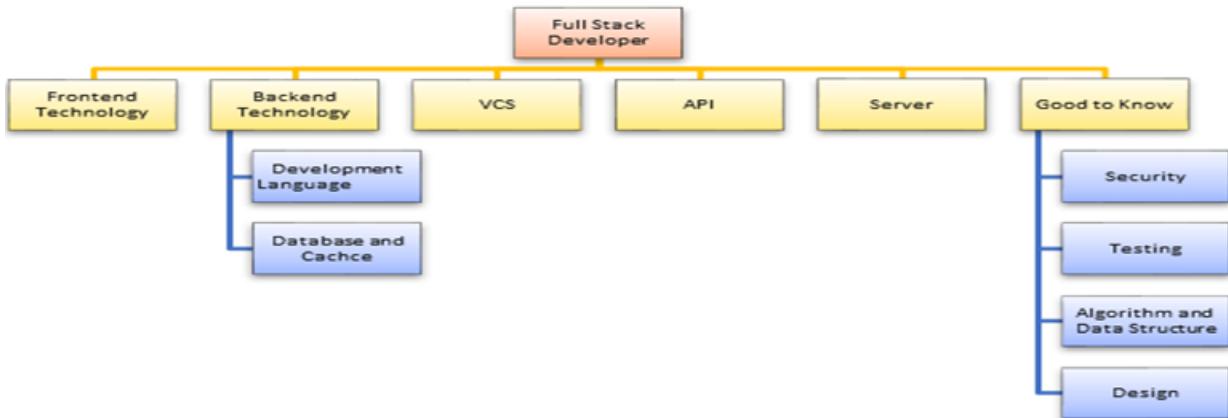


## UNIT-1

**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.

**Full Stack Web Developer:** A full stack web developer is a person who can develop both client and server software. He works on both front-end development and back-end development. He translates user requirements into the overall architecture and implement the new systems.



**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 and ReactJS 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, SQL Server

**4) Basic design ability:** In order to become a successful Full Stack web developer, the knowledge of designing like basic UI design.

**5) Server:** He works on server-side also like Apache, Tomcat etc.

**Front end:** It is the visible part of website or web; the user directly interacts with the front-end portion of the web application or website. Eg: HTML, CSS, BOOTSTRAP, CSS, JAVASCRIPT.

**AngularJS:** AngularJS is a JavaScript open-source front-end framework that is mainly used to develop single page web applications (SPAs). 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.

**React.js:** 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:** 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:** jQuery is an open-source JavaScript. jQuery simplifies HTML document traversing and manipulation, browser event handling, DOM animations.

## **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.
- 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. It is responsible for managing the database through queries and APIs by client-side commands.

PHP, ASP, C++, C#, JAVA, PYTHON, NODE.JS, GO, SQL, MONGODB,

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

- **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++:** It is a general-purpose programming language and widely used now a days for competitive programming. It is also used as backend language.
- **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:** Python is a programming language that lets you work quickly and integrate systems more efficiently.
- **JavaScript:** JavaScript can be used as both (front end and back end) programming languages.
- **Node.js:** Node.js is an open. 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 by large companies such as Paypal, Uber, Netflix, and so on.

**Database:** 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 manage information.
- **MongoDB:** 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:** Structured Query Language is a standard Database language which is used to create, maintain and retrieve the relational database.

*Full Stack Developer is "Just like a trader but, master on none".*

## **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

### **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 using.....  
HTML, CSS, BOOTSTRAP, W3.CSS, JAVASCRIPT, ES5, HTML DOM, XML, JQUERY, ANGULAR, REACT, BABONE.JS, REDUX, STORYBOOK, GRAPHQL, METEOR.JS, GRUNT, GULF.

**HTML:** 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:** 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:** 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:** 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

**React.js:** 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:** 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:** 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.

**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.

The back end portion is built by using some libraries, frameworks, and languages.:

- **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++** It is a general purpose programming language and widely used now a days for competitive programming. It is also used as backend language.
- **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:** Python is a programming language that lets you work quickly and integrate systems more efficiently.
- **JavaScript:** Javascript can be used as both (front end and back end) programming languages.
- **Node.js:** 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.

