<th> email </th>
<th> header5 </th>
<th> header6 </th>
<th> header7 </th>
<th> header8 </th>
<th> header9 </th>
<th> header10 </th>
</tr>
</thead>
<tbody>
<tr>
<td> 458 </td>
<td> jack </td>
<td> Design </td>
<td> jack@jtp.com </td>
<td> value5 </td>
<td> value6 </td>
<td> value7 </td>
<td> value8 </td>
<td> value9 </td>
<td> value10 </td>
</tr>
<tr class = "table-primary">
```

```
<td> 284 </td>
<td> Sophia </td>
<td> Marketing </td>
<td> sophi@jtp.com </td>
<td> value5 </td>
<td> value6 </td>
<td> value7 </td>
<td> value8 </td>
<td> value9 </td>
<td> value10 </td>
</tr>
<tr class = "table-warning">
<td> 351 </td>
<td> Merry </td>
<td> HR </td>
<td> merryhr@jtp.com </td>
<td> value5 </td>
<td> value6 </td>
<td> value7 </td>
<td> value8 </td>
<td> value9 </td>
<td> value10 </td>
</tr>
</tbody>
</table>
</div>
</div>
</body>
</html>
```

Output

The table below shows a responsive format of the bootstrap 5 table output.



Bootstrap 5 Default Table

UserId	Username	Department	Contact
458	jack	Design	jack@jtp.com
284	Sophia	Marketing	sophi@jtp.com
122	Sam	Development	samd@jtp.com
351	Merry	HR	merryhr@jtp.com

Bootstrap 5 image

Bootstrap 5 image tags insert an image on a web page for reference and information. Technically, an image is not placed into a web application like text and table data. The image path places on the web page using the <img> tag and the "href" element. The image tag of the online site does not have a closing tag.

Rounded corner image

Bootstrap 5 image function places image with rounded shape corner. The bootstrap 5 image function uses the "rounded" class with the image tag. We can use the class = "rounded" class for the rounded corner image.

Syntax

The following syntax shows rounded corner image in web page.

```
<img src = "name.jpg(png,jpeg)" class = "rounded" height = "value" width = "value">
```

- The <img> tag with the "src" and the "class" elements must require setting image in page.
- The height and width elements are required but do not necessary.

Example

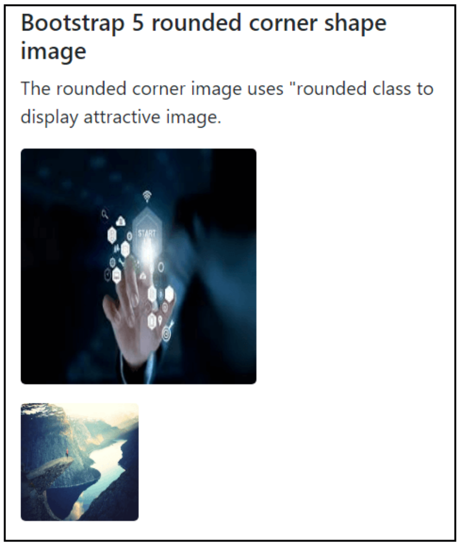
The following example uses rounded corner image in container tag. We can see the different size of the images place in one page.

```
<!DOCTYPE html>
<html lang = "en">
<head>
<title> Bootstrap 5 image Example </title>
<meta charset = "utf-8">
<meta name = "viewport" content = "width=device-width, initial-scale=1">
<link href = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel="stylesheet">
<script src = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js">
</script>
</head>
```

```
<body>
<div class = "container m-2 w-25" style = "border:2px solid black;">
<h5> Bootstrap 5 rounded corner shape image </h5>
<p> The rounded corner image uses "rounded class to display attractive image. </p>
<img src = "G:\ JavaTpoint\starts.jpg" class = "rounded mb-3" alt = "tech image" width = "200" height = "200"> <br>
<img src = "G: \JavaTpoint\nature.jpg" class = "rounded mb-3" alt = "nature image" width = "100" height = "100">
</div>
</div>
</body>
</html>
```

Output

The following output shows two different sizes of the image with rounded corners.



Bootstrap 5 rounded corner shape image

The rounded corner image uses "rounded class to display attractive image.

Circle image

Bootstrap 5 image function places images with circle shape border. The bootstrap 5 image function uses the "circle" class with the image tag. We can use the class = "rounded-circle" class for the circle shape image.

Syntax

The following syntax shows rounded corner image in web page.

```
<img src = "name.jpg(png,jpeg)" class = "rounded-circle" height = "value" width = "value">
```

- If height and width of the image is same then shape displays as a circle.
- If the height and width of the image do not match the same, then the shape displays as an elliptical.

Example

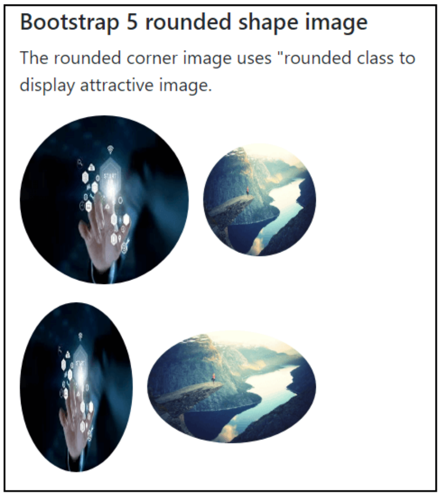
The following example uses circle shape image in container tag. We can see the different size and circular shape of the images place in one page.

```
<!DOCTYPE html>
<html lang = "en">
<head>
<title> Bootstrap 5 image Example </title>
<meta charset = "utf-8">
<meta name = "viewport" content = "width=device-width, initial-scale=1">
<link href = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel="stylesheet">
<script src = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js">
</script>
</head>
<body>
<div class = "container m-2 w-25" style = "border:2px solid black;">
<h5> Bootstrap 5 rounded shape image </h5>
<p> The rounded corner image uses "rounded class to display attractive image. </p>
<img src = "G:\JavaTpoint\starts.jpg" class = "rounded-circle mb-3" alt = "tech image" width = "150" height = "150">
<img src = "G:\JavaTpoint\nature.jpg" class = "rounded-circle mb-3" alt = "nature image" width = "100" height = "100"> <br>
<img src = "G:\JavaTpoint\starts.jpg" class = "rounded-circle mb-3" alt = "tech image" width = "100" height = "150">
```

```
<img src = "G:\JavaTpoint\nature.jpg" class = "rounded-circle mb-3" alt = "nature image" width = "150" height = "100">
</div>
</div>
</body>
</html>
```

Output

The following output shows an image of different sizes and shapes with height and width.



Bootstrap 5 rounded shape image

The rounded corner image uses "rounded class to display attractive image.

Thumbnail image

Bootstrap 5 image function places square shape image with its border. The bootstrap 5 image function uses the thumbnail class with the image tag. We can use class = "img-thumbnail" class for the stylish square shape image.

Syntax

The following syntax shows thumbnail image in web page.

```
<img src = "name.jpg" class = "img-thumbnail" height = "value" width = "value">
```

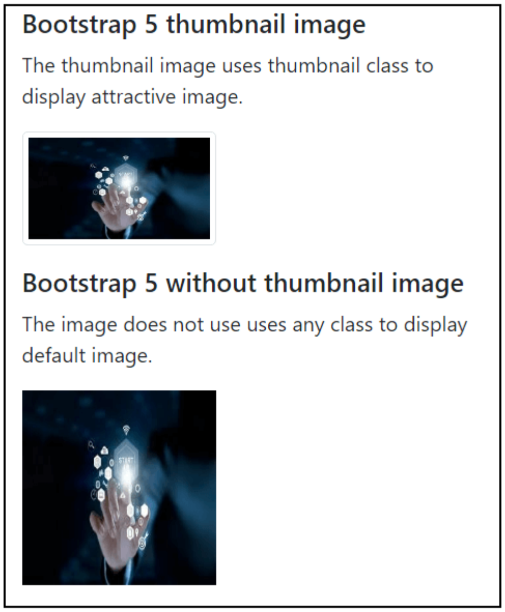
Example

The following example uses thumbnail and without thumbnail image in container tag. We can see the difference between thumbnail image and default image.

```
<!DOCTYPE html>
<html lang = "en">
<head>
<title> Bootstrap 5 image Example </title>
<meta charset = "utf-8">
<meta name = "viewport" content = "width=device-width, initial-scale=1">
<link href = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel="stylesheet">
<script src = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js">
</script>
</head>
<body>
<div class = "container m-2 w-25" style = "border:2px solid black;">
<h5> Bootstrap 5 thumbnail image </h5>
<p> The thumbnail image uses thumbnail class to display attractive image. </p>
<img src = "G:\JavaTpoint\022\may\starts.jpg" class = "img-thumbnail mb-3" alt = "tech image" width = "150" height = "150">
<h5> Bootstrap 5 without thumbnail image </h5>
<p> The image does not use uses any class to display default image. </p>
<img src = "G:\JavaTpoint\starts.jpg" class = "mb-3" alt = "tech image" width = "150" height = "150">
</div>
</body>
</html>
```

Output

The following output shows the default and thumbnail image height and width on a web page.



Bootstrap 5 thumbnail image

The thumbnail image uses thumbnail class to display attractive image.

Bootstrap 5 without thumbnail image

The image does not use uses any class to display default image.

Aligning image

Bootstrap 5 image function displays images in the required alignment of the web page. The image can place either start or end of the page. We can use the "float-start" and the "float-end" classes for the start and end of the page, respectively.

Syntax

The following syntax shows start alignment of the image in web page.

```
<img src = "name.jpg(png,jpeg)" class = "float-start" height = "value" width = "value">
```

The following syntax shows end alignment of the image in web page.

```
<img src = "name.jpg(png,jpeg)" class = "float-end" height = "value" width = "value">
```

Example

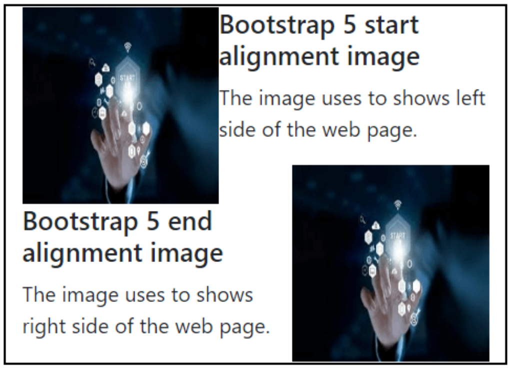
The following example uses start and end aligning image in container tag.

```
<!DOCTYPE html>
<html lang = "en">
<head>
```

```
<title> Bootstrap 5 image Example </title>
<meta charset = "utf-8">
<meta name = "viewport" content = "width=device-width, initial-scale=1">
<link href = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel="stylesheet">
<script src = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js">
</script>
</head>
<body>
<div class = "container m-2 w-25" style = "border:2px solid black;">
<img src = "G:\JavaTpoint\starts.jpg" class = "float-start" alt = "tech image" width = "150" height = "150">
<h5> Bootstrap 5 start alignment image </h5>
<p> The image uses to shows left side of the web page. </p>
<img src = "G:\JavaTpoint \starts.jpg" class = "float-end mb-3" alt = "tech image" width = "150" height = "150">
<br>
<h5> Bootstrap 5 end alignment image </h5>
<p> The image uses to shows right side of the web page. </p>
</div>
</body>
</html>
```

Output

The following output shows the start and end alignment of the image with the required height and width on a web page.



Bootstrap 5 start alignment image

The image uses to shows left side of the web page.

Bootstrap 5 end alignment image

The image uses to shows right side of the web page.

Centered image

Bootstrap 5 image function displays the image in the center alignment of the web page. The image places the center point of the container class or <div> tag using these two classes simultaneously. We can use the "mx-auto" and the "d-block" to automatically set the margin and display the image as a block.

Syntax

The following syntax shows center alignment image in web page.

```
<img src = "name.jpg" class = "mx-auto d-block" height = "value" width = "value">
```

Example

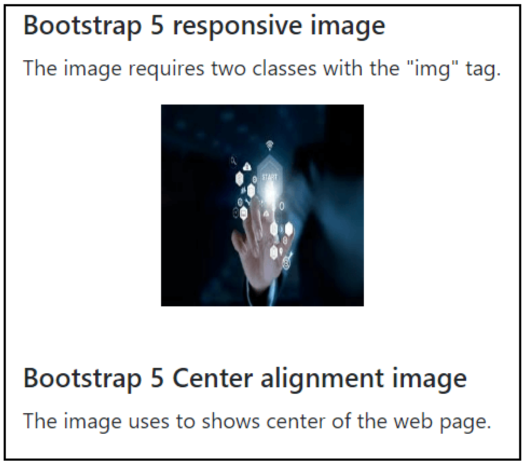
The following example uses center aligning of the image in container tag.

```
<!DOCTYPE html>
<html lang = "en">
<head>
<title> Bootstrap 5 image Example </title>
<meta charset = "utf-8">
<meta name = "viewport" content = "width=device-width, initial-scale=1">
<link href = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel="stylesheet">
<script src = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js">
</script>
</head>
<body>
```

```
<div class = "container m-2 w-25" style = "border:2px solid black;">
<h5> Bootstrap 5 responsive image </h5>
<p> The image requires two classes with the "img" tag. </p>
<img src = "G:\JavaTpoint\starts.jpg" class = "mx-auto d-block mb-3" alt = "tech image" width = "150" height = "150">
<br>
<h5> Bootstrap 5 Center alignment image </h5>
<p> The image uses to shows center of the web page. </p>
</div>
</body>
</html>
```

Output

The following output shows the center alignment of the image with the required height and width on the web page.



Bootstrap 5 responsive image

The image requires two classes with the "img" tag.

Bootstrap 5 Center alignment image

The image uses to shows center of the web page.

Responsive image

Images exist in various sizes, and they must fit in all sizes of the devices. Images that are responsive instantly change the size of the screen. Create responsive images by adding the "img-fluid" class to the <img> tag. The image will then scale proportionally to its parent. The "img-fluid" class sets the image's max-width upto 100 percent and its height to the "auto" value.

Syntax

The following syntax shows responsive image in web page.

```
<img src = "name.jpg(png,jpeg)" class = "img-fluid" height = "value" width = "value">
```

Example

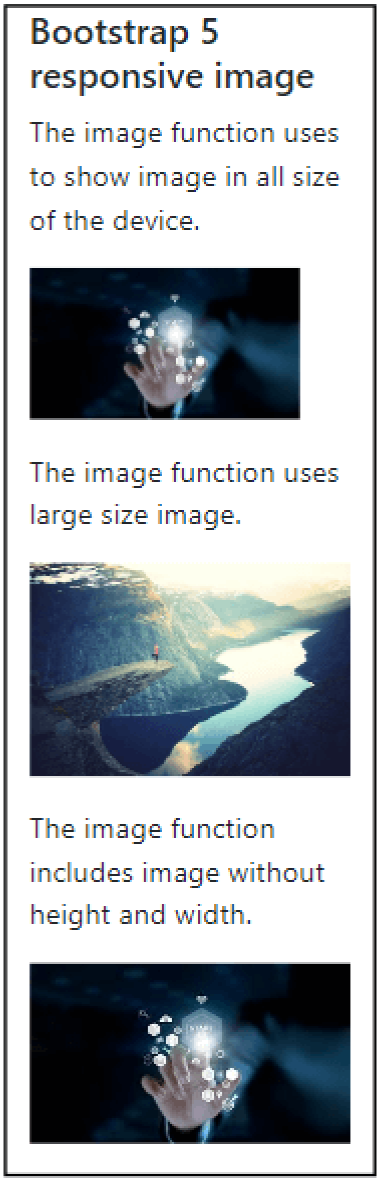
The following example places two images with different height and width in screen. The responsive image class helps to set in all size of the device. We can use image tag without height and width to create responsive image

```
<!DOCTYPE html>
<html lang = "en">
<head>
<title> Bootstrap 5 image Example </title>
<meta charset = "utf-8">
<meta name = "viewport" content = "width=device-width, initial-scale=1">
<link href = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel="stylesheet">
<script src = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js">
</script>
</head>
<body>
<div class = "container m-2 w-50" style = "border:2px solid black;">
<h5> Bootstrap 5 responsive image </h5>
<p> The image function uses to show image in all size of the device. </p>
<img src = "G:\JavaTpoint\starts.jpg" class = "img-fluid mb-3" alt = "tech image" width = "150" height = "150">
<br>
<p> The image function uses large size image. </p>
<img src = "G:\JavaTpoint\nature.jpg" class = "img-fluid mb-3" alt = "tech image" width = "1500" height = "1500">
<br>
<p> The image function includes image without height and width. </p>
<img src = "G:\JavaTpoint\starts.jpg" class = "img-fluid mb-3" alt = "nature image">
```

```
</div>
</body>
</html>
```

Output

The following output shows the responsive image to set in all screen sizes.



Bootstrap 5 responsive image

The image function uses to show image in all size of the device.

The image function uses large size image.

The image function includes image without height and width.

Bootstrap 5 Alerts

The bootstrap 5 alert component displays a preset message in reaction to a user's action. It sends information to end-users, such as errors, warnings, and confirmation messages. A limited set of configurable alert messages provide context-sensitive feedback for common user actions.

Bootstrap 5 alert function uses the default alert box with other elements. The alert box uses a link, closing button, and animation on the web page.

Default Alert function

We can create an alert box by using the .alert class and contextual classes like .alert-success, .alert-info, .alert-warning, .alert-danger, .alert-primary, .alert-secondary, .alert-light, or .alert-dark.

Syntax

The following syntax shows basic alert box in the web page.

```
<div class = "alert alert-success" id = "box1">
Short information!
</div>
```

Example

The following example uses basic alert box with different contextual classes. Here, we can place the short size information of the web page.

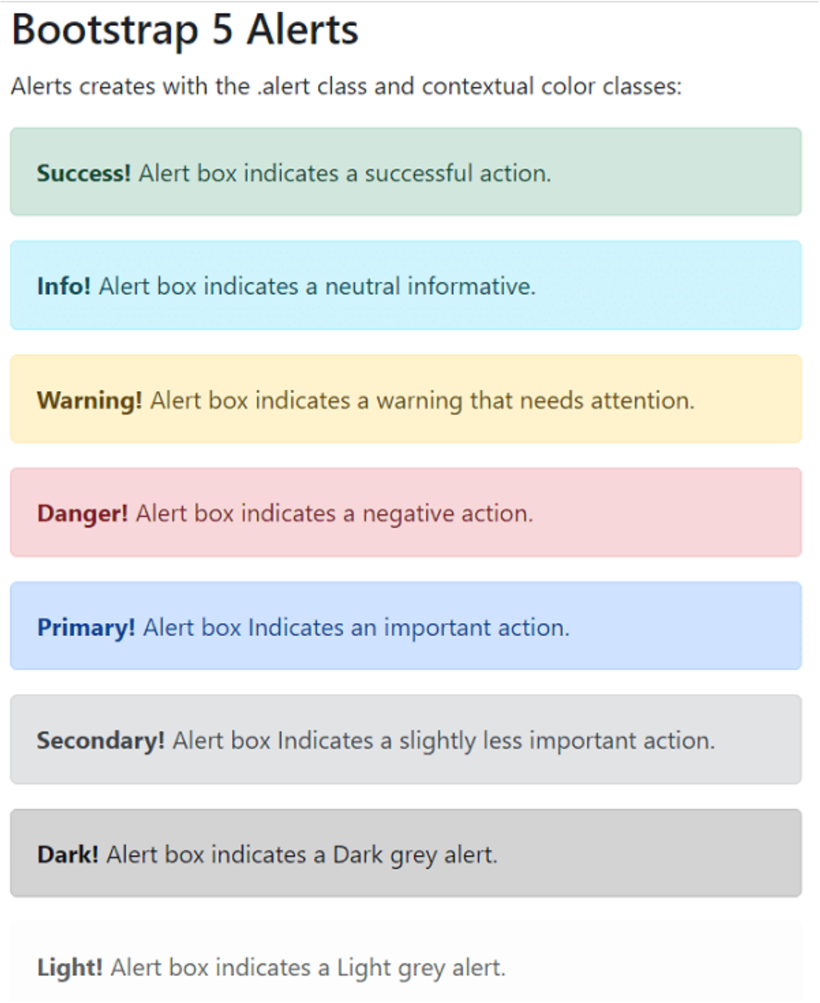
```
<!DOCTYPE html>
<html lang = "en">
<head>
<title> Bootstrap 5 alert Example </title>
<meta charset = "utf-8">
<meta name = "viewport" content = "width = device-width, initial-scale = 1">
  <link href = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel = "stylesheet">
  <script src = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js">
```

```
</script>
</head>
<body>
</head>
<body>
<div class = "container" id = "alertcont">
<h2> Bootstrap 5 Alerts </h2>
<p> Alerts creates with the .alert class and contextual color classes: </p>
<div class = "alert alert-success" id = "box1">
<b> Success! </b> Alert box indicates a successful action.
</div>
<div class = "alert alert-info" id = "box1">
<b> Info! </b> Alert box indicates a neutral informative.
</div>
<div class = "alert alert-warning" id = "box1">
<b> Warning! </b> Alert box indicates a warning that needs attention.
</div>
<div class = "alert alert-danger" id = "box1">
<b> Danger! </b> Alert box indicates a negative action.
</div>
<div class = "alert alert-primary" id = "box1">
<b> Primary! </b> Alert box Indicates an important action.
</div>
<div class = "alert alert-secondary" id = "box1">
<b> Secondary! </b> Alert box Indicates a slightly less important action.
</div>
<div class = "alert alert-dark" id = "box1">
<b> Dark! </b> Alert box indicates a Dark grey alert.
</div>
<div class = "alert alert-light" id = "box1">
<b> Light! </b> Alert box indicates a Light grey alert.
</div>
</div>
</body>
```

```
</html>
```

Output

The following output shows basic bootstrap 5 alert functions.



Bootstrap 5 Alerts

Alerts creates with the .alert class and contextual color classes:

Success! Alert box indicates a successful action.

Info! Alert box indicates a neutral informative.

Warning! Alert box indicates a warning that needs attention.

Danger! Alert box indicates a negative action.

Primary! Alert box Indicates an important action.

Secondary! Alert box Indicates a slightly less important action.

Dark! Alert box indicates a Dark grey alert.

Light! Alert box indicates a Light grey alert.

Bootstrap 5 Closing Alerts with a link

Add a ".alert-dismissible" class to the alert container to dismiss the alert message. The class = "btn-close" and data-bs-dismiss = "alert" attributes are applied to a link or a button element. When we click on the cross symbol, the alert box disappears.

To close the alert box, click the "x" icon to the right. The "&times;" is an HTML entity that, rather than the letter "x," is the chosen icon for close buttons.

Syntax

The following syntax shows closing alert box in the web page.

```
<div class = "alert alert-warning alert-dismissible" id = "box1">
<button type = "button" class = "btn-close" data-bs-dismiss = "alert" id = "close"> × </button>
Short information!
</div>
```

The following syntax shows alert box with link in the web page.

