**UNIT-1**

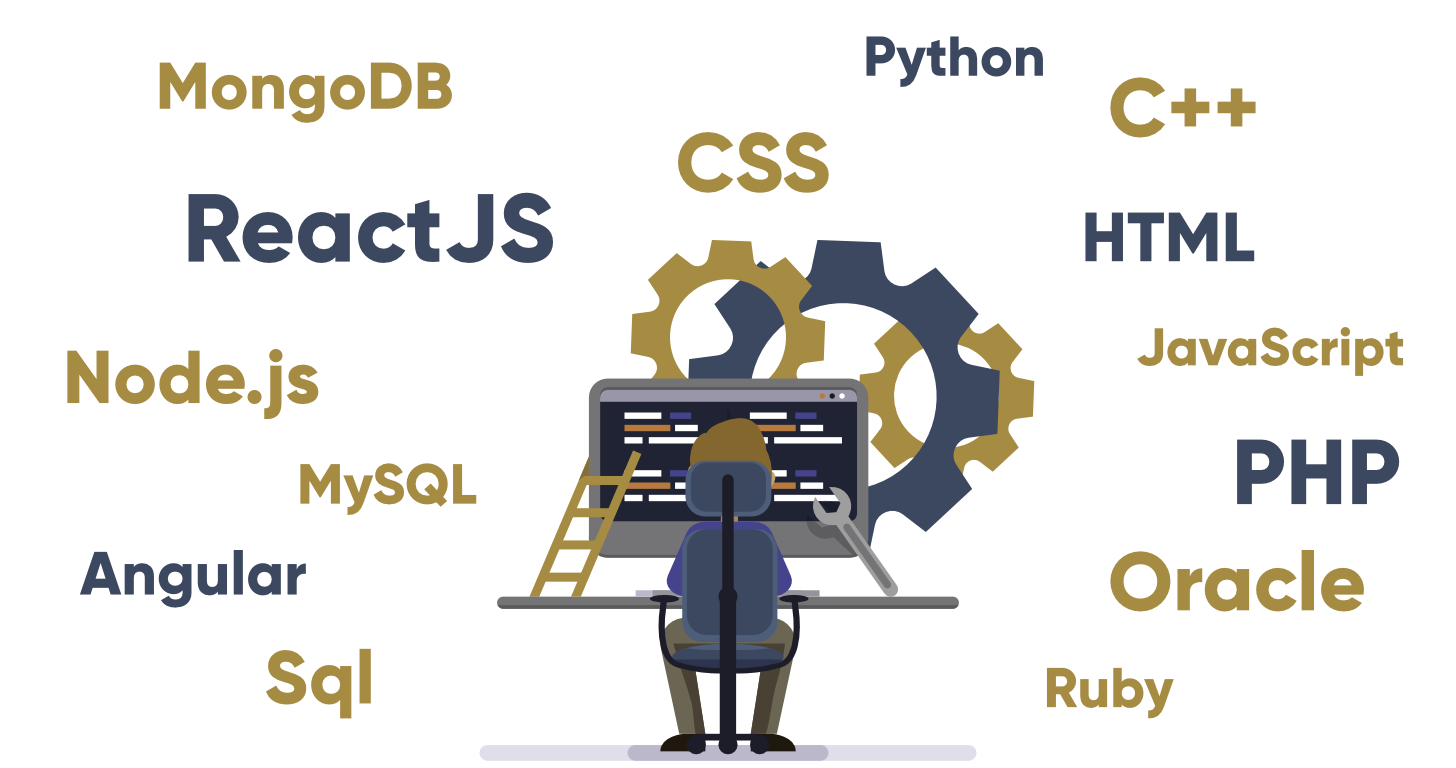
**Introduction:**

**Full stack:** In technology development, **full stack** refers to an **entire** computer system or application from the front end to the back end and the code that connects the two. The back end of a computer system encompasses “behind-the-scenes” technologies such as the database and operating system.

## Full Stack Web Developer: A full stack web developer is a person who can develop both client and server software. They are proficient in both front-end and back-end languages and frameworks, as well as in server, network and hosting environments.

## (or)

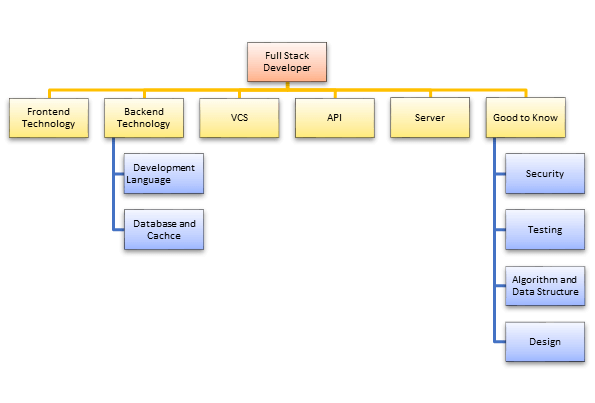
**Full Stack Developer** is an engineer who works on both client-side and server-side of the software application. This type of developer works on the Full Stack of a software application meaning Front end development, Back end development, Database, Server, API, and version controlling systems. Hence, the name "Full Stack" Developer.Full stack developer translates user requirements into the overall architecture and implement the new systems.



**Why Do You Need a Full-Stack Developer?**

* Full stack developer helps you to keep every part of the system running smoothly
* Full stack developer can provide help to everyone in the team and greatly reduce the time and technical costs of team communication
* If one person plays different roles, it saves your company's personnel, infrastructure and operational cost.

**Full Stack Developer Skills You Need to Know:**



**1) Front-end technology**

Full stack developer should be master of essential front-end technologies like HTML5, CSS3, JavaScript. Knowledge of third-party libraries like jQuery, LESS, [Angular](https://www.guru99.com/angularjs-tutorial.html) and [ReactJS](https://www.guru99.com/reactjs-tutorial.html) is desirable

**2) Development Languages**

Full stack engineer should know at least one server-side programming languages like Java, Python, Ruby, .Net etc.

**3) Database and cache**

Knowledge of various DBMS technology is another important need of full stack developer. MySQL, MongoDB, Oracle, SQLServer are widely used for this purpose. Knowledge of caching mechanisms like varnish, Memcached, Redis is a plus.

**4) Basic design ability**

In order to become a successful Full Stack web developer, the knowledge of designing is also recommended. Moreover, the person should know the principle of basic prototype design and UI /UX design.

**5) Server**

Exposure to handling Apache or nginx servers is desirable. A good background in Linux helps tremendously in administering servers.

**6) Version control system (VCS)**

A version control system allows full stack developers to keep track of all the changes made in the codebase. The knowledge of **Git** helps full stack developers to understand how to get the latest code, update parts of the code, make changes in other developer's code without breaking things.

**7) Working with API (REST & SOAP):**

Knowledge of web services or API is also important for full stack developers. Knowledge of creations and consumption of REST and SOAP services is desirable.

**SOAP** (Simple Object Access Protocol) is a standards-based web services access protocol that has been around for a long time. ... **REST** (Representational State Transfer) is another standard, made in response to **SOAP's** shortcomings. It seeks to fix the problems with **SOAP** and provide a simpler method of accessing web services.

**Other important points:**

1. Ability to write quality **unit tests**
2. He or she should have a complete understanding of automated processes for building testing, document, and deploying it at scale
3. An awareness of **security** concerns is important, as each layer has its own vulnerabilities
4. Knowledge of **Algorithms** and data structures is also an essential need for professional full stack developers.

**Java Full Stack Developer** :

A **Java Full Stack Developer** is a developer who has expertise and deep knowledge of frameworks and tools used in Java full stack development like Core Java, servlets, APIs, database, web architecture, etc. A Full Stack Java developer can build whole Java applications including front end, back-end, database, APIs, server and version control.

**Full Stack JavaScript:**

JavaScript has been around for over 20 years. It is the dominant programming language in web development.

In the beginning JavaScript was a language for the web client (browser). Then came the ability to use JavaScript on the web server (with Node.js).

Today the hottest buzzword is "Full Stack JavaScript".

The idea of "Full Stack JavaScript" is that all software in a web application, both client side and server side, should be written using JavaScript only.

**What is a Software Stack?**

Software stack is a collection of the programs which are used together to produce a specific result. It includes an operating system and its application. For example, a smartphone software stack includes OS along with the phone app, web browsers, and default applications.

**Popular Stacks:**

* LAMP stack: JavaScript - Linux - Apache - MySQL - PHP
* MERN: JavaScript - MongoDB - Express - React Js - Node.js
* MEAN stack: JavaScript - MongoDB - Express - AngularJS - Node.js
* Django stack: JavaScript - Python - Django - MySQL
* Ruby on Rails: JavaScript - Ruby - SQLite - Rails

**LAMP Stack:**

LAMP is a widely used model for web service stacks. Its name "LAMP" is an acronym of four open-source components.

* L= Linux: An open source operating system
* A= Apache: Widely used web server software
* M= MySQL: Popular open source database
* P=PHP: Server-side open source scripting language
* Many popular websites and web applications run on LAMP stack, Example: Facebook.

**MERN Stack:**

MERN is a collection of JavaScript-based technologies:

* M=MongoDB: Popular [nosql](https://www.guru99.com/nosql-tutorial.html) database
* E=Express: Light and portable web program framework
* R=React: A javascript library for building user interfaces
* N=Node.js: A server-side JavaScript run time

This stack currently in the huge demand as it is widely used to develop web applications.

**MEAN Stack:**

[MEAN](https://www.guru99.com/mean-stack-developer.html) Stack Application Development is witnessing a growing trend in usage. MEAN is an abbreviation of:

* M = MongoDB: nosql Database
* E = Express: Easy to use light and portable web program framework
* A = Angular.js: Robust framework for developing HTML5 and JavaScript- web programs
* N = Node.is: a server-side JavaScript run time

**What Does a Full Stack Developer Do?**

As a full stack developer, you may be involved in following activities:

* Translate user requirements into the overall architecture and implementation of new systems
* Manage Project and coordinate with the Client
* Write backend code in [Ruby](https://www.guru99.com/ruby-on-rails-tutorial.html), [Python](https://www.guru99.com/python-tutorials.html), [Java](https://www.guru99.com/java-tutorial.html), [PHP](https://www.guru99.com/php-tutorials.html) languages
* Writing optimized front end code HTML and [JavaScript](https://www.guru99.com/interactive-javascript-tutorials.html)
* Understand, create and debug database related queries
* Create test code to validate the applicaition against client requirement.
* Monitor the performance of web applications & infrastructure
* Troubleshooting web application with a fast and accurate a resolution

**Advantages:**

The advantage of being a full stack web developer is:

* You can master all the techniques involved in a development project
* You can make a prototype very rapidly
* You can provide help to all the team members
* You can reduce the cost of the project
* You can reduce the time used for team communication
* You can switch between front and back end development based on requirements
* You can better understand all aspects of new and upcoming technologies

**Disadvantages:**

* The solution chosen can be wrong for the project
* The solution chosen can be dependent on developer skills
* The solution can generate a key person risk
* Being a full stack developer is increasingly complex

**Technology related to full stack development:**

**Front end:** It is the visible part of website or web application which is responsible for user experience. The user directly interacts with the front end portion of the web application or website.

HTML,CSS,BOOTSTRAP,W3.CSS,JAVASCRIPT,ES5,HTML DOM,JSON,XML,JQUERY,ANGULAR,REACT,BACKBONE.JS,EMBER.JS,REDUX,STORYBOOK,GRAPHQL,METEOR.JS,GRUNT,GULF.

[**HTML:**](https://www.geeksforgeeks.org/html-tutorials/) HTML stands for Hyper Text Markup Language. It is used to design the front end portion of web pages using markup language. HTML is the combination of Hypertext and Markup language. Hypertext defines the link between the web pages. The markup language is used to define the text documentation within tag which defines the structure of web pages.

[**CSS:**](https://www.geeksforgeeks.org/css-tutorials/) Cascading Style Sheets, fondly referred to as CSS, is a simply designed language intended to simplify the process of making web pages presentable. CSS allows you to apply styles to web pages. More importantly, CSS enables you to do this independent of the HTML that makes up each web page.

[**JavaScript:**](https://www.geeksforgeeks.org/javascript-tutorial/) JavaScript is a famous scripting language used to create the magic on the sites to make the site interactive for the user. It is used to enhancing the functionality of a website to running cool games and web-based software.

**Front End Frameworks and Libraries:**

[AngularJS](https://www.geeksforgeeks.org/category/web-technologies/angular-js/)**:** AngularJs is a JavaScript open source front-end framework that is mainly used to develop single page web applications(SPAs). It is a continuously growing and expanding framework which provides better ways for developing web applications. It changes the static HTML to dynamic HTML. It is an open source project which can be freely used and changed by anyone. It extends HTML attributes with Directives, and data is bound with HTML.

[React.js](https://www.geeksforgeeks.org/react-js-introduction-working/)**:** React is a declarative, efficient, and flexible JavaScript library for building user interfaces. ReactJS is an open-source, component-based front end library responsible only for the view layer of the application. It is maintained by Facebook.

[Bootstrap](https://www.geeksforgeeks.org/bootstrap-tutorials/)**:** Bootstrap is a free and open-source tool collection for creating responsive websites and web applications. It is the most popular HTML, CSS, and JavaScript framework for developing responsive, mobile-first web sites.

[jQuery](https://www.geeksforgeeks.org/jquery-tutorials/)**:** jQuery is an open source JavaScript library that simplifies the interactions between an HTML/CSS document, or more precisely the Document Object Model (DOM), and JavaScript. Elaborating the terms, jQuery simplifies HTML document traversing and manipulation, browser event handling, DOM animations, Ajax interactions, and cross-browser JavaScript development.

**SASS:** It is the most reliable, mature and robust CSS extension language. It is used to extend the functionality of an existing CSS of a site including everything from variables, inheritance, and nesting with ease.Some other libraries and frameworks are: Semantic-UI, Foundation, Materialize, Backbone.js, Express.js, Ember.js etc.

**Other Important Points:**

* + Work with text editors to use shortcuts and its facilities i.e. Visual studio, Atom, Sublime etc.
  + Make UI responsible using grid system.
  + Git and git commands like init, add, commit etc for version control and to work with team.
  + Other tools like npm & yarn package managers, sass css pre-processor, browser DevTools i.e. chrome devtools.
  + Understand using HTTP, JSON, GraphQL APIs to fetch data using axios or other tools.
  + It also requires some design skill to make layout and look better.

**Back end:** It refers to the server-side development of web application or website with a primary focus on how the website works. It is responsible for managing the database through queries and APIs by client-side commands. This type of website mainly consists of three parts front end, back end, and database.

PHP,ASP,C++,C#,JAVA,PYTHON,NODE.JS,EXPRESS.JS,RUBY,REST,GO,SQL,MONGODB,FIREBASE.COM,SASS,LESS,PARSE.COM,

PASS(AZURE,HEROKU)

The back end portion is built by using some libraries, frameworks, and languages which are discussed below:

* [**PHP:**](https://www.geeksforgeeks.org/php/) PHP is a server-side scripting language designed specifically for web development. Since, PHP code executed on server side so it is called server side scripting language.
* [**C++**](https://www.geeksforgeeks.org/c-plus-plus/) It is a general purpose programming language and widely used now a days for competitive programming. It is also used as backend language.
* [**Java:**](https://www.geeksforgeeks.org/java/) Java is one of the most popular and widely used programming language and platform. It is highly scalable. Java components are easily available.
* [**Python:**](https://www.geeksforgeeks.org/python-programming-language/) Python is a programming language that lets you work quickly and integrate systems more efficiently.
* [**JavaScript:**](https://www.geeksforgeeks.org/javascript-tutorial/) Javascript can be used as both (front end and back end) programming languages.
  + [Node.js](https://www.geeksforgeeks.org/introduction-to-nodejs/)**:** Node.js is an open source and cross-platform runtime environment for executing JavaScript code outside of a browser. You need to remember that NodeJS is not a framework and it’s not a programming language. Most of the people are confused and understand it’s a framework or a programming language. We often use Node.js for building back-end services like APIs like Web App or Mobile App. It’s used in production by large companies such as Paypal, Uber, Netflix, Wallmart and so on.
  + **Back End Frameworks:** The list of back end frameworks are: Express, Django, Rails, Laravel, Spring etc.
  + The other back end program/scripting languages are: C#, Ruby, REST, GO etc.

**Other Important Points:**

* + Structuring the data in efficient way.
  + Handle request-response of APIs for storing and retrieve data.
  + Security of data is important.