**Database:** 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.

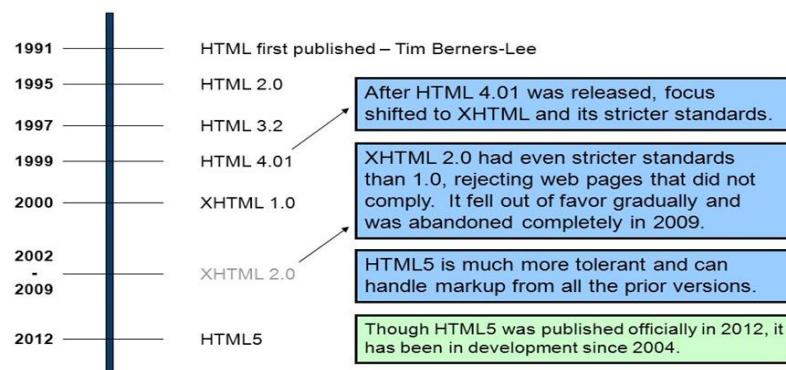
- **Oracle:** Oracle database is the collection of data which is treated as a unit. The purpose of this database is to store and retrieve information related to the query. It is a database server and used to manage information.
- **MongoDB:** MongoDB, the most popular NoSQL database, is an open source 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.s
- **Sql:** Structured Query Language is a standard Database language which is used to create, maintain and retrieve the relational database.

## HTML

HTML stands for **Hyper Text Markup Language**. It is the most widely used language on Web to develop web pages. HTML is a combination of Hypertext and Markup language. Hypertext defines the link between the web pages and markup language defines the text document within the tag that defines the structure of web pages.

**HTML** was created by Berners-Lee in late 1991 but "HTML 2.0" was the first standard HTML specification which was published in 1995. HTML 4.01 was a major version of HTML and it was published in late 1999. Though HTML 4.01 version is widely used but currently we are having HTML-5 version which is an extension to HTML 4.01, and this version was published in 2012.

### History of HTML



**Why to Learn HTML?** Originally, **HTML** was developed with the intent of defining the structure of documents like headings, paragraphs, lists, and so forth to facilitate the sharing of scientific information between researchers. Now, HTML is being widely used to format web pages with the help of different tags available in HTML language.

**HTML is Not Case Sensitive:** HTML tags are not case sensitive: <HTML> means the same as <html>.

**HTML** is a MUST for students and working professionals to become a great Software Engineer specially when they are working in Web Development Domain. To list down some of the key advantages of learning HTML:

- **Create Web site** - You can create a website or customize an existing web template if you know HTML well.
- **Become a web designer** - If you want to start a career as a professional web designer, HTML and CSS designing is a must skill.
- **Understand web** - If you want to optimize your website, to boost its speed and performance, it is good to know HTML to yield best results.
- **Learn other languages** - Once you understand the basic of HTML then other related technologies like JavaScript, php, or angular are become easier to understand.

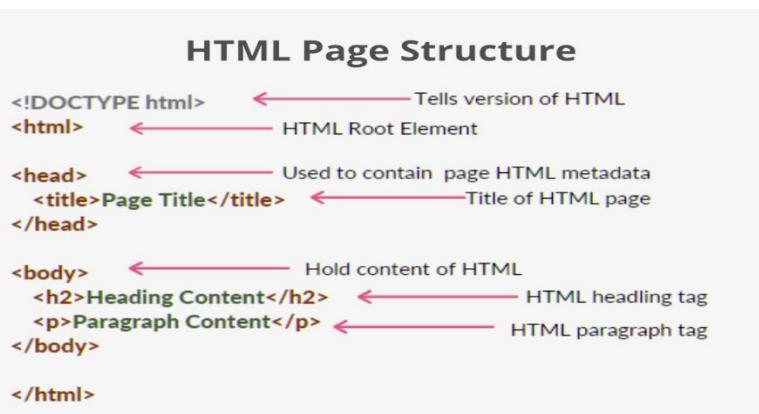
**Why HTML is used?** HTML is used to create the structure of web pages and website that are displayed on the Internet. It basically contains Tags and Attributes that are used to design the web pages. Also, we can link multiple pages using Hyperlinks.