```
<div class = "alert alert-success" id = "box1">
Short information!
<a href = "#" class = "alert-link" id = "links"> Click here... </a>.
</div>
```

Example

The following example shows basic bootstrap 5 alert boxes with a close button. Here, we can use the link in the alert box. The example shows some alert boxes with links and others without links. The "light" alert removes using the cross button.

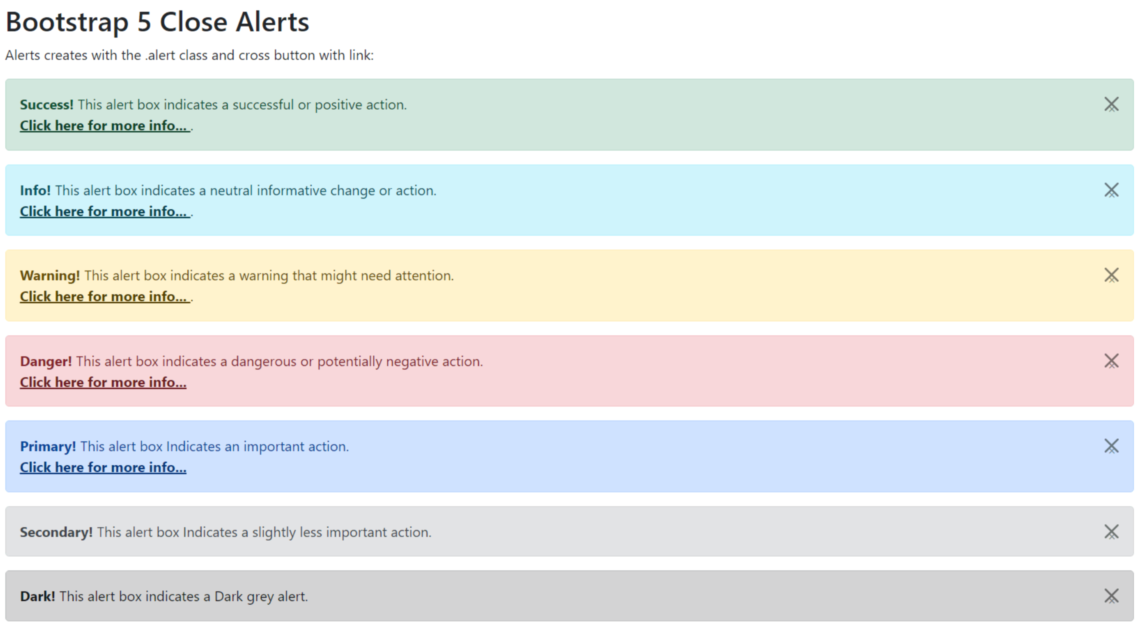
```
<!DOCTYPE html>
<html lang = "en">
<head>
<title> Bootstrap 5 alert Example </title>
<meta charset = "utf-8">
<meta name = "viewport" content="width=device-width, initial-scale=1">
<meta name = "viewport" content="width=device-width, initial-scale=1">
<script src = "https://cdn.jsdelivr.net/npm/jquery@3.6.0/dist/jquery.slim.min.js">
</script>
<link href = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel = "stylesheet">
<script src = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js">
</script>
</head>
<body>
```

```
<div class = "container" id = "alertcont">
<h2> Bootstrap 5 Close Alerts </h2>
<p> Alerts creates with the .alert class and cross button with link: </p>
<div class = "alert alert-success alert-dismissible" id = "box1">
<button type = "button" class = "btn-close" data-bs-dismiss = "alert" id = "close"> × </button>
<b> Success! </b> This alert box indicates a successful or positive action.
<br>
<a href = "#" class = "alert-link" id = "links"> Click here for more info... </a>.
</div>
<div class = "alert alert-info alert-dismissible" id = "box1">
<button type = "button" class = "btn-close" data-bs-dismiss = "alert" id = "close"> × </button>
<b> Info! </b> This alert box indicates a neutral informative change or action.
<br>
<a href = "#" class = "alert-link" id = "links"> Click here for more info... </a>.
</div>
<div class = "alert alert-warning alert-dismissible" id = "box1">
<button type = "button" class = "btn-close" data-bs-dismiss = "alert" id = "close"> × </button>
<b> Warning! </b> This alert box indicates a warning that might need attention.
<br>
<a href = "#" class = "alert-link" id = "links"> Click here for more info... </a>.
</div>
<div class = "alert alert-danger alert-dismissible" id = "box1">
<button type = "button" class = "btn-close" data-bs-dismiss = "alert" id = "close"> × </button>
<b> Danger! </b> This alert box indicates a dangerous or potentially negative action.
<br>
<a href = "#" class = "alert-link" id = "links"> Click here for more info... </a>
</div>
<div class = "alert alert-primary alert-dismissible" id = "box1">
<button type = "button" class = "btn-close" data-bs-dismiss = "alert" id = "close"> × </button>
<b> Primary! </b> This alert box Indicates an important action.
<br>
<a href = "https://www.javatpoint.com" class = "alert-link" id = "links"> Click here for more info... </a>
</div>
```

```
<div class = "alert alert-secondary alert-dismissible" id = "box1">
<button type = "button" class = "btn-close" data-bs-dismiss = "alert" id = "close"> × </button>
<b> Secondary! </b> This alert box Indicates a slightly less important action.
</div>
<div class = "alert alert-dark alert-dismissible" id = "box1">
<button type = "button" class = "btn-close" data-bs-dismiss = "alert" id = "close"> × </button>
<b> Dark! </b> This alert box indicates a Dark grey alert.
</div>
<div class = "alert alert-light alert-dismissible" id = "box1">
<button type = "button" class = "btn-close" data-bs-dismiss = "alert" id = "close"> × </button>
<b>Light!</b> This alert box indicates a Light grey alert.
</div>
</div>
</body>
</html>
```

Output

The following image represents bootstrap 5 close alert boxes with the link.



Bootstrap 5 Close Alerts

Alerts creates with the .alert class and cross button with link:

Success! This alert box indicates a successful or positive action.
Click here for more info... .

Info! This alert box indicates a neutral informative change or action.
Click here for more info... .

Warning! This alert box indicates a warning that might need attention.
Click here for more info... .

Danger! This alert box indicates a dangerous or potentially negative action.
Click here for more info...

Primary! This alert box Indicates an important action.
Click here for more info...

Secondary! This alert box Indicates a slightly less important action.

Dark! This alert box indicates a Dark grey alert.

Animated alert box

To close me, click the "x" sign to the right. I'm going to "fade" out. The .fade and .show classes create a fading effect when closing the alert message. The class = "btn-close" and data-bs-dismiss = "alert" attributes use in the <button> tag. When we click on the cross button, the alert box disappears with an animated effect.

Syntax

The following syntax shows animated alert box in the web page.

```
<div class = "alert alert-warning alert-dismissible fade show" id = "box1">
<button type = "button" class = "btn-close" data-bs-dismiss = "alert" id = "close"> × </button>
Short information!
</div>
```

Example

The following example shows alert box in animated format. Here, alert box shows and close functions use with special effect.

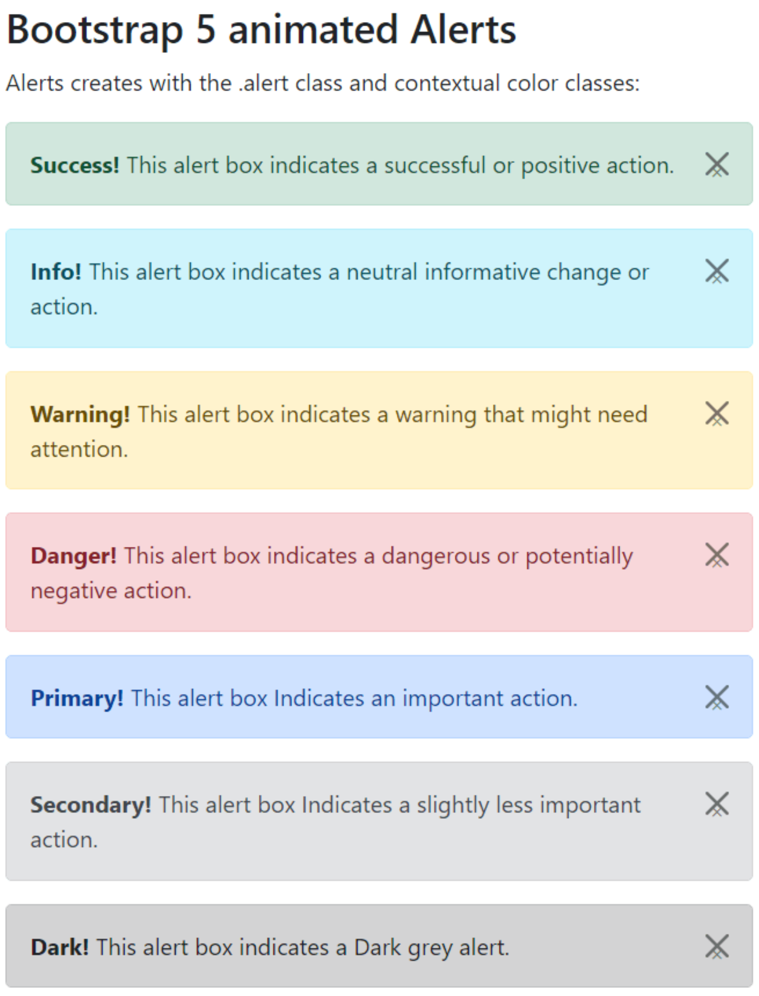
```
<!DOCTYPE html>
<html lang = "en">
<head>
<title> Bootstrap 5 alert Example </title>
<meta charset = "utf-8">
<meta name = "viewport" content="width=device-width, initial-scale=1">
<script src = "https://cdn.jsdelivr.net/npm/jquery@3.6.0/dist/jquery.slim.min.js">
</script>
<link href = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel = "stylesheet">
  <script src = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js">
</script>
</head>
<body>
<div class = "container" id = "alertcont">
```

```
<h2> Bootstrap 5 animated Alerts </h2>
<p> Alerts creates with the .alert class and contextual color classes: </p>
<div class = "alert alert-success alert-dismissible fade show" id = "box1">
<button type = "button" class = "btn-close" data-bs-dismiss = "alert" id = "close"> × </button>
<b> Success! </b> This alert box indicates a successful or positive action.
<br>
</div>
<div class = "alert alert-info alert-dismissible fade show" id = "box1">
<button type = "button" class = "btn-close" data-bs-dismiss = "alert" id = "close"> × </button>
<b> Info! </b> This alert box indicates a neutral informative change or action.
</div>
<div class = "alert alert-warning alert-dismissible fade show" id = "box1">
<button type = "button" class = "btn-close" data-bs-dismiss = "alert" id = "close"> × </button>
<b> Warning! </b> This alert box indicates a warning that might need attention.
</div>
<div class = "alert alert-danger alert-dismissible fade show" id = "box1">
<button type = "button" class = "btn-close" data-bs-dismiss = "alert" id = "close"> × </button>
<b> Danger! </b> This alert box indicates a dangerous or potentially negative action.
</div>
<div class = "alert alert-primary alert-dismissible fade show" id = "box1">
<button type = "button" class = "btn-close" data-bs-dismiss = "alert" id = "close"> × </button>
<b> Primary! </b> This alert box Indicates an important action.
</div>
<div class = "alert alert-secondary alert-dismissible fade show" id = "box1">
<button type = "button" class = "btn-close" data-bs-dismiss = "alert" id = "close"> × </button>
<b> Secondary! </b> This alert box Indicates a slightly less important action.
</div>
<div class  =  "alert alert-dark alert-dismissible fade show" id = "box1">
<button type = "button" class = "btn-close" data-bs-dismiss = "alert" id = "close"> × </button>
<b> Dark! </b> This alert box indicates a Dark grey alert.
</div>
</div>
</div>
</body>
```

```
</html>
```

Output

The following output shows an animated bootstrap 5 alert function.



Bootstrap 5 animated Alerts

Alerts creates with the .alert class and contextual color classes:

Success! This alert box indicates a successful or positive action. ×

Info! This alert box indicates a neutral informative change or action. ×

Warning! This alert box indicates a warning that might need attention. ×

Danger! This alert box indicates a dangerous or potentially negative action. ×

Primary! This alert box Indicates an important action. ×

Secondary! This alert box Indicates a slightly less important action. ×

Dark! This alert box indicates a Dark grey alert. ×

Bootstrap 5 Badges

Bootstrap 5 Badges uses to add extra information to any type of material. Use the ".badge" and contextual classes within <span> components to create rectangular badges. The scale of a badge must match the size of its parent element.

Badges in Heading

Bootstrap 5 badges use to know more information about heading. It places in the span tag according to heading size.

Syntax

The following syntax creates badges using bootstrap 5 classes.

```
<span class = "badge bg-info"> Value </span>
```

Example

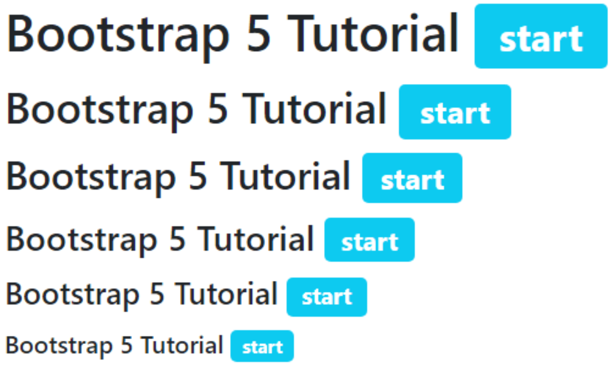
The following example shows badges inside of the heading tag. Each badge set its size according to the heading tag.

```
<!DOCTYPE html>
<html>
<head>
<title> Bootstrap 5 Badges Example </title>
<meta name = "viewport" content = "width=device-width, initial-scale=1">
<link href = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel = "stylesheet">
<script src = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js"></script>
</head>
<body>
<div class = "container">
<h1> Bootstrap 5 Tutorial <span class = "badge bg-info"> start </span>
</h1>
<h2> Bootstrap 5 Tutorial <span class = "badge bg-info"> start </span>
```

```
</h2>
<h3> Bootstrap 5 Tutorial <span class = "badge bg-info"> start </span>
</h3>
<h4> Bootstrap 5 Tutorial <span class = "badge bg-info"> start </span>
</h4>
<h5> Bootstrap 5 Tutorial <span class = "badge bg-info"> start </span>
</h5>
<h6> Bootstrap 5 Tutorial <span class = "badge bg-info"> start </span>
</h6>
</div>
</body>
</html>
```

Output

The following output shows badges with the heading tag.



Bootstrap 5 Tutorial start

Bootstrap 5 Tutorial start

Bootstrap 5 Tutorial start

Bootstrap 5 Tutorial start

Bootstrap 5 Tutorial start

Bootstrap 5 Tutorial start

Bootstrap 5 Contextual Badges

Bootstrap 5 badges give extra information about the parent element. Contextual badges provide different background colors as per user requirements.

Example

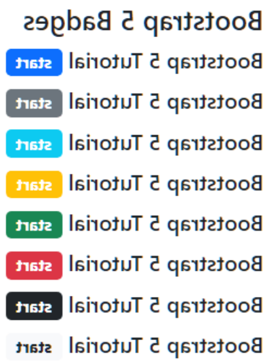
The following example shows different background color's badges inside of the heading tag. We can use contextual classes in the span tag with badge class.

```
<!DOCTYPE html>
<html>
```

```html
<head>
<title> Bootstrap 5 Badges Example </title>
<meta name = "viewport" content = "width=device-width, initial-scale=1">
<link href = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel = "stylesheet">
<script src = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js"></script>
</head>
<body>
<div class = "container">
<h5> Bootstrap 5 Badges </h5>
<h6> Bootstrap 5 Tutorial <span class = "badge bg-primary"> start </span>
</h6>
<h6> Bootstrap 5 Tutorial <span class = "badge bg-secondary"> start </span>
</h6>
<h6> Bootstrap 5 Tutorial <span class = "badge bg-info"> start </span>
</h6>
<h6> Bootstrap 5 Tutorial <span class = "badge bg-warning"> start </span>
</h6>
<h6> Bootstrap 5 Tutorial <span class = "badge bg-success"> start </span>
</h6>
<h6> Bootstrap 5 Tutorial <span class = "badge bg-danger"> start </span>
</h6>
<h6> Bootstrap 5 Tutorial <span class = "badge bg-dark"> start </span>
</h6>
<h6> Bootstrap 5 Tutorial <span class = "badge bg-light text-dark"> start </span>
</h6>
</div>
</body>
</html>
```

Output

The following output shows different background color badges with the <h6> tag.



Bootstrap 5 Badges

Bootstrap 5 Tutorial start

Bootstrap 5 Tutorial start

Bootstrap 5 Tutorial start

Bootstrap 5 Tutorial start

Bootstrap 5 Tutorial start

Bootstrap 5 Tutorial start

Bootstrap 5 Tutorial start

Bootstrap 5 Tutorial start

Bootstrap 5 pill Badges

Bootstrap 5 badges placed with the parent element for small information. We can use the "badge" class with the "rounded-pill" class to look pill size badges.

Syntax

The following syntax creates pill badges using bootstrap 5 classes.

```
<span class = "badge rounded-pill bg-info"> Value </span>
```

Example

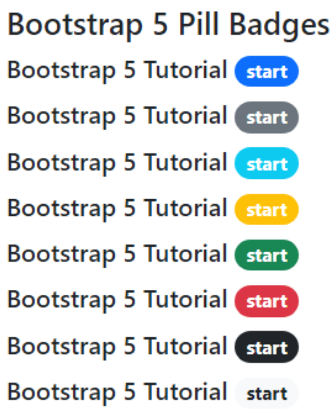
The following example shows pill style badges using bootstrap 5 classes in the web page.

```
<!DOCTYPE html>
<html>
<head>
<title> Bootstrap 5 Badges Example </title>
<meta name = "viewport" content = "width=device-width, initial-scale=1">
<link href = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel = "stylesheet">
<script src = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js"></script>
</head>
<body>
<div class = "container">
```

```
<h5> Bootstrap 5 Pill Badges </h5>
<h6> Bootstrap 5 Tutorial <span class = "badge rounded-pill bg-primary"> start </span>
</h6>
<h6> Bootstrap 5 Tutorial <span class = "badge rounded-pill bg-secondary"> start </span>
</h6>
<h6> Bootstrap 5 Tutorial <span class = "badge rounded-pill bg-info"> start </span>
</h6>
<h6> Bootstrap 5 Tutorial <span class = "badge rounded-pill bg-warning"> start </span>
</h6>
<h6> Bootstrap 5 Tutorial <span class = "badge rounded-pill bg-success"> start </span>
</h6>
<h6> Bootstrap 5 Tutorial <span class = "badge rounded-pill bg-danger"> start </span>
</h6>
<h6> Bootstrap 5 Tutorial <span class = "badge rounded-pill bg-dark"> start </span>
</h6>
<h6> Bootstrap 5 Tutorial <span class = "badge rounded-pill bg-light text-dark"> start </span>
</h6>
</div>
</body>
</html>
```

Output

The following output shows pill badges with the given tag.



Bootstrap 5 Pill Badges

Bootstrap 5 Tutorial start

Bootstrap 5 Tutorial start

Bootstrap 5 Tutorial start

Bootstrap 5 Tutorial start

Bootstrap 5 Tutorial start

Bootstrap 5 Tutorial start

Bootstrap 5 Tutorial start

Bootstrap 5 Tutorial start

Badges in Button

Bootstrap 5 badges are used inside of the button. The button function displays extra information about the button function. For example, the button displays badges with unread messages (numbers) of the web function.

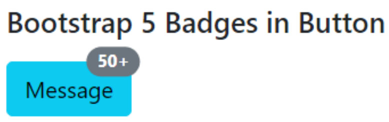
Example

The following example shows badges inside of the button function.

```
<!DOCTYPE html>
<html>
<head>
<title> Bootstrap 5 Badges Example </title>
<meta name = "viewport" content = "width=device-width, initial-scale=1">
<link href = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel = "stylesheet">
<script src = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js"></script>
</head>
<body>
<div class = "container">
<h5> Bootstrap 5 Badges in Button </h5>
<button type="button" class="btn btn-info position-relative mt-2">
  Message
  <span class="position-absolute top-1 start-95 translate-middle badge rounded-pill bg-secondary"> 50+
  </span>
</button>
</div>
</body>
</html>
```

Output

The following output shows badges in the button function.


Bootstrap 5 Badges in Button

Bootstrap 5 Button

The Bootstrap 5 button element is usable wherever text-level markup is permitted in the document body. Such an element is not required to be related to a form element.

The buttons in bootstrap allow the user to submit the form to the server for processing.

Utilize Bootstrap 5 custom button styles for actions in forms, dialogues, and more, supporting different sizes, states, and other options.

Bootstrap 5 contextual button

The bootstrap 5 button uses the button tag and types with classes. The "btn" class uses the button tag to display the default button format. The contextual button classes used on the web page such as a "btn-success", "btn-primary", "btn-secondary," and so on. Here, we can use contextual type for different colors and meanings of the button.

Syntax

The following syntax shows default button using bootstrap 5 classes.

```
<button type = "button" class = "btn"> Default Button </button>
```

The following syntax shows color button using bootstrap 5 classes.

```
<button type = "button" class = "btn btn-primary"> Primary Button </button>
```

- Here, we use primary button class in the <button> tag.
- Bootstrap 5 provides several contextual button classes for a different style.

Example

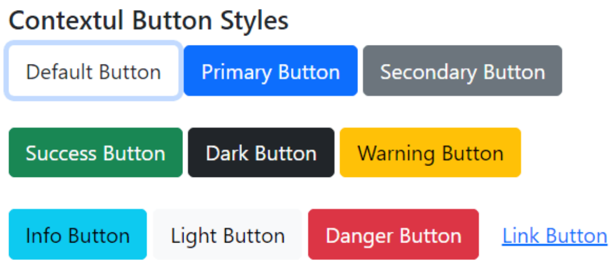
The following example shows different type of the button and its meaning. This button uses contextual classes to display specific button.

```
<!DOCTYPE html>
<html>
<head>
<title> Bootstrap Button Example </title>
```

```
<meta name = "viewport" content = "width=device-width, initial-scale=1">
<link href = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel = "stylesheet">
<script src = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js"></script>
</head>
<body>
<div class = "container mt-2">
<h5> Contextul Button Styles </h5>
<button type = "button" class = "btn"> Default Button </button>
<button type = "button" class = "btn btn-primary"> Primary Button </button>
<button type = "button" class = "btn btn-secondary"> Secondary Button </button>
<br><br>
<button type = "button" class = "btn btn-success"> Success Button </button>
<button type = "button" class = "btn btn-dark"> Dark Button </button>
<button type = "button" class = "btn btn-warning"> Warning Button </button>
<br><br>
<button type = "button" class = "btn btn-info"> Info Button </button>
<button type = "button" class = "btn btn-light"> Light Button </button>
<button type = "button" class = "btn btn-danger"> Danger Button </button>
<button type = "button" class = "btn btn-link"> Link Button </button>
</div>
</body>
</html>
```

Output

The following image shows a different button with its class on the html page.



Bootstrap 5 button tag

The bootstrap 5 button uses the different button tags and types with classes. The "btn" class requires in each button tag. If the form requires a color button, we can use contextual classes.

The <a> and <button> tag uses with closing tag to display button link and button function respectively. The <input> tag uses a different button element type without a closing tag. We can use the button, submit, and reset type in the input tag.

Syntax

The following syntax shows button using <a> tag for the link button.

```
<a href = "#" class = "btn"> Button </a>
```

The following syntax shows the button using the <button> tag for the button element.

```
<button type = "button" class = "btn"> Button </button>
```

The following syntax shows button using <input> tag for the submit form data.

```
<input type = "submit" class = "btn" value = "Submit">
```

The following syntax shows button using <input> tag for the reset form data.

```
<input type = "reset" class = "btn" value = "Reset">
```

The following syntax shows button using <input> tag for the button function.

```
<input type = "button" class = "btn" value = "Button">
```

Example

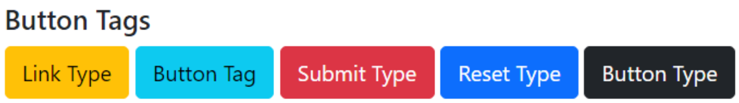
The following example shows different type of the buttons use in the bootstrap 5 form.

```
<!DOCTYPE html>
<html>
<head>
```

```
<title> Bootstrap Button Example </title>
<meta name = "viewport" content="width=device-width, initial-scale=1">
<link href = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel = "stylesheet">
<script src = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js"></script>
</head>
<body>
<div class = "container mt-2">
<h5> Button Tags </h5>
<a href = "#" class = "btn btn-warning"> Link Type </a>
<button type = "button" class = "btn btn-info"> Button Tag </button>
<input type = "submit" class = "btn btn-danger" value = "Submit Type">
<input type = "reset" class = "btn btn-primary" value = "Reset Type">
<input type = "button" class = "btn btn-dark" value = "Button Type">
</div>
</body>
</html>
```

Output

The following image shows a different button with its tags on the html page.



Bootstrap 5 button outline

The button outline shows a border around the button. The bootstrap 5 button uses the button tag with contextual outline classes. The "btn" class uses the button tag to display the default button format. We can use contextual class for different color borders and white color body buttons here.

Syntax

The following syntax shows a color bordered button using bootstrap 5 classes.

```
<button type = "button" class = "btn btn-outline-primary"> Primary Button </button>
```

- We use the primary button outline class in the <button> tag.
- Bootstrap 5 provide several contextual button outline classes for different border color.

Example

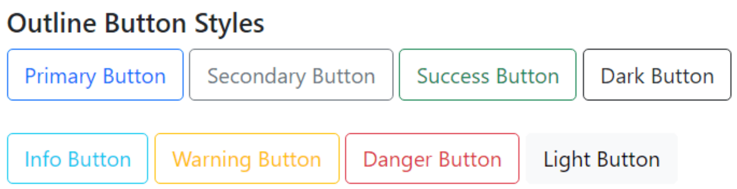
The following example shows different type of the button and its meaning. This button uses contextual classes to display specific button.