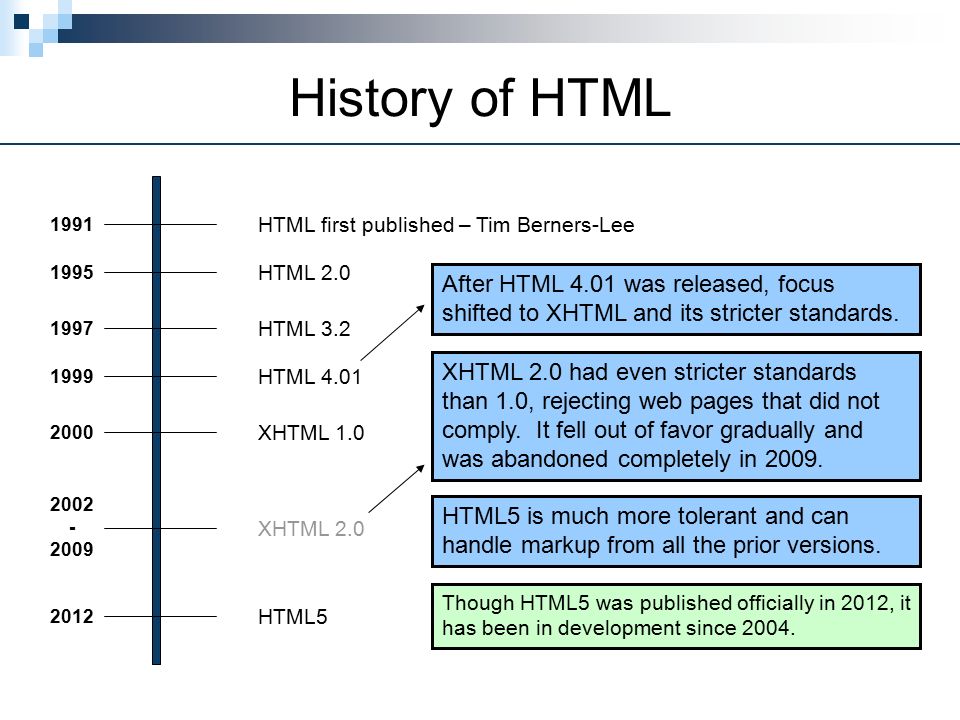
[Database](https://www.geeksforgeeks.org/dbms/)**:** Database is the collection of inter-related data which helps in efficient retrieval, insertion and deletion of data from database and organizes the data in the form of tables, views, schemas, reports etc.

* + **Oracle:** Oracle database is the collection of data which is treated as a unit. The purpose of this database is to store and retrieve information related to the query. It is a database server and used to manages information.
  + [MongoDB](https://www.geeksforgeeks.org/mongodb-an-introduction/)**:** MongoDB, the most popular NoSQL database, is an open source document-oriented database. The term ‘NoSQL’ means ‘non-relational’. It means that MongoDB isn’t based on the table-like relational database structure but provides an altogether different mechanism for storage and retrieval of data.
  + [Sql](https://www.geeksforgeeks.org/sql-tutorial/)**:** Structured Query Language is a standard Database language which is used to create, maintain and retrieve the relational database.

**Full Stack Developer is "jack of all trade, master on none".**

**Getting Started with HTML**

* 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.
* HTML **elements** are the building blocks of HTML pages.
* HTML elements are represented by **<> tags.**
* **An HTML file is simply a text file saved with an .html or .htm extension.**



**Creating First HTML Document:**

### Step 1: Creating the HTML file

Open up your computer's plain text editor and create a new file.

### Step 2: Type some HTML code

Start with an empty window and type the following code:

<!DOCTYPE html>

<html lang="en">

<head>

<title>A simple HTML document</title>

</head>

<body>

<p>Hello World!<p>

</body>

</html>

### Step 3: Saving the file

Now save the file on your desktop as "myfirstpage.html ".

To open the file in a browser. Navigate to your file then double click on it. It will open in your default Web browser. If it does not, open your browser and drag the file to it.

* The first line <!DOCTYPE html> is the [document type declaration](https://www.tutorialrepublic.com/html-tutorial/html-doctypes.php). It instructs the web browser that this document is an HTML5 document. It is case-insensitive.
* The <head> element is a container for the tags that provides information about the document, for example, <title> tag defines the title of the document.
* The <body> element contains the document's actual content (paragraphs, links, images, tables, and so on) that is rendered in the web browser and displayed to the user.

**HTML Tags and Elements:**

HTML is written in the form of HTML elements consisting of markup tags. These markup tags are the fundamental characteristic of HTML. Every markup tag is composed of a keyword, surrounded by angle brackets, such as <html>, <head>, <body>, <title>, <p>, and so on.

HTML tags normally come in pairs like <html> and </html>. The first tag in a pair is often called the opening tag (or start tag), and the second tag is called the closing tag (or end tag).

An opening tag and a closing tag are identical, except a slash (/) after the opening angle bracket of the closing tag, to tell the browser that the command has been completed.

**Example:**

**<!DOCTYPE html>**

**<html lang="en">**

**<head>**

**<title>A simple HTML document</title>**

**</head>**

**<body>**

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

**<!-- Paragraph with nested element -->**

**<p>**

**This is <b>another</b> paragraph.**

**</p>**

**</body>**

**</html>**

**HTML TAGS:**

1. <html> </html> [HTML open / close tags]
2. <h1> </h1> [Heading 1 open / close tags]
3. <h2> </h2> [Heading 2 open / close tags]
4. <h3> </h3> [Heading 3 open / close tags]
5. <h4> </h4> [Heading 4 open / close tags]
6. <h5> </h5> [Heading 5 open / close tags]
7. <h6> </h6> [Heading 6 open / close tags]
8. <title> </title> [Title open/ close tags]
9. <center> </center> [Center alignment open/close tags]
10. <body> </body> [Body open/ close tags]
11. <body bgcolor=”colorname”>
12. <body background=”image.jpg”>
13. <input type=”text” > [input type tags]
14. <input type=”number”>
15. <input type=”radio”>
16. <input type=”checkbox”>
17. <input type=”password”>
18. <input type=”button”>
19. <input type=”submit”>
20. <input type=”reset”>
21. <input type=”date”>
22. <input type=”email”>
23. <input type=”file”>
24. <input type=”color”>
25. <img src=”name.jpg”>
26. <marquee> Type text </marquee>
27. <p> </p> [Paragraph tag]
28. <q> </q> [Quotation tag]
29. <table> </table> [Table tag]
30. <tr> </tr> [Table row]
31. <td> </td> [Table data]
32. <ul> </ul> [Unordered List]
33. <li> </li> [List]
34. <u> </u> [Underline]
35. <br> </br> [Line Break]
36. <hr> [horizontal row]
37. <del> </del> [Strikethrough]
38. <sup> </sup> [Super Script]
39. <sub> </sub> [Sub Script]
40. <select> [Drop down List tag]  
       <option value="Volvo">Volvo</option>  
       <option value="Saab">Saab</option>  
       <option value="Mercedes">Mercedes</option>  
       <option value="Audi">Audi</option>  
     </select>
41. <a href=”name.html”>Click Here</a> [anchor tag]
42. <font size="3" color="red">This is some text!</font> [Font tags]

<font size="2" color="blue">This is some text! </font>  
<font face="verdana" color="green">This is some text! </font>

1. <textarea rows="4" cols="50"> ………………..</textarea> [Text area]
2. <form action=”pagename.html”> </form> [Action tag]

**HTML5**

HTML5 is the latest and most enhanced version of HTML. Technically, HTML is not a programming language, but rather a markup language.

HTML5 is the next major revision of the HTML standard superseding HTML 4.01, XHTML 1.0, and XHTML 1.1. HTML5 is a standard for structuring and presenting content on the World Wide Web.

HTML5 is a cooperation between the World Wide Web Consortium (W3C) and the Web Hypertext Application Technology Working Group (WHATWG).

The new standard incorporates features like video playback and drag-and-drop that have been previously dependent on third-party browser plug-ins such as Adobe Flash, Microsoft Silverlight, and Google Gears.

## Browser Support:

The latest versions of Apple Safari, Google Chrome, Mozilla Firefox, and Opera all support many HTML5 features and Internet Explorer 9.0 will also have support for some HTML5 functionality.

The mobile web browsers that come pre-installed on iPhones, iPads, and Android phones all have excellent support for HTML5.

**New Features**:HTML5 introduces a number of new elements and attributes that can help you in building modern websites.

* **New Semantic Elements** − These are like <header>, <footer>, and <section>.
* **Forms 2.0** − Improvements to HTML web forms where new attributes have been introduced for <input> tag.
* **Persistent Local Storage** − To achieve without resorting to third-party plugins.
* **WebSocket** − A next-generation bidirectional communication technology for web applications.
* **Server-Sent Events** − HTML5 introduces events which flow from web server to the web browsers and they are called Server-Sent Events (SSE).
* **Canvas** − This supports a two-dimensional drawing surface that you can program with JavaScript.
* **Audio & Video** − You can embed audio or video on your webpages without resorting to third-party plugins.
* **Geolocation** − Now visitors can choose to share their physical location with your web application.
* **Microdata** − This lets you create your own vocabularies beyond HTML5 and extend your web pages with custom semantics.
* **Drag and drop** − Drag and drop the items from one location to another location on the same webpage.

**Why use HTML5:**It is enriched with advance features which makes it easy and interactive for designer/developer and users.

It allows you to play a video and audio file.

It allows you to draw on a canvas.

It facilitate you to design better forms and build web applications that work offline.

It provides you advance features for that you would normally have to write JavaScript to do.

The most important reason to use HTML 5 is, we believe it is not going anywhere. It will be here to serve for a long time according to W3C recommendation.

**HTML5 Syntax:**

The HTML 5 language has a "custom" HTML syntax that is compatible with HTML 4 and XHTML1 documents published on the Web, but is not compatible with the more esoteric SGML features of HTML 4.

HTML 5 does not have the same syntax rules as XHTML where we needed lower case tag names, quoting our attributes, an attribute had to have a value and to close all empty elements.

HTML5 comes with a lot of flexibility and it supports the following features −

* Uppercase tag names.
* Quotes are optional for attributes.
* Attribute values are optional.
* Closing empty elements are optional.

**The DOCTYPE:**

DOCTYPEs in older versions of HTML were longer because the HTML language was SGML based and therefore required a reference to a DTD.

 DOCTYPE declaration appears at the top of a web page before all other elements. According to the HTML specification or standards, every HTML document requires a valid document type declaration to insure that your web pages are displayed the way they are intended to be displayed.

**syntax:**

**<!DOCTYPE html>**

* The above syntax is case-insensitive.

# HTML Meta:

* The <meta> tags are typically used to provide structured metadata such as a document's keywords, description, author name, character encoding, and other metadata. Any number of meta tags can be placed inside the [head section](https://www.tutorialrepublic.com/html-tutorial/html-head.php) of an HTML or XHTML document.
* Metadata will not be displayed on the web page, but will be machine parsable, and can be used by the browsers, search engines like Google or other web services.

**syntax:**

<meta charset = "UTF-8">

The above syntax is case-insensitive.

**Declaring Character Encoding in HTML:**

<!DOCTYPE html>

<html lang="en">

<head>

<title>Declaring Character Encoding</title>

<meta charset="utf-8">

</head>

<body>

<h1>Hello World!</h1>

</body>

</html>

## The <script> tag

It's common practice to add a type attribute with a value of "text/javascript" to script elements as follows −

<script type = "text/javascript" src = "scriptfile.js"></script>

HTML 5 removes extra information required and you can use simply following syntax −

<script src = "scriptfile.js"></script>

## The <link> tag

So far we were writing <link> as follows −

<link rel = "stylesheet" type = "text/css" href = "stylefile.css">

HTML 5 removes extra information required and you can simply use the following syntax −

<link rel = "stylesheet" href = "stylefile.css">

# HTML 5 Tags:

There is a list of newly included tags in HTML 5. These HTML 5 tags (elements) provide a better document structure. This list shows all HTML 5 tags in alphabetical order with description.

## List of HTML 5 Tags

|  |  |
| --- | --- |
| **Tag** | **Description** |
| <article> | This element is used to define an independent piece of content in a document, that may be a blog, a magazine or a newspaper article. |
| <aside> | It specifies that article is slightly related to the rest of the whole page. |
| <audio> | It is used to play audio file in HTML. |
| <bdi> | The bdi stands for bi-directional isolation. It isolates a part of text that is formatted in other direction from the outside text document. |
| <canvas> | It is used to draw canvas. |
| <data> | It provides machine readable version of its data. |
| <datalist> | It provides auto complete feature for textfield. |
| <details> | It specifies the additional information or controls required by user. |
| <dialog> | It defines a window or a dialog box. |
| <figcaption> | It is used to define a caption for a <figure> element. |
| <figure> | It defines a self-contained content like photos, diagrams etc. |
| <footer> | It defines a footer for a section. |
| <header> | It defines a header for a section. |
| <main> | It defines the main content of a document. |
| <mark> | It specifies the marked or highlighted content. |
| <menuitem> | It defines a command that the user can invoke from a popup menu. |
| <meter> | It is used to measure the scalar value within a given range. |
| <nav> | It is used to define the navigation link in the document. |
| <progress> | It specifies the progress of the task. |
| <rp> | It defines what to show in browser that don't support ruby annotation. |
| <rt> | It defines an explanation/pronunciation of characters. |
| <ruby> | It defines ruby annotation along with <rp> and <rt>. |
| <section> | It defines a section in the document. |
| <summary> | It specifies a visible heading for <detailed> element. |
| <svg> | It is used to display shapes. |
| <time> | It is used to define a date/time. |
| <video> | It is used to play video file in HTML. |
| <wbr> | It defines a possible line break. |

|  |  |
| --- | --- |
| **HTML5 Form Tags** | |
|  |  |
| <datalist> | It represent predefined list for input <option> element. |
| <output> | It is used a container element to represent the output of a calculation or outcome of user action. |
| Graphics Tags | |
|  |  |
| <canvas> | It allows drawing graphics and animations via scripting. |
| <svg> | It is used to draw scalable vector graphics. |
| **HTML5 Media Tags** | |
|  |  |
| <audio> | It defines sound content. |
| <embed> | It defines a container for external files/application/media. |
| <source> | It defines multiple media resources for the media elements. |
| <track> | It defines text tracks for <audio> and <video> files |
| <video> | It defines video content within HTML document. |

## HTML5 New <input> types:

|  |  |
| --- | --- |
| **Type** | **Description** |
| color | It represents an input field which defines a color selector. |
| date | It represents an input field to define a date selector. |
| datetime | It defines full date and time display with time zone information. |
| datetime-local | It defines date and time without time zone information. |
| email | It defines an input field with email pattern Validation. |
| month | It defines the input field to enter month for the particular year |
| number | It defines field which selects a numeric value only. |
| range | It defines a numeric value selector with a given range of 1 to 100. |
| search | It is used to define a search field. |
| tel | It represents a control to enter a telephone number. |
| time | It represents a control to enter time value with no time zone. |
| url | It represents an input field to enter a URL |
| week | It defines a selector for week value for the particular year. |

## HTML5 Elements:

HTML5 elements are marked up using start tags and end tags. Tags are delimited using angle brackets with the tag name in between.

The difference between start tags and end tags is that the latter includes a slash before the tag name.

Following is the example of an HTML5 element −

<p>...</p>

HTML5 tag names are case insensitive and may be written in all uppercase or mixed case, although the most common convention is to stick with lowercase.

Most of the elements contain some content like <p>...</p> contains a paragraph. Some elements, however, are forbidden from containing any content at all and these are known as void elements. For example, **br, hr, link, meta**, etc.

## HTML5 Attributes

Elements may contain attributes that are used to set various properties of an element.

Some attributes are defined globally and can be used on any element, while others are defined for specific elements only.

The example of an HTML5 attribute which illustrates how to mark up a div element with an attribute named class using a value of "example" −

<div class = "example">...</div>

Attributes may only be specified within start tags and must never be used in end tags.

HTML5 attributes are case insensitive and may be written in all uppercase or mixed case, although the most common convention is to stick with lowercase.

## Global Attributes in HTML5:

In addition to the element specific attributes, the HTML5 defines few attributes that are common to all elements. These attributes may be specified on all elements, with some exceptions where it is not relevant, such as elements found inside the [<head>](https://www.tutorialrepublic.com/html-reference/html-head-tag.php) section of the document, e.g. [<base>](https://www.tutorialrepublic.com/html-reference/html-base-tag.php), [<script>](https://www.tutorialrepublic.com/html-reference/html-script-tag.php), [<title>](https://www.tutorialrepublic.com/html-reference/html-title-tag.php) etc.

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Value** | **Description** |
| accesskey | *shortcut key* | Specifies a keyboard shortcut to activate or focus the element. |
| class | *classname* | Assigns a class name or space-separated list of class names to an element. |
| contenteditable | true false | Indicates whether the content of an element is editable by the user or not. |
| contextmenu | *menu-id* | Specifies a context menu for an element. A context menu is a menu that appears when the user clicks the right mouse button on the element. |
| data-\* | *data* | Specified on any HTML element, to store custom data specific to the page. |
| dir | ltr rtl | Specifies the base direction of directionality of the element's text. |
| draggable | true false | Specifies whether an element is draggable or not. |
| dropzone | copy move link | Specifies whether the dragged data is copied, moved, or linked, when dropped. |
| hidden | hidden | Indicates that the element is not yet, or is no longer, relevant. |
| id | *name* | Specifies a unique identifier (ID) for an element which must be unique in the whole document. |
| lang | *language-code* | Specifies the primary language for the element's text content. |
| spellcheck | true false | Specifies whether the element may be checked for spelling errors or not. |
| style | *style* | Specifies inline style information for an element. |
| tabindex | *number* | Specifies the tabbing order of an element. |
| title | *text* | Provides advisory information related to the element. It would be appropriate for a tooltip. |
| translate | yes no | Specifies whether the text content of an element should be translated or not. |
| xml:lang | *language-code* | Specifies the primary language for the element's text content, in XHTML documents. |

## Event Attributes in HTML5

The following event attributes can be applied to the most of the elements for the execution of JavaScript when certain events occur, with some exceptions where it is not relevant such as elements found inside the [<head>](https://www.tutorialrepublic.com/html-reference/html-head-tag.php) section, e.g. [<title>](https://www.tutorialrepublic.com/html-reference/html-title-tag.php), [<base>](https://www.tutorialrepublic.com/html-reference/html-base-tag.php), [<link>](https://www.tutorialrepublic.com/html-reference/html-link-tag.php) etc.