**HTML Editors:** HTML text editors are used to create and modify web pages. HTML codes can be written in any text editor including the **notepad**. One just needs to write HTML in any text editor and save the file with an extension ".html" or ".htm". Some of the popular HTML text editors are given below:

- **Notepad:** notepad is a simple text editor. It is an inbuilt desktop application available in Windows OS. Create new file: File->New File or Ctrl+N. Write HTML code in text editor. Save the file with a suitable name of your choice and a **.html** extension. Open the saved HTML file in your favourite browser (double-click on the file, or right-click – and choose “Open with”).
- **Brackets:** Brackets is an open-source software primarily used for Web development. It provides live HTML, CSS, JavaScript editing functionality. Similar to Notepad editor, create new file and save with .html extension to run HTML file.
- **Sublime Text Editor:** Sublime is a cross platform code editor tool. It supports all markup languages. Similar to Notepad editor, create new file and save with .html extension to run HTML file.
- **Atom:** Atom is an open-source code editor tool for MAC, Linux and Windows. Similar to Notepad editor, create new file and save with .html extension to run HTML file.
- **Visual Studio Code:** Visual Studio Code. It is the one of the most popular code editors of Today's generation. Many companies and Software developer preferred these online code Editor.

**Step 1:** Open VS code Editor and Install Live Server on clicking the extension button simply search live server on search bar and download. Live server extension helps to run the code and display output.

**Step 2:** Create a new File and save with .html extension and use the open live server button to click right button.



## Basic Tags

- **<DOCTYPE! html>** – A doctype or document type declaration is an instruction that tells the web browser about the markup language in which the current page is written. It is not an element or tag. The doctype declaration is not case-sensitive.
- **<html>** – This tag is used to define the root element of an HTML document. This tag tells the browser that it is an HTML document. It is the second outer container element that contains all other elements within it.
- **<head>** – This tag is used to define the head portion of the HTML document that contains information related to the document. Elements within the head tag are not visible on the front end of a webpage.
- **<body>** – The body tag is used to enclose all the visible content of a webpage. In other words, the body content is what the browser will show on the front end.

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

```

<!-- Head Section content -->
<head>
    <!-- Page title -->
    <title>Basic Web Page</title>
</head>
<!-- Body Section content -->
<body>
    <!-- Used to display heading content -->
    <h1>Welcome to CMR Engineering College</h1>
    <!-- Used to display paragraph content -->
    <p>A computer science wing for CMR</p>
</body>
</html>

```

**HTML Element:** The HTML element consists of 3 parts.

- **Opening tag:** It is used to tell the browser where the content starts.
- **Closing tag:** It is used to tell the browser where the content ends.
- **Content:** It is the actual content material inside the opening and closing tags.  
Combining all these 3 parts results in an overall HTML Element

In this example `<p>` is a starting tag, `</p>` is an ending tag and it contains some content between the tags, which form an element.

```

<!DOCTYPE html>
<html>
<head>
    <title>HTML Elements</title>
</head>
<body>
    <p>Welcome to CMRIT!</p>
</body>
</html>

```

### **HTML Documents:**

All HTML documents must start with a document type declaration: `<!DOCTYPE html>`.

The HTML document itself begins with `<html>` and ends with `</html>`.

The visible part of the HTML document is between `<body>` and `</body>`.

Example

```

<!DOCTYPE html>
<html>
<body>
<h1>Welcome to CMR Engineering College</h1>
</body>
</html>

```

### **The `<!DOCTYPE>` Declaration:**

The `<!DOCTYPE>` declaration represents the document type, and helps browsers to display web pages correctly.

It must only appear once, at the top of the page (before any HTML tags).

The <!DOCTYPE> declaration is not case sensitive.