```
<!DOCTYPE html>
<html>
<head>
 <title> Bootstrap Button Example </title>
<meta name = "viewport" content = "width=device-width, initial-scale=1">
<link href = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel = "stylesheet">
<script src = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js"></script>
</head>
<body>
<div class = "container mt-2">
 <h5> Outline Button Styles </h5>
  <button type = "button" class = "btn btn-outline-primary"> Primary Button </button>
  <button type = "button" class = "btn btn-outline-secondary"> Secondary Button </button>
  <button type = "button" class = "btn btn-outline-success"> Success Button </button>
  <button type = "button" class = "btn btn-outline-dark"> Dark Button </button>
 <br><br>
  <button type = "button" class = "btn btn-outline-info"> Info Button </button>
  <button type = "button" class = "btn btn-outline-warning"> Warning Button </button>
  <button type = "button" class = "btn btn-outline-danger"> Danger Button </button>
  <button type = "button" class = "btn btn-outline-light"> Light Button </button>
</div>
</body>
```

```
</html>
```

Output

The following image shows a different button with its class on the html page.



Button Size

Bootstrap 5 button function provides classes for button size. The button uses large, default, and small sizes. If we do not use button size classes, then bootstrap 5 shows the default size of the button.

Syntax

The following syntax shows a color bordered button using bootstrap 5 classes.

<button type = "button" class = "btn btn-primary btn-lg|btn-sm"> Primary Button </button>

- The "btn-lg" button class uses for large-size buttons.
- The "btn-sm" button class uses for small-size buttons.

Example

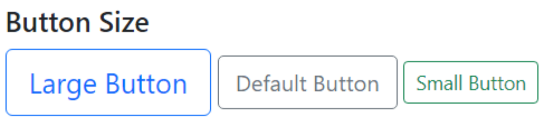
The following example shows different size of the button in the web page. Here, first button shows large size button, second button shows default size button and third button shows small size button.

```
<!DOCTYPE html>
<html>
<head>
<title> Bootstrap Button Example </title>
<meta name = "viewport" content = "width=device-width, initial-scale=1">
```

```
<link href = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel = "stylesheet">
<script src = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js"></script>
</head>
<body>
<div class = "container mt-2">
<h5> Button Size </h5>
<button type = "button" class = "btn btn-outline-primary btn-lg"> Large Button </button>
<button type = "button" class = "btn btn-outline-secondary"> Default Button </button>
<button type = "button" class = "btn btn-outline-success btn-sm"> Small Button </button>
</div>
</body>
</html>
```

Output

The following image shows a different button with its class on the html page.



Button Size

Bootstrap 5 button group

Bootstrap 5 provides one class to create multiple buttons simultaneously. Group a collection of multiple buttons on a single line or in a column. Button group displays in one line but handles different operations on the web page. We can create different contextual button groups, radio button groups, and checkbox button groups vertically or horizontally.

Basic button group

Bootstrap 5 button group creates a different button in one single line group. All buttons have their operation in the form.

Syntax

The following syntax shows to create single group of the multiple buttons.

```
<div class = "btn-group">
  <button type = "button" class = "btn"> Name </button>
</div>
```

Example

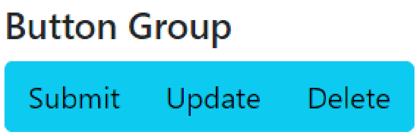
The following example creates three similar buttons in single row with different values. Here, we can see submit, update and delete buttons in single tab.

```
<!DOCTYPE html>
<html>
<head>
<title> Bootstrap Button Example </title>
<meta name = "viewport" content = "width=device-width, initial-scale=1">
<link href = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel = "stylesheet">
<script src = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js"></script>
</head>
<body>
<div class = "container mt-2">
```

```
<h5> Button Group </h5>
<div class = "btn-group">
<button type = "button" class = "btn btn-info"> Submit </button>
<button type = "button" class = "btn btn-info"> Update </button>
<button type = "button" class = "btn btn-info"> Delete </button>
</div>
</div>
</body>
</html>
```

Output

The following output image shows one tab for different buttons.



Contextual button group

Bootstrap 5 button group creates different color buttons in one single line group. All buttons have their operation and meaning in the form.

Example

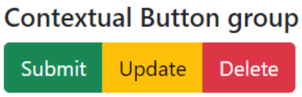
The following example creates three different buttons in single line with different values. Here, we can see submit, update and delete buttons with different color.

```
<!DOCTYPE html>
<html>
<head>
<title> Bootstrap Button group Example </title>
<meta name = "viewport" content = "width=device-width, initial-scale=1">
<link href = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel = "stylesheet">
<script src = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js"></script>
```

```
</head>
<body>
<div class = "container mt-2">
<h5> Contextual Button group </h5>
<div class = "btn-group">
<button type = "button" class = "btn btn-success"> Submit </button>
<button type = "button" class = "btn btn-warning"> Update </button>
<button type = "button" class = "btn btn-danger"> Delete </button>
</div>
</div>
</body>
</html>
```

Output

The following output image shows one tab for different contextual buttons.



Contextual Button group

Button group Sizes

Bootstrap 5 button group creates a different size of the button group.

Syntax

The following syntax shows to large size button group.

```
<div class = "btn-group btn-group-lg"> Buttons. </div>
```

The following syntax shows to small size button group.

```
<div class = "btn-group btn-group-sm"> Buttons. </div>
```

Example

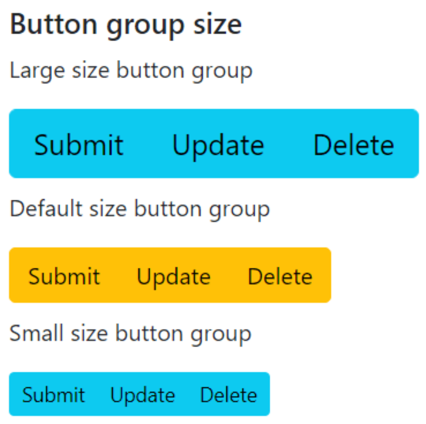
The following example shows three sizes of the button groups using bootstrap 5 classes.

```
<!DOCTYPE html>
```

```
<html>
<head>
<title> Bootstrap5 Button group Example </title>
<meta name = "viewport" content = "width=device-width, initial-scale=1">
<link href = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel = "stylesheet">
<script src = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js"></script>
</head>
<body>
<div class = "container mt-2">
<h5> Button group size </h5>
<p> Large size button group </p>
<div class = "btn-group btn-group-lg mb-2">
<button type = "button" class = "btn btn-info"> Submit </button>
<button type = "button" class = "btn btn-info"> Update </button>
<button type = "button" class = "btn btn-info"> Delete </button>
</div>
<p> Default size button group </p>
<div class = "btn-group mb-2">
<button type = "button" class = "btn btn-warning"> Submit </button>
<button type = "button" class = "btn btn-warning "> Update </button>
<button type = "button" class = "btn btn-warning "> Delete </button>
</div>
<p> Small size button group </p>
<div class = "btn-group btn-group-sm mb-2">
<button type = "button" class = "btn btn-info"> Submit </button>
<button type = "button" class = "btn btn-info"> Update </button>
<button type = "button" class = "btn btn-info"> Delete </button>
</div>
</div>
</body>
</html>
```

Output

The following output image shows one tab for different buttons.



Button group size

Large size button group

Default size button group

Small size button group

Vertical button group

Bootstrap 5 button groups create a different button in one single column group. All buttons have a different operation in the form.

Syntax

The following syntax shows to create single group of the multiple buttons.

```
<div class = "btn-group-vertical">
  <button type = "button" class = "btn"> Name </button>
</div>
```

Example

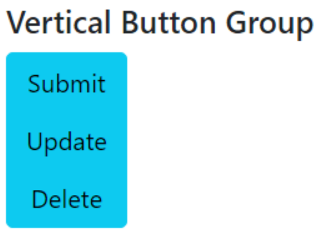
The following example creates three similar types of the button in single column with different values. Here, bootstrap 5 creates submit, update and delete buttons vertically.

```
<!DOCTYPE html>
<html>
<head>
<title> Bootstrap Button Example </title>
<meta name = "viewport" content = "width=device-width, initial-scale=1">
```

```
<link href = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel = "stylesheet">
<script src = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js">
</script>
</head>
<body>
<div class = "container mt-2">
<h5> Vertical Button Group </h5>
<div class = "btn-group-vertical">
<button type = "button" class = "btn btn-info"> Submit </button>
<button type = "button" class = "btn btn-info"> Update </button>
<button type = "button" class = "btn btn-info"> Delete </button>
</div>
</div>
</body>
</html>
```

Output

The following output image shows one vertical tab for different buttons.



Outline button group

Bootstrap 5 button group creates a similar bordered button in one single line group. All buttons have their operation in the form.

Syntax

The following syntax shows to create single group of the multiple buttons with border style. Here, we can use the "btn-outline-info" class to display bordered button.

```
<div class = "btn-group">
  <button type = "button" class = "btn btn-outline-info"> Name </button>
</div>
```

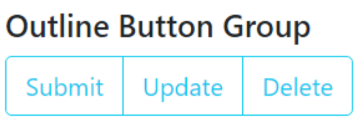
Example

The following example creates three buttons in single row with border. Here, we can see submit, update and delete buttons in single tab.

```
<!DOCTYPE html>
<html>
<head>
<title> Bootstrap Button Example </title>
<meta name = "viewport" content = "width=device-width, initial-scale=1">
<link href = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel = "stylesheet">
<script src = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js"></script>
</head>
<body>
<div class = "container mt-2">
<h5> Outline Button Group </h5>
<div class = "btn-group">
<button type = "button" class = "btn btn-outline-info"> Submit </button>
<button type = "button" class = "btn btn-outline-info"> Update </button>
<button type = "button" class = "btn btn-outline-info"> Delete </button>
</div>
</div>
</body>
</html>
```

Output

The following output image shows bordered button group in one row.


Outline Button Group

Checkbox button group

Bootstrap 5 button groups create multiple checkbox buttons in one single row. The button groups either select a single checkbox or select multiple checkboxes.

Syntax

The following syntax shows to create single group of the checkbox buttons.

```
<div class = "btn-group">
  <input type = "checkbox" class = "btn-check" id = "check1" autocomplete = "off">
  <label class = "btn btn-primary" for = "check1"> Select1 </label>
</div>
```

Example

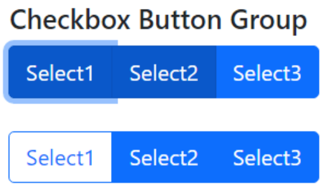
The following example creates three similar types of the button in single column with different values. Here, bootstrap 5 creates submit, update and delete buttons vertically.

```
<!DOCTYPE html>
<html>
<head>
<title> Bootstrap Button Example </title>
<meta name = "viewport" content = "width=device-width, initial-scale=1">
<link href = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel = "stylesheet">
<script src = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js">
</script>
</head>
<body>
<div class = "container mt-2">
<h5> Checkbox Button Group </h5>
<div class = "btn-group ">
<input type = "checkbox" class = "btn-check" id = "check1" autocomplete = "off">
```

```
<label class = "btn btn-primary" for = "check1"> Select1 </label>
<input type = "checkbox" class = "btn-check" id = "check2" autocomplete = "off">
<label class = "btn btn-primary" for = "check2"> Select2 </label>
<input type = "checkbox" class = "btn-check" id = "check3" autocomplete = "off">
<label class = "btn btn-primary" for = "check3"> Select3 </label>
</div>
<br><br>
<div class = "btn-group">
<input type = "checkbox" class = "btn-check" id = "checks1" autocomplete = "off">
<label class = "btn btn-outline-primary" for = "checks1"> Select1 </label>
<input type = "checkbox" class = "btn-check" id = "checks2" autocomplete = "off">
<label class = "btn btn-outline-primary" for = "checks2"> Select2 </label>
<input type = "checkbox" class = "btn-check" id = "checks3" autocomplete = "off">
<label class = "btn btn-outline-primary" for = "checks3"> Select3 </label>
</div>
</div>
</body>
</html>
```

Output

The following output image shows one button group for multiple checkbox buttons.



Radio button group

Bootstrap 5 button groups create multiple radio buttons in one single row. The button groups select a single radio button at one time.

Syntax

The following syntax shows to create single group of the radio buttons.

```
<div class = "btn-group">
  <input type = "radio" class = "btn-check" name = "radiobtn" id = "rdio1" autocomplete = "off">
  <label class = "btn btn-primary" for = "rdio1"> Value </label>
</div>
```

Example

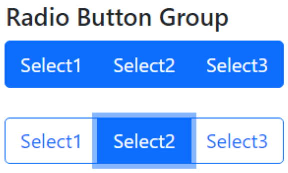
The following example shows radio button group with contextual button class and outline button class.

```
<!DOCTYPE html>
<html>
<head>
<title> Bootstrap Button Example </title>
<meta name = "viewport" content = "width=device-width, initial-scale=1">
<link href = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel = "stylesheet">
<script src = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js">
</script>
</head>
<body>
<div class = "container mt-2">
<h5> Radio Button Group </h5>
<div class = "btn-group ">
<input type = "radio" class = "btn-check" name = "radiobtn" id = "check1" autocomplete = "off">
<label class = "btn btn-primary" for = "check1"> Select1 </label>
<input type = "radio" class = "btn-check" name = "radiobtn" id = "check2" autocomplete = "off">
<label class = "btn btn-primary" for = "check2"> Select2 </label>
<input type = "radio" class = "btn-check" name = "radiobtn" id = "check3" autocomplete = "off">
<label class = "btn btn-primary" for = "check3"> Select3 </label>
</div>
<br><br>
<div class = "btn-group">
<input type = "radio" class = "btn-check" name = "radiobtn" id = "checks1" autocomplete = "off">
<label class = "btn btn-outline-primary" for = "checks1"> Select1 </label>
```

```
<input type = "radio" class = "btn-check" name = "radiobtn" id = "checks2" autocomplete = "off">
<label class = "btn btn-outline-primary" for = "checks2"> Select2 </label>
<input type = "radio" class = "btn-check" name = "radiobtn" id = "checks3" autocomplete = "off">
<label class = "btn btn-outline-primary" for = "checks3"> Select3 </label>
</div>
</div>
</body>
</html>
```

Output

The following output image shows one button group for multiple radio buttons.



Button Toolbar

Bootstrap 5 button group contains multiple buttons in one tool. Bootstrap 5 button toolbar contains multiple button groups in a single row. We can contain multiple different button groups in one button toolbar.

Combine sets of button groups into button toolbars for components with a higher level of complexity. Use utility classes to spread out groups, buttons, and other elements.

Syntax

The following syntax shows to create button toolbar for the multiple button groups.

```
<div class = "btn-toolbar">
<div class = "btn-group">
   <button type = "button" class = "btn"> Name </button>
</div>
</div>
```

Example

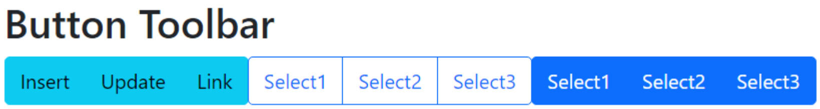
The following example creates three similar buttons in single button groups with different values. Here, we can place multiple button groups like button, checkboxes, and outline radio buttons in toolbar.

```
<!DOCTYPE html>
<html>
<head>
<title> Bootstrap Button Example </title>
<meta name = "viewport" content = "width=device-width, initial-scale=1">
<link href = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel = "stylesheet">
<script src = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js"></script>
</head>
<body>
<div class = "container mt-2">
<h2> Button group </h2>
<div class = "btn-toolbar" role = "toolbar">
<div class = "btn-group" role = "group">
<button type = "button" class = "btn btn-info"> Insert </button>
<button type = "button" class = "btn btn-info"> Update </button>
<button type = "button" class = "btn btn-info"> Link </button>
</div>
<div class = "btn-group" role = "group">
<input type = "radio" class = "btn-check" name = "radiobtn" id = "checks1" autocomplete = "off">
<label class = "btn btn-outline-primary" for = "checks1"> Select1 </label>
<input type = "radio" class = "btn-check" name = "radiobtn" id = "checks2" autocomplete = "off">
<label class = "btn btn-outline-primary" for = "checks2"> Select2 </label>
<input type = "radio" class = "btn-check" name = "radiobtn" id = "checks3" autocomplete = "off">
<label class = "btn btn-outline-primary" for = "checks3"> Select3 </label>
</div>
<div class = "btn-group" role = "group">
<input type = "checkbox" class = "btn-check" id = "check1" autocomplete = "off">
```

```
<label class = "btn btn-primary" for = "check1"> Select1 </label>
<input type = "checkbox" class = "btn-check" id = "check2" autocomplete = "off">
<label class = "btn btn-primary" for = "check2"> Select2 </label>
<input type = "checkbox" class = "btn-check" id = "check3" autocomplete = "off">
<label class = "btn btn-primary" for = "check3"> Select3 </label>
</div>
</div>
</div>
</body>
</html>
```

Output

The following output image shows different button groups in a single toolbar.



Bootstrap 5 Progress Bar

Bootstrap 5 progress bars show the data and ratings about the particular functions. If we want to display a value out of 100%, then use the progress bar. You can set the final value of the function and display the progress value.

Basic Progress Bar

The "progress" and "progress-bar" apply to a container element and its child element, respectively. Set the width of the progress bar using the width property of CSS. The style width is used to display the progress bar's value.

Syntax

The following syntax shows how to create progress bar using bootstrap 5 classes and css stayle.

```
<div class = "progress">
<div class = "progress-bar" style = "width:80%">
</div>
</div>
```

Example

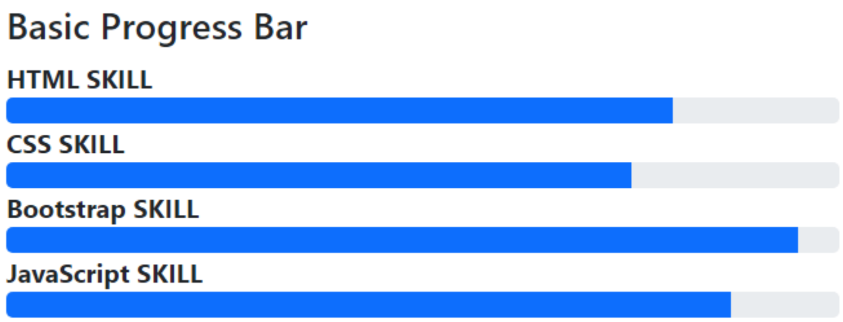
The example shows basic and default progress bar according to the skill set. The width property sets value of the function in the progress bar.

```
<!DOCTYPE html>
<html>
<head>
<title> Bootstrap 5 Progress bar Example </title>
<meta name = "viewport" content = "width=device-width, initial-scale=1">
<link href = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel = "stylesheet">
<script src = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js"></script>
</head>
<body>
```

```
<div class = "container">
<h4> Basic Progress Bar </h4>
<b> HTML SKILL </b>
<div class = "progress">
<div class = "progress-bar" style = "width:80%">
</div>
</div>
<b> CSS SKILL </b>
<div class = "progress">
<div class = "progress-bar" style = "width:75%">
</div>
</div>
<b> Bootstrap SKILL </b>
<div class = "progress">
<div class = "progress-bar" style = "width:95%">
</div>
</div>
<b> JavaScript SKILL </b>
<div class = "progress">
<div class = "progress-bar" style = "width:87%">
</div>
</div>
</div>
</body>
</html>
```

Output

The following image shows a basic progress bar with different values.



Basic Progress Bar

HTML SKILL

CSS SKILL

Bootstrap SKILL

JavaScript SKILL

Height of Progress Bar

The "progress" and "progress-bar" apply height to the progress bar and its value. The height property sets with the "progress" class then bootstrap sets the height of the progress bar. The height property is set with the "progress-bar" class then the bootstrap function sets the height of the progress value. The heights of the "progress" and "progress-bar" classes are similar values.

Syntax

The following syntax shows height of the progress bar.

```
<div class = "progress" style = "height:20px">
<div class = "progress-bar" style = "width:80%">
</div>
</div>
```

The following syntax shows height of the progress bar and its value.

```
<div class = "progress" style = "height:20px">
<div class = "progress-bar" style = "width:80%; height:20px">
</div>
</div>
```

Example

The example shows progress bar with the required height. The height property sets value of the function in the progress class.

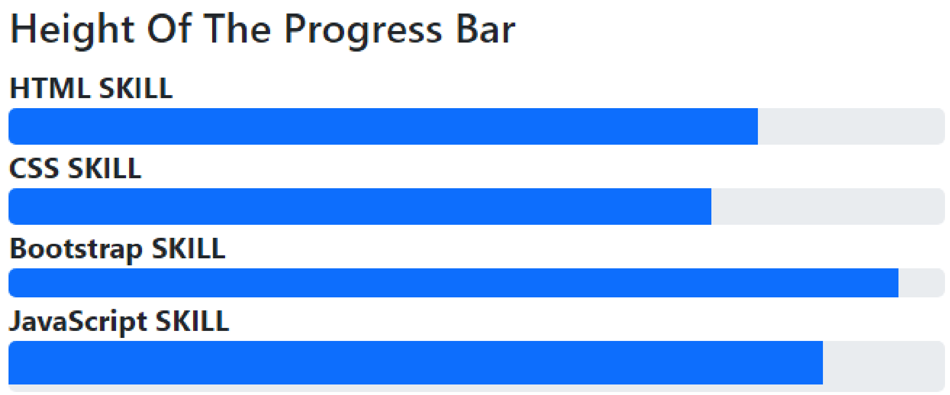
```
<!DOCTYPE html>
```

```html
<html>
<head>
<title> Bootstrap 5 Progress bar Example </title>
<meta name = "viewport" content = "width=device-width, initial-scale=1">
<link href = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel = "stylesheet">
<script src = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js"></script>
</head>
<body>
<div class = "container">
<h4> Height Of The Progress Bar </h4>
<b> HTML SKILL </b>
<div class = "progress" style = "height: 20px">
<div class = "progress-bar" style = "width:80%">
</div>
</div>
<b> CSS SKILL </b>
<div class = "progress" style = "height: 20px">
<div class = "progress-bar" style = "width:75%; height:20px">
</div>
</div>
<b> Bootstrap SKILL </b>
<div class = "progress">
<div class = "progress-bar" style = "width:95%">
</div>
</div>
<b> JavaScript SKILL </b>
<div class = "progress" style = "height: 28px">
<div class = "progress-bar" style = "width:87%; height:24px">
</div>
</div>
</div>
</body>
```

```
</html>
```

Output

The following image shows the different heights of the progress bar.



Height Of The Progress Bar

HTML SKILL

CSS SKILL

Bootstrap SKILL

JavaScript SKILL

Label of Progress Bar

The label of the progress bar shows extra and essential information inside of the bar. We can place the value of the progress or any information in text format.

Syntax

The following syntax shows how to create progress bar with label using bootstrap 5 classes.

```
<div class = "progress">
<div class = "progress-bar" style = "width:80%"> Label data </div>
</div>
```

Example

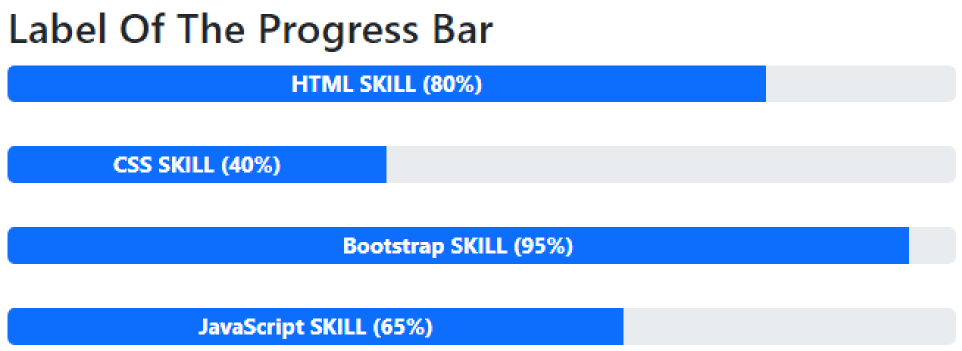
The example shows label of the progress bar. Here, we can write information, value and requires small data of the bar.

```
<!DOCTYPE html>
<html>
<head>
<title> Bootstrap 5 Progress bar Example </title>
```

```
<meta name = "viewport" content = "width=device-width, initial-scale=1">
<link href = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel = "stylesheet">
<script src = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js">
</script>
</head>
<body>
<div class = "container">
<h4> Label Of The Progress Bar </h4>
<div class = "progress" style = "height: 20px">
<div class = "progress-bar" style = "width:80%">
 <b> HTML SKILL (80%) </b>
</div>
</div> <br>
<div class = "progress" style = "height: 20px">
<div class = "progress-bar" style = "width:40%">
<b> CSS SKILL (40%) </b> </div>
</div> <br>
<div class = "progress" style = "height: 20px">
<div class = "progress-bar" style = "width: 95%">
<b> Bootstrap SKILL (95%) </b> </div>
</div> <br>
<div class = "progress" style = "height: 20px">
<div class = "progress-bar" style = "width: 65%;">
<b> JavaScript SKILL (65%) </b> </div>
</div>
</div>
</body>
</html>
```

Output

The following output image shows the progress bar and its required labels.



Label Of The Progress Bar

HTML SKILL (80%)

CSS SKILL (40%)

Bootstrap SKILL (95%)

JavaScript SKILL (65%)

Contextual Progress Bar

The "progress" and "progress-bar" apply to a parent and its child element, respectively. This progress bar shows different background colors with contextual background classes. The default progress bar shows the primary background color of the bootstrap 5 class.

This contextual progress bar shows the meaning of the information. For example, warning background contextual class (bg-warning) shows a warning about the function or element.

Syntax

The following syntax shows how to create progress bar using bootstrap 5 classes and css stayle.