## Window Events

Events related to the window object (applies to the [<body>](https://www.tutorialrepublic.com/html-reference/html-body-tag.php) tag):

| **Attribute** | **Value** | **Description** |
| --- | --- | --- |
| Onafterprint | *script* | Fires after the associated document is printed. |
| Onbeforeprint | *script* | Fires before the associated document is printed. |
| Onbeforeunload | *script* | Fires before a document being unloaded. |
| Onerror | *script* | Fires when document errors occur. |
| Onhashchange | *script* | Fires when the fragment identifier part of the document's URL i.e. the portion of a URL that follows the sign (#) changes. |
| Onload | *script* | Fires when the document has finished loading. |
| Onmessage | *script* | Fires when the message event occurs i.e. when user sends a cross-document message or a message is sent from a worker with postMessage() method. See [HTML5 Web Workers](https://www.tutorialrepublic.com/html-tutorial/html5-web-workers.php). |
| Onoffline | *script* | Fires when the network connection fails and the browser starts working offline. |
| Ononline | *script* | Fires when the network connections returns and the browser starts working online. |
| Onpagehide | *script* | Fires when the page is hidden, such as when a user is moving to another webpage. |
| Onpageshow | *script* | Fires when the page is shown, such as when a user navigates to a webpage. |
| Onpopstate | *script* | Fires when changes are made to the active history. |
| Onresize | *script* | Fires when the browser window is resized. |
| Onstorage | *script* | Fires when a [Web Storage](https://www.tutorialrepublic.com/html-tutorial/html5-web-storage.php) area is updated. |
| Onunload | *script* | Fires immediately before the document is unloaded or the browser window is closed. |

## Form Events

Events that occur due to the user interacting with the HTML form controls.

| **Attribute** | **Value** | **Description** |
| --- | --- | --- |
| Onblur | *script* | Fires when an element loses focus. |
| Onchange | *script* | Fires when the value or state of the element is changed. |
| Onfocus | *script* | Fires when the element receives focus. |
| Oninput | *script* | Fires when the value of an element is changed by the user. |
| Oninvalid | *script* | Fires when a submittable element do not satisfy their constraints during form validation. |
| Onreset | *script* | Fires when the user resets a form. |
| Onselect | *script* | Fires when some text is being selected or the current selection is changed by the user. |
| Onsearch | *script* | Fires when the user writes something in a [search input](https://www.tutorialrepublic.com/html-tutorial/html5-new-input-types.php#search-input) field. |
| Onsubmit | *script* | Fires when a form is submitted. |

## Mouse Events

Events that occur due to the user interacting with a pointing device such as a mouse:

| **Attribute** | **Value** | **Description** |
| --- | --- | --- |
| Onclick | *script* | Fires when the user clicks the left mouse button on the element. |
| Ondblclick | *script* | Fires when the user double-clicks on the element. |
| Oncontextmenu | *script* | Fires when a context menu is triggered by the user through right-click on the element. |
| Ondrag | *script* | Fires when the user drags an element. The ondrag event fires throughout the drag operation. |
| Ondragend | *script* | Fires when the user releases the mouse button at the end of a drag operation. |
| Ondragenter | *script* | Fires when the user drags an element to a valid drop target. |
| Ondragleave | *script* | Fires when an element leaves a valid drop target during a drag operation. |
| Ondragover | *script* | Fires when an element is being dragged over a valid drop target. |
| Ondragstart | *script* | Fires when the user starts to drag a text selection or selected element. |
| Ondrop | *script* | Fires when the mouse button is released during a drag-and-drop operation i.e. when dragged element is being dropped. |
| Onmousedown | *script* | Fires when the mouse button is pressed over an element. |
| Onmousemove | *script* | Fires when the user moves the mouse pointer over an element. |
| Onmouseout | *script* | Fires when the user moves the mouse pointer outside the boundaries of an element. |
| Onmouseover | *script* | Fires when the user moves the mouse pointer onto an element. |
| Onmouseup | *script* | Fires when the user releases the mouse button while the mouse is over an element. |
| Onmousewheel | *script* | Use the onwheel attribute instead. |
| Onscroll | *script* | Fires when the user scrolls the contents of an element by scrolling the element's scrollbar. |
| Onshow | *script* | Fires when a [contextmenu](https://www.tutorialrepublic.com/html-reference/html5-event-attributes.php#contextmenu) event was fired onto an element that has a contextmenu attribute. |
| Ontoggle | *script* | Fires when the user opens or closes the [<details>](https://www.tutorialrepublic.com/html-reference/html5-details-tag.php) element. |
| Onwheel | *script* | Fires when the user scrolls the contents of an element by rolling the mouse wheel up or down over an element. |

## Keyboard Events

Events that occur by the user interaction with the keyboard:

| **Attribute** | **Value** | **Description** |
| --- | --- | --- |
| Onkeydown | *script* | Fires when the user presses a key. |
| Onkeypress | *script* | Fires when the user presses an alphanumeric key. |
| Onkeyup | *script* | Fires when the user releases a key. |

## Clipboard Events

Events related to modification of the clipboard, that is copy, cut and paste:

| **Attribute** | **Value** | **Description** |
| --- | --- | --- |
| Oncopy | *script* | Fires when the user copies the element or selection, adding it to the system clipboard. |
| Oncut | *script* | Fires when the element or selection is removed from the document and added to the system clipboard. |
| Onpaste | *script* | Fires when the user pastes data, transferring the data from the system clipboard to the document. |

## Media Events

Events that occur when handling media elements that are embedded inside the HTML documents, such as [<audio>](https://www.tutorialrepublic.com/html-reference/html5-audio-tag.php) and [<video>](https://www.tutorialrepublic.com/html-reference/html5-video-tag.php) elements:

| **Attribute** | **Value** | **Description** |
| --- | --- | --- |
| Onabort | *script* | Fires when playback is aborted, but not due to an error. |
| Oncanplay | *script* | Fires when enough data is available to play the media, at least for a couple of frames, but would require further buffering. |
| Oncanplaythrough | *script* | Fires when entire media can be played to the end without requiring to stop for further buffering. |
| Oncuechange | *script* | Fires when the text track cue in a [<track>](https://www.tutorialrepublic.com/html-reference/html5-track-tag.php) element changes. |
| Ondurationchange | *script* | Fires when the duration of the media changes. |
| Onemptied | *script* | Fires when the media element is reset to its initial state, either because of a fatal error during load, or because the load() method is called to reload it. |
| Onended | *script* | Fires when the end of playback is reached. |
| Onerror | *script* | Fires when an error occurs while fetching the media data. |
| Onloadeddata | *script* | Fires when media data is loaded at the current playback position. |
| onloadedmetadata | *script* | Fires when metadata of the media (like duration and dimensions) has finished loading. |
| Onloadstart | *script* | Fires when loading of the media begins. |
| Onpause | *script* | Fires when playback is paused, either by the user or programmatically. |
| Onplay | *script* | Fires when playback of the media starts after having been paused i.e. when the play() method is requested. |
| Onplaying | *script* | Fires when the audio or video has started playing. |
| Onprogress | *script* | Fires periodically to indicate the progress while downloading the media data. |
| Onratechange | *script* | Fires when the playback rate or speed is increased or decreased, like slow motion or fast forward mode. |
| Onseeked | *script* | Fires when the seek operation ends. |
| Onseeking | *script* | Fires when the current playback position is moved. |
| Onstalled | *script* | Fires when the download has stopped unexpectedly. |
| Onsuspend | *script* | Fires when the loading of the media is intentionally stopped. |
| Ontimeupdate | *script* | Fires when the playback position changed, like when the user fast forwards to a different playback position. |
| Onvolumechange | *script* | Fires when the volume is changed, or playback is muted or unmuted. |
| Onwaiting | *script* | Fires when playback stops because the next frame of a video resource is not available. |

**Difference between HTML and HTML5:**

|  |  |
| --- | --- |
| **HTML** | **HTML5** |
| HTML Doctype declaration is lengthy. | DOCTYPE declaration in HTML5 is simple. |
| HTML Character encoding is longer. | HTML5 Character encoding declaration is simple. |
| Audio and video are not HTML parts. | Audio and video are HTML5 part. |
| It is possible to draw a vector with the help of other technologies like Silverlight, Flash, VML, etc. | Vector graphics are a part of HTML5, e.g., canvas, SVG. |
| It is impossible to get the actual Geolocation of a person browsing any website. | JS Geolocation API in HTML5 enables you to identify the location of the user browsing any website. |
| HTML offers local storage instead of cookies. | Html5 uses cookies to store data. |
| In HTML, it is not possible to draw basic shapes. | In Html5, it is possible to draw basic shapes. |
| It allows you to run JavaScript in a browser. | It enables you to run JavaScript code in the background. |
| You can use HTML with all old browsers. | You can use HTML5 with all new browsers. |
| You can use browser cache as temporary storage. | You can use application (database and web storage) Cache as temporary storage. |
| Web Socket is not available. | You can establish full-duplex communication channels with a server using Web Sockets. |
| There is no process to handlestructurally incorrect HTML codes. | HTML5 supports persistent error handling via the improvised error handling process. |
| HTML is less mobile-friendly. | HTML5 is mobile friendly. |
| Attributes like async, charset, and ping are not present in HTML. | Attributes of async, ping, charset, and are a part of HTML5. |
| HTML does not allow drag and drop effects | HTML5 allows drag and drop effects. |
| Offer new attributes like tabinex, id, tabinex, etc. | These are certain attributes which are applied to HTML 5 elements. |

**Video And Audio**

# HTML Video:

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

HTML 5 supports <video> tag also. The HTML video tag is used for streaming video files such as a movie clip, song clip on the web page.

Currently, there are three video formats supported for HTML video tag:

1. mp4
2. webM
3. ogg

|  |  |  |  |
| --- | --- | --- | --- |
| **Browser** | **mp4** | **webM** | **ogg** |
| Internet Explorer | yes | no | no |
| Google Chrome | yes | yes | yes |
| Mozilla Firefox | yes | yes | yes |
| Opera | no | yes | yes |
| Apple Safari | yes | no | no |

## HTML Video Tag Example:

Let's see the code to play mp4 file using HTML video tag.

<!DOCTYPE>

<html>

<body>

<video controls>

<source src="movie.mp4" type="video/mp4">

</video>

</body>

</html>

## Attributes of HTML Video Tag:

Let's see the list of HTML 5 video tag attributes.

|  |  |
| --- | --- |
| **Attribute** | **Description** |
| controls | It defines the video controls which is displayed with play/pause buttons. |
| height | It is used to set the height of the video player. |
| width | It is used to set the width of the video player. |
| poster | It specifies the image which is displayed on the screen when the video is not played. |
| autoplay | It specifies that the video will start playing as soon as it is ready. |
| loop | It specifies that the video file will start over again, every time when it is completed. |
| muted | It is used to mute the video output. |
| preload | It specifies the author view to upload video file when the page loads. |
| src | It specifies the source URL of the video file. |

## HTML Video Tag Attribute Example

Let's see the example of video tag in HTML where are using height, width, autoplay, controls and loop attributes.

<!DOCTYPE>

<html>

<body>

<video width="320" height="240" controls autoplay loop>

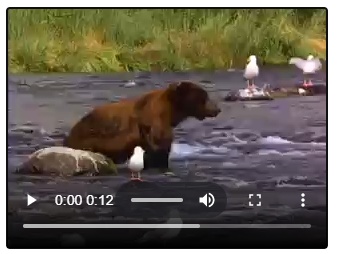
<source src="movie.mp4" type="video/mp4">

</video>

</body>

</html>

**Output:**



**HTML5 Audio Tag:**

**HTML audio tag** is used to define sounds such as music and other audio clips. Currently there are three supported file format for HTML 5 audio tag.

1. mp3
2. wav
3. ogg

HTML5 supports <video> and <audio> controls. The Flash, Silverlight and similar technologies are used to play the multimedia items.

This table defines that which web browser supports which audio file format.

|  |  |  |  |
| --- | --- | --- | --- |
| **Browser** | **mp3** | **wav** | **ogg** |
| Internet Explorer | yes | no | no |
| Google Chrome | yes | yes | yes |
| Mozilla Firefox | yes\* | yes | yes |
| Opera | no | yes | yes |
| Apple Safari | yes | yes | no |

## HTML Audio Tag Example:

## <!DOCTYPE>

## <html>

## <body>

## <audio controls>

## <source src="koyal.mp3" type="audio/mpeg">

## </audio>

## </body>

## </html>

## Attributes of HTML Audio Tag:

There is given a list of HTML audio tag.

|  |  |
| --- | --- |
| **Attribute** | **Description** |
| controls | It defines the audio controls which is displayed with play/pause buttons. |
| autoplay | It specifies that the audio will start playing as soon as it is ready. |
| loop | It specifies that the audio file will start over again, every time when it is completed. |
| muted | It is used to mute the audio output. |
| preload | It specifies the author view to upload audio file when the page loads. |
| src | It specifies the source URL of the audio file. |

## HTML Audio Tag Attribute Example:

Here we are going to use controls, autoplay, loop and src attributes of HTML audio tag.

<!DOCTYPE>

<html>

<body>

<audio controls autoplay loop>

<source src="koyal.mp3" type="audio/mpeg"></audio>

</body>

</html>

# HTML Canvas Tag:

The HTML element is used to draw graphics on the fly, via scripting (usually JavaScript). The element is only a container for graphics. You must use a script to actually draw the graphics. Canvas has several methods for drawing paths, boxes, circles, text, and adding images.

## How to create a HTML canvas?

A canvas is a rectangle like area on an HTML page. It is specified with canvas element. By default, the <canvas> element has no border and no content, it is like a container.

**<canvas** id = "mycanvas" width ="200" height ="100"**>** **</canvas>**

## HTML 5 Canvas Tag Example:

## <!DOCTYPE>

## <html>

## <body>

## <canvas id="myCanvas" width="300" height="200" style="border:2px solid;">

## </canvas>

## </body>

## </html>

## output:

## canvas op.png

## Supporting Browsers:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Element | Chrome | IE | Firefox | Opera | Safari |
| <canvas> | Yes | Yes | Yes | Yes | Yes |

## HTML Canvas Tag with JavaScript

A canvas is a two dimensional grid.

Coordinates (0,0) defines the upper left corner of the canvas. The parameters (0,0,200,100) is used for fillRect() method. This parameter will fill the rectangle start with the upper-left corner (0,0) and draw a 200 \* 100 rectangle.

1. **<canvas** id="myCanvas" width="250" height="150" style="border:1px solid #c3c3c3;"**>**
2. **</canvas>**
3. **<script>**
4. var c = document.getElementById("myCanvas");
5. var cctx = c.getContext("2d");
6. ctx.fillStyle = "#FF0000";
7. ctx.fillRect(0,0,200,100);
8. **</script>**

## Drawing Line on Canvas:

If you want to draw a straight line on the canvas, you can use the following two methods.

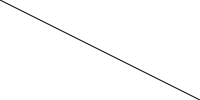
**moveTo(x,y):** It is used to define the starting point of the line.

**lineTo(x,y):** It is used to define the ending point of the line.

If you draw a line which starting point is (0,0) and the end point is (200,100), use the stroke method to draw the line.

1. **<canvas** id="myCanvasLine" width="200" height="100" style="border:1px solid #d3d3d3;"**>**
2. **</canvas>**
3. **<script>**
4. var c = document.getElementById("myCanvasLine");
5. var cctx = c.getContext("2d");
6. ctx.moveTo(0,0);
7. ctx.lineTo(200,100);
8. ctx.stroke();
9. **</script>**

1. **Output:**



## Drawing Circle on Canvas

If you want to draw a circle on the canvas, you can use the arc() method:

1. arc(x, y, r, start, stop)

To sketch circle on HTML canvas, use one of the ink() methods, like stroke() or fill().

1. **<canvas** id="myCanvasCircle" width="200" height="100" style="border:1px solid #d3d3d3;"**>**
2. .**</canvas>**
3. **<script>**
4. var c = document.getElementById("myCanvasCircle");
5. var cctx = c.getContext("2d");
6. ctx.beginPath();
7. ctx.arc(95,50,40,0,2\*Math.PI);
8. ctx.stroke();
9. **</script>**

**Output:**

## Drawing text on canvas

There are property and methods used for drawing text on the canvas.