The <!DOCTYPE> declaration for HTML5 is:

```
<!DOCTYPE html>
```

**Necessary to add an end tag:** It is necessary to add the end tag of an element. Otherwise, the displayed content may or may not be displayed correctly. It is a good practice if you add closing tags to the non-void HTML Elements.

```
<!DOCTYPE html>
<html>
<head>
    <title>HTML Elements</title>
</head>
<body>
    <p>Hi CMR!
</body></html>
```

**HTML Headings:** These tags help us to give headings to the content of a webpage. These tags are mainly written inside the body tag. HTML provides us with six heading tags from <h1> to <h6>. Every tag displays the heading in a different style and font size.

**Example:** The use of 6 heading tags from <h1> to <h6> in HTML.

```
<!DOCTYPE html>
<html>
<head>
    <title>CMRGROUP</title>
</head>
<body>
    <h1> Hello CMR Engineering College </h1>
    <h2> Hello CMR Engineering College </h2>
    <h3> Hello CMR Engineering College </h3>
    <h4> Hello CMR Engineering College </h4>
    <h5> Hello CMR Engineering College </h5>
    <h6> Hello CMR Engineering College </h6>
</body> </html>
```

**HTML Paragraph:** These tags help us to write paragraph statements on a webpage. They start with the <p> tag and ends with </p>.

**HTML Break:** – These tags are used for inserting a single line type break. It does not have any closing tag. In HTML the break tag is written as <br>.

**Example:** This example illustrates the use of the <p> tag for writing a paragraph statement in HTML.

```

<!DOCTYPE html>
<html>
<head>
    <title>CMR GROUP</title>
</head>
<body>
    <h1>Hello CMRIT</h1>
    <p> A Computer Science portal for CMR<br>
        A Computer Science portal for CMR<br>
        A Computer Science portal for CMR<br>
    </p>
</body> </html>

```

**Horizontal Lines:** Horizontal lines are used to visually break-up sections of a document. The `<hr>` tag creates a line from the current position in the document to the right margin and breaks the line accordingly. If we want to give a line between two paragraphs-

### Example

```

<!DOCTYPE html>
<html>
<head>
    <title>Horizontal Line Example</title>
</head>
<body>
    <p> This is my first paragraph keep it on top</p>
    <hr />
    <p> This is my second paragraph keep it on bottom </p>
</body>
</html>

```

**o/p:** This is my first paragraph keep it on top

---

This is my second paragraph keep it on bottom

**Preserve Formatting:** Sometimes, you want your text to follow the exact format of how it is written in the HTML document. In these cases, you can use the preformatted tag `<pre>`. Any text between the opening `<pre>` tag and the closing `</pre>` tag will preserve the formatting of the source document.

### Example

```

<!DOCTYPE html>
<html>
<head>
    <title>Preserve Formatting Example</title>
</head>
<body>

```

```

<pre>
    function myFunction(){
        int a=10;
        System.out.Println("Hello");
        System.exit(0);
    </pre>
</body>
</html>

```

### O/P:

```

function myFunction(){
    int a=10;
    System.out.Println("Hello");
    System.exit(0);      }

```

**Nonbreaking Spaces:** The &nbsp; character entity is used to denote a non-breaking space which is a fixed space. This may be perceived as twice the space of a normal space. It is used to create a space in a line that cannot be broken by word wrap.

If we want to use Two spaces gap, we have to use &ensp; and to use four space – &emsp; is used.

### Syntax:

```

<!DOCTYPE html>
<html>
    <head>
        <title>Nonbreaking Spaces Example</title>
    </head>
    <body>
        <p>This is original text<br>
<h1>CMREngineeringCollege<h1> <br>
        <p>The text appears is "CMR&nbsp;Engineering &nbsp;College "</p>
    </body>
</html>

```

**O/P:** An example of this technique appears in the text is " CMR Engineering College"

### Nested HTML Elements

It is very much allowed to keep one HTML element inside another HTML element –

```

<body>
    <h1>I am studying in <i>CMR</i> Engineering College</h1>
    <p> I am studying in <u>CMR</u> Engineering College </p>
</body>

```

**Example 3:** This example describes the use of Nested HTML elements. Here, the <html> tag contains the <head> and <body>. The <head> and <body> tag contain other elements so it is called a nested element.