```
<div class = "progress">
<div class = "progress-bar bg-warning" style = "width:80%">
</div>
</div>
```

Example

The example shows different background color of the progress bar as per requirement.

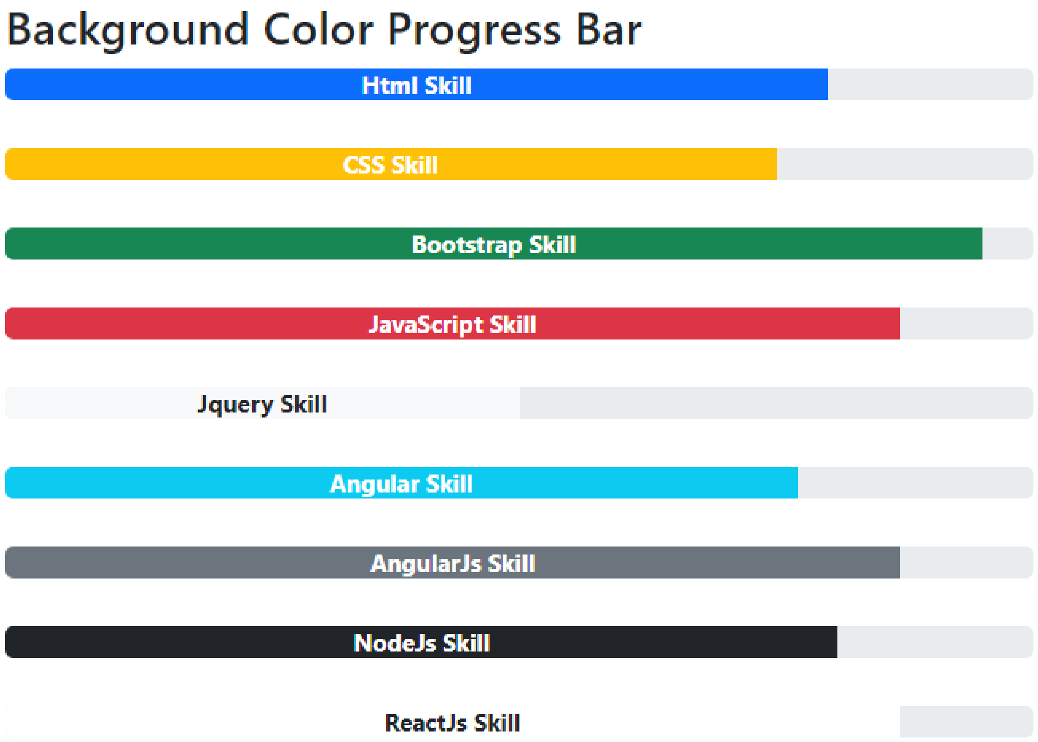
```
<!DOCTYPE html>
<html>
<head>
<title> Bootstrap 5 Progress bar Example </title>
<meta name = "viewport" content = "width=device-width, initial-scale=1">
```

```
<link href = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel = "stylesheet">
<script src = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js"></script>
</head>
<body>
<div class = "container">
<h4> Background Color Progress Bar </h4>
<div class = "progress">
<div class = "progress-bar" style = "width:80%">
<b> Html Skill </b>
</div>
</div>
<br>
<div class = "progress">
<div class = "progress-bar bg-warning" style = "width:75%">
<b> CSS Skill </b>
</div>
</div>
<br>
<div class = "progress">
<div class = "progress-bar bg-success" style = "width:95%">
<b> Bootstrap Skill </b>
</div>
</div>
<br>
<div class = "progress">
<div class = "progress-bar bg-danger" style = "width:87%">
<b> JavaScript Skill </b>
</div>
</div>
<br>
<div class = "progress">
<div class = "progress-bar bg-light text-dark" style = "width:50%">
```

```html
<b> Jquery Skill </b>
</div>
</div>
<br>
<div class = "progress">
<div class = "progress-bar bg-info" style = "width:77%">
<b> Angular Skill </b>
</div>
</div>
<br>
<div class = "progress">
<div class = "progress-bar bg-secondary" style = "width:87%">
<b> AngularJs Skill </b>
</div>
</div>
<br>
<div class = "progress">
<div class = "progress-bar bg-dark" style = "width:81%">
<b> NodeJs Skill </b>
</div>
</div>
<br>
<div class = "progress">
<div class = "progress-bar bg-white text-dark" style = "width:87%">
<b> ReactJs Skill </b>
</div>
</div>
</div>
</body>
</html>
```

Output

The following image shows a basic progress bar with different values.



Background Color Progress Bar

Striped Progress Bar

The striped progress bar shows stripe style in the progress bar. The striped progress bar comes with contextual classes to look attractive. We can set the height and apply the label on the progress bar as per the requirement.

Syntax

The following syntax shows how to create striped progress bar using bootstrap 5 classes and css stayle.

```
<div class = "progress">
<div class = "progress-bar progress-bar-striped" style = "width:80%">
</div>
</div>
```

Example

The example shows different background color of the progress bar as per requirement.

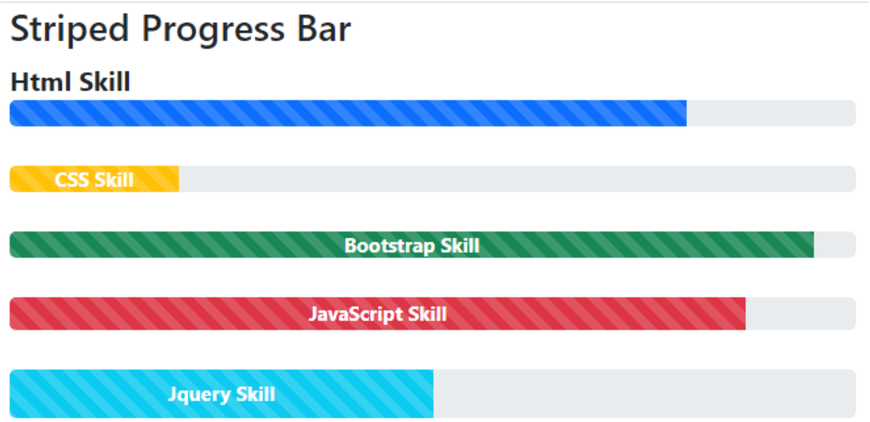
```
<!DOCTYPE html>
```

```
<html>
<head>
<title> Bootstrap 5 Progress bar Example </title>
<meta name = "viewport" content = "width=device-width, initial-scale=1">
<link href = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel = "stylesheet">
<script src = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js"></script>
</head>
<body>
<div class = "container">
<h4> Striped Progress Bar </h4>
<b> Html Skill </b>
<div class = "progress">
<div class = "progress-bar progress-bar-striped" style = "width:80%">
</div>
</div>
<br>
<div class = "progress">
<div class = "progress-bar bg-warning progress-bar-striped" style = "width:20%">
<b> CSS Skill </b>
</div>
</div>
<br>
<div class = "progress">
<div class = "progress-bar bg-success progress-bar-striped" style = "width:95%">
<b> Bootstrap Skill </b>
</div>
</div>
<br>
<div class = "progress" style = "height:20px">
<div class = "progress-bar bg-danger progress-bar-striped" style = "width:87%">
<b> JavaScript Skill </b>
</div>
```

```
</div>
<br>
<div class = "progress" style = "height:30px">
<div class = "progress-bar bg-info progress-bar-striped" style = "width:50%; height:30px ">
<b> Jquery Skill </b>
</div>
</div>
</div>
</body>
</html>
```

Output

The following image shows a striped progress bar with different values.



Animated Progress Bar

The Animated progress bar shows stripe style with animated effect in the function. The animated progress bar requires a striped (progress-bar-striped) class; otherwise progress-bar-animated class does not work on the basic function bar. We can set the height and apply the label on the progress bar as per the requirement.

Syntax

The following syntax shows how to create animated progress bar using bootstrap 5 classes and css stayle.

```
<div class = "progress">
```

```
<div class = "progress-bar progress-bar-striped progree-bar-animated" style = "width:80%">
</div>
</div>
```

Example

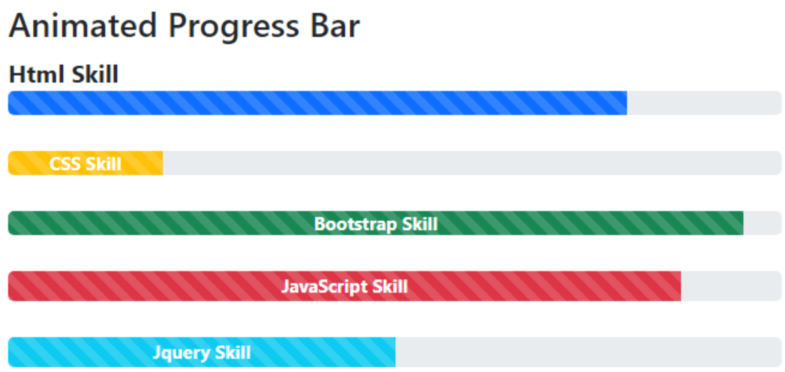
The example shows different background color of the progress bar as per requirement.

```
<!DOCTYPE html>
<html>
<head>
<title> Bootstrap 5 Progress bar Example </title>
<meta name = "viewport" content = "width=device-width, initial-scale=1">
<link href = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel = "stylesheet">
<script src = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js"></script>
</head>
<body>
<div class = "container">
<h4> Animated Progress Bar </h4>
<b> Html Skill </b>
<div class = "progress">
<div class = "progress-bar progress-bar-striped progress-bar-animated" style = "width:80%">
</div>
</div>
<br>
<div class = "progress">
<div class = "progress-bar bg-warning progress-bar-striped progress-bar-animated" style = "width:20%">
<b> CSS Skill </b>
</div>
</div>
<br>
<div class = "progress">
```

```
<div class = "progress-bar bg-success progress-bar-striped progress-bar-animated" style = "width:95%">
<b> Bootstrap Skill </b>
</div>
</div>
<br>
<div class = "progress" style = "height:20px">
<div class = "progress-bar bg-danger progress-bar-striped progress-bar-animated" style = "width:87%">
<b> JavaScript Skill </b>
</div>
</div>
<br>
<div class = "progress" style = "height:20px">
<div class = "progress-bar bg-info progress-bar-striped progress-bar-animated" style = "width:50%; height:20px ">
<b> Jquery Skill </b>
</div>
</div>
</div>
</body>
</html>
```

Output

The following image shows an animated progress bar with different values.



Multiple Progress Bar

The bootstrap 5 class uses default, striped, and animated progress bars to store multiple progress values. The multiple progress bar function contains several progress data in a single line. We can place the progress bar value or any information in text format in a single bar.

Syntax

The following syntax shows how to create multiple progress bars with label using bootstrap 5 classes. the "progress" class contains various "progress-bar"

```
<div class = "progress">
<div class = "progress-bar" style = "width:80%"> Label data1 </div>
<div class = "progress-bar" style = "width:10%"> Label data2 </div>
</div>
```

Example

The example shows the multiple progress bar. Here, we can write information, value and requires small data of the several progress bar in single line.

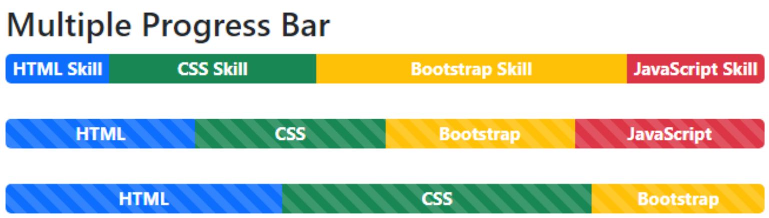
```
<!DOCTYPE html>
<html>
<head>
<title> Bootstrap 5 Progress bar Example </title>
<meta name = "viewport" content = "width=device-width, initial-scale=1">
<link href = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel = "stylesheet">
<script src = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js">
</script>
</head>
<body>
<div class = "container">
<h4> Multiple Progress Bar </h4>
<div class = "progress" style = "height: 20px">
<div class = "progress-bar" style = "width:15%">
```

```
 <b> HTML Skill </b>
</div>
<div class = "progress-bar bg-success" style = "width:30%">
<b> CSS Skill </b> </div>
<div class = "progress-bar bg-warning" style = "width: 45%">
<b> Bootstrap Skill </b> </div>
<div class = "progress-bar bg-danger" style = "width: 20%;">
<b> JavaScript Skill </b> </div>
</div>
<br>
<div class = "progress" style = "height: 20px">
<div class = "progress-bar progress-bar-striped" style = "width:25%">
 <b> HTML </b>
</div>
<div class = "progress-bar bg-success progress-bar-striped" style = "width:25%">
<b> CSS </b> </div>
<div class = "progress-bar bg-warning progress-bar-striped" style = "width: 25%">
<b> Bootstrap </b> </div>
<div class = "progress-bar bg-danger progress-bar-striped" style = "width: 25%;">
<b> JavaScript </b> </div>
</div>
<br>
<div class = "progress" style = "height: 20px">
<div class = "progress-bar progress-bar-striped progress-bar-animated" style = "width:40%">
 <b> HTML </b>
</div>
<div class = "progress-bar bg-success progress-bar-striped progress-bar-animated" style = "width:45%">
<b> CSS </b> </div>
<div class = "progress-bar bg-warning progress-bar-striped progress-bar-animated" style = "width: 25%">
<b> Bootstrap </b> </div>
</div>
</div>
```

```
</body>
</html>
```

Output

The following output image shows multiple progress bars and their required labels.



Multiple Progress Bar

Bootstrap 5 Spinners

Bootstrap 5 classes use the "spinners" to indicate the loading status in your applications. They are constructed just using HTML and CSS; therefore, you do not require JavaScript to make them. Using our incredible utility classes, you can simply modify their appearance, alignment, and size. To toggle their visibility, you will need to write some custom JavaScript.

Basic spinners

The basic spinner uses the "spinner-border" class on the web page. We can use text with the spinner. The spinner text can be visible or hidden as per user requirements.

Syntax

The following syntax shows how to create spinner using bootstrap 5 class.

```
<div class = "spinner-border" role = "status">
</div>
```

Example

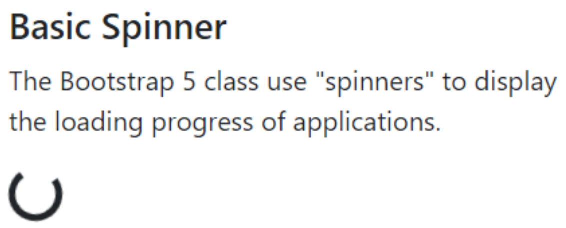
The example shows basic and default spinners according to the user requirement.

```
<!DOCTYPE html>
<html>
<head>
<title> Bootstrap 5 Spinners Example </title>
<meta name = "viewport" content = "width=device-width, initial-scale=1">
<link href = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel = "stylesheet">
<script src = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js"></script>
</head>
<body>
<div class = "container">
<h4> Basic Spinner </h4>
<p> The Bootstrap 5 class use "spinners" to display the loading progress of applications.
```

```
</p>
<div class = "spinner-border" role = "status">
</div>
</div>
</body>
</html>
```

Output

The following image shows the basic spinner on the web page.



Basic Spinner

The Bootstrap 5 class use "spinners" to display the loading progress of applications.

Bootstrap 5 Colored spinners

The colored spinner uses the "spinner-border" with the "text-*" classes on the web page. We can use bootstrap 5 contextual classes such as primary, secondary, and muted with the spinner. The colored spinner uses as per web application format and user requirements.

Syntax

The following syntax shows how to create colored spinner using bootstrap 5 class.

```
<div class = "spinner-border text-muted" role = "wait">
</div>
```

Example

The example shows basic and default and colored spinners according to the user requirement.

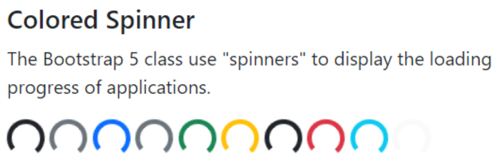
```
<!DOCTYPE html>
<html>
<head>
```

```
<title> Bootstrap 5 Spinners Example </title>
<meta name = "viewport" content = "width=device-width, initial-scale=1">
<link href = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel = "stylesheet">
<script src = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js"></script>
</head>
<body>
<div class = "container">
<h4> Colored Spinner </h4>
<p> The Bootstrap 5 class use "spinners" to display the loading progress of applications.
</p>
<div class = "spinner-border" role = "wait">
</div>
<div class = "spinner-border text-muted" role = "wait">
</div>
<div class = "spinner-border text-primary" role = "wait">
</div>
<div class = "spinner-border text-secondary" role = "wait">
</div>
<div class = "spinner-border text-success" role = "wait">
</div>
<div class = "spinner-border text-warning" role = "wait">
</div>
<div class = "spinner-border text-dark" role = "wait">
</div>
<div class = "spinner-border text-danger" role = "wait">
</div>
<div class = "spinner-border text-info" role = "wait">
</div>
<div class = "spinner-border text-light" role = "wait">
</div>
</div>
</body>
```

```
</html>
```

Output

The following image shows colored spinners on the web page.



Colored Spinner

The Bootstrap 5 class use "spinners" to display the loading progress of applications.

Bootstrap 5 growing spinners

The growing spinner uses the "spinner-grow" class on the web page. The growing spinner uses the contextual color in the <div> tag.

Syntax

The following syntax shows how to create growing spinner using bootstrap 5 class.

```
<div class = "spinner-grow text-muted" role = "wait">
</div>
```

Example

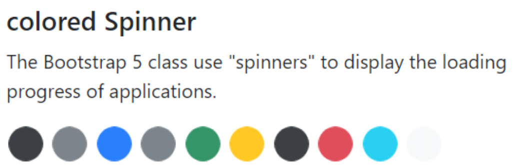
The example shows basic and default and colored spinners according to the user requirement.

```
<!DOCTYPE html>
<html>
<head>
<title> Bootstrap 5 Spinners Example </title>
<meta name = "viewport" content = "width=device-width, initial-scale=1">
<link href = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel = "stylesheet">
<script src = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js"></script>
</head>
<body>
```

```
<div class = "container">
<h4> colored Spinner </h4>
<p> The Bootstrap 5 class use "spinners" to display the loading progress of applications.
</p>
<div class = "spinner-grow" role = "wait">
</div>
<div class = "spinner-grow text-muted" role = "wait">
</div>
<div class = "spinner-grow text-primary" role = "wait">
</div>
<div class = "spinner-grow text-secondary" role = "wait">
</div>
<div class = "spinner-grow text-success" role = "wait">
</div>
<div class = "spinner-grow text-warning" role = "wait">
</div>
<div class = "spinner-grow text-dark" role = "wait">
</div>
<div class = "spinner-grow text-danger" role = "wait">
</div>
<div class = "spinner-grow text-info" role = "wait">
</div>
<div class = "spinner-grow text-light" role = "wait">
</div>
</div>
</body>
</html>
```

Output

The following image shows grow spinners on the web page.



colored Spinner

The Bootstrap 5 class use "spinners" to display the loading progress of applications.

Spinner sizes

The bootstrap 5 spinner function uses on a web page with the required size. We can set the spinner as per large, default, and small size. The spinner size uses with the spinner-border and spinner-grow classes.

Syntax

The following syntax shows how to create spinner with required size using bootstrap 5 class.

```
<div class = "spinner-border spinner-border-lg|sm" role = "wait">
</div>
```

The following syntax shows how to create growing spinner with required size.

```
<div class = "spinner-grow spinner-grow-lg|sm" role = "wait">
</div>
```

Example

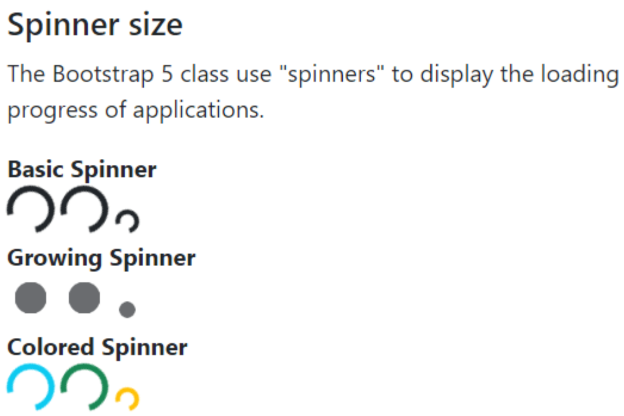
The example shows basic and default and colored spinners according to the user requirement.

```
<!DOCTYPE html>
<html>
<head>
<title> Bootstrap 5 Spinners Example </title>
<meta name = "viewport" content = "width=device-width, initial-scale=1">
<link href = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel = "stylesheet">
<script src = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js"></script>
</head>
<body>
<div class = "container">
```

```
<h4> Spinner size </h4>
<p> The Bootstrap 5 class use "spinners" to display the loading progress of applications.
</p>
<b> Basic Spinner </b><br>
<div class = "spinner-border spinner-border-lg" role = "wait">
</div>
<div class = "spinner-border" role = "wait">
</div>
<div class = "spinner-border spinner-border-sm" role = "wait">
</div><br>
<b> Growing Spinner </b><br>
<div class = "spinner-grow spinner-grow-lg" role = "wait">
</div>
<div class = "spinner-grow" role = "wait">
</div>
<div class = "spinner-grow spinner-grow-sm" role = "wait">
</div><br>
<b> Colored Spinner </b><br>
<div class = "spinner-border text-info spinner-border-lg" role = "wait">
</div>
<div class = "spinner-border text-success" role = "wait">
</div>
<div class = "spinner-border text-warning spinner-border-sm" role = "wait">
</div>
</div>
</body>
</html>
```

Output

The following image shows grow spinners on the web page.



Spinner size

The Bootstrap 5 class use "spinners" to display the loading progress of applications.

Basic Spinner

Growing Spinner

Colored Spinner

Button Spinner

Bootstrap 5 button spinner works with the button function and sizes. We can create a button and place the spinner tag in the button function. The spinner chooses size as per button size and web application requirements.

Syntax

The following syntax shows how to colored spinners in the button with required size using bootstrap 5 class.

```
<button class = "btn btn-success">
<span class = "spinner-border spinner-border-sm text-warning">
</span>
Waiting...
</button>
```

Example

The example shows colored spinners in the button according to the user requirement.

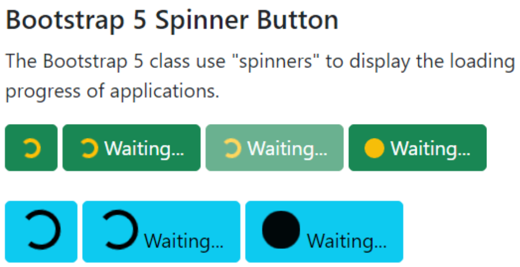
```
<!DOCTYPE html>
<html>
<head>
<title> Bootstrap 5 Spinners Example </title>
<meta name = "viewport" content = "width=device-width, initial-scale=1">
<link href = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel = "stylesheet">
```

```
<script src = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js"></script>
</head>
<body>
<div class = "container">
<h4> Bootstrap 5 Spinner Button </h4>
<p> The Bootstrap 5 class use "spinners" to display the loading progress of applications.
</p>
<button class = "btn btn-success">
<span class = "spinner-border spinner-border-sm text-warning">
</span>
</button>
<button class = "btn btn-success">
<span class = "spinner-border spinner-border-sm text-warning">
</span>
Waiting...
</button>
<button class = "btn btn-success" disabled>
<span class = "spinner-border spinner-border-sm text-warning">
</span>
Waiting...
</button>
<button class = "btn btn-success">
<span class = "spinner-grow spinner-grow-sm text-warning">
</span>
Waiting...
</button>
<br> <br>
<button class = "btn btn-info">
<span class = "spinner-border spinner-border-lg">
</span>
</button>
<button class = "btn btn-info">
<span class = "spinner-border spinner-border-lg">
```

```
</span>
Waiting...
</button>
<button class = "btn btn-info">
<span class = "spinner-grow spinner-grow-lg">
</span>
Waiting...
</button>
</div>
</body>
</html>
```

Output

The following image shows button spinners on the web page.



Bootstrap 5 Spinner Button

The Bootstrap 5 class use "spinners" to display the loading progress of applications.

Spinner margin

The spinner uses the "spinner-border" and "spinner-grow" classes with the required margin.

Syntax

The following syntax shows spinner with margin size using bootstrap 5 class.

```
<div class = "spinner-border m-1" role = "wait">
</div>
```

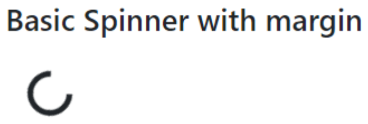
Example

The example shows basic spinners with margin according to the user requirement.

```
<!DOCTYPE html>
<html>
<head>
<title> Bootstrap 5 Spinners Example </title>
<meta name = "viewport" content = "width=device-width, initial-scale=1">
<link href = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel = "stylesheet">
<script src = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js"></script>
</head>
<body>
<div class = "container border">
<h4> Basic Spinner with margin </h4>
<div class = "spinner-border m-3" role = "status">
</div>
</div>
</body>
</html>
```

Output

The following image shows the basic spinner on the web page.



Basic Spinner with margin

Spinner position

The spinner uses the "spinner-border" and "spinner-grow" classes with the required position. We can place it at the start or end of the side of the web page.

Syntax

The following syntax shows spinner with required position using bootstrap 5 class.