**font property:** It is used to define the font property for the text.

**fillText(text,x,y) method:** It is used to draw filled text on the canvas. It looks like bold font.

**strokeText(text,x,y) method:** It is also used to draw text on the canvas, but the text is unfilled.

Let's see **fillText()** method example.

1. **<canvas** id="myCanvasText1" width="300" height="100" style="border:1px solid #d3d3d3;"**>**
2. .**</canvas>**
3. **<script>**
4. var c = document.getElementById("myCanvasText1");
5. var cctx = c.getContext("2d");
6. ctx.font = "30px Arial";
7. ctx.fillText("Hello JavaTpoint",10,50);
8. **</script>**

Let's see **strokeText()** method example.

1. **<canvas** id="myCanvasText2" width="300" height="100" style="border:1px solid #d3d3d3;"**>**
2. .**</canvas>**
3. **<script>**
4. var c = document.getElementById("myCanvasText2");
5. var cctx = c.getContext("2d");
6. ctx.font = "30px Arial";
7. ctx.strokeText("Hello JavaTpoint",10,50);
8. **</script>**

**Output:**

**HELLO WORLD**

**SVG**

The Scalable Vector Graphics (SVG) is an XML-based image format that is used to define two-dimensional vector-based graphics for the web. Unlike raster image (Ex .jpg, .gif, .png, etc.), a vector image can be scaled up or down to any extent without losing the image quality.

An SVG image is drawn out using a series of statements that follow the XML schema — that means SVG images can be created and edited with any text editor, such as Notepad. There are several other advantages of using SVG over other image formats like JPEG, GIF, PNG, etc.

* SVG images can be searched, indexed, scripted, and compressed.
* SVG images can be created and modified using JavaScript in real time.
* SVG images can be printed with high quality at any resolution.
* SVG content can be animated using the built-in animation elements.
* SVG images can contain [hyperlinks](https://www.tutorialrepublic.com/html-tutorial/html-links.php) to other documents.

## HTML SVG Circle Example:

## <!DOCTYPE html>

## <html>

## <body>

## <svg width="100" height="100">

## <circle cx="50" cy="50" r="40" stroke="yellow" stroke-width="4" fill="red" />

## </svg>

## </body>

## </html>

## Output:

**HTML SVG Rectangle:**

<!DOCTYPE html>

<html>

<body>

<svg width="200" height="100">

<rect width="200" height="100" stroke="yellow" stroke-width="4" fill="red" />

</svg>

</body>

</html>

**output:**

**HTML SVG polygon Example:**

<!DOCTYPE html>

<html>

<body>

<svg height="210" width="500">

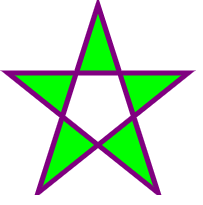
<polygon points="100,10 40,198 190,78 10,78 160,198"

style="fill:red;stroke:yellow;stroke-width:5;fill-rule:nonzero;" />

</svg>

</body>

</html>



**Difference between SVG and HTML5 Canvas:**

| SVG | Canvas |
| --- | --- |
| Vector based (composed of shapes) | Raster based (composed of pixel) |
| SVG has better scalability. So it can be printed with high quality at any resolution. | Canvas has poor scalability. Hence it is not suitable for printing on higher resolution. |
| SVG gives better performance with smaller number of objects or larger surface. | Canvas gives better performance with smaller surface or larger number of objects. |
| SVG can be modified through script and CSS. | Canvas can be modified through script only. |
| Multiple graphical elements, which become the part of the page’s DOM tree. | Single element similar to <img> in behavior. Canvas diagram can be saved to PNG or JPG format. |

## Web Storage

The Web Storage is one of the great features of HTML5. With the Web Storage feature, web applications can locally store data within the browser on the client side. It stores data in the form of key/value pair on the browser. Web Storage sometimes also known as DOM storage.

Storing data with the help of web storage is similar to cookies, but it is better and faster than cookies storage.

In compared to cookies Web Storage has Following Advantages:

* Web Storage can use storage space upto 5MB per domain. (The browser software may prompt the user if the space limit is reached).
* It will not send data to the server side, hence it is faster than cookies storage.
* The data stored by local Storage never expires, but cookies data expires after some time or session.
* Web Storage is more secure than cookies.

## Types of Web Storage:

There are two types of web storage with different scope and lifetime.

* **Local Storage:** Local Storages uses Windows.localStaorage object which stores data and available for every page. But data persist even if the browser is closed and reopened (Stores data with no Expiration).
* **Session Storage:** Session Storage uses Windows.sessionStorage object which stores data for one session and data will be lost if the window or browser tab will be closed.

## 1.HTML Web Storage Objects

HTML web storage provides two objects for storing data on the client:

* window.localStorage - stores data with no expiration date
* window.sessionStorage - stores data for one session (data is lost when the browser tab is closed)

Before using web storage, check browser support for localStorage and sessionStorage:

if (typeof(Storage) !== "undefined") {  
  // *Code for localStorage/sessionStorage.*  
} else {  
  // Sorry! No Web Storage support..  
}

**i) The localStorage Object:**

The localStorage object stores the data with no expiration date. The data will not be deleted when the browser is closed, and will be available the next day, week, or year.

**Example:**

<!DOCTYPE html>

<html>

<body>

<div id="result"></div>

<script>

// Check browser support

if (typeof(Storage) !== "undefined") {

// Store

localStorage.setItem("lastname", "Smith");

// Retrieve

document.getElementById("result").innerHTML = localStorage.getItem("lastname");

} else {

document.getElementById("result").innerHTML = "Sorry, your browser does not support Web Storage...";

}

</script>

</body>

</html>

**Output:** Smith

**Example 2:**

<!DOCTYPE html>

<html>

<head>

<title>Web Storage API</title>

<style>

body{

color: green;

text-align: center;

font-size: 30px;

margin-top: 30px;

font-style: italic;

}

</style>

</head>

<body>

<script>

if(typeof(Storage)!=="undefined") {

localStorage.setItem("name","Harshita");

localStorage.setItem("Country", "India");

document.write("Hi"+" "+localStorage.name+" "+"from" +" "+ localStorage.Country);

}

else{

alert("Sorry! your browser is not supporting the browser")

}

</script>

</body>

</html>

**Output:** *Hi Harshita from India*

## ii) The sessionStorage Object:

The sessionStorage object is equal to the localStorage object, **except** that it stores the data for only one session. The data is deleted when the user closes the specific browser tab.

The following example counts the number of times a user has clicked a button, in the current session:

<!DOCTYPE html>

<html>

<head>

<script>

function clickCounter() {

if (typeof(Storage) !== "undefined") {

if (sessionStorage.clickcount) {

sessionStorage.clickcount = Number(sessionStorage.clickcount)+1;

} else {

sessionStorage.clickcount = 1;

}

document.getElementById("result").innerHTML = "You have clicked the button " + sessionStorage.clickcount + " time(s) in this session.";

} else {

document.getElementById("result").innerHTML = "Sorry, your browser does not support web storage...";

}

}

</script>

</head>

<body>

<p><button onclick="clickCounter()" type="button">Click me!</button></p>

<div id="result"></div>

<p>Click the button to see the counter increase.</p>

<p>Close the browser tab (or window), and try again, and the counter is reset.</p>

**Output:**

Click me!

Click the button to see the counter increase.

Close the browser tab (or window), and try again, and the counter is reset.

## 2. Remove Web Storage:

As we have seen the session storage data will automatically be deleted, when you close the browser but the data saved by local storage will remain in the browser even if you close it.

Hence to delete the local storage data, you need to call two methods:

* **localStorage.removeItem('key'):** If you want to delete the value on a particular key, then you can use the "key," and that value will be deleted.
* **localStorage.clear():** If you want to delete or clear all settings with key/value pair, then you can call this method.

### Example:

<!DOCTYPE html>

<html>

<head>

<title>Web Storage API</title>

<style>

body{

color: green;

text-align: center;

font-size: 30px;

margin-top: 30px;

font-style: italic;

}

</style>

</head>

<body>

<button onclick="remove();">Remove item</button>

<div id="output"></div>

<script>

if(typeof(Storage)!=="undefined") {

localStorage.setItem("name","Harshita");

localStorage.setItem("Country", "India");

document.getElementById('output').innerHTML= "Hii, my name is"+ " "+ localStorage.name +" "+"and i belongs to"+" "+localStorage.Country;

}

else{

alert("Sorry! your browser is not supporting the browser")

}

function remove() {

localStorage.removeItem("name");

document.getElementById('output').innerHTML= "Hii, my name is"+ " "+ localStorage.name +" "+"and i belongs to"+" "+localStorage.Country;

}

</script>

</body>

</html>

**Output:**

**Remove item**

*Hii, my name is undefined and i belongs to India*

# HTML Drag and Drop

**HTML Drag and Drop** (DnD) is a feature of HTML5. It is a powerful user interface concept which *is used to copy, reorder and delete items with the help of mouse*. You can hold the mouse button down over an element and drag it to another location. If you want to drop the element there, just release the mouse button.

Drag and drop is a very common feature. It is when you "grab" an object and drag it to a different location. HTML 5 DnD is supported by all the major browsers like Chrome, Firefox 3.5 and Safari 4.

## Drag and Drop Events:

There are number of events which are fired during various stages of the drag and drop operation. These events are listed below −

|  |  |
| --- | --- |
| **Sr.No.** | **Events & Description** |
| 1 | **dragstart**  Fires when the user starts dragging of the object. |
| 2 | **dragenter**  Fired when the mouse is first moved over the target element while a drag is occurring. A listener for this event should indicate whether a drop is allowed over this location. If there are no listeners, or the listeners perform no operations, then a drop is not allowed by default. |
| 3 | **dragover**  This event is fired as the mouse is moved over an element when a drag is occurring. Much of the time, the operation that occurs during a listener will be the same as the dragenter event. |
| 4 | **dragleave**  This event is fired when the mouse leaves an element while a drag is occurring. Listeners should remove any highlighting or insertion markers used for drop feedback. |
| 5 | **drag**  Fires every time the mouse is moved while the object is being dragged. |
| 6 | **drop**  The drop event is fired on the element where the drop was occurred at the end of the drag operation. A listener would be responsible for retrieving the data being dragged and inserting it at the drop location. |
| 7 | **dragend**  Fires when the user releases the mouse button while dragging an object. |

## The DataTransfer Object:

The event listener methods for all the drag and drop events accept **Event** object which has a readonly attribute called **dataTransfer**.

The **event.dataTransfer** returns **DataTransfer** object associated with the event as follows −

function EnterHandler(event) {

DataTransfer dt = event.dataTransfer;

.............

}

The *DataTransfer* object holds data about the drag and drop operation. This data can be retrieved and set in terms of various attributes associated with DataTransfer object as explained below −

|  |  |
| --- | --- |
| **Sr.No.** | **DataTransfer attributes and their description** |
| 1 | **dataTransfer.dropEffect [ = value ]**   * Returns the kind of operation that is currently selected. * This attribute can be set, to change the selected operation. * The possible values are none, copy, link, and move. |
| 2 | **dataTransfer.effectAllowed [ = value ]**   * Returns the kinds of operations that are to be allowed. * This attribute can be set, to change the allowed operations. * The possible values are **none, copy, copyLink, copyMove, link, linkMove, move, all** and **uninitialized**. |
| 3 | **dataTransfer.types**  Returns a DOMStringList listing the formats that were set in the dragstart event. In addition, if any files are being dragged, then one of the types will be the string "Files". |
| 4 | **dataTransfer.clearData ( [ format ] )**  Removes the data of the specified formats. Removes all data if the argument is omitted. |
| 5 | **dataTransfer.setData(format, data)**  Adds the specified data. |
| 6 | **data = dataTransfer.getData(format)**  Returns the specified data. If there is no such data, returns the empty string. |
| 7 | **dataTransfer.files**  Returns a FileList of the files being dragged, if any. |
| 8 | **dataTransfer.setDragImage(element, x, y)**  Uses the given element to update the drag feedback, replacing any previously specified feedback. |
| 9 | **dataTransfer.addElement(element)**  Adds the given element to the list of elements used to render the drag feedback. |

## Drag and Drop Process:

Following are the steps to be carried out to implement Drag and Drop operation −

### Step 1 - Making an Object Draggable:

Here are steps to be taken −

* If you want to drag an element, you need to set the **draggable** attribute to **true** for that element.
* Set an event listener for **dragstart** that stores the data being dragged.
* The event listener **dragstart** will set the allowed effects (copy, move, link, or some combination).

Following is the example to make an object draggable −

<!DOCTYPE HTML>

<html>

<head>

<style type = "text/css">

#boxA, #boxB {

float:left;padding:10px;margin:10px; -moz-user-select:none;

}

#boxA { background-color: #6633FF; width:75px; height:75px; }

#boxB { background-color: #FF6699; width:150px; height:150px; }

</style>

<script type = "text/javascript">

function dragStart(ev) {

ev.dataTransfer.effectAllowed = 'move';

ev.dataTransfer.setData("Text", ev.target.getAttribute('id'));

ev.dataTransfer.setDragImage(ev.target,0,0);

return true;

}

</script>

</head>

<body>

<center>

<h2>Drag and drop HTML5 demo</h2>

<div>Try to drag the purple box around.</div>

<div id = "boxA" draggable = "true"

ondragstart = "return dragStart(ev)">

<p>Drag Me</p>

</div>

<div id = "boxB">Dustbin</div>

</center>

</body>

</html>

**Output:**

## Drag and drop HTML5 demo

Try to drag the purple box around.

Drag Me

Dustbin

### Step 2 - Dropping the Object:

To accept a drop, the drop target has to listen to at least three events.

* The **dragenter** event, which is used to determine whether or not the drop target is to accept the drop. If the drop is to be accepted, then this event has to be canceled.
* The **dragover** event, which is used to determine what feedback is to be shown to the user. If the event is canceled, then the feedback (typically the cursor) is updated based on the dropEffect attribute's value.
* Finally, the **drop** event, which allows the actual drop to be performed.

Following is the example to drop an object into another object −

<html>

<head>

<style type="text/css">

#boxA, #boxB {

float:left;padding:10px;margin:10px;-moz-user-select:none;

}

#boxA { background-color: #6633FF; width:75px; height:75px; }

#boxB { background-color: #FF6699; width:150px; height:150px; }

</style>

<script type="text/javascript">

function dragStart(ev) {

ev.dataTransfer.effectAllowed='move';

ev.dataTransfer.setData("Text", ev.target.getAttribute('id'));

ev.dataTransfer.setDragImage(ev.target,0,0);

return true;

}

function dragEnter(ev) {

event.preventDefault();

return true;

}

function dragOver(ev) {

return false;

}

function dragDrop(ev) {

var src = ev.dataTransfer.getData("Text");

ev.target.appendChild(document.getElementById(src));

ev.stopPropagation();

return false;

}

</script>

</head>

<body>

<center>

<h2>Drag and drop HTML5 demo</h2>

<div>Try to move the purple box into the pink box.</div>

<div id="boxA" draggable="true" ondragstart="return dragStart(event)">

<p>Drag Me</p>

</div>

<div id="boxB" ondragenter="return dragEnter(event)" ondrop="return dragDrop(event)" ondragover="return dragOver(event)">Dustbin</div>

</center>

</body>

</html>

## Output:

## Drag and drop HTML5 demo

Try to move the purple box into the pink box.

Drag Me

Dustbin

**Stages during Drag and Drop operations:**

**1) Make an element draggable:**

If you want to make an element draggable, set the draggable attribute to "true" on the element. For example:

**<img** draggable = "true"**>**

**2) What to drag:**

Use ondragstart and setData () methods.

Specify what should happen when the element is dragged.

**3) Where to Drop:**

Use ondragover event.

**4) Do the Drop:**

Use ondrop event.

# HTML5 Geolocation

The Geolocation is one of the best HTML5 API which is used to identify the user's geographic location for the web application.

This new feature of HTML5 allows you to navigate the latitude and longitude coordinates of the current website's visitor. These coordinates can be captured by JavaScript and send to the server which can show your current location on the website

Most of the geolocation services use Network routing addresses such as IP addresses, RFID, WIFI and MAC addresses or internal GPS devices to identify the user's location.

## User privacy:

The user's location is the privacy concern, so geolocation API protects the user's privacy by taking the user's permission before getting the location. Geolocation API sends a notification prompt box which user can allow or deny, and if the user allows then only his location will be identified.

## Locate the User's Position:

The HTML Geolocation API is used to get the geographical position of a user.

Since this can compromise privacy, the position is not available unless the user approves it.

**Geolocation object**

The Geolocation API is work with the navigation.geolocation object. Its read-only property returns a Geolocation object which identifies the location of the user and can generate a customized result based on user location.

### Syntax:

geo=navigator. geolocation;

If this object is present, then you can get the geolocation services.