## **Empty HTML Elements:**

HTML Elements without any content i.e., that do not print anything are called Empty elements. Empty HTML elements do not have an ending tag. For instance. <br>, <hr>,<input> etc are HTML elements.

**Example 5:** In this example <br> tag doesn't print anything. It is used as a line break that breaks the line between <h2> and <p> tags.

```
<!DOCTYPE html>
<html>
<head>
    <title>Empty HTML Elements</title>
</head>
<body>
    <h2>Welcome To CMR</h2>
    <br />
    <p>Hello CMR.</p>
</body>
</html>
```

**HTML Paragraph:** The <p> tag in HTML defines a paragraph. These have both opening and closing tags. So anything mentioned within <p> and </p> is treated as a paragraph. A paragraph is a block-level element so a new paragraph always begins on a new line.

**Syntax:** <p> Content </p>

**<br tag>:** The HTML <br> tag element creates a line break, giving you a new line without starting a new paragraph. Use `<br>` when you want to move to the next line without beginning a whole new paragraph.

Syntax: <br>

**<hr tag>:** The HTML <hr> tag is used to create a horizontal rule or line, visually separating content on a webpage. Use <hr> when you want to insert a horizontal line to signify a division between sections or elements, providing a clear visual break in the page.

Syntax: <hr>

**Align Attribute:** The <p> tag specifically supports the alignment attribute and allows us to align our paragraphs in left, right, or center alignment.

Syntax: <p align="value">

```
<!DOCTYPE html>
<html>
<body>
    <p align="center">
        Welcome to CMR
    </p>
    <p align="left">
        A Computer Science and Engineering.
    </p>
</body>
</html>
```

```

</p>
<p align="right">
    Now the alignment is right.
</p>
</body>
</html>

```

## HTML Text Formatting

HTML facilitates the ability for formatting text just like we do in MS Word. To display text in HTML strong, small, and Highlight formatting respectively.

```

<!DOCTYPE html>
<html> <body>
    <h2>Welcome To CMRCollege</h2>
        <!--Text in Strong-->
        <strong>Hello CMR Group</strong>
        <br>
            <!--Text in small-->
        <small>Hello CMR</small>
        <br>
            <!--Text in Highlight-->
        <mark>Hello CMRIT</mark>
</body> </html>

```

**Making text Bold or Strong:** We can make the text **bold** using the **<b>** tag. The tag uses both opening and closing tags. The text that needs to be made bold must be within **<b>** and **</b>** tag. We can also use the **<strong>** tag to make the text strong, with added semantic importance. It also opens with **<strong>** and ends with **</strong>** tag.

```

<!DOCTYPE html>
<html>
<head>
    <title>Bold Text</title>
</head>
<body>
    <!--Normal text-->
    <p>Hello CMR Group</p>
        <!--Text in Bold-->
    <p> <b>Hello CMR</b> </p>
        <!--Text in Strong-->
    <p><strong>Hello CMRIT</strong></p>
</body>
</html>

```

**Making text Italic or emphasize:** The <i> tag is used to italicise the text. It opens with <i> and ends with </i> tag. The <em> tag is used to emphasize the text, with added semantic importance. It opens with <em> and ends with </em> tag. It is also possible to highlight a text in HTML using the <mark> tag.

```
<!DOCTYPE html>
<html> <head>
    <title>Italic</title>
</head>
<body>
    <!--Normal text-->
    <p>Hello CMR Group</p>
    <!--Text in Italics-->
    <p>
        <i>Hello CMR</i>
    </p>
    <!--Text in Emphasize-->
    <p>
        <em>Hello CMRIT</em>
    </p>
    <!--Text in Highlight-->
    <p>
        <mark>Hello CMRCOLLEGE</mark>
    </p>
</body> </html>
```