```
<div class = "clearfix">
```

```
<div class = "spinner-border float-start" role = "wait">
</div>
</div>
```

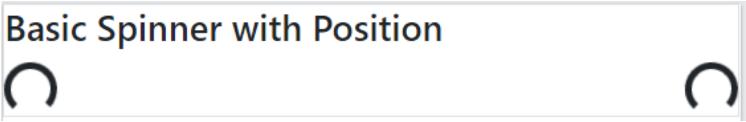
Example

The example shows basic and default spinners according to the user requirement.

```
<!DOCTYPE html>
<html>
<head>
<title> Bootstrap 5 Spinners Example </title>
<meta name = "viewport" content = "width=device-width, initial-scale=1">
<link href = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel = "stylesheet">
<script src = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js"></script>
</head>
<body>
<h4> Basic Spinner with Position </h4>
<div class = "clearfix border">
<h4> Basic Spinner with Position </h4>
<div class = "spinner-border float-end" role = "wait">
</div>
<div class = "spinner-border float-start" role = "wait">
</div>
</div>
</body>
</html>
```

Output

The following image shows the basic spinner on the web page.



Basic Spinner with Position

Bootstrap 5 Pagination

Bootstrap 5 List class and HTML elements are used for pagination so that screen readers may announce the number of available links. Use a bootstrap 5 classes in the <div> element to indicate to screen readers and other assistive technology that it is a pagination section. The pagination function shows the active page link before and next pages in the application. The pagination works for large web applications, eCommerce websites, and tutorial websites for linking multiple pages.

Basic pagination

Bootstrap5 pagination function uses the "pagination" class inside of the div tag. The bootstrap uses list tag such as a <ul> and <li> for linking pages. The <li> and <a> tag uses the "page-item" and "page-link" classes respectively. The <li> tag uses the anchor element to place require link in the list value.

Syntax

The following syntax shows how to create pagination using bootstrap 5 classes.

```
<ul class = "pagination">
<li class = "page-item">
<a class = "page-link" href = "#"> value </a>
</li>
</ul>
```

Examples

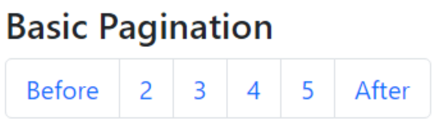
The following example shows basic pagination using bootstrap 5 classes. here, we can link before display page, after page, and available display pages link.

```
<!DOCTYPE html>
<html>
<head>
<title> Bootstrap 5 Pagination Example </title>
<meta name = "viewport" content = "width=device-width, initial-scale=1">
<link href = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel = "stylesheet">
```

```
<script src = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js"></script>
</head>
<body>
<div class = "container m-2">
<h4> Basic Pagination </h4>
<ul class = "pagination">
 <li class = "page-item"> <a class = "page-link" href = "#"> Before </a> </li>
 <li class = "page-item"> <a class = "page-link" href = "#"> 2 </a> </li>
 <li class = "page-item"> <a class = "page-link" href = "#"> 3 </a> </li>
 <li class = "page-item"> <a class = "page-link" href = "#"> 4 </a> </li>
 <li class = "page-item"> <a class = "page-link" href = "#"> 5 </a> </li>
 <li class = "page-item"> <a class = "page-link" href = "#"> After </a> </li>
</ul>
</div>
</body>
</html>
```

Output

The following image shows the basic pagination output for the web page.



Active Pagination

The pagination function uses the "pagination" class inside the div tag with the list function. The <li> tag of list function uses the "page-item" with the "active" class.

Syntax

The following syntax shows active pagination using bootstrap 5 classes in web page.

```
<ul class = "pagination">
<li class = "page-item active">
```

```
<a class = "page-link" href = "#"> value </a>
</li>
</ul>
```

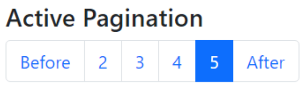
Example

The following example shows active pagination using bootstrap 5 classes. the "5" page value uses the "active" class. In the web application, active page changes as per display page.

```
<!DOCTYPE html>
<html>
<head>
<title> Bootstrap 5 Pagination Example </title>
<meta name = "viewport" content = "width=device-width, initial-scale=1">
<link href = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel = "stylesheet">
<script src = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js"></script>
</head>
<body>
<div class = "container m-2">
<h4> Active Pagination </h4>
<ul class = "pagination">
  <li class = "page-item"> <a class = "page-link" href = "#"> Before </a> </li>
  <li class = "page-item"> <a class = "page-link" href = "#"> 2 </a> </li>
  <li class = "page-item"> <a class = "page-link" href = "#"> 3 </a> </li>
  <li class = "page-item"> <a class = "page-link" href = "#"> 4 </a> </li>
  <li class = "page-item active"> <a class = "page-link" href = "#"> 5 </a> </li>
  <li class = "page-item"> <a class = "page-link" href = "#"> After </a> </li>
</ul>
</div>
</body>
</html>
```

Output

The following image shows the Active pagination output for the web page.


Active Pagination

Disabled Pagination

Bootstrap 5 pagination function uses the "pagination" class inside of the div tag with list function. The <li> tag of list function uses the "page-item" with the "disabled" class.

Syntax

The following syntax shows disabled pagination using bootstrap 5 classes in web page.

```
<ul class = "pagination">
<li class = "page-item disabled">
<a class = "page-link" href = "#"> value </a>
</li>
</ul>
```

Example

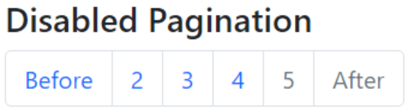
The following example shows disabled pagination using bootstrap 5 classes. we use disabled class in the last link of the pagination.

```
<!DOCTYPE html>
<html>
<head>
<title> Bootstrap 5 Pagination Example </title>
<meta name = "viewport" content = "width=device-width, initial-scale=1">
<link href = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel = "stylesheet">
<script src = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js"></script>
</head>
<body>
<div class = "container m-2">
<h4> Disabled Pagination </h4>
```

```
<ul class = "pagination">
  <li class = "page-item"> <a class = "page-link" href = "#"> Before </a> </li>
  <li class = "page-item"> <a class = "page-link" href = "#"> 2 </a> </li>
  <li class = "page-item"> <a class = "page-link" href = "#"> 3 </a> </li>
  <li class = "page-item"> <a class = "page-link" href = "#"> 4 </a> </li>
  <li class = "page-item disabled"> <a class = "page-link" href = "#"> 5 </a> </li>
  <li class = "page-item disabled"> <a class = "page-link" href = "#"> After </a> </li>
</ul>
</div>
</body>
</html>
```

Output

The following image shows disabled pagination output for the web page.



Pagination Size

The pagination function uses the "pagination", "page-item" and "page-link" classes inside of the div tag. Bootstrap 5 pagination displays with the required size, such as large and small. The size class comes with the "pagination" class.

Syntax

The following syntax shows pagination size using bootstrap 5 classes in web page.

```
<ul class = "pagination pagination-lg| pagination-sm">
<li class = "page-item">
<a class = "page-link" href = "#"> value </a>
</li>
</ul>
```

Example

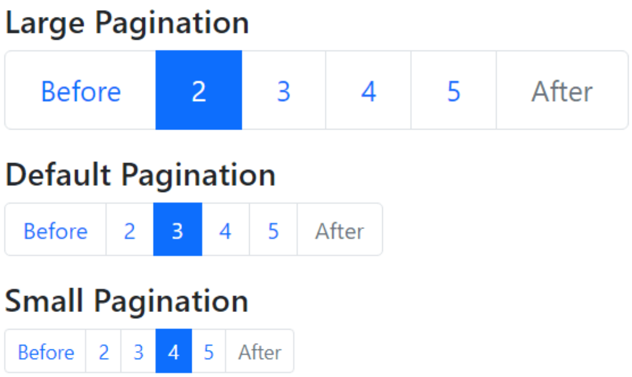
The following example shows disabled pagination using bootstrap 5 classes. we can use large and small size pagination.

```
<!DOCTYPE html>
<html>
<head>
<title> Bootstrap 5 Pagination Example </title>
<meta name = "viewport" content = "width=device-width, initial-scale=1">
<link href = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel = "stylesheet">
<script src = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js"></script>
</head>
<body>
<div class = "container m-2">
<h4> Large Pagination </h4>
<ul class = "pagination pagination-lg">
  <li class = "page-item"> <a class = "page-link" href = "#"> Before </a> </li>
  <li class = "page-item active"> <a class = "page-link" href = "#"> 2 </a> </li>
  <li class = "page-item"> <a class = "page-link" href = "#"> 3 </a> </li>
  <li class = "page-item"> <a class = "page-link" href = "#"> 4 </a> </li>
  <li class = "page-item"> <a class = "page-link" href = "#"> 5 </a> </li>
  <li class = "page-item disabled"> <a class = "page-link" href = "#"> After </a> </li>
</ul>
<h4> Default Pagination </h4>
<ul class = "pagination">
  <li class = "page-item"> <a class = "page-link" href = "#"> Before </a> </li>
  <li class = "page-item"> <a class = "page-link" href = "#"> 2 </a> </li>
  <li class = "page-item active"> <a class = "page-link" href = "#"> 3 </a> </li>
  <li class = "page-item"> <a class = "page-link" href = "#"> 4 </a> </li>
  <li class = "page-item"> <a class = "page-link" href = "#"> 5 </a> </li>
  <li class = "page-item disabled"> <a class = "page-link" href = "#"> After </a> </li>
</ul>
```

```
<h4> Small Pagination </h4>
<ul class = "pagination pagination-sm">
  <li class = "page-item"> <a class = "page-link" href = "#"> Before </a> </li>
  <li class = "page-item"> <a class = "page-link" href = "#"> 2 </a> </li>
  <li class = "page-item"> <a class = "page-link" href = "#"> 3 </a> </li>
  <li class = "page-item active"> <a class = "page-link" href = "#"> 4 </a> </li>
  <li class = "page-item"> <a class = "page-link" href = "#"> 5 </a> </li>
  <li class = "page-item disabled"> <a class = "page-link" href = "#"> After </a> </li>
</ul>
</div>
</body>
</html>
```

Output

The following image shows the different sizes of the pagination output for a web page.



Alignment of Pagination

The pagination function uses the "pagination" with alignment classes inside the <ul> tag. Bootstrap 5 pagination displays with the required alignment, such as a start, center, and end. The pagination does not require any class to display the start point of the page. The web page displays left-aligned pagination by default, but the center and right-aligned pagination require the bootstrap 5 alignment class.

Syntax

The following syntax shows center alignment pagination using bootstrap 5 classes.

```
<ul class = "pagination justify-content-center">
<li class = "page-item">
<a class = "page-link" href = "#"> value </a>
</li>
</ul>
```

The following syntax shows end alignment pagination using bootstrap 5 classes.

```
<ul class = "pagination justify-content-end">
Write list tag here!
</ul>
```

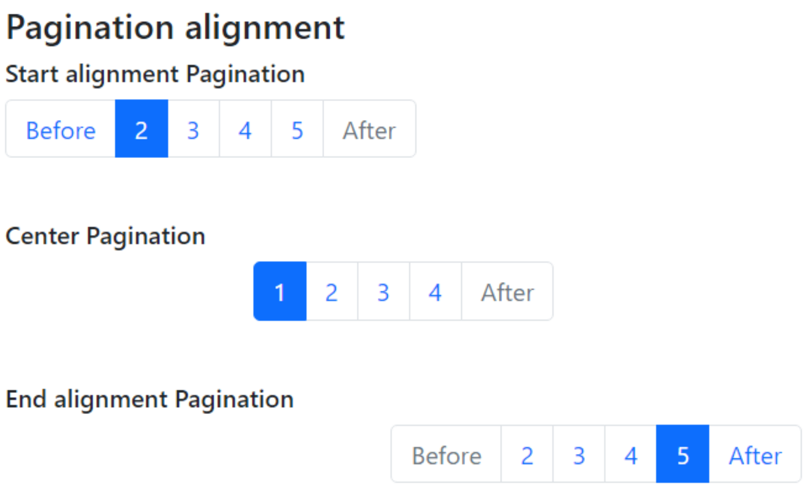
Example

The following example shows pagination alignment using bootstrap 5 classes. we can see pagination function in a starting point, center and end point of the page.

```
<!DOCTYPE html>
<html>
<head>
<title> Bootstrap 5 Pagination Example </title>
<meta name = "viewport" content = "width=device-width, initial-scale=1">
<link href = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel = "stylesheet">
<script src = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js"></script>
</head>
<body>
<div class = "container m-2">
<h4> Pagination alignment </h4>
<h6> Start alignment Pagination </h6>
<ul class = "pagination">
  <li class = "page-item"> <a class = "page-link" href = "#"> Before </a> </li>
  <li class = "page-item active"> <a class = "page-link" href = "#"> 2 </a> </li>
  <li class = "page-item"> <a class = "page-link" href = "#"> 3 </a> </li>
  <li class = "page-item"> <a class = "page-link" href = "#"> 4 </a> </li>
  <li class = "page-item"> <a class = "page-link" href = "#"> 5 </a> </li>
  <li class = "page-item disabled"> <a class = "page-link" href = "#"> After </a> </li>
</ul>
<br>
<h6> Center Pagination </h6>
<ul class = "pagination justify-content-center">
  <li class = "page-item active"> <a class = "page-link" href = "#"> 1 </a> </li>
  <li class = "page-item"> <a class = "page-link" href = "#"> 2 </a> </li>
  <li class = "page-item"> <a class = "page-link" href = "#"> 3 </a> </li>
  <li class = "page-item"> <a class = "page-link" href = "#"> 4 </a> </li>
  <li class = "page-item disabled"> <a class = "page-link" href = "#"> After </a> </li>
</ul>
<br>
<h6> End alignment Pagination </h6>
<ul class = "pagination justify-content-end">
  <li class = "page-item disabled "> <a class = "page-link" href = "#"> Before </a> </li>
  <li class = "page-item"> <a class = "page-link" href = "#"> 2 </a> </li>
  <li class = "page-item"> <a class = "page-link" href = "#"> 3 </a> </li>
  <li class = "page-item"> <a class = "page-link" href = "#"> 4 </a> </li>
  <li class = "page-item active"> <a class = "page-link" href = "#"> 5 </a> </li>
  <li class = "page-item"> <a class = "page-link" href = "#"> After </a> </li>
</ul>
</div>
</body>
</html>
```

Output

The following image shows a different alignment of the pagination output for the web page.



Pagination alignment

Start alignment Pagination

Center Pagination

End alignment Pagination

Breadcrumb pagination

Bootstrap5 pagination function uses the "breadcrumb" class inside of the <ul> tag. The bootstrap uses list tag such as a <li> and <a> for linking pages. The list tags such as a <li> use the "breadcrumb-item" class.

Syntax

The following syntax shows how to create breadcrumb pagination using bootstrap 5 classes.

```
<ul class = "breadcrumb">
<li class = "breadcrumb-item">
<a href = "#"> value </a>
</li>
</ul>
```

Examples

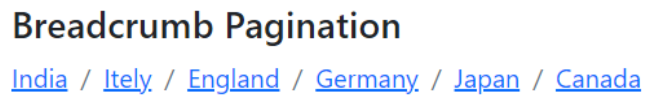
The following example shows breadcrumb pagination using bootstrap 5 classes. We can link different values in breadcrumb format. The pagination uses sliding line between two values to differentiate values.

```
<!DOCTYPE html>
<html>
<head>
<title> Bootstrap 5 Pagination Example </title>
```

```
<meta name = "viewport" content = "width=device-width, initial-scale=1">
<link href = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel = "stylesheet">
<script src = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js"></script>
</head>
<body>
<div class = "container m-2">
<h4> Breadcrumb Pagination </h4>
<ul class = "breadcrumb">
 <li class = "breadcrumb-item"> <a href = "#"> India </a> </li>
 <li class = "breadcrumb-item"> <a href = "#"> Itely </a> </li>
 <li class = "breadcrumb-item"> <a href = "#"> England </a> </li>
 <li class = "breadcrumb-item"> <a href = "#"> Germany </a> </li>
 <li class = "breadcrumb-item"> <a href = "#"> Japan </a> </li>
 <li class = "breadcrumb-item"> <a href = "#"> Canada </a> </li>
</ul>
</div>
</body>
</html>
```

Output

The following image shows the basic pagination output for the web page.



Breadcrumb Pagination

India / Itely / England / Germany / Japan / Canada

Bootstrap 5 Cards

In Bootstrap 5, a card is a bordered box with padding surrounding its text. There are options for headers, footers, content, and colors. The card uses to contain web application functions such as form, table and other user interactive features.

Basic card

The "card" class uses to construct a basic card, and the "card-body" class uses for card content. The card function creates a basic card, and the card body contains the required information for the application.

Syntax

The following syntax helps to create basic card on the web page.

```
<div class = "card">
<div class = "card-body">
Information!
</div>
</div>
```

Examples

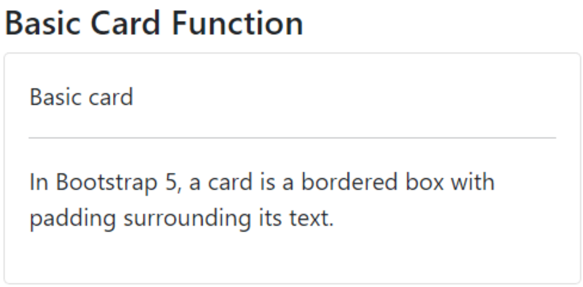
The following example shows card function using bootstrap 5 classes.

```
<!DOCTYPE html>
<html>
<head>
<title> Bootstrap 5 Card Example </title>
<meta name = "viewport" content = "width=device-width, initial-scale=1">
<link href = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel = "stylesheet">
<script src = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js">
</script>
</head>
<body>
```

```
<div class = "container m-2">
<h4> Basic Card Function </h4>
<div class = "card">
<div class = "card-body">
Basic card <hr>
<p> In Bootstrap 5, a card is a bordered box with padding surrounding its text. </p>
</div>
</div>
</div>
</body>
</html>
```

Output

The following image shows the basic bootstrap 5 card function.



Basic Card Function

Basic card

In Bootstrap 5, a card is a bordered box with padding surrounding its text.

Card with header and footer

The "card" class creates a basic card, and the "card-body" contains the required information for the application. The "card-header" contains and displays the heading of the card information. The "card-footer" displays footer information of the page or function.

Syntax

The following syntax helps to create bootstrap 5 card with header and footer on the web page.

```
<div class = "card">
<div class = "card-header"> </div>
```

```
<div class = "card-body"> Information. </div>
<div class = "card-footer"> </div>
</div>
```

Examples

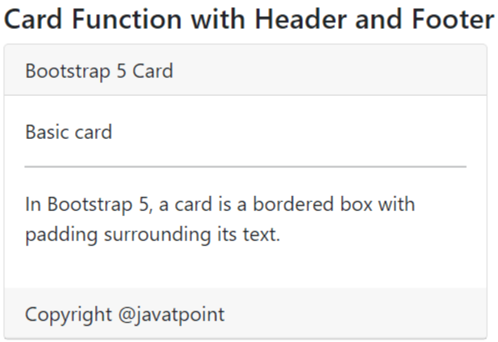
The following example shows card function using bootstrap 5 classes. we can place card header, body, and footer class in the "card" class.

```
<!DOCTYPE html>
<html>
<head>
<title> Bootstrap 5 Card Example </title>
<meta name = "viewport" content = "width=device-width, initial-scale=1">
<link href = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel = "stylesheet">
<script src = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js">
</script>
</head>
<body>
<div class = "container m-2">
<h4> Card Function with Header and Footer </h4>
<div class = "card">
<div class = "card-header">
Bootstrap 5 Card
</div>
<div class = "card-body">
Basic card <hr>
<p> In Bootstrap 5, a card is a bordered box with padding surrounding its text. </p>
</div>
<div class = "card-header">
 Copyright @javatpoint
</div>
</div>
</div>
```

```
</body>
</html>
```

Output

The following image shows the basic bootstrap 5 card function with header and footer.



Card Function with Header and Footer

Bootstrap 5 Card

Basic card

In Bootstrap 5, a card is a bordered box with padding surrounding its text.

Copyright @javatpoint

Contextual Card Function

The "card" class creates cards with different background colours using the "bg-primary" contextual class. The "card-body" and other card classes are contained in the card class. The contextual card displays different background colours and the card's meaning. The "bg-info" class shows informative information about the card.

Syntax

The following syntax helps to create bootstrap 5 contextual card on the web page.

```
<div class = "card bg-warning">
<div class = "card-body"> Information. </div>
</div>
```

Examples

The following example shows contextual card function using bootstrap 5 classes. we can use different background color for the card.

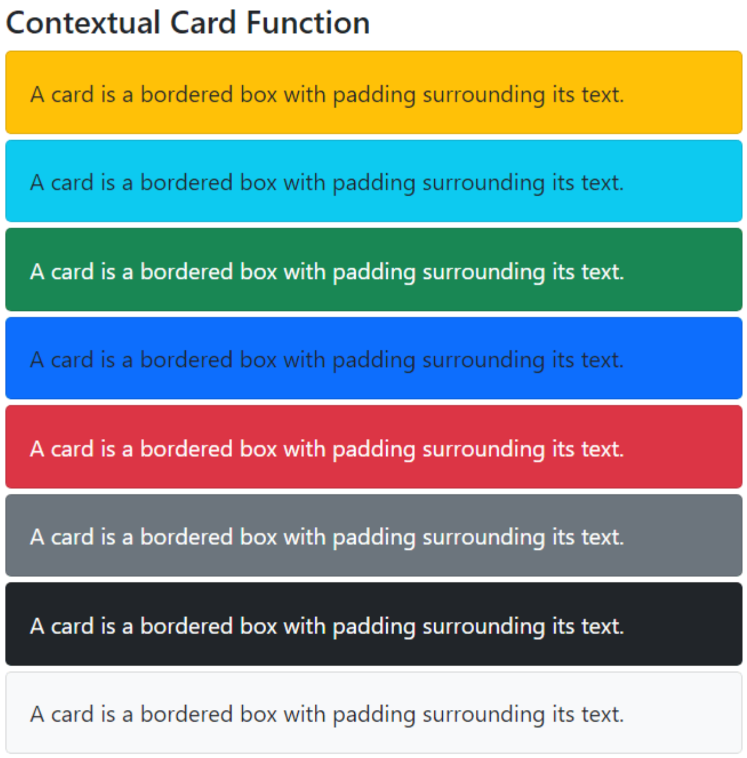
```
<!DOCTYPE html>
<html>
<head>
```

```
<title> Bootstrap 5 Card Example </title>
<meta name = "viewport" content = "width=device-width, initial-scale=1">
<link href = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel = "stylesheet">
<script src = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js">
</script>
</head>
<body>
<div class = "container m-2">
<h4> Contextual Card Function </h4>
<div class = "card bg-warning">
<div class = "card-body">
 A card is a bordered box with padding surrounding its text.
</div>
</div>
<div class = "card bg-info mb-1 mt-1">
<div class = "card-body">
A card is a bordered box with padding surrounding its text.
</div>
</div>
<div class = "card bg-success text-white">
<div class = "card-body">
A card is a bordered box with padding surrounding its text.
</div>
</div>
<div class = "card bg-primary mb-1 mt-1">
<div class = "card-body">
A card is a bordered box with padding surrounding its text.
</div>
</div>
<div class = "card bg-danger text-white mb-1">
<div class = "card-body">
A card is a bordered box with padding surrounding its text.
</div>
```

```
</div>
<div class = "card bg-secondary text-white mb-1">
<div class = "card-body">
A card is a bordered box with padding surrounding its text.
</div>
</div>
<div class = "card bg-dark text-white mb-1">
<div class = "card-body">
A card is a bordered box with padding surrounding its text.
</div>
</div>
<div class = "card bg-light">
<div class = "card-body">
A card is a bordered box with padding surrounding its text.
</div>
</div>
</div>
</body>
</html>
```

Output

The following image shows the bootstrap 5 contextual card function.



Contextual Card Function

A card is a bordered box with padding surrounding its text.

A card is a bordered box with padding surrounding its text.

A card is a bordered box with padding surrounding its text.

A card is a bordered box with padding surrounding its text.

A card is a bordered box with padding surrounding its text.

A card is a bordered box with padding surrounding its text.

A card is a bordered box with padding surrounding its text.

A card is a bordered box with padding surrounding its text.

Card with title, link and text

The card contains text, title, and link using the "card-text", "card-title", and "card-link" classes, respectively. This class uses in the "card-body" class to display functional data. The card title shows the heading of the card function, and the link shows another source on the page. The card text displays core and detailed information about the function.

Syntax

The following syntax helps to create bootstrap 5 card with title, link, and text on the web page.