## Geolocation Methods:

The Geolocation API uses three methods of Geolocation interface which are given following:

|  |  |
| --- | --- |
| **Methods** | **Description** |
| getCurrentPosition() | It identifies the device or the user's current location and returns a position object with data. |
| watchPosition() | Return a value whenever the device location changes. |
| clearWatch() | It cancels the previous watchPosition() call |

## Checking for browser support:

The geolocation property of navigator.geolcation object helps to determine the browser support for the Geolocation API.

**Example:**

<!DOCTYPE html>

<html>

<head>

<title>Geolocation API</title>

</head>

<body>

<h1>Find your Current location</h1>

<button onclick="getlocation()">Click me</button>

<div id="location"></div>

<script>

var x= document.getElementById("location");

function getlocation() {

if(navigator.geolocation){

alert("your browser is supporting Geolocation API")

}

else

{

alert("Sorry! your browser is not supporting")

}

}

</script>

</body>

</html>

# Output: Find your Current location

Click me

## Using HTML Geolocation:

The getCurrentPosition() method is used to return the user's position.

The example below returns the latitude and longitude of the user's position:

<!DOCTYPE html>

<html>

<body>

<p>Click the button to get your coordinates.</p>

<button onclick="getLocation()">Try It</button>

<p id="demo"></p>

<script>

var x = document.getElementById("demo");

function getLocation() {

if (navigator.geolocation) {

navigator.geolocation.getCurrentPosition(showPosition);

}

else {

x.innerHTML = "Geolocation is not supported by this browser.";

}

}

function showPosition(position) {

x.innerHTML = "Latitude: " + position.coords.latitude +

"<br>Longitude: " + position.coords.longitude;

}

</script>

</body>

</html>

**Output:** Click the button to get your coordinates.

Try it

Latitude: 17.6028245  
Longitude: 78.48654909999999

**Example explained:**

* Check if Geolocation is supported
* If supported, run the getCurrentPosition() method. If not, display a message to the user
* If the getCurrentPosition() method is successful, it returns a coordinates object to the function specified in the parameter (showPosition)
* The showPosition() function outputs the Latitude and Longitude

The example above is a very basic Geolocation script, with no error handling.

## Getting the User's current position:

To get the user's current location, getCurrentPosition() method of the navigator.geolocation object is used. This method accepts three parameters:

* **success:** A success callback function to get the location of the user
* **error:** An error callback function which takes "Position Error" object as input.
* **options:** It defines various options for getting the location.

## Handling Errors and Rejections:

The second parameter of the getCurrentPosition() method is used to handle errors. It specifies a function to run if it fails to get the user's location:

<!DOCTYPE html>

<html>

<body>

<p>Click the button to get your coordinates.</p>

<button onclick="getLocation()">Try It</button>

<p id="demo"></p>

<script>

var x = document.getElementById("demo");

function getLocation() {

if (navigator.geolocation) {

navigator.geolocation.getCurrentPosition(showPosition, showError);

} else {

x.innerHTML = "Geolocation is not supported by this browser.";

}

}

function showPosition(position) {

x.innerHTML = "Latitude: " + position.coords.latitude +

"<br>Longitude: " + position.coords.longitude;

}

function showError(error) {

switch(error.code) {

case error.PERMISSION\_DENIED:

x.innerHTML = "User denied the request for Geolocation."

break;

case error.POSITION\_UNAVAILABLE:

x.innerHTML = "Location information is unavailable."

break;

case error.TIMEOUT:

x.innerHTML = "The request to get user location timed out."

break;

case error.UNKNOWN\_ERROR:

x.innerHTML = "An unknown error occurred."

break;

}

}

</script>

</body>

</html>

**Output:**

**Click the button to get your coordinates.**

Try It

## Displaying the Result in a Map:

To display the result in a map, you need access to a map service, like Google Maps.

In the example below, the returned latitude and longitude is used to show the location in a Google Map (using a static image):

### Example:

function showPosition(position) {  
  var latlon = position.coords.latitude + "," + position.coords.longitude;  
  
  var img\_url = "https://maps.googleapis.com/maps/api/staticmap?center=  
  "+latlon+"&zoom=14&size=400x300&sensor=false&key=YOUR\_KEY";  
  
  document.getElementById("mapholder").innerHTML = "<img src='"+img\_url+"'>";  
}

## Location-specific Information

This page has demonstrated how to show a user's position on a map.

Geolocation is also very useful for location-specific information, like:

* Up-to-date local information
* Showing Points-of-interest near the user
* Turn-by-turn navigation (GPS)

## The getCurrentPosition() Method - Return Data:

The getCurrentPosition() method returns an object on success. The latitude, longitude and accuracy properties are always returned. The other properties are returned if available:

|  |  |
| --- | --- |
| **Property** | **Returns** |
| coords.latitude | The latitude as a decimal number (always returned) |
| coords.longitude | The longitude as a decimal number (always returned) |
| coords.accuracy | The accuracy of position (always returned) |
| coords.altitude | The altitude in meters above the mean sea level (returned if available) |
| coords.altitudeAccuracy | The altitude accuracy of position (returned if available) |
| coords.heading | The heading as degrees clockwise from North (returned if available) |
| coords.speed | The speed in meters per second (returned if available) |
| timestamp | The date/time of the response (returned if available) |

## Geolocation Object - Other interesting Methods:

The Geolocation object also has other interesting methods:

* watchPosition() - Returns the current position of the user and continues to return updated position as the user moves (like the GPS in a car).
* clearWatch() - Stops the watchPosition() method.

The example below shows the watchPosition() method. You need an accurate GPS device to test this (like smartphone):

**Example:**

<!DOCTYPE html>

<html>

<body>

<p>Click the button to get your coordinates.</p>

<button onclick="getLocation()">Try It</button>

<p id="demo"></p>

<script>

var x = document.getElementById("demo");

function getLocation() {

if (navigator.geolocation) {

navigator.geolocation.watchPosition(showPosition);

} else {

x.innerHTML = "Geolocation is not supported by this browser.";

}

}

function showPosition(position) {

x.innerHTML="Latitude: " + position.coords.latitude +

"<br>Longitude: " + position.coords.longitude;

}

</script>

</body>

</html>

**Output**

Click the button to get your coordinates.

Try It

**Basic Styling using CSS3**

Cascading Style Sheets (CSS) is a language that is used to illustrate the look, style, and format of a document written in any markup language. In simple words, it is used to style and organize the layout of Web pages. CSS3 is the latest version of an earlier CSS version, CSS2.

CSS3 is the latest standard for CSS.

CSS3 is completely backwards-compatible with earlier versions of CSS.

CSS3 has been split into "modules". It contains the "old CSS specification" (which has been split into smaller pieces).

Some of the most important CSS3 modules are:

* Selectors
* Box Model
* Backgrounds and Borders
* Image Values and Replaced Content
* Text Effects
* 2D/3D Transformations
* Animations
* Multiple Column Layout
* User Interface

### Features of CSS3:

The features of the CSS3 are as follows:

#### 1. Selectors

Selectors allow the designer to select on more precise levels of the web page. They are structural pseudo-classes that perform partial matches to help match attribute and attribute values. New selectors target a pseudo-class to style the elements targeted in the URL. Selectors also include a checked pseudo-class to style checked elements such as checkboxes and radio buttons.

#### 2. Text Effects and Layout

With CSS3, we can change the justification of text, whitespace adjustment of the document, and style the hyphenation of words.

#### 3. First-Letter and First-Line Pseudo-Classes

CSS 3 includes properties that help with kerning (adjusting the spacing between characters to achieve a visually pleasing effect) and positioning drop-caps (large decorative capital letter at the starting of a paragraph).

#### 4. Paged Media and Generated Content

CSS 3 has additional choices in Paged Media, such as page numbers and running headers and footers. There are additional properties for printing Generated Content as well, like properties for cross-references and footnotes.

#### 5. Multi-Column Layout

This feature includes properties to allow designers to present their content in multiple columns with options like the column-count, column-gap, and column-width.

### Advantages of CSS3

* CSS3 provides a consistent and precise positioning of navigable elements.
* It is easy to customize a web page as it can be done by merely altering a modular file.
* Graphics are easier in CSS3, thus making it easy to make the site appealing.
* It permits online videos to be seen without using third-party plug-ins.
* CSS3 is economical, time-saving, and most browsers support it.

## Including CSS in HTML Documents:

CSS can either be attached as a separate document or embedded in the HTML document itself. There are three methods of including CSS in an HTML document:

* **Inline styles** — Using the style attribute in the HTML start tag.
* **Embedded styles** — Using the [<style>](https://www.tutorialrepublic.com/html-reference/html-style-tag.php) element in the head section of a document.
* **External style sheets** — Using the [<link>](https://www.tutorialrepublic.com/html-reference/html-link-tag.php) element, pointing to an external CSS file.

**i) Inline Styles:**

Inline styles are used to apply the unique style rules to an element by putting the CSS rules directly into the start tag. It can be attached to an element using the style attribute.

The style attribute includes a series of CSS property and value pairs. Each "property: value" pair is separated by a semicolon (;), just as you would write into an embedded or external style sheets. But it needs to be all in one line i.e. no line break after the semicolon, as shown here:

**Example:** <!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<title>Example of CSS Inline Styles</title>

</head>

<body>

<h1 style="color:red; font-size:30px;">This is a heading</h1>

<p style="color:green; font-size:22px;">This is a paragraph.</p>

<div style="color:blue; font-size:14px;">This is some text content.</div>

</body>

</html>

**ii) Embedded Style Sheets:**

Embedded or internal style sheets only affect the document they are embedded in.

Embedded style sheets are defined in the [<head>](https://www.tutorialrepublic.com/html-tutorial/html-head.php) section of an HTML document using the [<style>](https://www.tutorialrepublic.com/html-reference/html-style-tag.php) element. You can define any number of <style> elements in an HTML document but they must appear between the <head> and </head> tags.

**Example:**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<title>Example of CSS Embedded Style Sheet</title>

<style>

body { background-color: YellowGreen; }

p { color: #fff; }

</style>

</head>

<body>

<h1>This is a heading</h1>

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

</body>

</html>

**iii) External Style Sheets:**

An external style sheet is ideal when the style is applied to many pages of the website.

An external style sheet holds all the style rules in a separate document that you can link from any HTML file on your site. External style sheets are the most flexible because with an external style sheet, you can change the look of an entire website by changing just one file.

An external style sheet can be linked to an HTML document using the [<link>](https://www.tutorialrepublic.com/html-reference/html-link-tag.php) tag. The <link> tag goes inside the [<head>](https://www.tutorialrepublic.com/html-reference/html-head-tag.php) section.

**Example:**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<title>Example of CSS External Style Sheet</title>

<link rel="stylesheet" href="/examples/css/style.css">

</head>

<body>

<h1>This is a heading</h1>

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

</body>

</html>

## Positioning And Background Images

## CSS Positioning Methods:

Positioning elements appropriately on the web pages is a necessity for a good layout design. There are several methods in CSS that you can use for positioning elements.

**i) Static Positioning:**

A static positioned element is always positioned according to the normal flow of the page. HTML elements are positioned static by default. Static positioned elements are not affected by the [top](https://www.tutorialrepublic.com/css-reference/css-top-property.php), [bottom](https://www.tutorialrepublic.com/css-reference/css-bottom-property.php), [left](https://www.tutorialrepublic.com/css-reference/css-left-property.php), [right](https://www.tutorialrepublic.com/css-reference/css-right-property.php), and [z-index](https://www.tutorialrepublic.com/css-reference/css-z-index-property.php) properties.

**Example:**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<title>Example of CSS Static Positioning</title>

<style>

.box{

color: #fff;

background: #7dc765;

padding: 20px;

}

.container{

padding: 50px;

margin: 50px;

position: relative;

border: 5px solid black;

font-family: Arial, sans-serif;

}

.container p{

line-height: 50px;

}

</style>

</head>

<body>

<div class="container">

<div class="box">

<h2>Static Positioned Box</h2>

<div><strong>Note:</strong> This box is positioned static, which is default. It is always positioned according to the normal flow of the page.</div>

</div>

<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nam eu sem tempor, varius quam at, luctus dui. Mauris magna metus, dapibus nec turpis vel, semper malesuada ante. Vestibulum id metus ac nisl bibendum scelerisque non non purus. Suspendisse varius nibh non aliquet sagittis. In tincidunt orci sit amet elementum vestibulum. Vivamus fermentum in arcu in aliquam. Quisque aliquam porta odio in fringilla. Vivamus nisl leo, blandit at bibendum eu, tristique eget risus. Integer aliquet quam ut elit suscipit, id interdum neque porttitor. Integer faucibus ligula.</p>

<p>Quis quam ut magna consequat faucibus. Pellentesque eget nisi a mi suscipit tincidunt. Ut tempus dictum risus. Pellentesque viverra sagittis quam at mattis. Suspendisse potenti. Aliquam sit amet gravida nibh, facilisis gravida odio. Phasellus auctor velit at lacus blandit, commodo iaculis justo viverra. Etiam vitae est arcu. Mauris vel congue dolor. Aliquam eget mi mi. Fusce quam tortor, commodo ac dui quis, bibendum viverra erat. Maecenas mattis lectus enim, quis tincidunt dui molestie euismod. Curabitur et diam tristique, accumsan nunc eu.</p>

</div>

</body></html>

**ii) Relative Positioning:**

A relative positioned element is positioned relative to its normal position.

In the relative positioning scheme the element's box position is calculated according to the normal flow. Then the box is shifted from this normal position according to the properties — top or bottom and/or left or right.

## <!DOCTYPE html>

## <html lang="en">

## <head>

## <meta charset="utf-8">

## <title>Example of CSS Relative Positioning</title>

## <style>

## .box{

## position: relative;

## left: 100px;

## color: #fff;

## background: #00c4cc;

## padding: 20px;

## }

## .container{

## padding: 50px;

## margin: 50px;

## border: 5px solid black;

## font-family: Arial, sans-serif;

## }

## .container p{

## line-height: 50px;

## }

## </style>

## </head>

## <body>

## <div class="container">

## <div class="box">

## <h2>Relative Positioned Box</h2>

## <div><strong>Note:</strong> The left margin edge of this DIV box is shifted to right by 100px from its original position. The whitespace generated is preserved.</div>

## </div>

## <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nam eu sem tempor, varius quam at, luctus dui. Mauris magna metus, dapibus nec turpis vel, semper malesuada ante. Vestibulum id metus ac nisl bibendum scelerisque non non purus. Suspendisse varius nibh non aliquet sagittis. In tincidunt orci sit amet elementum vestibulum. Vivamus fermentum in arcu in aliquam. Quisque aliquam porta odio in fringilla. Vivamus nisl leo, blandit at bibendum eu, tristique eget risus. Integer aliquet quam ut elit suscipit, id interdum neque porttitor. Integer faucibus ligula.</p>

## <p>Quis quam ut magna consequat faucibus. Pellentesque eget nisi a mi suscipit tincidunt. Ut tempus dictum risus. Pellentesque viverra sagittis quam at mattis. Suspendisse potenti. Aliquam sit amet gravida nibh, facilisis gravida odio. Phasellus auctor velit at lacus blandit, commodo iaculis justo viverra. Etiam vitae est arcu. Mauris vel congue dolor. Aliquam eget mi mi. Fusce quam tortor, commodo ac dui quis, bibendum viverra erat. Maecenas mattis lectus enim, quis tincidunt dui molestie euismod. Curabitur et diam tristique, accumsan nunc eu.</p>

## </div>

## </body>

## </html>

## iii) Absolute Positioning:

An absolutely positioned element is positioned relative to the first parent element that has a position other than static. If no such element is found, it will be positioned on a page relative to the 'top-left' corner of the browser window. The box's offsets further can be specified using one or more of the properties top, right, bottom, and left.

Absolutely positioned elements are taken out of the normal flow entirely and thus take up no space when placing sibling elements. However, it can overlap other elements depending on the [z-index](https://www.tutorialrepublic.com/css-reference/css-z-index-property.php) property value. Also, an absolutely positioned element can have [margins](https://www.tutorialrepublic.com/css-reference/css-margin-property.php), and they do not collapse with any other margins.

**Example:**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<title>Example of CSS Absolute Positioning</title>

<style>

.box{

position: absolute;

top: 200px;

left: 100px;

color: #fff;

width: 60%;

background: #4cafdf;

padding: 20px;

}

.container{

padding: 50px;

margin: 50px;

position: relative;

border: 5px solid black;

font-family: Arial, sans-serif;

}

.container p{

line-height: 50px;

}

</style>

</head>

<body>

<div class="container">

<div class="box">

<h2>Absolute Positioned Box</h2>

<div><strong>Note:</strong> This box is absolutely positioned relative to the container DIV element. It is scroll with the page.</div>

</div>

<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nam eu sem tempor, varius quam at, luctus dui. Mauris magna metus, dapibus nec turpis vel, semper malesuada ante. Vestibulum id metus ac nisl bibendum scelerisque non non purus. Suspendisse varius nibh non aliquet sagittis. In tincidunt orci sit amet elementum vestibulum. Vivamus fermentum in arcu in aliquam. Quisque aliquam porta odio in fringilla. Vivamus nisl leo, blandit at bibendum eu, tristique eget risus. Integer aliquet quam ut elit suscipit, id interdum neque porttitor. Integer faucibus ligula.</p>

<p>Quis quam ut magna consequat faucibus. Pellentesque eget nisi a mi suscipit tincidunt. Ut tempus dictum risus. Pellentesque viverra sagittis quam at mattis. Suspendisse potenti. Aliquam sit amet gravida nibh, facilisis gravida odio. Phasellus auctor velit at lacus blandit, commodo iaculis justo viverra. Etiam vitae est arcu. Mauris vel congue dolor. Aliquam eget mi mi. Fusce quam tortor, commodo ac dui quis, bibendum viverra erat. Maecenas mattis lectus enim, quis tincidunt dui molestie euismod. Curabitur et diam tristique, accumsan nunc eu.</p>

</div>

</body>

</html>

**iv) Fixed Positioning:**

Fixed positioning is a subcategory of absolute positioning.