**Making a text Subscript or Superscript:** The <sup> element is used to superscript a text and the <sub> element is used to subscript a text. They both have an opening and a closing tag.

```
<!DOCTYPE html>
<html>
<head><title>Superscript and Subscript</title>
</head><body>
    <!--Text in Normal-->
    <p>Hello CMR Group</p>
    <!--Text in Superscript-->
    <p>Hello
        <sup>CMR</sup>
    </p>
    <!--Text in Subscript-->
    <p>Hello
        <sub>CMRIT</sub>
    </p>
</body></html>
```

The **<small>** element is used to make the text smaller. The text that needs to be displayed smaller should be written inside `<small>` and `</small>` tag.

```
<!DOCTYPE html>
<html><head>
    <title>Small</title>
</head>
<body>
    <!--Text in Normal-->
    <p>Hello CMR Group</p>
    <!--Text in small-->
    <p>
        <small>Hello CMRIT</small>
    </p>
</body> </html>
```

**Striking through the text:** The **<del>** element is used to strike through the text marking the part as deleted. It also has an opening and a closing tag.

```
<!DOCTYPE html>
<html>
<head>
    <title>Delete</title>
</head>
<body>
    <!--Text in Normal-->
    <p>Hello CMR Group</p>
    <!--Text in Delete-->
    <p> <del>Hello CMRIT</del> </p>
</body>
</html>
```

**Adding a text:** The **<ins>** element is used to underline a text marking the part as inserted or added. It also has an opening and a closing tag.

```
<!DOCTYPE html>
<html><head>
    <title>Inserting the Text</title>
</head>
<body>
    <!--Text in Normal-->
    <p>Hello CMR Group</p>
    <!--Text in Insert-->
    <p>
        <ins>Hello CMRIT</ins>
    </p>
</body></html>
```

**HTML Quotations:** The Quotation elements in HTML are used to insert quoted texts in a web page, that is the portion of texts different from the normal texts in the web page. Below are some of the most used quotation elements of HTML:

**<q> element:** The `<q>` element is used to set a set of text inside the quotation marks. It has both opening and closing tags.

```
<!DOCTYPE html>
<html>
<head>
    <title>Quotations</title>
</head>
<body>
    <h3>CMR Group</h3>
    <p>I am studying at CMR<br></p>
    <!--Inside quotes-->
    <p><q>I am studying at CMR</q></p>
</body></html>
```

**<blockquote> element:** The `<blockquote>` element is also used for quotations in a different way. Instead of putting the text in quotes, it adds space before the start of the sentence, with this tag, we can also indent the start of the new paragraph. It has both opening and closing tags.

```
<!DOCTYPE html>
<html>
<head>
    <title>Blockquote</title>
</head>
<body>
    <h3>CMRCollage</h3>
    <p>The quick brown fox jumps over the lazy dog<br></p>
    <!--Inside blockquotes-->
    <p>
        <blockquote>The quick brown fox jumps over the lazy dog</blockquote>
    </p>
</body>
</html>
```

#### **HTML <mark> Element:**

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

**Example**

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

#### **HTML <del> Element:**

The HTML `<del>` element defines text that has been deleted from a document. Browsers will usually strike a line through deleted text:

## Example

```
<p>My favorite color is <del>blue</del> red.</p>
```

O/p: My favorite color is blue red.

## HTML <ins> Element:

The HTML <ins> element defines a text that has been inserted into a document. Browsers will usually underline inserted text:

**<abbr> element:** The <abbr> element is used to define a text as an acronym or abbreviation. The title attribute can be used to show the full version of the abbreviation/acronym when you hover the mouse over the <abbr> element. It has both opening and closing tags. This is useful for browsers and search engines.

```
<p>Welcome to <abbr title="CMRGroupofInstitutions">CGI</abbr></p>
```