```
<div class = "card">
<div class = "card-body">
<h5 class = "card-title"> Title </h5>
<p class = "card-text"> Text. </p>
<a href = "#" class = "card-link"> Link </a>
</div>
```

```
</div>
```

Examples

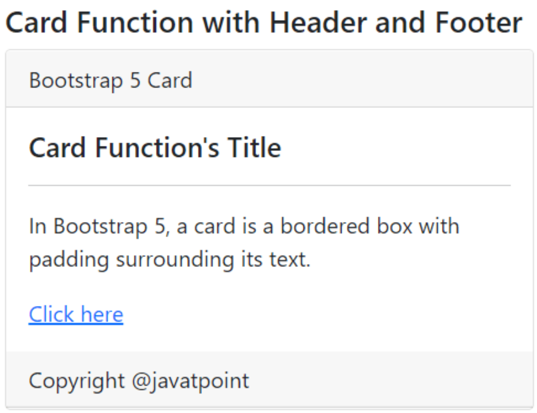
The following example shows card function with other functions using bootstrap 5 classes. we can place link and other card information in the "card-body" class.

```
<!DOCTYPE html>
<html>
<head>
<title> Bootstrap 5 Card Example </title>
<meta name = "viewport" content = "width=device-width, initial-scale=1">
<link href = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel = "stylesheet">
<script src = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js">
</script>
</head>
<body>
<div class = "container m-2">
<h4> Card Function with Header and Footer </h4>
<div class = "card">
<div class = "card-header">
Bootstrap 5 Card
</div>
<div class = "card-body">
<h5 class = "card-title"> Card Function's Title </h5> <hr>
<p class = "card-text"> In Bootstrap 5, a card is a bordered box with padding surrounding its text. </p>
<a href = "#" class = "card-link"> Click here </a>
</div>
<div class = "card-header">
 Copyright @javatpoint
</div>
</div>
</div>
```

```
</body>
</html>
```

Output

The following image shows the bootstrap 5 card function with title, link and information.



Card Function with Header and Footer

Bootstrap 5 Card

Card Function's Title

In Bootstrap 5, a card is a bordered box with padding surrounding its text.

Click here

Copyright @javatpoint

Card with image

The card contains an image on the top, and bottom of the function using the "card-img-top" and "card-img-bottom" classes, respectively. This class uses in the <img> tag in the <card> class. The <img> tag is required to set an image on the card with the required width in a CSS style.

Syntax

The following syntax helps to create bootstrap 5 card with image on the web page.

```
<div class = "card">
<img class = "card-img-
top" src = "C:\Users\hp\Pictures\tools.png" alt = "Card images" style = "width:50%">
<div class = "card-body"> Information! </div>
</div>
```

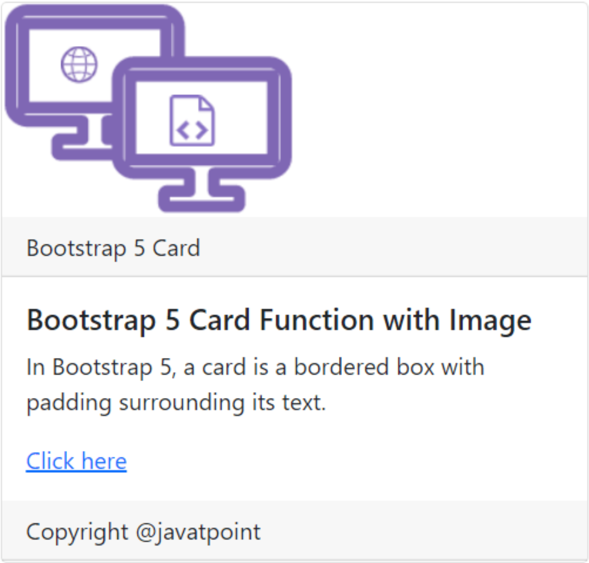
Examples

The following example shows card function with image using bootstrap 5 classes. we can place image top side of the card. The <img> tag uses in the "card" class.

```
<!DOCTYPE html>
<html>
<head>
<title> Bootstrap 5 Card Example </title>
<meta name = "viewport" content = "width=device-width, initial-scale=1">
<link href = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel = "stylesheet">
<script src = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js">
</script>
</head>
<body>
<div class = "container m-2">
<div class = "card" style="width:400px">
 <img class="card-img-top" src="C:\Users\hp\Pictures\Phototastic\tools.png" alt="Card images" style="width:50%">
<div class = "card-header">
Bootstrap 5 Card
</div>
<div class = "card-body">
<h5 class = "card-title"> Bootstrap 5 Card Function with Image </h5>
<p class = "card-text"> In Bootstrap 5, a card is a bordered box with padding surrounding its text. </p>
<a href = "#" class = "card-link"> Click here </a>
</div>
<div class = "card-header">
 Copyright @javatpoint
</div>
</div>
</div>
</body>
</html>
```

Output

The following image shows the bootstrap 5 card function with the image and other information.



Bootstrap 5 Card

Bootstrap 5 Card Function with Image

In Bootstrap 5, a card is a bordered box with padding surrounding its text.

Click here

Copyright @javatpoint

Card with image overlay

The card contains an image of the function's top, bottom, and centre alignment. The image sets as a background of the card using the "card-img-overlay" class. The <img> tag is required to set an image on the card. The width requires 100% in CSS style for the given image.

Syntax

The following syntax helps to create bootstrap 5 card with image on the web page.

```
<div class = "card">
<img class = "card-img-top" src = "C:\Users\hp\Pictures\tools.png" alt = "Card images" style = "width:100%">
<div class = "card-img-overlay">
<div class = "card-body"> Information! </div>
</div>
</div>
```

Examples

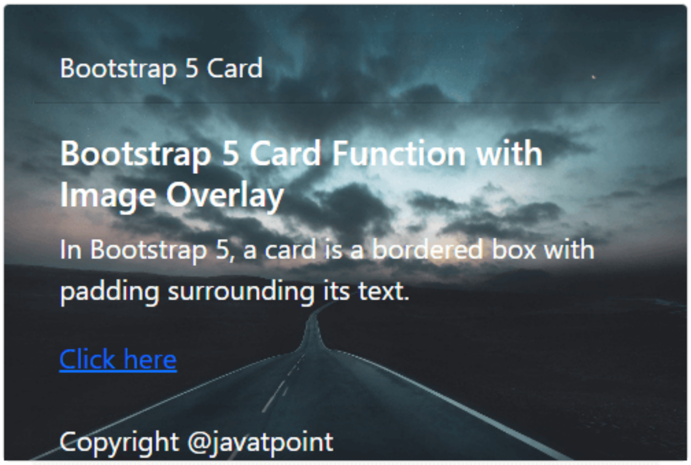
The following example shows card function with image using bootstrap 5 classes. we can place image top side of the card. The <img> tag uses in the "card" class.

```
<!DOCTYPE html>
<html>
<head>
<title> Bootstrap 5 Card Example </title>
<meta name = "viewport" content = "width=device-width, initial-scale=1">
<link href = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel = "stylesheet">
<script src = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js">
</script>
</head>
<body>
<div class = "container m-2">
<div class = "card text-white" style = "width:400px">
 <img class = "card-img-top" src = "C:\Users\hp\Pictures\Phototastic\timespace.jpg" alt = "Card images" style = "width:100%" >
 <div class = "card-img-overlay">
<div class = "card-header">
Bootstrap 5 Card
</div>
<div class = "card-body">
<h5 class = "card-title"> Bootstrap 5 Card Function with Image Overlay </h5>
<p class = "card-text"> In Bootstrap 5, a card is a bordered box with padding surrounding its text. </p>
<a href = "#" class = "card-link"> Click here </a>
</div>
<div class = "card-header">
 Copyright @javatpoint
</div>
</div>
```

```
</div>
</div>
</body>
</html>
```

Output

The following image shows the bootstrap 5 card function with the background image and other information.



Bootstrap 5 Card

Bootstrap 5 Card Function with Image Overlay

In Bootstrap 5, a card is a bordered box with padding surrounding its text.

Click here

Copyright @javatpoint

Bootstrap 5 Navs

The base ".nav" class shares the overall markup and styles of the Bootstrap navigation with a link. It navigates all important and required pages in the bar's small size. We get the overall structure of the web application and its pages.

Bootstrap 5 nav function is used to design the navigation component, which is a solid basis for creating various navigation component kinds. It adds style overrides, basic disabled styling, and link padding for wider hit areas.

There is no ".active" state in the base ".nav" function component. We can use the "disabled" function to inactive or disable lint content by default.

Basic Navs

The default setting for a web page's basic navigation function is horizontal alignment. The ".nav" class is inserted into the "ul" element, and then each "li" element's ".nav-item" and ".nav-link" classes are added.

Syntax

The following syntax shows basic nav function.

```
<ul class = "nav">
<li class = "nav-item">
<a class = "nav-link" href = "#"> link value </a>
</li>
</ul>
```

Example

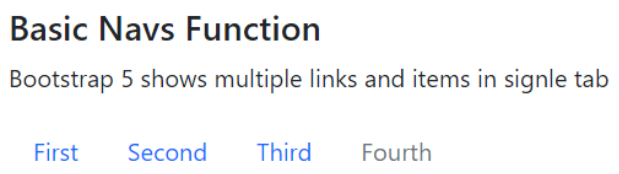
The following example shows basic navs using bootstrap 5 classes.

```
<!DOCTYPE html>
<html lang="en">
<head>
<title> Bootstrap Navs Example </title>
<meta charset = "utf-8">
<meta name = "viewport" content = "width=device-width, initial-scale=1">
```

```
<link href = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel = "stylesheet">
<script src = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js"></script>
</head>
<body>
<div class = "container mt-2">
<h4> Basic Navs Function </h4>
<p> Bootstrap 5 shows multiple links and items in signle tab </p>
<ul class = "nav nav-pills">
<li class = "nav-item">
<a class = "nav-link" href = "#"> First </a>
</li>
<li class = "nav-item">
<a class = "nav-link" href = "#"> Second </a>
</li>
<li class = "nav-item">
<a class = "nav-link" href = "#"> Third </a>
</li>
<li class = "nav-item">
<a class = "nav-link disabled" href = "#"> Fourth </a>
</li>
</ul>
</div>
</body>
</html>
```

Output

The basic navs shows in below image.



Basic Navs Function

Bootstrap 5 shows multiple links and items in signle tab

First Second Third Fourth

Bootstrap 5 Nav alignments

The nav function is a basic navigation feature web page. We can align the nav function at the pages' start, center, and end as per requirement. The nav function uses the "justify-content-center" and "justify-content-end" classes for the center and end position, respectively.

Syntax

The following syntax shows basic nav function for center alignment.

```
<ul class = "nav justify-content-center">
<li class = "nav-item">
<a class = "nav-link" href = "#"> link value </a>
</li>
</ul>
```

Example

The following image shows start, middle and end alignment of the nav function.

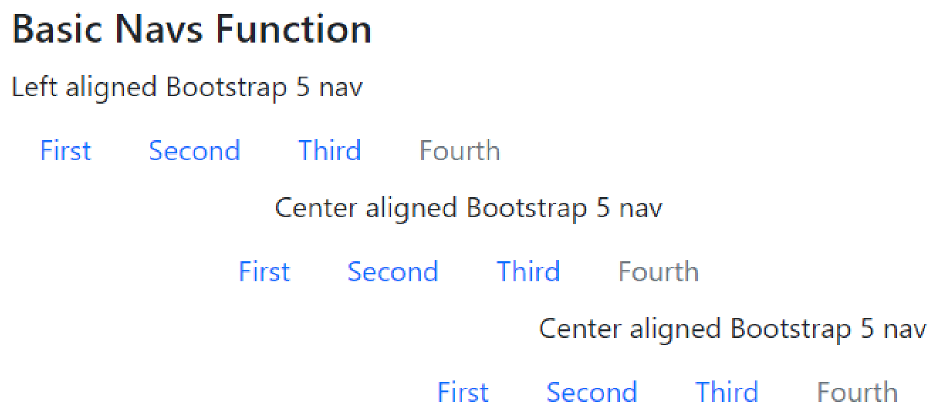
```
<!DOCTYPE html>
<html lang="en">
<head>
<title>Bootstrap Example</title>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1">
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel="stylesheet">
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js"></script>
</head>
<body>
<div class = "container mt-2">
<h4> Basic Navs Function </h4>
<p class = "mb-1"> Left aligned Bootstrap 5 nav </p>
<ul class = "nav">
```

```
<li class = "nav-item">
<a class = "nav-link" href = "#"> First </a>
</li>
<li class = "nav-item">
<a class = "nav-link" href = "#"> Second </a>
</li>
<li class = "nav-item">
<a class = "nav-link" href = "#"> Third </a>
</li>
<li class = "nav-item">
<a class = "nav-link disabled" href = "#"> Fourth </a>
</li>
</ul>
<p class = "text-center mb-1"> Center aligned Bootstrap 5 nav </p>
<ul class = "nav justify-content-center">
<li class = "nav-item">
<a class = "nav-link" href = "#"> First </a>
</li>
<li class = "nav-item">
<a class = "nav-link" href = "#"> Second </a>
</li>
<li class = "nav-item">
<a class = "nav-link" href = "#"> Third </a>
</li>
<li class = "nav-item">
<a class = "nav-link disabled" href = "#"> Fourth </a>
</li>
</ul>
<p class = "text-end mb-1"> Center aligned Bootstrap 5 nav </p>
<ul class = "nav justify-content-end">
<li class = "nav-item">
<a class = "nav-link" href = "#"> First </a>
</li>
<li class = "nav-item">
```

```
<a class = "nav-link" href = "#"> Second </a>
</li>
<li class = "nav-item">
<a class = "nav-link" href = "#"> Third </a>
</li>
<li class = "nav-item">
<a class = "nav-link disabled" href = "#"> Fourth </a>
</li>
</ul>
</div>
</body>
</html>
```

Output

The alignment of the nav shows in below image.



Basic Navs Function

Left aligned Bootstrap 5 nav

First Second Third Fourth

Center aligned Bootstrap 5 nav

First Second Third Fourth

Center aligned Bootstrap 5 nav

First Second Third Fourth

Vertical Navs

The "flext-column" class works for the vertical alignment of the nav function on the html element like <div> tag.

Syntax

The following syntax shows basic nav function for vertical alignment.

```
<ul class = "nav flex-column">
<li class = "nav-item">
```

```
<a class = "nav-link" href = "#"> link value </a>
</li>
</ul>
```

Example

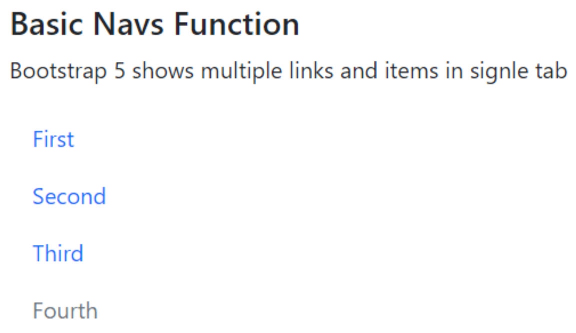
The following example shows vertical navs using bootstrap 5 classes.

```
<!DOCTYPE html>
<html lang="en">
<head>
<title> Bootstrap Navs Example </title>
<meta charset = "utf-8">
<meta name = "viewport" content = "width=device-width, initial-scale=1">
<link href = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel = "stylesheet">
<script src = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js"></script>
</head>
<body>
<div class = "container mt-2">
<h4> Basic Navs Function </h4>
<p> Bootstrap 5 shows multiple links and items in signle tab </p>
<ul class = "nav flex-column">
<li class = "nav-item">
<a class = "nav-link" href = "#"> First </a>
</li>
<li class = "nav-item">
<a class = "nav-link" href = "#"> Second </a>
</li>
<li class = "nav-item">
<a class = "nav-link" href = "#"> Third </a>
</li>
<li class = "nav-item">
<a class = "nav-link disabled" href = "#"> Fourth </a>
```

```
</li>
</ul>
</div>
</body>
</html>
```

Output

The vertical nav function shows in below image.



Basic Navs Function

Bootstrap 5 shows multiple links and items in signle tab

First

Second

Third

Fourth

Bootstrap 5 Nav tabs

The "flext-tab" class works for the tabular format of the nav function on the html element like <div> tag.

Syntax

The following syntax shows basic nav function with tabs.

```
<ul class = "nav nav-tab">
<li class = "nav-item">
<a class = "nav-link" href = "#"> link value </a>
</li>
</ul>
```

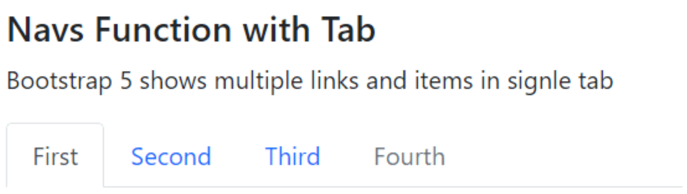
Example

The following example shows vertical navs using bootstrap 5 classes.

```
<!DOCTYPE html>
<html lang="en">
<head>
<title> Bootstrap Navs Example </title>
<meta charset = "utf-8">
<meta name = "viewport" content = "width=device-width, initial-scale=1">
<link href = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel = "stylesheet">
<script src = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js"></script>
</head>
<body>
<div class = "container mt-2">
<h4> Navs Function with Tab </h4>
<p> Bootstrap 5 shows multiple links and items in signle tab </p>
<ul class = "nav nav-tabs">
<li class = "nav-item">
<a class = "nav-link active" href = "#"> First </a>
</li>
<li class = "nav-item">
<a class = "nav-link" href = "#"> Second </a>
</li>
<li class = "nav-item">
<a class = "nav-link" href = "#"> Third </a>
</li>
<li class = "nav-item">
<a class = "nav-link disabled" href = "#"> Fourth </a>
</li>
</ul>
</div>
</body>
</html>
```

Output

The nav tab function shows in below image.



Navs Function with Tab

Bootstrap 5 shows multiple links and items in signle tab

Bootstrap 5 Nav pills

The "flext-pills" class works for the pills format of the nav function on the html element like <div> tag.

Syntax

The following syntax shows basic nav function with pills.

```
<ul class = "nav nav-pills">
<li class = "nav-item">
<a class = "nav-link" href = "#"> link value </a>
</li>
</ul>
```

Example

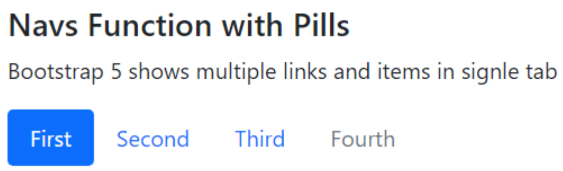
The following example shows navs function with pills using bootstrap 5 classes.

```
<!DOCTYPE html>
<html lang="en">
<head>
<title> Bootstrap Navs Example </title>
<meta charset = "utf-8">
<meta name = "viewport" content = "width=device-width, initial-scale=1">
<link href = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel = "stylesheet">
<script src = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js"></script>
```

```
</head>
<body>
<div class = "container mt-2">
<h4> Navs Function with Pills </h4>
<p> Bootstrap 5 shows multiple links and items in signle tab </p>
<ul class = "nav nav-pills">
<li class = "nav-item">
<a class = "nav-link active" href = "#"> First </a>
</li>
<li class = "nav-item">
<a class = "nav-link" href = "#"> Second </a>
</li>
<li class = "nav-item">
<a class = "nav-link" href = "#"> Third </a>
</li>
<li class = "nav-item">
<a class = "nav-link disabled" href = "#"> Fourth </a>
</li>
</ul>
</div>
</body>
</html>
```

Output

The nav pill function shows in below image.



Navs Function with Pills

Bootstrap 5 shows multiple links and items in signle tab

Navs function with fills Tabs/pills

The " nav-fills " class works to acquire appropriate space for the horizontal nav items. This class works on the "tab" and "pills" of the nav link function. The nav function shows the link content with the appropriate width per the web page.

Syntax

The following syntax shows justify nav function with pills.

```
<ul class = "nav nav-pills nav-fills">
<li class = "nav-item">
<a class = "nav-link" href = "#"> link value </a>
</li>
</ul>
```

The following syntax shows justify nav function with tab.

```
<ul class = "nav nav-tab nav-fills">
<li class = "nav-item">
<a class = "nav-link" href = "#"> link value </a>
</li>
</ul>
```

Example

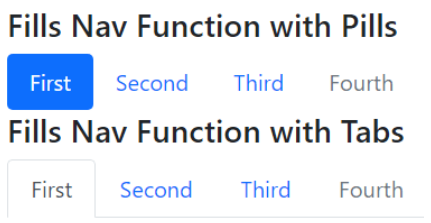
The following example shows navs function with fills using bootstrap 5 classes.

```
<!DOCTYPE html>
<html lang="en">
<head>
<title> Bootstrap Navs Example </title>
<meta charset = "utf-8">
<meta name = "viewport" content = "width=device-width, initial-scale=1">
<link href = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel = "stylesheet">
<script src = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js"></script>
</head>
<body>
<div class = "container mt-2">
<h4> Fills Nav Function with Pills </h4>
<ul class = "nav nav-pills nav-fills ">
```

```
<li class = "nav-item">
<a class = "nav-link active" href = "#"> First </a>
</li>
<li class = "nav-item">
<a class = "nav-link" href = "#"> Second </a>
</li>
<li class = "nav-item">
<a class = "nav-link" href = "#"> Third </a>
</li>
<li class = "nav-item">
<a class = "nav-link disabled" href = "#"> Fourth </a>
</li>
</ul>
<h4> Fills Nav Function with Tabs </h4>
<ul class = "nav nav-tabs nav-fills ">
<li class = "nav-item">
<a class = "nav-link active" href = "#"> First </a>
</li>
<li class = "nav-item">
<a class = "nav-link" href = "#"> Second </a>
</li>
<li class = "nav-item">
<a class = "nav-link" href = "#"> Third </a>
</li>
<li class = "nav-item">
<a class = "nav-link disabled" href = "#"> Fourth </a>
</li>
</ul>
</div>
</body>
</html>
```

Output

The nav function with fills class shows the below image.



Fills Nav Function with Pills

First Second Third Fourth

Fills Nav Function with Tabs

First Second Third Fourth

Navs function with justifying Tabs/pills

The " nav-justified" class works to acquire space for the horizontal nav function. This class works on the "tab" and "pills" of the nav function. The justified nav function shows the link content with equal width.

Syntax

The following syntax shows justify nav function with pills.

```
<ul class = "nav nav-pills nav-justified">
<li class = "nav-item">
<a class = "nav-link" href = "#"> link value </a>
</li>
</ul>
```

The following syntax shows justify nav function with tab.

```
<ul class = "nav nav-tab nav-justified">
<li class = "nav-item">
<a class = "nav-link" href = "#"> link value </a>
</li>
</ul>
```

Example

The following example shows navs function with justified width using bootstrap 5 classes.

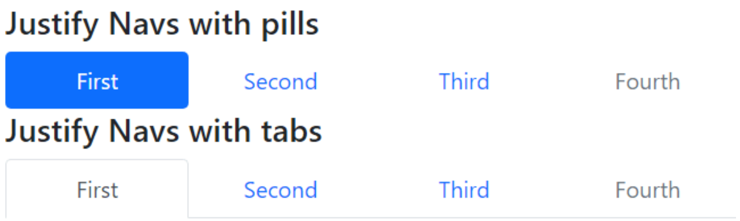
```
<!DOCTYPE html>
<html lang="en">
<head>
<title> Bootstrap Navs Example </title>
```

```
<meta charset = "utf-8">
<meta name = "viewport" content = "width=device-width, initial-scale=1">
<link href = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel = "stylesheet">
<script src = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js"></script>
</head>
<body>
<div class = "container mt-2">
<h4> Justify Navs with pills </h4>
<ul class = "nav nav-pills nav-justified ">
<li class = "nav-item">
<a class = "nav-link active" href = "#"> First </a>
</li>
<li class = "nav-item">
<a class = "nav-link" href = "#"> Second </a>
</li>
<li class = "nav-item">
<a class = "nav-link" href = "#"> Third </a>
</li>
<li class = "nav-item">
<a class = "nav-link disabled" href = "#"> Fourth </a>
</li>
</ul>
<h4> Justify Navs with tabs </h4>
<ul class = "nav nav-tabs nav-justified ">
<li class = "nav-item">
<a class = "nav-link active" href = "#"> First </a>
</li>
<li class = "nav-item">
<a class = "nav-link" href = "#"> Second </a>
</li>
<li class = "nav-item">
<a class = "nav-link" href = "#"> Third </a>
```

```
</li>
<li class = "nav-item">
<a class = "nav-link disabled" href = "#"> Fourth </a>
</li>
</ul>
</div>
</body>
</html>
```

Output

The justified nav function shows in below image.



Justify Navs with pills

Justify Navs with tabs

Navs function with dropdown

The dropdown function uses inside of the bootstrap 5 nav function. We can click on the required link to show dropdown functions. The "dropdown" class uses in the <li> tag with the "nav-item" class. The dropdown toggle uses with the "nav-link" class inside of the <a> tag. we can place the dropdown inside of the nav tag.

Example

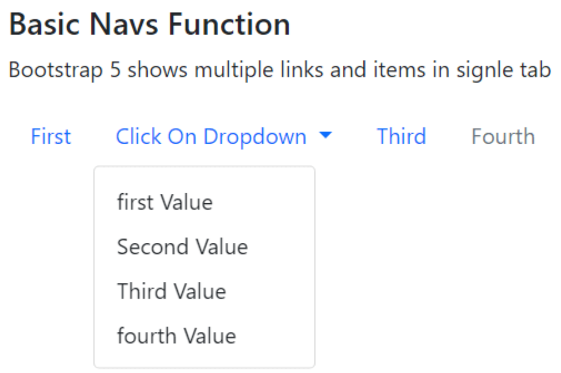
The following example shows basic navs with dropdown using bootstrap 5 classes.