The only difference is, a fixed positioned element is fixed with respect to the browser's [viewport](https://www.tutorialrepublic.com/definitions.php#viewport) and does not move when scrolled.

**Example:**

## <!DOCTYPE html>

## <html lang="en">

## <head>

## <meta charset="utf-8">

## <title>Example of CSS Fixed Positioning</title>

## <style>

## .box{

## position: fixed;

## top: 200px;

## left: 100px;

## color: #fff;

## width: 60%;

## background: #f44712;

## padding: 20px;

## }

## .container{

## padding: 50px;

## margin: 50px;

## position: relative;

## border: 5px solid black;

## font-family: Arial, sans-serif;

## }

## .container p{

## line-height: 50px;

## }

## </style>

## </head>

## <body>

## <div class="container">

## <div class="box">

## <h2>Fixed Positioned Box</h2>

## <div><strong>Note:</strong> The position of this box is fixed relative to the document's viewport. It doesn't scroll with the page.</div>

## </div>

## <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nam eu sem tempor, varius quam at, luctus dui. Mauris magna metus, dapibus nec turpis vel, semper malesuada ante. Vestibulum id metus ac nisl bibendum scelerisque non non purus. Suspendisse varius nibh non aliquet sagittis. In tincidunt orci sit amet elementum vestibulum. Vivamus fermentum in arcu in aliquam. Quisque aliquam porta odio in fringilla. Vivamus nisl leo, blandit at bibendum eu, tristique eget risus. Integer aliquet quam ut elit suscipit, id interdum neque porttitor. Integer faucibus ligula.</p>

## <p>Quis quam ut magna consequat faucibus. Pellentesque eget nisi a mi suscipit tincidunt. Ut tempus dictum risus. Pellentesque viverra sagittis quam at mattis. Suspendisse potenti. Aliquam sit amet gravida nibh, facilisis gravida odio. Phasellus auctor velit at lacus blandit, commodo iaculis justo viverra. Etiam vitae est arcu. Mauris vel congue dolor. Aliquam eget mi mi. Fusce quam tortor, commodo ac dui quis, bibendum viverra erat. Maecenas mattis lectus enim, quis tincidunt dui molestie euismod. Curabitur et diam tristique, accumsan nunc eu.</p>

## </div>

## </body>

## </html>

## Background Images: The CSS3 provides several new properties to manipulate the background of an element like background clipping, multiple backgrounds, and the option to adjust the background size.

## i) CSS3 background-size Property:

The background-size property can be used to specify the size of the background images.  Prior to CSS3, the size of the background images was determined by the actual size of the images. The background image size can be specified using the pixels or percentage values as well as the keywords auto, contain, and cover. Negative values are not allowed.

## Example:

## <!DOCTYPE html>

## <html lang="en">

## <head>

## <meta charset="utf-8">

## <title>Example of Setting background-size of an Element</title>

## <style>

## .box {

## width: 250px;

## height: 150px;

## background: url("/examples/images/sky.jpg") no-repeat;

## background-size: contain;

## border: 6px solid #333;

## }

## </style>

## </head>

## <body>

## <div class="box"></div>

## <p><strong>Note:</strong> The original size of the background image is 500x300 pixels, but using the background-size CSS property we are still able to show the complete image inside the smaller box.</p>

## </body>

## </html>

## ii) CSS3 background-clip Property:

The background-clip property can be used to specify whether an element's background extends into the border or not. The background-clip property can take the three values: border-box, padding-box, content-box.

## Example:

## <!DOCTYPE html>

## <html lang="en">

## <head>

## <meta charset="utf-8">

## <title>Example of CSS3 Background Clipping</title>

## <style>

## .box {

## width: 250px;

## height: 150px;

## padding: 10px;

## border: 6px dashed #333;

## background: orange;

## }

## .clip1 {

## background-clip: border-box;

## }

## .clip2 {

## background-clip: padding-box;

## }

## .clip3 {

## background-clip: content-box;

## }

## </style>

## </head>

## <body>

## <h2>Default Background Behavior</h2>

## <div class="box"></div>

## <h2>Background Clipping Using border-box</h2>

## <div class="box clip1"></div>

## <h2>Background Clipping Using padding-box</h2>

## <div class="box clip2"></div>

## <h2>Background Clipping Using content-box</h2>

## <div class="box clip3"></div>

## </body>

## </html>

## iii) CSS3 background-origin Property:

The background-origin property can be used to specify the positioning area of the background images. It can take the same values as background-clip property: border-box, padding-box, content-box.

**Example:**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<title>Example of Setting background-origin of an Element</title>

<style>

.box {

width: 250px;

height: 150px;

padding: 10px;

border: 6px dashed #333;

background: url("/examples/images/sky.jpg") no-repeat;

background-size: contain;

background-origin: content-box;

}

</style>

</head>

<body>

<div class="box"></div>

</body>

</html>

**iv) CSS3 Multiple Backgrounds:**

CSS3 gives you ability to add multiple backgrounds to a single element. The backgrounds are layered on the top of one another. The number of layers is determined by the number of comma-separated values in the [background-image](https://www.tutorialrepublic.com/css-reference/css-background-image-property.php) or [background](https://www.tutorialrepublic.com/css-reference/css-background-property.php) shorthand property.

**Example:**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<title>Example of CSS3 Multiple Backgrounds</title>

<style>

.box {

width: 100%;

height: 500px;

background: url("/examples/images/birds.png") no-repeat center, url("/examples/images/clouds.png") no-repeat center, url("/examples/images/sun.png") no-repeat 10% 30%, lightblue;

}

</style>

</head>

<body>

<div class="box"></div>

</body></html>

## BootStrap

## Bootstrap: Bootstrap 4 is the newest version of Bootstrap. [Bootstrap](https://www.javatpoint.com/bootstrap-tutorial) can be defined as a free and open-source framework that can be used to create responsive, mobile-first, front-end web pages.

### Advantages of Bootstrap:

1. The first and foremost advantage of using Bootstrap is that it is very easy to use and implement. If a person has some basic knowledge of HTML and CSS, that user can easily use Bootstrap.
2. The fact that Bootstrap can adapt to the size of any phone, tablet, desktop and so on is also very interesting feature.
3. Bootstrap 4 is also useful because it is compatible with all modern browsers which include Google Chrome, Firefox, Internet Explorer 10+, Edge, Safari, and Opera.
4. It also produces less cross-browser bugs.
5. It is light weighted and consequently it can be widely used as a framework for creating responsive sites.
6. Lastly, Bootstrap 4 is a very simple and yet very effective grid system.

**Usages of Bootstrap 4 :**

1. **Supported by various Browsers**: It can be supported by every browser.
2. **Simple to start and implement**: It is very easy to start and implement when the user has a fair amount of knowledge about HTML and CSS. In addition to that, the documentation is readily available on the official site.
3. **Responsive design and looks**: The web pages that are created by using the Bootstrap framework are responsive and it can adapt to any screen size like mobile, desktop, etc.
4. **Easily Customized**: It also provides some built-in components and functionalities that can be used for the purpose of easily customizing the web pages.

**Disadvantages**:

1.In many cases, Bootstrap cannot be considered very practical for businesses that need a big amount of investment.

2.Moreover, Bootstrap 4 can take a lot of time to create a website. Therefore, it is not a very bright idea to use bootstrap 4 when there is no investment.

3.A person using Bootstrap 4 is not likely to earn any money even after investment.

4.And this is one of the reasons why a user can very easily end up in a lot of debt.

# Environment Setup

Bootstrap 4 is the latest version of Bootstrap. In this section, it is shown how to download and setup Bootstrap 4.

There are two ways of using Bootstrap 4 in the Website:

a) The first method that can be used is to start using Bootstrap 4 by inserting it from the CDN which stands for **Content Delivery Network**.

b) And the second method is to download it from getbootstrap.com.

### Using CDN: The [Bootstrap 4](https://www.javatpoint.com/bootstrap-4) can be used in the website by inserting it from the CDN which is short for - Content Delivery Network.

The code given below can be used to compile the Bootstrap's CDN of CSS and JS in the project.

<!-- Compiled and Minified Bootstrap CSS -->**<link** rel = "stylesheet"    href = "https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css"   integrity = "sha384-Gn5384xqQ1aoWXA+058RXPxPg6fy4IWvTNh0E263XmFcJlSAwiGgFAW/dAiS6JXm"    crossorigin = "anonymous"**>**<!-- jQuery Library -->**<script** src = "https://code.jquery.com/jquery-3.2.1.slim.min.js"    integrity = "sha384-KJ3o2DKtIkvYIK3UENzmM7KCkRr/rE9/Qpg6aAZGJwFDMVNA/GpGFF93hXpG5KkN"    crossorigin = "anonymous"**></script>**<!-- Popper -->**<script** src = "https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.12.9/umd/popper.min.js"    integrity = "sha384-ApNbgh9B+Y1QKtv3Rn7W3mgPxhU9K/ScQsAP7hUibX39j7fakFPskvXusvfa0b4Q"    crossorigin = "anonymous"**></script>**<!-- Compiled and Minified Bootstrap JavaScript -->**<script** src = "https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/js/bootstrap.min.js"    integrity = "sha384-JZR6Spejh4U02d8jOt6vLEHfe/JQGiRRSQQxSfFWpi1MquVdAyjUar5+76PVCmYl"    crossorigin = "anonymous"**></script>**

In this process, also insert the CDN versions of [**jQuery**](https://www.javatpoint.com/jquery-tutorial) and **Popper.js** which is used for the purpose of using the different JavaScript components, for example - modals, tooltips, popovers, and so on. This needs to be before the minified Bootstrap JavaScript, if the compiled version of [JavaScript](https://www.javatpoint.com/javascript-tutorial) is being used.

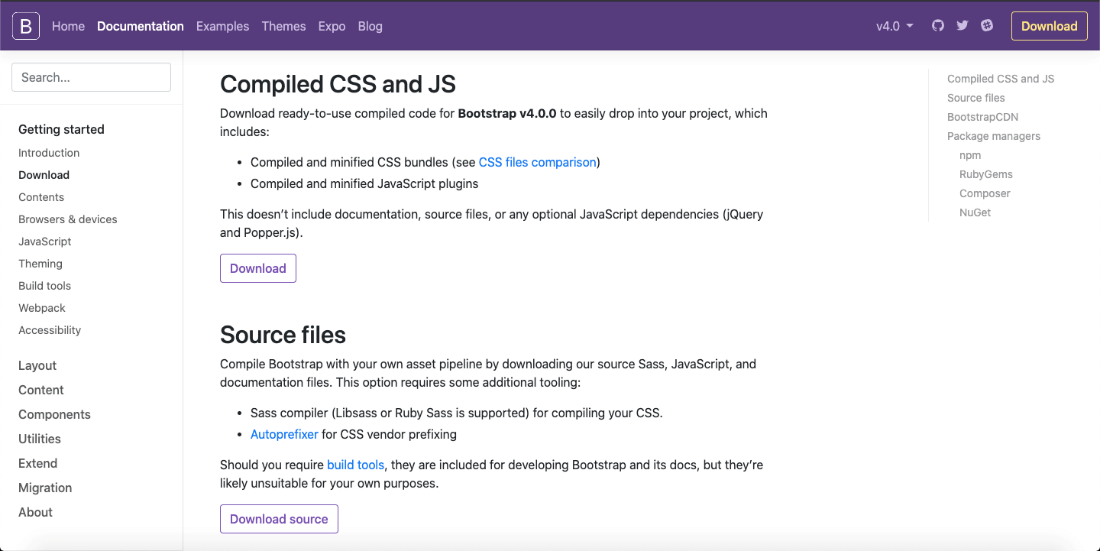
There are some components that extensively require the jQuery. Following are some of these components -

* For using the closable alerts
* For toggling the states with the help of the buttons and the checkboxes or radio buttons and using the collapse for toggling content
* For the purpose of carousel for slides, controls, and indicators
* For using Dropdowns (the Popper.js can be used for perfect positioning)
* For opening and closing the Modals
* For collapsing the Navbar
* For using the tooltips and popovers (the Popper.js can be used for perfect positioning)
* For using various other components

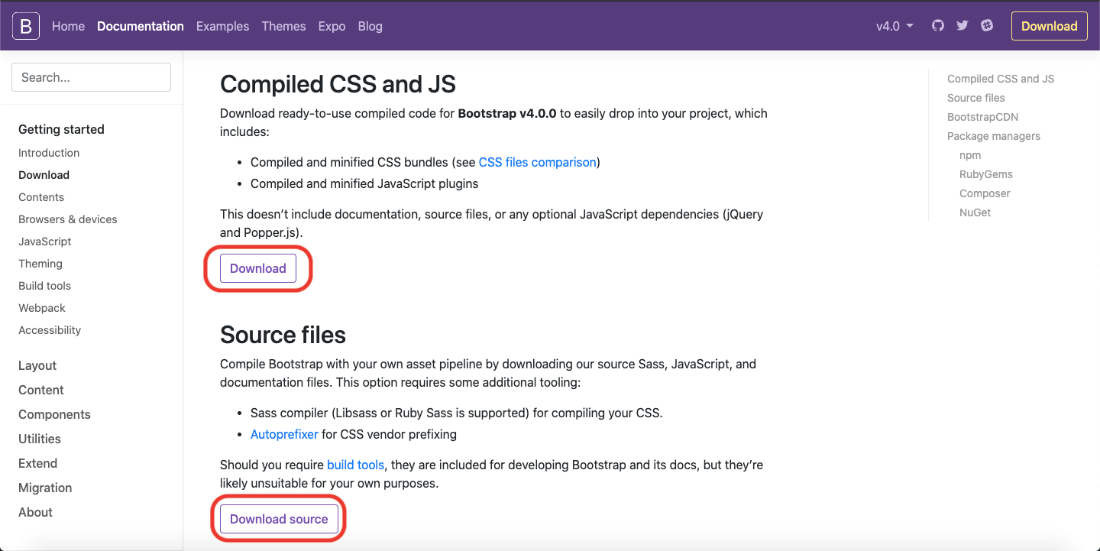
### Downloading Bootstrap 4:

The Bootstrap 4 can be downloaded from this link given below: <https://getbootstrap.com/docs/4.0/getting-started/download/>.

Once the link has been opened, a window will open:



In this window, there are two given options, one is download ready to use compiled code for Bootstrap v4.0.0 and the other option is downloading the source files that compiles [Bootstrap](https://www.javatpoint.com/bootstrap-tutorial) with a person's own asset pipeline by downloading Bootstrap 4's source Sass, JavaScript, and documentation files.



**Download** - By choosing this option, a user can download the pre-compiled and minified versions of Bootstrap's CSS and JavaScript. No documentation or original source code files are included. However, this does not consist of the documentation, the source files, or any other optional JavaScript dependencies such as jQuery and Popper.js.

**Download Source** - By choosing this option, a person can get the latest Bootstrap SCSS, JavaScript source code and documentation files. Moreover, this option also needs some additional tooling such as:

a )Sass compiler like Libsass or Ruby Sass is supported for compiling the CSS in the website.

b)Auto prefixer for CSS vendor prefixing.

**Precompiled Bootstrap 4:**

Once the compiled version Bootstrap 4 is downloaded, extract the ZIP file, and you will see the following file/directory structure -

1. bootstrap/
2. ├── css/
3. │   ├── bootstrap.css
4. │   ├── bootstrap.css.map
5. │   ├── bootstrap.min.css
6. │   ├── bootstrap.min.css.map
7. │   ├── bootstrap-theme.css
8. │   ├── bootstrap-theme.css.map
9. │   ├── bootstrap-theme.min.css
10. │   └── bootstrap-theme.min.css.map
11. ├── js/
12. │   ├── bootstrap.js
13. │   └── bootstrap.min.js
14. └── fonts/
15. ├── glyphicons-halflings-regular.eot
16. ├── glyphicons-halflings-regular.svg
17. ├── glyphicons-halflings-regular.ttf
18. ├── glyphicons-halflings-regular.woff
19. └── glyphicons-halflings-regular.woff2

There are compiled CSS and JS (bootstrap.\*), as well as compiled and minified CSS and JS (bootstrap.min.\*).