```
<!DOCTYPE html>
<html lang="en">
<head>
<title> Bootstrap Navs Example </title>
<meta charset = "utf-8">
<meta name = "viewport" content = "width=device-width, initial-scale=1">
```

```
<link href = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel = "stylesheet">
<script src = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js"></script>
</head>
<body>
<div class = "container mt-2">
<h4> Basic Navs Function </h4>
<p> Bootstrap 5 shows multiple links and items in signle tab </p>
<ul class = "nav">
<li class = "nav-item">
<a class = "nav-link" href = "#"> First </a>
</li>
<li class = "nav-item dropdown">
<a class = "nav-link dropdown-toggle" data-bs-toggle = "dropdown" value=""> Click On Dropdown </a>
<ul class = "dropdown-menu">
<li> <a class = "dropdown-item" href = "#"> first Value </a> </li>
<li> <a class = "dropdown-item" href = "#"> Second Value </a> </li>
<li> <a class = "dropdown-item" href = "#"> Third Value </a> </li>
<li> <a class = "dropdown-item" href = "#"> fourth Value </a> </li>
</ul>
</li>
<li class = "nav-item">
<a class = "nav-link" href = "#"> Third </a>
</li>
<li class = "nav-item">
<a class = "nav-link disabled" href = "#"> Fourth </a>
</li>
</ul>
</div>
</body>
</html>
```

Output

The nav tab function with dropdown shows the below image.



Basic Navs Function

Bootstrap 5 shows multiple links and items in signle tab

Navs function with Toggleable

The toggleable function uses to display a particular tab of the pages after clicking on the nav link. The data-bs-toggle = "tab" function requires inside of the <a> tag with the ".nav-link" class. The " tab-content" class shows which data shows for nav function links. The id and href are required for the respective div container and nav item.

syntax

The following example shows navs with toggleable function using bootstrap 5 classes.

```
<ul class = "nav">
<li class = "nav-item">
<a class = "nav-link" data-bs-toggle = "tab" href = "#f1"> First </a>
</li>
</ul>
</div>
<div class="tab-content">
<div id="f1" class="container tab-pane"> <br>
//informatiom
</div>
</div>
```

Example

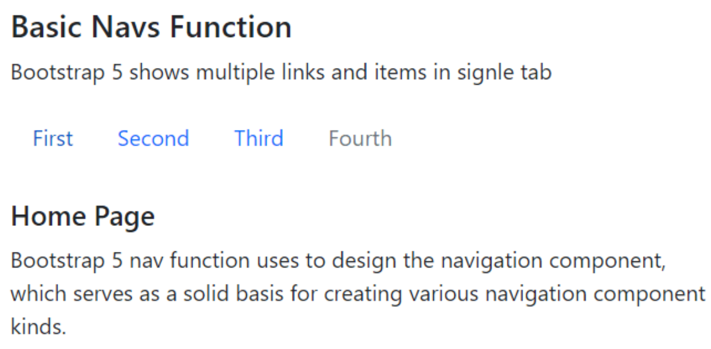
The following example shows navs with toggleable function using bootstrap 5 classes.

```
<!DOCTYPE html>
<html lang="en">
<head>
<title> Bootstrap Navs Example </title>
<meta charset = "utf-8">
<meta name = "viewport" content = "width=device-width, initial-scale=1">
<link href = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel = "stylesheet">
<script src = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js"></script>
</head>
<body>
<div class = "container mt-2">
<h4> Basic Navs Function </h4>
<p> Bootstrap 5 shows multiple links and items in signle tab </p>
<ul class = "nav">
<li class = "nav-item">
<a class = "nav-link active" data-bs-toggle = "tab" href = "#f1"> First </a>
</li>
<li class = "nav-item">
<a class = "nav-link" data-bs-toggle = "tab" href = "#f2"> Second </a>
</li>
<li class = "nav-item">
<a class = "nav-link" data-bs-toggle = "tab" href = "#f3"> Third </a>
</li>
<li class = "nav-item">
<a class = "nav-link disabled" href = "#f4"> Fourth </a>
</li>
</ul>
</div>
<div class="tab-content">
```

```
<div id="f1" class="container tab-pane active"> <br>
<h5> Home Page </h5>
<p> Bootstrap 5 nav function uses to design the navigation component, which serves as a solid basis for creating various navigation component kinds. <p>
</div>
<div id="f2" class="container tab-pane"> <br>
<h5> Second Page </h5>
<p> There is no ".active" state in the base ".nav" function component. We can use the "disabled" function to inactive or disabling lint content by default. <p>
</div>
<div id="f3" class="container tab-pane"> <br>
<h5> Third Page </h5>
<p> Bootstrap 5 nav function uses to design the navigation component, which serves as a solid basis for creating various navigation component kinds. <p>
</div>
</div>
</body>
</html>
```

Output

The nav tab function with the toggleable function shows the below image.



Basic Navs Function

Bootstrap 5 shows multiple links and items in signle tab

Home Page

Bootstrap 5 nav function uses to design the navigation component, which serves as a solid basis for creating various navigation component kinds.

Bootstrap 5 carousel

Bootstrap 5 carousel is a slideshow that cycles through various content using classes. It works with a collection of pictures, text, or unique markup elements. Additionally, it supports previous and next indicators and buttons.

The following table shows carousel classes with a detailed description.

Class	Description
"carousel"	It creates a carousel function on the web page.
"carousel-indicators"	This class provides more carousel indicators. It indicates the number of slides in the carousel and which slide the user is currently showing or working on. The tiny dots on each slide's bottom represent the slide's indicator.
"carousel-inner"	It inserts slides in the carousel class.
"carousel-item"	It specifies the data or information on the carousel slide.
"carousel-control-prev"	It provides a previous or left button to the carousel. It allows the user to go to the previous slide of the carousel.
"carousel-control-next"	It provides a next or right button to the carousel. It allows the user to go next slide of the carousel.
"carousel-control-next-icon"	It comes with the carousel-control-next to create a "next or right side" button.

"carousel-control-prev-icon"	It comes with the "carousel-control-prev" class to create a "previous or left side" button.
"slide"	It provides a CSS style transition and animation effect to the carousel function. It slides from one item to the next. The carousel function removes the class if you do not want this effect.

Basic carousel

The basic carousel uses different images in the slides with the next and previous buttons. This function uses on top of the web application. It attracts users' interaction and shows essential information.

Syntax

The following syntax shows basic carousel function using bootstrap 5 classes.

```
<div id = "carouseldemo" class = "carousel slide" data-bs-ride = "carousel">
<div class = "carousel-inner">
<div class = "carousel-item">
<img src = "image.jpg" alt = "image2" class = "d-block" style = "width:100%;">
</div>
</div>
</div>
```

The following syntax shows basic previous and next button for carousel function.

```
<div id = "carouseldemo" class = "carousel slide" data-bs-ride = "carousel">
<button class = "carousel-control-prev" type = "button" data-bs-target = "#carouseldemo" data-bs-slide = "prev"> <span class = "carousel-control-prev-icon"></span>
</button>
<button class = "carousel-control-next" type = "button" data-bs-target = "#carouseldemo" data-bs-slide = "next"> <span class = "carousel-control-next-icon"></span>
</button>
</div>
```

Example

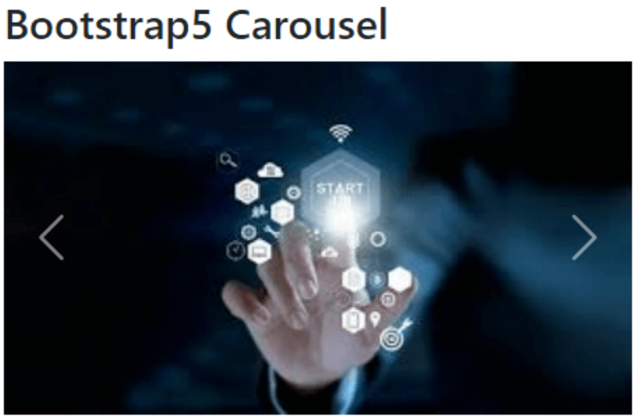
The following example uses four slides with images. It shows next and previous button.

```
<!DOCTYPE html>
<html lang = "en">
<head>
<title> Bootstrap 5 Carousel Example </title>
<meta charset = "utf-8">
<meta name = "viewport" content = "width = device-width, initial-scale = 1">
<link href = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel = "stylesheet">
<script src = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js">
</script>
</head>
<body>
</head>
<body>
<div class = "container" id = "alertcont" style = "width:50%;height:50%;">
<h2> Bootstrap5 Carousel </h2>
<div id = "carouseldemo" class = "carousel slide" data-bs-ride = "carousel">
<!-- The slideshow with images -->
<div class = "carousel-inner">
<div class = "carousel-item active">
<img src = "G:\writing stuff\content-writing\JavaTpoint\2022\may\starts.jpg" alt = "image1" class = "d-block" style = "width:100%;">
</div>
<div class = "carousel-item">
<img src = "G:\writing stuff\content-writing\JavaTpoint\2022\may\guudu.jpg" alt = "image2" class = "d-block" style = "width:100%;">
</div>
<div class = "carousel-item">
<img src = "G:\writing stuff\content-writing\JavaTpoint\2022\may\starts.jpg" alt = "image3" class = "d-block" style = "width:100%;">
</div>
```

```
<div class = "carousel-item">
<img src = "G:\writing stuff\content-writing\JavaTpoint\2022\may\guudu.jpg" alt = "image4" class = "d-block" style = "width:100%;">
</div>
</div>
<!-- previous and next icons -->
<button class = "carousel-control-prev" type = "button" data-bs-target = "#carouseldemo" data-bs-slide = "prev">
<span class="carousel-control-prev-icon"></span>
</button>
<button class = "carousel-control-next" type = "button" data-bs-target = "#carouseldemo" data-bs-slide = "next">
<span class = "carousel-control-next-icon"></span>
</button>
</div>
</div>
</body>
</html>
```

Output

The following output shows the carousel function with images.



Bootstrap5 Carousel

Basic carousel with indicator

The carousel function includes an indicator to show many slides are available in the function. The indicator shows the current working slide, previous slides and next slides.

Syntax

The following syntax shows carousel function with indicator using bootstrap 5 classes.

```
<div id = "carouseldemo" class = "carousel slide" data-bs-ride = "carousel">
<div class="carousel-indicators">
<button type = "button" data-bs-target = "#carouseldemo" data-bs-slide-to = "0"></button>
</div>
//other carousel function code
</div>
```

Example

The following example uses four slides with indicator. The first slide is active with active indicator. The other slides can use with next and previous buttons and imdicators.

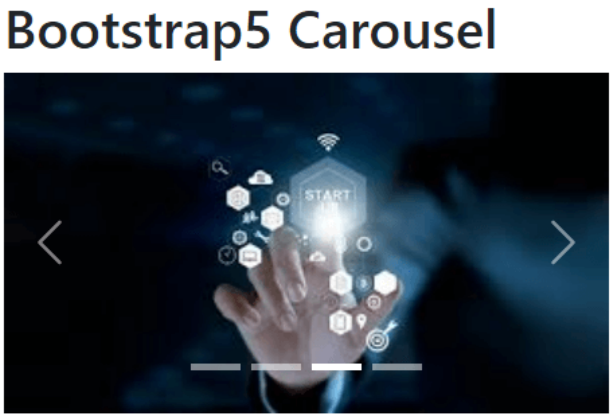
```
<!DOCTYPE html>
<html lang = "en">
<head>
<title> Bootstrap 5 Carousel Example </title>
<meta charset = "utf-8">
<meta name = "viewport" content = "width = device-width, initial-scale = 1">
<link href = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel = "stylesheet">
<script src = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js">
</script>
</head>
<body>
</head>
<body>
<div class = "container" id = "alertcont" style = "width:50%;height:50%;">
<h2> Bootstrap5 Carousel </h2>
```

```
<div id = "carouseldemo" class = "carousel slide" data-bs-ride = "carousel">
<!-- Indicators of carousel -->
<div class="carousel-indicators">
<button type = "button" data-bs-target = "#carouseldemo" data-bs-slide-to = "0" class="active"></button>
<button type = "button" data-bs-target = "#carouseldemo" data-bs-slide-to = "1"></button>
<button type = "button" data-bs-target = "#carouseldemo" data-bs-slide-to = "2"></button>
<button type = "button" data-bs-target = "#carouseldemo" data-bs-slide-to = "3"></button>
</div>
<!-- The slideshow with images -->
<div class = "carousel-inner">
<div class = "carousel-item active">
<img src = "G:\writing stuff\content-writing\JavaTpoint\2022\may\starts.jpg" alt = "image1" class = "d-block" style = "width:100%;">
</div>
<div class = "carousel-item">
<img src = "G:\writing stuff\content-writing\JavaTpoint\2022\may\guudu.jpg" alt = "image2" class = "d-block" style = "width:100%;">
</div>
<div class = "carousel-item">
<img src = "G:\writing stuff\content-writing\JavaTpoint\2022\may\starts.jpg" alt = "image2" class = "d-block" style = "width:100%;">
</div>
<div class = "carousel-item">
<img src = "G:\writing stuff\content-writing\JavaTpoint\2022\may\guudu.jpg" alt = "image2" class = "d-block" style = "width:100%;">
</div>
</div>
<!-- previous and next icons -->
<button class = "carousel-control-prev" type = "button" data-bs-target = "#carouseldemo" data-bs-slide = "prev">
<span class="carousel-control-prev-icon"></span>
</button>
```

```
<button class = "carousel-control-next" type = "button" data-bs-target = "#carouseldemo" data-bs-slide = "next">
<span class = "carousel-control-next-icon"></span>
</button>
</div>
</div>
</body>
</html>
```

Output

The following output shows the carousel function with indicators.


Bootstrap5 Carousel

Basic carousel with the caption

The carousel function includes extra information with the slide. It shows information, content and heading of the carousel function.

Syntax

The following syntax shows carousel function with caption using bootstrap 5 classes.

```
<div id = "carouseldemo" class = "carousel slide" data-bs-ride = "carousel">
<div class = "carousel-inner">
<div class = "carousel-item">
<img src = "image.jpg" alt = "image2" class = "d-block" style = "width:100%;">
<div class = "carousel-caption d-none d-md-block">
```

```
//information
</div>
</div>
</div>
</div>
```

Example

The following example uses four slides with indicator. The first slide is active with active indicator.

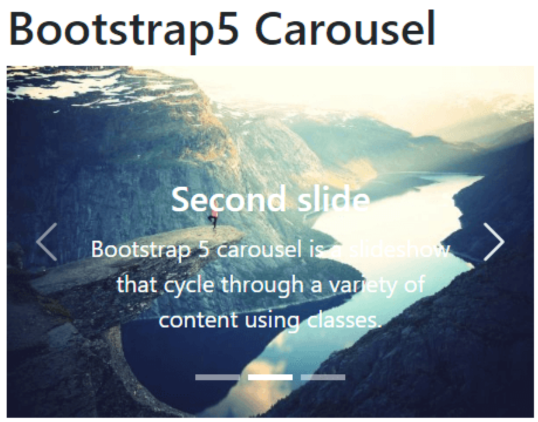
```
<!DOCTYPE html>
<html lang = "en">
<head>
<title> Bootstrap 5 Carousel Example </title>
<meta charset = "utf-8">
<meta name = "viewport" content = "width = device-width, initial-scale = 1">
<link href = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel = "stylesheet">
<script src = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js">
</script>
</head>
<body>
</head>
<body>
<div class = "container" id = "alertcont" style = "width:25%;height:25%;">
<h2> Bootstrap5 Carousel </h2>
<div id = "carouseldemo" class = "carousel slide" data-bs-ride = "carousel">
<!-- Indicators of carousel -->
<div class = "carousel-indicators">
<button type = "button" data-bs-target = "#carouseldemo" data-bs-slide-to = "0" class = "active"></button>
<button type = "button" data-bs-target = "#carouseldemo" data-bs-slide-to = "1"></button>
<button type = "button" data-bs-target = "#carouseldemo" data-bs-slide-to = "2"></button>
</div>
```

```
<div class = "carousel-inner">
<div class = "carousel-item active">
<img src = "G:\writing stuff\content-writing\JavaTpoint\2022\may\timespace.jpg" alt = "image1" class = "d-block" style = "width:100%;">
<div class = "carousel-caption d-none d-md-block">
<h4> First slide </h4>
<p> Bootstrap 5 carousel is a slideshow that cycle through a variety of content using classes.  </p>
</div>
</div>
<div class = "carousel-item">
<img src = "G:\writing stuff\content-writing\JavaTpoint\2022\may\guudu.jpg" alt = "image2" class = "d-block" style = "width:100%;">
<div class = "carousel-caption d-none d-md-block">
<h4> Second slide </h4>
<p> Bootstrap 5 carousel is a slideshow that cycle through a variety of content using classes.  </p>
</div>
</div>
<div class = "carousel-item">
<img src = "G:\writing stuff\content-writing\JavaTpoint\2022\may\timespace.jpg" alt = "image3" class = "d-block" style = "width:100%;">
<div class = "carousel-caption d-none d-md-block">
<h4> Third slide </h4>
<p> Bootstrap 5 carousel is a slideshow that cycle through a variety of content using classes.  </p>
</div>
</div>
</div>
<button class = "carousel-control-prev" type = "button" data-bs-target = "#carouseldemo" data-bs-slide = "prev">
<span class="carousel-control-prev-icon"></span>
</button>
<button class = "carousel-control-next" type = "button" data-bs-target = "#carouseldemo" data-bs-slide = "next">
```

```
<span class = "carousel-control-next-icon"></span>
</button>
</div>
</div>
</body>
</html>
```

Output

The following output shows the carousel function with caption.



Bootstrap5 Carousel

Second slide

Bootstrap 5 carousel is a slideshow that cycle through a variety of content using classes.

Basic carousel with interval

The carousel function shows essential information with multiple slides. The slides move automatically after some interval. We can set a time limit on the image displayed on the web screen.

Syntax

The following syntax shows carousel function with interval using bootstrap 5 classes.

```
<div id = "carouseldemo" class = "carousel slide" data-bs-ride = "carousel">
<div class = "carousel-inner">
<div class = "carousel-item" data-bs-interval = "5000">
</div>
```

```
</div>
</div>
```

Example

The following example uses three slides with interval. The first slide is active with large size interval than the other slides.

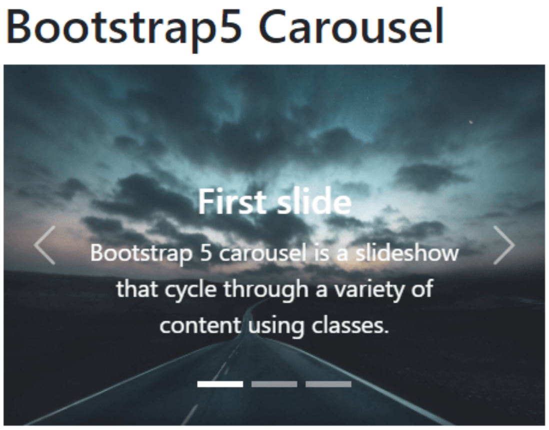
```
<!DOCTYPE html>
<html lang = "en">
<head>
<title> Bootstrap 5 Carousel Example </title>
<meta charset = "utf-8">
<meta name = "viewport" content = "width = device-width, initial-scale = 1">
<link href = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel = "stylesheet">
<script src = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js">
</script>
</head>
<body>
</head>
<body>
<div class = "container" id = "alertcont" style = "width:25%;height:25%;">
<h2> Bootstrap5 Carousel </h2>
<div id = "carouseldemo" class = "carousel slide" data-bs-ride = "carousel">
<!-- Indicators of carousel -->
<div class = "carousel-indicators">
<button type = "button" data-bs-target = "#carouseldemo" data-bs-slide-to = "0" class = "active"></button>
<button type = "button" data-bs-target = "#carouseldemo" data-bs-slide-to = "1"></button>
<button type = "button" data-bs-target = "#carouseldemo" data-bs-slide-to = "2"></button>
</div>
<div class = "carousel-inner">
<div class = "carousel-item active" data-bs-interval = "5000">
```

```
<img src = "G:\writing stuff\content-writing\JavaTpoint\2022\may\timespace.jpg" alt = "image1" class = "d-block" style = "width:100%;">
<div class = "carousel-caption d-none d-md-block">
<h4> First slide </h4>
<p> Bootstrap 5 carousel is a slideshow that cycle through a variety of content using classes.  </p>
</div>
</div>
<div class = "carousel-item" data-bs-interval = "3000">
<img src = "G:\writing stuff\content-writing\JavaTpoint\2022\may\guudu.jpg" alt = "image2" class = "d-block" style = "width:100%;">
<div class = "carousel-caption d-none d-md-block">
<h4> Second slide </h4>
<p> Bootstrap 5 carousel is a slideshow that cycle through a variety of content using classes.  </p>
</div>
</div>
<div class = "carousel-item">
<img src = "G:\writing stuff\content-writing\JavaTpoint\2022\may\timespace.jpg" alt = "image3" class = "d-block" style = "width:100%;">
<div class = "carousel-caption d-none d-md-block">
<h4> Third slide </h4>
<p> Bootstrap 5 carousel is a slideshow that cycle through a variety of content using classes.  </p>
</div>
</div>
</div>
<button class = "carousel-control-prev" type = "button" data-bs-target = "#carouseldemo" data-bs-slide = "prev">
<span class="carousel-control-prev-icon"></span>
</button>
<button class = "carousel-control-next" type = "button" data-bs-target = "#carouseldemo" data-bs-slide = "next">
<span class = "carousel-control-next-icon"></span>
</button>
```

```
</div>
</div>
</body>
</html>
```

Output

The following output shows the carousel function with intervals.



Bootstrap5 Carousel

First slide

Bootstrap 5 carousel is a slideshow that cycle through a variety of content using classes.

Dark Carousel

The carousel function uses the "carousel-dark" to display indicators, buttons and other info in dark colours. According to the web page template, it gives an attractive, stylish, and interactive carousel function.

Syntax

The following syntax shows carousel function with interval using bootstrap 5 classes.

```
<div id = "carouseldemo" class = "carousel carousel-dark slide" data-bs-ride = "carousel">
</div>
```

Example

The following example uses two slides with interval and dark carousel.

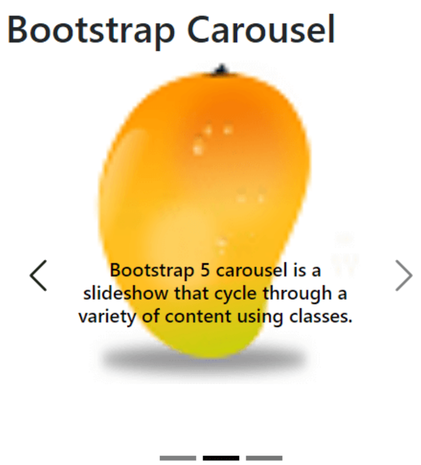
```
<!DOCTYPE html>
```

```html
<html lang = "en">
<head>
<title> Bootstrap 5 Carousel Example </title>
<meta charset = "utf-8">
<meta name = "viewport" content = "width = device-width, initial-scale = 1">
<link href = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel = "stylesheet">
<script src = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js">
</script>
</head>
<body>
</head>
<body>
<div class = "container" id = "alertcont" style = "width:25%;height:25%;">
<h2> Bootstrap Carousel </h2>
<div id = "carouseldemo" class = "carousel carousel-dark slide" data-bs-ride = "carousel">
<div class = "carousel-indicators">
<button type = "button" data-bs-target = "#carouseldemo" data-bs-slide-to = "0" class = "active"></button>
<button type = "button" data-bs-target = "#carouseldemo" data-bs-slide-to = "1"></button>
<button type = "button" data-bs-target = "#carouseldemo" data-bs-slide-to = "2"></button>
</div>
<div class = "carousel-inner">
<div class = "carousel-item active" data-bs-interval = "5000">
<img src = "G:\writing stuff\content-writing\JavaTpoint\2022\may\pinapple.png" alt = "image1" class = "d-block" style = "width:100%;">
<div class = "carousel-caption d-none d-md-block">
<h6> Bootstrap 5 carousel is a slideshow that cycle through a variety of content using classes.  </h6>
</div>
</div>
<div class = "carousel-item" data-bs-interval = "3000">
```

```
<img src = "G:\writing stuff\content-writing\JavaTpoint\2022\may\mango.png" alt = "image2" class = "d-block" style = "width:100%;">
<div class = "carousel-caption d-none d-md-block">
<h6> Bootstrap 5 carousel is a slideshow that cycle through a variety of content using classes. </h6>
</div>
</div>
<div class = "carousel-item" data-bs-interval = "2000">
<img src = "G:\writing stuff\content-writing\JavaTpoint\2022\may\pinapple.png" alt = "image2" class = "d-block" style = "width:100%;">
<div class = "carousel-caption d-none d-md-block">
<h6> Bootstrap 5 carousel is a slideshow that cycle through a variety of content using classes. </h6>
</div>
</div>
</div>
<button class = "carousel-control-prev" type = "button" data-bs-target = "#carouseldemo" data-bs-slide = "prev">
<span class="carousel-control-prev-icon"></span>
</button>
<button class = "carousel-control-next" type = "button" data-bs-target = "#carouseldemo" data-bs-slide = "next">
<span class = "carousel-control-next-icon"></span>
</button>
</div>
</div>
</body>
</html>
```

Output

The following output shows the dark carousel function with intervals.



Bootstrap Carousel

Bootstrap 5 carousel is a slideshow that cycle through a variety of content using classes.

Bootstrap 5 navbar

The Nav Bar in bootstrap 5 connects and links all of a website's relevant web pages, making it simple for users to navigate between them. A website's navigation bar, which includes the page names and links to them and a search box tailored to user needs, is a crucial element.

For responsive collapsing and color scheme classes, navbars need to wrap the ".navbar" with the ".navbar-expand-sm|-md|-lg|-xl|-xxl." By default, navbars and their contents are fluid.

Basic navbar function

The container can be modified to limit its horizontal width in many ways. Use our utility classes for flex and spacing to adjust the spacing and alignment within navbars.

By default, navbars are responsive, but you can simply change that with the bootstrap 5 class. The responsive navigation bar function depends on our Collapse JavaScript plugin.

Use a "nav" element to ensure accessibility, or, if using a more general element like a "div," add the attribute role = "navigation". This role uses each navbar to make it clear to users of assistive technologies, such as a landmark area.

Use aria-current = "page" to indicate the current page or aria-current="true" to indicate the current item in a set.

Syntax

The following syntax shows navbar function using bootstrap 5 classes.

```
<nav class = "navbar navbar-expand-sm bg-dark">
<div class = "container-fluid">
<ul class = "navbar-nav">
<li class = "nav-item">
<a class = "nav-link" href = "#"> First </a>
</li>
</ul>
</div>
</nav>
```

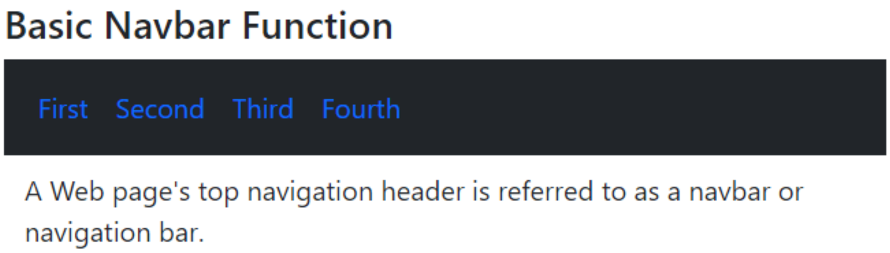
Example

```html
<!DOCTYPE html>
<html lang="en">
<head>
<title> Bootstrap Navbar Example </title>
<meta charset = "utf-8">
<meta name = "viewport" content = "width=device-width, initial-scale=1">
<link href = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel = "stylesheet">
<script src = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js"></script>
</head>
<body>
<div class = "container mt-2">
<h4> Basic Navbar Function </h4>
<nav class = "navbar navbar-expand-sm bg-dark">
<div class = "container-fluid">
<ul class = "navbar-nav">
<li class = "nav-item">
<a class = "nav-link" href = "#"> First </a>
</li>
<li class = "nav-item">
<a class = "nav-link" href = "#"> Second </a>
</li>
<li class = "nav-item">
<a class = "nav-link" href = "#"> Third </a>
</li>
<li class = "nav-item">
<a class = "nav-link" href = "#"> Fourth </a>
</li>
</ul>
</div>
</nav>
<div class = "container mt-2">
```

```
<p> A Web page's top navigation header is referred to as a navbar or navigation bar. </p>
</div>
</div>
</body>
</html>
```

Output

The basic navbar function shows in below image.



Basic Navbar Function

A Web page's top navigation header is referred to as a navbar or navigation bar.

Vertical navbar

The ".navbar-expand-sm|-md|-lg|-xl|-xxl" class uses the responsive horizontal navbar. If we do not use the navbar expand class, the navbar function automatically displays in vertical format. The <nav> tag uses the navbar class with the required background class to show the vertical navbar.

Syntax

The following syntax shows vertical navbar function using bootstrap 5 classes.

```
<nav class = "navbar bg-light">
//write navbar functions code here!
</nav>=
```

Example

The following example shows vertical navbar using bootstrap 5 classes.

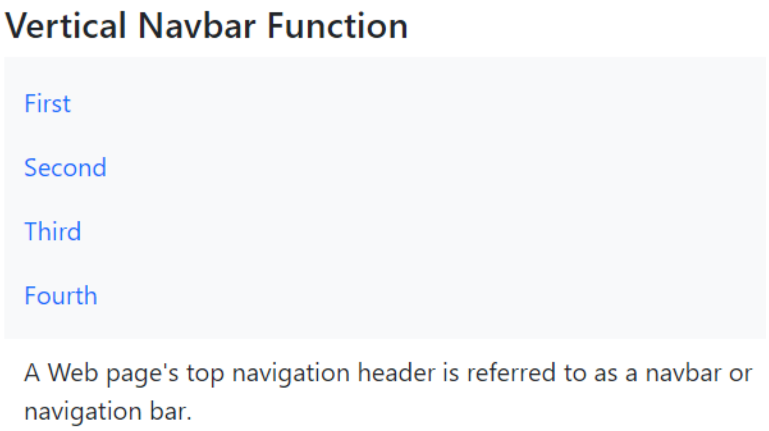
```
<!DOCTYPE html>
<html lang="en">
<head>
```

```
<title> Bootstrap Navbar Example </title>
<meta charset = "utf-8">
<meta name = "viewport" content = "width=device-width, initial-scale=1">
<link href = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel = "stylesheet">
<script src = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js"></script>
</head>
<body>
<div class = "container mt-2">
<h4> Vertical Navbar Function </h4>
<nav class = "navbar bg-light">
<div class = "container-fluid">
<ul class = "navbar-nav">
<li class = "nav-item">
<a class = "nav-link" href = "#"> First </a>
</li>
<li class = "nav-item">
<a class = "nav-link" href = "#"> Second </a>
</li>
<li class = "nav-item">
<a class = "nav-link" href = "#"> Third </a>
</li>
<li class = "nav-item">
<a class = "nav-link" href = "#"> Fourth </a>
</li>
</ul>
</div>
</nav>
<div class = "container mt-2">
<p> A Web page's top navigation header is referred to as a navbar or navigation bar. </p>
</div>
</div>
</body>
```

```
</html>
```

Output

The vertical navbar function shows below image.



Vertical Navbar Function

A Web page's top navigation header is referred to as a navbar or navigation bar.

Centered navbar

The "justify-content-center" class places navbar items in the center position. If we want to place items in the end position, use the "justify-content-end" class. This class is placed inside of the <nav> tag with the "navbar" class.

Syntax

The following syntax shows vertical navbar function using bootstrap 5 classes.

```
<nav class = "navbar bg-dark navbar-expand-sm justify-content-center">
//write navbar functions code here!
</nav>
```

Example

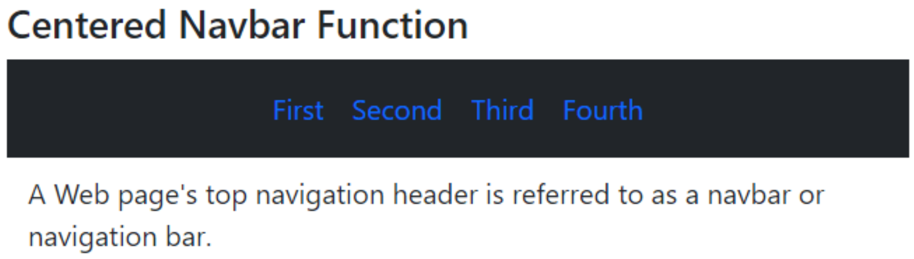
The following example shows vertical navbar using bootstrap 5 classes.

```
<!DOCTYPE html>
<html lang="en">
<head>
<title> Bootstrap Navbar Example </title>
<meta charset = "utf-8">
```

```
<meta name = "viewport" content = "width=device-width, initial-scale=1">
<link href = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel = "stylesheet">
<script src = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js"></script>
</head>
<body>
<div class = "container mt-2">
<h4> Centered Navbar Function </h4>
<nav class = "navbar bg-dark navbar-expand-sm justify-content-center">
<ul class = "navbar-nav">
<li class = "nav-item">
<a class = "nav-link" href = "#"> First </a>
</li>
<li class = "nav-item">
<a class = "nav-link" href = "#"> Second </a>
</li>
<li class = "nav-item">
<a class = "nav-link" href = "#"> Third </a>
</li>
<li class = "nav-item">
<a class = "nav-link" href = "#"> Fourth </a>
</li>
</ul>
</nav>
<div class = "container mt-2">
<p> A Web page's top navigation header is referred to as a navbar or navigation bar. </p>
</div>
</div>
</body>
</html>
```

Output

The centered navbar function shows below image.



Centered Navbar Function

A Web page's top navigation header is referred to as a navbar or navigation bar.

Colored navbar function

The "contextual" classes are placed in the <nav> tag with the navbar class to display the colored navbar. The navbar changes the background color per the required bg-* class. The colored navbar either gives the meaning of the navigational function or shows the navbar as per the application's theme. We can modify with CSS style and bootstrap 5 classes.

Syntax

The following syntax shows navbar function using bootstrap 5 classes.

```
<nav class = "navbar navbar-expand-sm bg-primary">
<div class = "container-fluid">
<ul class = "navbar-nav">
<li class = "nav-item">
<a class = "nav-link" href = "#"> First </a>
</li>
</ul>
</div>
</nav>
```

Example

The following example shows colored navbar using bootstrap 5 classes. we can see different background colors navbar.

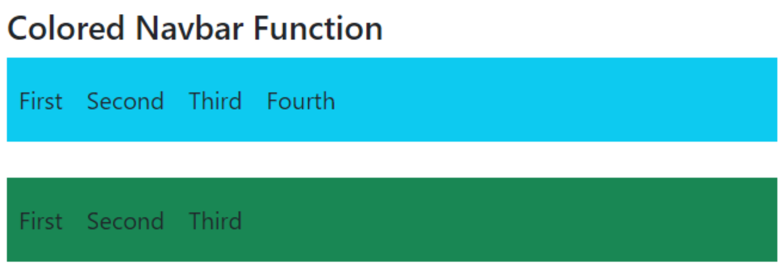
```
<!DOCTYPE html>
<html lang="en">
```

```
<head>
<title> Bootstrap Navbar Example </title>
<meta charset = "utf-8">
<meta name = "viewport" content = "width=device-width, initial-scale=1">
<link href = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel = "stylesheet">
<script src = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js"></script>
</head>
<body>
<div class = "container mt-2">
<h4> Colored Navbar Function </h4>
<nav class = "navbar bg-info navbar-expand-sm">
<ul class = "navbar-nav">
<li class = "nav-item">
<a class = "nav-link text-dark" href = "#"> First </a>
</li>
<li class = "nav-item">
<a class = "nav-link text-dark" href = "#"> Second </a>
</li>
<li class = "nav-item">
<a class = "nav-link text-dark" href = "#"> Third </a>
</li>
<li class = "nav-item">
<a class = "nav-link text-dark" href = "#"> Fourth </a>
</li>
</ul>
<br>
</nav>
<nav class = "navbar bg-success navbar-expand-sm">
<ul class = "navbar-nav">
<li class = "nav-item">
<a class = "nav-link text-dark" href = "#"> First </a>
</li>
```

```
<li class = "nav-item">
<a class = "nav-link text-dark" href = "#"> Second </a>
</li>
<li class = "nav-item">
<a class = "nav-link text-dark" href = "#"> Third </a>
</li>
</ul>
</nav>
</div>
</body>
</html>
```

Output

The colored navbar function shows in below image.



Navbar function with Logo

The "navbar-brand" shows the navigation bar's logo, images, and animated function. This function helps to represent the company's slogan or logo. This class is placed in the <a> tag to modify the nav item with the required logo.

Syntax

The following syntax shows navbar function using bootstrap 5 classes.

```
<nav class = "navbar navbar-expand-sm bg-primary">
<div class = "container-fluid">
<a class = "navbar-brand" href = "#"> Logos </a>
```

```
</div>
<ul class = "navbar-nav">
<li class = "nav-item">
<a class = "nav-link" href = "#"> First </a>
</li>
</ul>
</nav>
```

Example

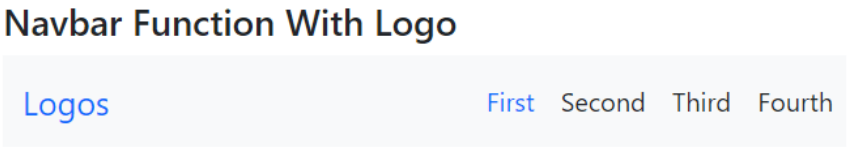
The following example shows navbar with logo using bootstrap 5 classes.

```
<!DOCTYPE html>
<html lang="en">
<head>
<title> Bootstrap Navbar Example </title>
<meta charset = "utf-8">
<meta name = "viewport" content = "width=device-width, initial-scale=1">
<link href = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel = "stylesheet">
<script src = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js"></script>
</head>
<body>
<div class = "container mt-2">
<h4> Navbar Function With Logo </h4>
<nav class = "navbar bg-light navbar-expand-sm justify-content-center">
<div class = "container-fluid">
<a class = "navbar-brand" href = "#"> Logos </a>
</div>
<ul class = "navbar-nav">
<li class = "nav-item">
<a class = "nav-link" href = "#"> First </a>
</li>
<li class = "nav-item">
```

```
<a class = "nav-link text-dark" href = "#"> Second </a>
</li>
<li class = "nav-item">
<a class = "nav-link text-dark" href = "#"> Third </a>
</li>
<li class = "nav-item">
<a class = "nav-link text-dark" href = "#"> Fourth </a>
</li>
</ul>
</nav>
</div>
</body>
</html>
```

Output

The navbar with the logo function shows the below image.



Navbar function with Form

The navbar function uses a form for immediate user interaction. The navbar uses a form to submit users' data, like an email address or contact number. If we want to search for anything from a web page, the navigation bar uses a search function with the submit button.

Example

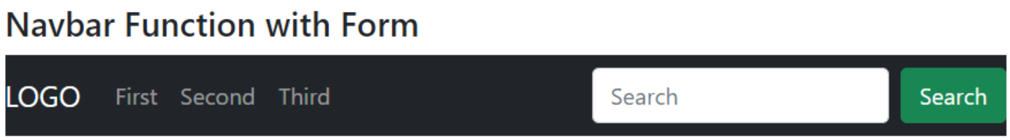
The following example shows navbar with form using bootstrap 5 classes.

```
<!DOCTYPE html>
<html lang="en">
<head>
<title> Bootstrap Navbar Example </title>
```

```
<meta charset = "utf-8">
<meta name = "viewport" content = "width=device-width, initial-scale=1">
<link href = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel = "stylesheet">
<script src = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js"></script>
</head>
<body>
<div class = "container mt-2">
<h4> Navbar Function with Form </h4>
<nav class = "navbar bg-dark navbar-dark navbar-expand-sm">
<a class = "navbar-brand" href="#"> LOGO </a>
<ul class = "navbar-nav me-auto">
<li class = "nav-item">
<a class = "nav-link" href = "#"> First </a>
</li>
<li class = "nav-item">
<a class = "nav-link" href = "#"> Second </a>
</li>
<li class = "nav-item">
<a class = "nav-link" href = "#"> Third </a>
</li>
</ul>
<form class="d-flex">
<input class="form-control me-2" type="text" placeholder="Search">
<button class="btn btn-success" type="button">Search</button>
</form>
</nav>
</div>
</body>
</html>
```

Output

The navbar with the form function shows the below image.



Navbar Function with Form

Toggle Navbar function

The toggle navbar helps to create a responsive function for all sizes of devices. The navbar shows on large-size devices with default size. If the device's screen is xl or sm size, then the toggle button shows on the screen. This button works as a dropdown with a nav link.

Syntax

The following syntax shows toggle navbar using bootstrap 5 classes.

```
<nav class = "navbar bg-dark navbar-dark navbar-expand-sm">
<button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#collapsible">
<span class="navbar-toggler-icon"></span>
</button>
<div class="collapse navbar-collapse" id="collapsible">
<ul class = "navbar-nav me-auto">
//list item!
</ul>
</div>
</nav>
```

Example

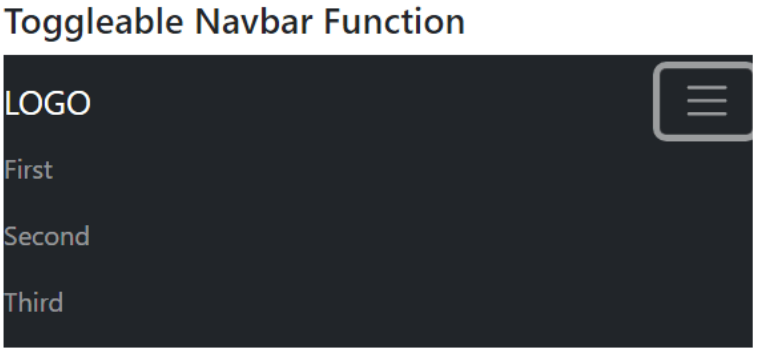
The following example shows navbar with Toggleable function using bootstrap 5 classes.

```
<!DOCTYPE html>
<html lang="en">
<head>
<title> Bootstrap Navbar Example </title>
<meta charset = "utf-8">
<meta name = "viewport" content = "width=device-width, initial-scale=1">
```

```
<link href = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel = "stylesheet">
<script src = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js"></script>
</head>
<body>
<div class = "container mt-2">
<h4> Toggleable Navbar Function </h4>
<nav class = "navbar bg-dark navbar-dark navbar-expand-sm">
<a class = "navbar-brand" href="#"> LOGO </a>
<button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#collapsible">
<span class="navbar-toggler-icon"></span>
</button>
<div class="collapse navbar-collapse" id="collapsible">
<ul class = "navbar-nav me-auto">
<li class = "nav-item">
<a class = "nav-link" href = "#"> First </a>
</li>
<li class = "nav-item">
<a class = "nav-link" href = "#"> Second </a>
</li>
<li class = "nav-item">
<a class = "nav-link" href = "#"> Third </a>
</li>
</ul>
</div>
</nav>
</div>
</body>
</html>
```

Output

The navbar with the toggle function shows the below image.



Toggleable Navbar Function

Sticky Navbar function

The navbar function places somewhere on the web pages. The web page scrolls up with information, and then the navbar function sticks to the top of the page and scrolls only to the body content. We can create a nav bar with a sticky top or bottom position using the "sticky-top" or the "sticky-bottom" position.

Syntax

The following syntax shows navbar with sticky top position.

```
<nav class = "navbar bg-dark navbar-dark navbar-expand-sm sticky-top">
</nav>
```

The following syntax shows navbar with sticky bottom position.

```
<nav class = "navbar bg-dark navbar-dark navbar-expand-sm sticky-bottom">
</nav>
```

Example

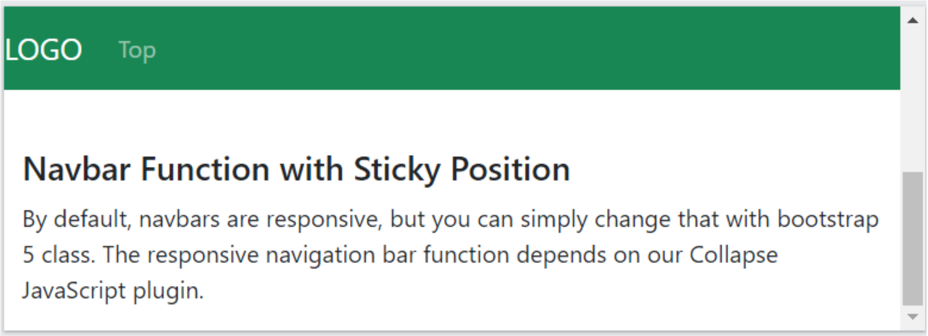
The following example shows navbar with sticky top position using bootstrap 5 classes.

```
<!DOCTYPE html>
<html lang="en">
<head>
<title> Bootstrap Navbar Example </title>
<meta charset = "utf-8">
<meta name = "viewport" content = "width=device-width, initial-scale=1">
```

```
<link href = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel = "stylesheet">
<script src = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js"></script>
</head>
<body>
<div class = "container-fluid mt-5"><br>
<h4> Navbar Function with Sticky Top Position </h4>
<p>
Use a "nav" element to ensure accessibility, or, if using a more general element like a "div," add the attribute role = "navigation". This role uses to each navbar to make it clear to users of assistive technologies such as a landmark area.
</p>
</div>
<nav class = "navbar bg-success navbar-dark navbar-expand-sm sticky-top">
<a class = "navbar-brand" href="#"> LOGO </a>
<ul class = "navbar-nav">
<li class = "nav-item">
<a class = "nav-link" href = "#"> Top </a>
</li>
</ul>
</nav>
<div class = "container-fluid mt-5"><br>
<h4> Navbar Function with Sticky Position </h4>
<p> By default, navbars are responsive, but you can simply change that with bootstrap 5 class. The responsive navigation bar function depends on our Collapse JavaScript plugin.
</p>
</div>
</body>
</html>
```

Output

The navbar function with a sticky top position shows the below image.



Navbar Function with Sticky Position

By default, navbars are responsive, but you can simply change that with bootstrap 5 class. The responsive navigation bar function depends on our Collapse JavaScript plugin.

Navbar function with Fixed Position

The "fixed-top" or the "fixed-bottom" class places the navbar function on the top or bottom by default. The navbar function displays a fixed position on the web page.

Syntax

The following syntax shows navbar with fixed top position.

```
<nav class = "navbar bg-dark navbar-dark navbar-expand-sm fixed-top">
</nav>
```

The following syntax shows navbar with fixed bottom position.

```
<nav class = "navbar bg-dark navbar-dark navbar-expand-sm fixed-bottom">
</nav>
```

Example

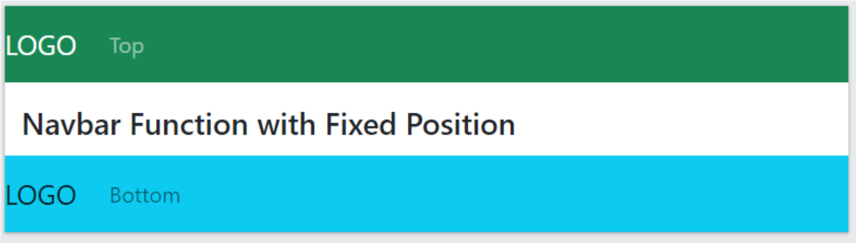
The following example shows navbar with fixed position using bootstrap 5 classes. the first navbar shows on top side of the application and second navbar shows on bottom side of the application. We can see the other container data with navigation bar function.

```
<!DOCTYPE html>
<html lang="en">
<head>
<title> Bootstrap Navbar Example </title>
<meta charset = "utf-8">
<meta name = "viewport" content = "width=device-width, initial-scale=1">
```

```
<link href = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel = "stylesheet">
<script src = "https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js"></script>
</head>
<body>
<div class = "container mt-2">
<nav class = "navbar bg-success navbar-dark navbar-expand-sm fixed-top">
<a class = "navbar-brand" href="#"> LOGO </a>
<ul class = "navbar-nav">
<li class = "nav-item">
<a class = "nav-link" href = "#"> Top </a>
</li>
</ul>
</nav>
<nav class = "navbar bg-info navbar-light navbar-expand-sm fixed-bottom">
<a class = "navbar-brand" href="#"> LOGO </a>
<ul class = "navbar-nav">
<li class = "nav-item">
<a class = "nav-link" href = "#"> Bottom </a>
</li>
</ul>
</nav>
</div>
<div class = "container-fluid mt-5"><br>
<h4> Navbar Function with Fixed Position </h4>
</div>
</body>
</html>
```

Output

The navbar function with a fixed position shows the below image.



Navbar Function with Fixed Position