### Bootstrap 4 Source Code:

The Bootstrap source code download includes the precompiled CSS, JavaScript, and font assets, along with source Less, JavaScript, and documentation. More specifically, it includes the following and more -

1. bootstrap/
2. ├── less/
3. ├── js/
4. ├── fonts/
5. ├── dist/
6. │      ├── css/
7. │      ├── js/
8. │      └── fonts/
9. └── docs/
10. └── examples/
11. The files under the js/ and scss/ are the source code responsible for Bootstrap, CSS and JavaScript.
12. The dist/ folder consists of everything that is listed in the pre-compiled download section that is given above.
13. The docs/examples/ only consists of the source code for Bootstrap documentation and examples of Bootstrap usage respectively.
14. The less/ in the source code given above is basically responsible for the source code for the icon fonts in Bootstrap 4.
15. Apart from that, the other included files provide support for other important components such as the packages, the information about the license, and anything related to development.

# Bootstrap Templates

# Navbar

**Navigation Bars:**A navigation bar is a navigation header that is placed at the top of the page:

**i) Basic Navbar:**

With Bootstrap, a navigation bar can extend or collapse, depending on the screen size.

A standard navigation bar is created with the .navbar class, followed by a responsive collapsing class: .navbar-expand-xl|lg|md|sm (stacks the navbar vertically on extra large, large, medium or small screens).

To add links inside the navbar, use a <ul> element with class="navbar-nav". Then add <li> elements with a .nav-item class followed by an <a> element with a .nav-link class:

[**Link 1**](javascript:void(0))[**Link 2**](javascript:void(0))[**Link 3**](javascript:void(0))

## Example:

## <!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 rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css">

## <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>

## <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.16.0/umd/popper.min.js"></script>

## <script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"></script>

## </head>

## <body>

## <nav class="navbar navbar-expand-sm bg-light">

## <ul class="navbar-nav">

## <li class="nav-item">

## <a class="nav-link" href="#">Link 1</a>

## </li>

## <li class="nav-item">

## <a class="nav-link" href="#">Link 2</a>

## </li>

## <li class="nav-item">

## <a class="nav-link" href="#">Link 3</a>

## </li>

## </ul>

## </nav>

## <br>

## <div class="container-fluid">

## <h3>Basic Navbar Example</h3>

## <p>A navigation bar is a navigation header that is placed at the top of the page.</p>

## <p>The navbar-expand-xl|lg|md|sm class determines when the navbar should stack vertically (on extra large, large, medium or small screens).</p>

## </div>

## </body>

## </html>

## ii) Vertical Navbar:

Remove the .navbar-expand-xl|lg|md|sm class to create a vertical navigation bar:

* [**Link 1**](javascript:void(0))
* [**Link 2**](javascript:void(0))
* [**Link 3**](javascript:void(0))

**Example**: <!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 rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css">

## <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>

## <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.16.0/umd/popper.min.js"></script>

## <script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"></script>

## </head>

## <body>

## <nav class="navbar bg-light">

## <ul class="navbar-nav">

## <li class="nav-item">

## <a class="nav-link" href="#">Link 1</a>

## </li>

## <li class="nav-item">

## <a class="nav-link" href="#">Link 2</a>

## </li>

## <li class="nav-item">

## <a class="nav-link" href="#">Link 3</a>

## </li>

## </ul>

## </nav>

## <br>

## <div class="container-fluid">

## <h3>Vertical Navbar Example</h3>

## <p>A navigation bar is a navigation header that is placed at the top of the page.</p>

## </div>

## </body>

## </html>

## iii) Centered Navbar:

Add the .justify-content-center class to center the navigation bar.

The following example will center the navigation bar on medium, large and extra large screens. On small screens it will be displayed vertically and left-aligned (because of the .navbar-expand-sm class):

[**Link 1**](javascript:void(0))[**Link 2**](javascript:void(0))[**Link 3**](javascript:void(0))

## Example:

## <!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 rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css">

## <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>

## <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.16.0/umd/popper.min.js"></script>

## <script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"></script>

## </head>

## <body>

## <nav class="navbar navbar-expand-sm bg-light justify-content-center">

## <ul class="navbar-nav">

## <li class="nav-item">

## <a class="nav-link" href="#">Link 1</a>

## </li>

## <li class="nav-item">

## <a class="nav-link" href="#">Link 2</a>

## </li>

## <li class="nav-item">

## <a class="nav-link" href="#">Link 3</a>

## </li>

## </ul>

## </nav>

## <br>

## <div class="container-fluid">

## <h3>Centered Navbar</h3>

## <p>Use the .justify-content-center class to center the navigation bar.</p>

## <p>In this example, the navbar will be centered on medium, large and extra large screens. On small screens it will be displayed vertically and left-aligned (because of the .navbar-expand-sm class).</p>

## </div>

## </body>

## </html>

## iv) Colored Navbar:

Use any of the .bg-color classes to change the background color of the navbar (.bg-primary, .bg-success, .bg-info, .bg-warning, .bg-danger, .bg-secondary, .bg-dark and .bg-light)

**Tip:** Add a **white** text color to all links in the navbar with the .navbar-dark class, or use the .navbar-light class to add a **black** text color.

## Example:

## <!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 rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css">

## <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>

## <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.16.0/umd/popper.min.js"></script>

## <script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"></script>

## </head>

## <body>

## <div class="container">

## <h3>Colored Navbar</h3>

## <p>Use any of the .bg-color classes to add a background color to the navbar.</p>

## <p>Tip: Add a white text color to all links in the navbar with the .navbar-dark class, or use the .navbar-light class to add a black text color.</p>

## </div>

## <nav class="navbar navbar-expand-sm bg-light navbar-light">

## <ul class="navbar-nav">

## <li class="nav-item active">

## <a class="nav-link" href="#">Active</a>

## </li>

## <li class="nav-item">

## <a class="nav-link" href="#">Link</a>

## </li>

## <li class="nav-item">

## <a class="nav-link" href="#">Link</a>

## </li>

## <li class="nav-item">

## <a class="nav-link disabled" href="#">Disabled</a>

## </li>

## </ul>

## </nav>

## <nav class="navbar navbar-expand-sm bg-dark navbar-dark">

## <ul class="navbar-nav">

## <li class="nav-item active">

## <a class="nav-link" href="#">Active</a>

## </li>

## <li class="nav-item">

## <a class="nav-link" href="#">Link</a>

## </li>

## <li class="nav-item">

## <a class="nav-link" href="#">Link</a>

## </li>

## <li class="nav-item">

## <a class="nav-link disabled" href="#">Disabled</a>

## </li>

## </ul>

## </nav>

## <nav class="navbar navbar-expand-sm bg-primary navbar-dark">

## <ul class="navbar-nav">

## <li class="nav-item active">

## <a class="nav-link" href="#">Active</a>

## </li>

## <li class="nav-item">

## <a class="nav-link" href="#">Link</a>

## </li>

## <li class="nav-item">

## <a class="nav-link" href="#">Link</a>

## </li>

## <li class="nav-item">

## <a class="nav-link disabled" href="#">Disabled</a>

## </li>

## </ul>

## </nav>

## <nav class="navbar navbar-expand-sm bg-success navbar-dark">

## <ul class="navbar-nav">

## <li class="nav-item active">

## <a class="nav-link" href="#">Active</a>

## </li>

## <li class="nav-item">

## <a class="nav-link" href="#">Link</a>

## </li>

## <li class="nav-item">

## <a class="nav-link" href="#">Link</a>

## </li>

## <li class="nav-item">

## <a class="nav-link disabled" href="#">Disabled</a>

## </li>

## </ul>

## </nav>

## <nav class="navbar navbar-expand-sm bg-info navbar-dark">

## <ul class="navbar-nav">

## <li class="nav-item active">

## <a class="nav-link" href="#">Active</a>

## </li>

## <li class="nav-item">

## <a class="nav-link" href="#">Link</a>

## </li>

## <li class="nav-item">

## <a class="nav-link" href="#">Link</a>

## </li>

## <li class="nav-item">

## <a class="nav-link disabled" href="#">Disabled</a>

## </li>

## </ul>

## </nav>

## <nav class="navbar navbar-expand-sm bg-warning navbar-dark">

## <ul class="navbar-nav">

## <li class="nav-item active">

## <a class="nav-link" href="#">Active</a>

## </li>

## <li class="nav-item">

## <a class="nav-link" href="#">Link</a>

## </li>

## <li class="nav-item">

## <a class="nav-link" href="#">Link</a>

## </li>

## <li class="nav-item">

## <a class="nav-link disabled" href="#">Disabled</a>

## </li>

## </ul>

## </nav>

## <nav class="navbar navbar-expand-sm bg-danger navbar-dark">

## <ul class="navbar-nav">

## <li class="nav-item active">

## <a class="nav-link" href="#">Active</a>

## </li>

## <li class="nav-item">

## <a class="nav-link" href="#">Link</a>

## </li>

## <li class="nav-item">

## <a class="nav-link" href="#">Link</a>

## </li>

## <li class="nav-item">

## <a class="nav-link disabled" href="#">Disabled</a>

## </li>

## </ul>

## </nav>

## <nav class="navbar navbar-expand-sm bg-secondary navbar-dark">

## <ul class="navbar-nav">

## <li class="nav-item active">

## <a class="nav-link" href="#">Active</a>

## </li>

## <li class="nav-item">

## <a class="nav-link" href="#">Link</a>

## </li>

## <li class="nav-item">

## <a class="nav-link" href="#">Link</a>

## </li>

## <li class="nav-item">

## <a class="nav-link disabled" href="#">Disabled</a>

## </li>

## </ul>

## </nav>

## </body>

## </html>

## v) Brand / Logo:

The .navbar-brand class is used to highlight the brand/logo/project name of your page:

## Example:

## <!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 rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css">

## <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>

## <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.16.0/umd/popper.min.js"></script>

## <script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"></script>

## </head>

## <body>

## <nav class="navbar navbar-expand-sm bg-dark navbar-dark">

## <!-- Brand/logo -->

## <a class="navbar-brand" href="#">Logo</a>

## 

## <!-- Links -->

## <ul class="navbar-nav">

## <li class="nav-item">

## <a class="nav-link" href="#">Link 1</a>

## </li>

## <li class="nav-item">

## <a class="nav-link" href="#">Link 2</a>

## </li>

## <li class="nav-item">

## <a class="nav-link" href="#">Link 3</a>

## </li>

## </ul>

## </nav>

## <div class="container-fluid">

## <h3>Brand / Logo</h3>

## <p>The .navbar-brand class is used to highlight the brand/logo/project name of your page.</p>

## </div>

## </body>

## </html>

**vi) Collapsing The Navigation Bar:** Very often, especially on small screens, you want to hide the navigation links and replace them with a button that should reveal them when clicked on.

To create a collapsible navigation bar, use a button with class="navbar-toggler", data-toggle="collapse" and data-target="#thetarget". Then wrap the navbar content (links, etc) inside a div element with class="collapse navbar-collapse", followed by an id that matches the data-target of the button: "thetarget".

**Example:**

<!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 rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css">

<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>

<script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.16.0/umd/popper.min.js"></script>

<script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"></script>

</head>

<body>

<nav class="navbar navbar-expand-md bg-dark navbar-dark">

<a class="navbar-brand" href="#">Navbar</a>

<button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#collapsibleNavbar">

<span class="navbar-toggler-icon"></span>

</button>

<div class="collapse navbar-collapse" id="collapsibleNavbar">

<ul class="navbar-nav">

<li class="nav-item">

<a class="nav-link" href="#">Link</a>

</li>

<li class="nav-item">

<a class="nav-link" href="#">Link</a>

</li>

<li class="nav-item">

<a class="nav-link" href="#">Link</a>

</li>

</ul>

</div>

</nav>

<br>

<div class="container">

<h3>Collapsible Navbar</h3>

<p>In this example, the navigation bar is hidden on small screens and replaced by a button in the top right corner (try to re-size this window).</p>

<p>Only when the button is clicked, the navigation bar will be displayed.</p>

<p>Tip: You can also remove the .navbar-expand-md class to ALWAYS hide navbar links and display the toggler button.</p>

</div>

</body>

</html>

**vii) Navbar With Dropdown:**

**Example:**

<!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 rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css">

<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>

<script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.16.0/umd/popper.min.js"></script>

<script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"></script>

</head>

<body>

<nav class="navbar navbar-expand-sm bg-dark navbar-dark">

<!-- Brand -->

<a class="navbar-brand" href="#">Logo</a>

<!-- Links -->

<ul class="navbar-nav">

<li class="nav-item">

<a class="nav-link" href="#">Link 1</a>

</li>

<li class="nav-item">

<a class="nav-link" href="#">Link 2</a>

</li>

<!-- Dropdown -->

<li class="nav-item dropdown">

<a class="nav-link dropdown-toggle" href="#" id="navbardrop" data-toggle="dropdown">

Dropdown link

</a>

<div class="dropdown-menu">

<a class="dropdown-item" href="#">Link 1</a>

<a class="dropdown-item" href="#">Link 2</a>

<a class="dropdown-item" href="#">Link 3</a>

</div>

</li>

</ul>

</nav>

<br>

<div class="container">

<h3>Navbar With Dropdown</h3>

<p>This example adds a dropdown menu in the navbar.</p>

</div>

</body>

</html>

**viii) Navbar Forms and Buttons:** Add a <form> element with class="form-inline" to group inputs and buttons side-by-side:

<!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 rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css">

<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>

<script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.16.0/umd/popper.min.js"></script>

<script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"></script>

</head>

<body>

<nav class="navbar navbar-expand-sm bg-dark navbar-dark">

<form class="form-inline" action="/action\_page.php">

<input class="form-control mr-sm-2" type="text" placeholder="Search">

<button class="btn btn-success" type="submit">Search</button>

</form>

</nav>

<br>

<div class="container">

<h3>Navbar Forms</h3>

<p>Use the .form-inline class to align form elements side by side inside the navbar.</p>

</div>

</body>

</html>

**ix) Navbar Text:** Use the .navbar-text class to vertical align any elements inside the navbar that are not links (ensures proper padding and text color).

**Example:** <!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 rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css">

<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>

<script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.16.0/umd/popper.min.js"></script>

<script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"></script>

</head>

<body>

<nav class="navbar navbar-expand-sm bg-dark navbar-dark">

<!-- Links -->

<ul class="navbar-nav">

<li class="nav-item">

<a class="nav-link" href="#">Link 1</a>

</li>

<li class="nav-item">

<a class="nav-link" href="#">Link 2</a>

</li>

</ul>

<!-- Navbar text-->

<span class="navbar-text">

Navbar text

</span>

</nav>

<br>

<div class="container">

<h3>Navbar Text</h3>

<p>Use the .navbar-text class to vertical align any elements inside the navbar that are not links (ensures proper padding).</p>

</div>

</body>

</html>

**x)Fixed Navigation Bar:** The navigation bar can also be fixed at the top or at the bottom of the page.A fixed navigation bar stays visible in a fixed position (top or bottom) independent of the page scroll. The .fixed-top class makes the navigation bar fixed at the **top**:

**Example:** <!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 rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css">

<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>

<script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.16.0/umd/popper.min.js"></script>

<script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"></script>

</head>

<body style="height:1500px">

<nav class="navbar navbar-expand-sm bg-dark navbar-dark fixed-top">

<a class="navbar-brand" href="#">Logo</a>

<ul class="navbar-nav">

<li class="nav-item">

<a class="nav-link" href="#">Link</a>

</li>

<li class="nav-item">

<a class="nav-link" href="#">Link</a>

</li>

</ul>

</nav>

<div class="container-fluid" style="margin-top:80px">

<h3>Top Fixed Navbar</h3>

<p>A fixed navigation bar stays visible in a fixed position (top or bottom) independent of the page scroll.</p>

<h1>Scroll this page to see the effect</h1>

</div>

</body>

</html>

# Typography

# i) <h1> - <h6>:

Bootstrap 4 styles HTML headings (<h1> to <h6>)  with a bolder font-weight and an increased font-size:

**Example:**

<!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 rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css">

<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>

<script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.16.0/umd/popper.min.js"></script>

<script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"></script>

</head>

<body>

<div class="container">

<h1>h1 Bootstrap heading (2.5rem = 40px)</h1>

<h2>h2 Bootstrap heading (2rem = 32px)</h2>

<h3>h3 Bootstrap heading (1.75rem = 28px)</h3>

<h4>h4 Bootstrap heading (1.5rem = 24px)</h4>

<h5>h5 Bootstrap heading (1.25rem = 20px)</h5>

<h6>h6 Bootstrap heading (1rem = 16px)</h6>

</div>

</body>

</html>

## ii) Display Headings:

Display headings are used to stand out more than normal headings (larger font-size and lighter font-weight), and there are four classes to choose from: .display-1, .display-2, .display-3, .display-4

**Example:** <!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 rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css">

<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>

<script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.16.0/umd/popper.min.js"></script>

<script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"></script>

</head>

<body>

<div class="container">

<h1>Display Headings</h1>

<p>Display headings are used to stand out more than normal headings (larger font-size and lighter font-weight):</p>

<h1 class="display-1">Display 1</h1>

<h1 class="display-2">Display 2</h1>

<h1 class="display-3">Display 3</h1>

<h1 class="display-4">Display 4</h1>

</div>

</body>

</html>

**iii) <small>:**

In Bootstrap 4 the HTML <small> element is used to create a lighter, secondary text in any heading:

**Example:**

<!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 rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css">

<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>

<script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.16.0/umd/popper.min.js"></script>

<script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"></script>

</head>

<body>

<div class="container">

<h1>Lighter, Secondary Text</h1>

<p>The small element is used to create a lighter, secondary text in any heading:</p>

<h1>h1 heading <small>secondary text</small></h1>

<h2>h2 heading <small>secondary text</small></h2>

<h3>h3 heading <small>secondary text</small></h3>

<h4>h4 heading <small>secondary text</small></h4>

<h5>h5 heading <small>secondary text</small></h5>

<h6>h6 heading <small>secondary text</small></h6>

</div>

</body>

</html>

**iv) <mark>:**

Bootstrap 4 will style the HTML <mark> element with a yellow background color and some padding:

**Example:**

<!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 rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css">

<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>

<script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.16.0/umd/popper.min.js"></script>

<script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"></script>

</head>

<body>

<div class="container">

<h1>Highlight Text</h1>

<p>Use the mark element to <mark>highlight</mark> text.</p>

</div>

</body>

</html>

**v) <abbr>:**

Bootstrap 4 will style the HTML <abbr> element with a dotted border bottom:

**Example:**

<!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 rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css">

<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>

<script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.16.0/umd/popper.min.js"></script>

<script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"></script>

</head>

<body>

<div class="container">

<h1>Abbreviations</h1>

<p>The abbr element is used to mark up an abbreviation or acronym:</p>

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

</div>

</body>

</html>

**vi) <blockquote>:**

Add the .blockquote class to a <blockquote> when quoting blocks of content from another source:

**Example:**

<!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 rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css">

<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>

<script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.16.0/umd/popper.min.js"></script>

<script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"></script>

</head>

<body>

<div class="container">

<h1>Blockquotes</h1>

<p>The blockquote element is used to present content from another source:</p>

<blockquote class="blockquote">

<p>For 50 years, WWF has been protecting the future of nature. The world's leading conservation organization, WWF works in 100 countries and is supported by 1.2 million members in the United States and close to 5 million globally.</p>

<footer class="blockquote-footer">From WWF's website</footer>

</blockquote>

</div>

</body>

</html>

**vii) <dl>:**

Bootstrap 4 will style the HTML <dl> element in the following way:

**Example:**

<!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 rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css">

<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>

<script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.16.0/umd/popper.min.js"></script>

<script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"></script>

</head>

<body>

<div class="container">

<h1>Description Lists</h1>

<p>The dl element indicates a description list:</p>

<dl>

<dt>Coffee</dt>

<dd>- black hot drink</dd>

<dt>Milk</dt>

<dd>- white cold drink</dd>

</dl>

</div>

</body>

</html>

**viii) <code>:**

Bootstrap 4 will style the HTML <code> element in the following way:

**Example:** <!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 rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css">

<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>

<script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.16.0/umd/popper.min.js"></script>

<script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"></script>

</head>

<body>

<div class="container">

<h1>Code Snippets</h1>

<p>Inline snippets of code should be embedded in the code element:</p>

<p>The following HTML elements: <code>span</code>, <code>section</code>, and <code>div</code> defines a section in a document.</p>

</div>

</body>

</html>

**ix) <kbd>:**

Bootstrap 4 will style the HTML <kbd> element in the following way:

**Example:**

<!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 rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css">

<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>

<script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.16.0/umd/popper.min.js"></script>

<script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"></script>

</head>

<body>

<div class="container">

<h1>Keyboard Inputs</h1>

<p>To indicate input that is typically entered via the keyboard, use the kbd element:</p>

<p>Use <kbd>ctrl + p</kbd> to open the Print dialog box.</p>

</div>

</body>

</html>

**X) <pre>:**

Bootstrap 4 will style the HTML <pre> element in the following way:

**Example:**

<!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 rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css">

<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>

<script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.16.0/umd/popper.min.js"></script>

<script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"></script>

</head>

<body>

<div class="container">

<h1>Multiple Code Lines</h1>

<p>For multiple lines of code, use the pre element:</p>

<pre>

Text in a pre element

is displayed in a fixed-width

font, and it preserves

both spaces and

line breaks.

</pre>

</div>

</body>

</html>

# Forms And Tables

Form controls automatically receive some global styling with Bootstrap:

All textual <input>, <textarea>, and <select> elements with class .form-control have a width of 100%.

**Form Layouts:**

Bootstrap provides two types of form layouts:

* Stacked (full-width) form
* Inline form

**i) Stacked Form:**

**Example:**

<!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 rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css">

<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>

<script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.16.0/umd/popper.min.js"></script>

<script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"></script>

</head>

<body>

<div class="container">

<h2>Stacked form</h2>

<form action="/action\_page.php">

<div class="form-group">

<label for="email">Email:</label>

<input type="email" class="form-control" id="email" placeholder="Enter email" name="email">

</div>

<div class="form-group">

<label for="pwd">Password:</label>

<input type="password" class="form-control" id="pwd" placeholder="Enter password" name="pswd">

</div>

<div class="form-group form-check">

<label class="form-check-label">

<input class="form-check-input" type="checkbox" name="remember"> Remember me

</label>

</div>

<button type="submit" class="btn btn-primary">Submit</button>

</form>

</div>

</body>

</html>

**ii)Inline Form:**

**Example:**

<!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 rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css">

<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>

<script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.16.0/umd/popper.min.js"></script>

<script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"></script>

</head>

<body>

<div class="container">

<h2>Inline form</h2>

<p>Make the viewport larger than 576px wide to see that all of the form elements are inline and left-aligned. On small screens, the form groups will stack horizontally.</p>

<form class="form-inline" action="/action\_page.php">

<label for="email">Email:</label>

<input type="email" class="form-control" id="email" placeholder="Enter email" name="email">

<label for="pwd">Password:</label>

<input type="password" class="form-control" id="pwd" placeholder="Enter password" name="pswd">

<div class="form-check">

<label class="form-check-label">

<input class="form-check-input" type="checkbox" name="remember"> Remember me

</label>

</div>

<button type="submit" class="btn btn-primary">Submit</button>

</form>

</div>

</body>

</html>

**iii) Form Row/Grid:**

You can also use columns (.col) to control the width and alignment of form inputs without using spacing utilities. Just remember to put them inside a .row container.

**Example:**

<!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 rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css">

<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>

<script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.16.0/umd/popper.min.js"></script>

<script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"></script>

</head>

<body>

<div class="container">

<h2>Form Grid</h2>

<p>Create two form elements that appear side by side with .row and .col:</p>

<form action="/action\_page.php">

<div class="row">

<div class="col">

<input type="text" class="form-control" id="email" placeholder="Enter email" name="email">

</div>

<div class="col">

<input type="password" class="form-control" placeholder="Enter password" name="pswd">

</div>

</div>

<button type="submit" class="btn btn-primary mt-3">Submit</button>

</form>

</div>

</body>

</html>

**iv) Form Validation**: You can use different validation classes to provide valuable feedback to users. Add either .was-validated or .needs-validation to the <form> element, depending on whether you want to provide validation feedback before or after submitting the form. The input fields will have a green (valid) or red (invalid) border to indicate what's missing in the form. You can also add a .valid-feedback or .invalid-feedback message to tell the user explicitly what's missing, or needs to be done before submitting the form.

**Example:**

<!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 rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css">

<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>

<script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.16.0/umd/popper.min.js"></script>

<script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"></script>

</head>

<body>

<div class="container">

<h2>Form Validation</h2>

<p>In this example, we use <code>.was-validated</code> to indicate what's missing before submitting the form:</p>

<form action="/action\_page.php" class="was-validated">

<div class="form-group">

<label for="uname">Username:</label>

<input type="text" class="form-control" id="uname" placeholder="Enter username" name="uname" required>

<div class="valid-feedback">Valid.</div>

<div class="invalid-feedback">Please fill out this field.</div>

</div>

<div class="form-group">

<label for="pwd">Password:</label>

<input type="password" class="form-control" id="pwd" placeholder="Enter password" name="pswd" required>

<div class="valid-feedback">Valid.</div>

<div class="invalid-feedback">Please fill out this field.</div>

</div>

<div class="form-group form-check">

<label class="form-check-label">

<input class="form-check-input" type="checkbox" name="remember" required> I agree on blabla.

<div class="valid-feedback">Valid.</div>

<div class="invalid-feedback">Check this checkbox to continue.</div>

</label>

</div>

<button type="submit" class="btn btn-primary">Submit</button>

</form>

</div>

</body>

</html>

**Tables:** A basic Bootstrap 4 table has a light padding and horizontal dividers.

The .table class adds basic styling to a table:

**i) Striped Rows:**

The .table-striped class adds zebra-stripes to a table:

**Example:** <!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 rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css">

<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>

<script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.16.0/umd/popper.min.js"></script>

<script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"></script>

</head>

<body>

<div class="container">

<h2>Striped Rows</h2>

<p>The .table-striped class adds zebra-stripes to a table:</p>

<table class="table table-striped">

<thead>

<tr>

<th>Firstname</th>

<th>Lastname</th>

<th>Email</th>

</tr>

</thead>

<tbody>

<tr>

<td>John</td>

<td>Doe</td>

<td>john@example.com</td>

</tr>

<tr>

<td>Mary</td>

<td>Moe</td>

<td>mary@example.com</td>

</tr>

<tr>

<td>July</td>

<td>Dooley</td>

<td>july@example.com</td>

</tr>

</tbody>

</table>

</div>

</body>

</html>

**ii) Bordered Table:**

The .table-bordered class adds borders on all sides of the table and cells:

**Example:** <!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 rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css">

<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>

<script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.16.0/umd/popper.min.js"></script>

<script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"></script>

</head>

<body>

<div class="container">

<h2>Bordered Table</h2>

<p>The .table-bordered class adds borders on all sides of the table and the cells:</p>

<table class="table table-bordered">

<thead>

<tr>

<th>Firstname</th>

<th>Lastname</th>

<th>Email</th>

</tr>

</thead>

<tbody>

<tr>

<td>John</td>

<td>Doe</td>

<td>john@example.com</td>

</tr>

<tr>

<td>Mary</td>

<td>Moe</td>

<td>mary@example.com</td>

</tr>

<tr>

<td>July</td>

<td>Dooley</td>

<td>july@example.com</td>

</tr>

</tbody>

</table>

</div>

</body>

</html>

**iv) Hover Rows**

The .table-hover class adds a hover effect (grey background color) on table rows:

**Example:**

<!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 rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css">

<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>

<script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.16.0/umd/popper.min.js"></script>

<script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"></script>

</head>

<body>

<div class="container">

<h2>Hover Rows</h2>

<p>The .table-hover class enables a hover state (grey background on mouse over) on table rows:</p>

<table class="table table-hover">

<thead>

<tr>

<th>Firstname</th>

<th>Lastname</th>

<th>Email</th>

</tr>

</thead>

<tbody>

<tr>

<td>John</td>

<td>Doe</td>

<td>john@example.com</td>

</tr>

<tr>

<td>Mary</td>

<td>Moe</td>

<td>mary@example.com</td>

</tr>

<tr>

<td>July</td>

<td>Dooley</td>

<td>july@example.com</td>

</tr>

</tbody>

</table>

</div>

</body>

</html>

**v) Black/Dark Table:**

The .table-dark class adds a black background to the table:

<!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 rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css">

<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>

<script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.16.0/umd/popper.min.js"></script>

<script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"></script>

</head>

<body>

<div class="container">

<h2>Black/Dark Table</h2>

<p>The .table-dark class adds a black background to the table:</p>

<table class="table table-dark">

<thead>

<tr>

<th>Firstname</th>

<th>Lastname</th>

<th>Email</th>

</tr>

</thead>

<tbody>

<tr>

<td>John</td>

<td>Doe</td>

<td>john@example.com</td>

</tr>

<tr>

<td>Mary</td>

<td>Moe</td>

<td>mary@example.com</td>

</tr>

<tr>

<td>July</td>

<td>Dooley</td>

<td>july@example.com</td>

</tr>

</tbody>

</table>

</div>

</body>

</html>

**vi) Dark Striped Table:**

Combine .table-dark and .table-striped to create a dark, striped table:

**Example:**    </thead>

   <tbody>

     <tr>

       <td>John</td>

       <td>Doe</td>

       <td>john@example.com</td>

     </tr>

     <tr>

       <td>Mary</td>

       <td>Moe</td>

       <td>mary@example.com</td>

     </tr>

     <tr>

       <td>July</td>

       <td>Dooley</td>

       <td>july@example.com</td>

     </tr>

   </tbody>

 </table>

</div>

​

</body>

</html>

​

## vii) Hoverable Dark Table:

The .table-hover class adds a hover effect (grey background color) on table rows:

**Example:**

<!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 rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css">

<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>

<script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.16.0/umd/popper.min.js"></script>

<script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"></script>

</head>

<body>

<div class="container">

<h2>Hoverable Dark Table</h2>

<p>The .table-hover class adds a hover effect (grey background color) on table rows:</p>

<table class="table table-dark table-hover">

<thead>

<tr>

<th>Firstname</th>

<th>Lastname</th>

<th>Email</th>

</tr>

</thead>

<tbody>

<tr>

<td>John</td>

<td>Doe</td>

<td>john@example.com</td>

</tr>

<tr>

<td>Mary</td>

<td>Moe</td>

<td>mary@example.com</td>

</tr>

<tr>

<td>July</td>

<td>Dooley</td>

<td>july@example.com</td>

</tr>

</tbody>

</table>

</div>

</body>

</html>

**viii) Borderless Table:**

The .table-borderless class removes borders from the table:

**Example:** <!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 rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css">

<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>

<script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.16.0/umd/popper.min.js"></script>

<script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"></script>

</head>

<body>

<div class="container">

<h2>Borderless Table</h2>

<p>The .table-borderless class removes borders from the table:</p>

<table class="table table-borderless">

<thead>

<tr>

<th>Firstname</th>

<th>Lastname</th>

<th>Email</th>

</tr>

</thead>

<tbody>

<tr>

<td>John</td>

<td>Doe</td>

<td>john@example.com</td>

</tr>

<tr>

<td>Mary</td>

<td>Moe</td>

<td>mary@example.com</td>

</tr>

<tr>

<td>July</td>

<td>Dooley</td>

<td>july@example.com</td>

</tr>

</tbody>

</table>

</div>

</body>

</html**>**

**ix) Small table:**

The .table-sm class makes the table smaller by cutting cell padding in half:

**Example:**

<!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 rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css">

<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>

<script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.16.0/umd/popper.min.js"></script>

<script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"></script>

</head>

<body>

<div class="container">

<h2>Small Table</h2>

<p>The .table-sm class makes the table smaller by cutting cell padding in half:</p>

<table class="table table-bordered table-sm">

<thead>

<tr>

<th>Firstname</th>

<th>Lastname</th>

<th>Email</th>

</tr>

</thead>

<tbody>

<tr>

<td>John</td>

<td>Doe</td>

<td>john@example.com</td>

</tr>

<tr>

<td>Mary</td>

<td>Moe</td>

<td>mary@example.com</td>

</tr>

<tr>

<td>July</td>

<td>Dooley</td>

<td>july@example.com</td>

</tr>

</tbody>

</table>

</div>

</body>

</html>

x) **Responsive Tables:**

The .table-responsive class adds a scrollbar to the table when needed (when it is too big horizontally):

**Example:**

<!DOCTYPE html>

<html>

<head>

<meta name="viewport" content="width=device-width, initial-scale=1">

<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css">

<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>

<script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.16.0/umd/popper.min.js"></script>

<script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"></script>

</head>

<body>

<div class="container">

<h2>Responsive Table</h2>

<p>The .table-responsive class adds a scrollbar to the table when needed:</p>

<div class="table-responsive">

<table class="table table-bordered">

<thead>

<tr>

<th>#</th>

<th>Firstname</th>

<th>Lastname</th>

<th>Age</th>

<th>City</th>

<th>Country</th>

<th>Sex</th>

<th>Example</th>

<th>Example</th>

<th>Example</th>

<th>Example</th>

<th>Example</th>

<th>Example</th>

<th>Example</th>

<th>Example</th>

<th>Example</th>

<th>Example</th>

<th>Example</th>

<th>Example</th>

</tr>

</thead>

<tbody>

<tr>

<td>1</td>

<td>Anna</td>

<td>Pitt</td>

<td>35</td>

<td>New York</td>

<td>USA</td>

<td>Female</td>

<td>Yes</td>

<td>Yes</td>

<td>Yes</td>

<td>Yes</td>

<td>Yes</td>

<td>Yes</td>

<td>Yes</td>

<td>Yes</td>

<td>Yes</td>

<td>Yes</td>

<td>Yes</td>

<td>Yes</td>

</tr>

</tbody>

</table>

</div>

</div>

</body>

</html>

**Top of Form**

**Bottom of Form**