**<bdo> element:** The <bdo> element is used to define a bidirectional override which means that, the text is written from right to left or left to right. It has both opening and closing tags. It is used to over-ride the current text direction. It takes an attribute "rtl" to display the text from right to left.

```
<!DOCTYPE html>
<html>
<head>
<title>CMR Engg College</title>
<h1><center>CMR GROUP<center></h1>
</head>
<body>
<p> I am stdyding in CMRIT</p>
<b> I am stdyding in CMRIT1</b><br>
<i> I am stdyding in CMRIT1</i><br>
<u> I am stdyding in CMRIT1</u><br>
<small> I am stdyding in CMRIT1</small><br>
<strong> I am stdyding in CMRIT1</strong><br>
<big> I am stdyding in CMRIT3</big><br>
<strike> I am stdyding in CMRIT1</strike><br>
<p> This is the top of the line<hr>
This is the bottom of the line</p>
<br>
<em>I am stdyding in CMRIT</em>
<pre>
function abc(){
int a=20;
System.out.println(a);
System.exit(0);
}
</pre><br>
I love my <del>Dad</del>Mom <br>
I love my <del>Dad</del><br>
<ins>Mom</ins><br>
<del>blue</del>red<br>
<p> I will come to <mark>college</mark>regularly</p>
<p>CMR Engineering College</p><br>
<bdo dir="rtl">CMR Engineering College</bdo><br>
<q>I am from CMRIT</q><br>
H<sub>2</sub>O<sub>2</sub>
<abbr title="CMR Engg College">CMR</abbr>
CMR&nbsp;Engineering&nbsp;College
</body>
</html>
```

## CMR GROUP

I am stdyding in CMRIT

I am stdyding in CMRIT1  
I am stdyding in CMRIT3  
I am stdyding in CMRIT1

This is the top of the line

This is the bottom of the line

I am stdyding in CMRIT

function abc(){
int a=20;
System.out.println(a);
System.exit(0);
}

I love my DadMom  
I love my Dad  
Mom  
blueed

I will come to college regularly  
CMR Engineering College

egelloC gnireenignE RMC  
"I am from CMRIT"  
H<sub>2</sub>O, CMR CMR Engineering College

## Example:

```
<!DOCTYPE html>
<html><head>
    <title>Bidirectional</title>
</head>
<body>
    <h3>CMR GROUP</h3>
    <!--Normal text-->
    <p>I am Studying at CMRIT<br></p>
    <!--Inside <bdo> tag-->
    <p>
        <bdo dir="rtl"> I am studying at CMRIT </bdo>
    </p>
</body> </html>
```

O/P: TIRMC ta gniydu ma I

## **HTML LINK**

It is a connection from one web resource to another. A link has two ends, An anchor and direction. The link starts at the “source” anchor and points to the “destination” anchor, which may be any Web resource such as an image, a video clip, a sound bite, a program, an HTML document or an element within an HTML document.

This basically means that by using the ‘a’ tag, you can link 1 element of the code to another element that may/may not be in your code.

Links are specified in HTML using the “a” tag.

```
<a href="url" > Text</a>
```

### **Syntax Explanation:**

**href :** The href attribute is used to specify the destination address of the link used.

“href” stands for Hypertext reference.

**Text link :** The text link is the visible part of the link. It is what the viewer clicks on

```
<!DOCTYPE html>
<html>
<h3>Example of Adding a link</h3>
<body>
    <p>Click on the following link</p>
    <a href="https://www.cmrithyderabad.ac.in">CMRIT</a>
</body></html>
```

### **Using Image as a Link in HTML**

An image can be used to create a link to a specified URL. When the viewer clicks on the link, it redirects them to another page.

The code is `<a href="url">`

```
</a>
```

**Note: img src** stands for image source ( i.e URL or file address )

```
<!DOCTYPE html>
<html>
<body>
    <h3>Using Image as a link</h3>
    <p>Click on the image to visit cmrit homepage.</p>
    <a href= "https://www.cmrithyderabad.ac.in">
        </a>
</body> </html>
```

### **HTML Image**

Here, we will know how to add images to the web page that will make the website attractive & various methods to insert the images.

There are 2 ways to insert the images into a webpage:

- By providing a full path or address (URL) to access an internet file.
- By providing the file path relative to the location of the current web page file.

**Adding images on a webpage:** The `<img>` tag is used to add or embed the images to a webpage/website. The “img” tag is an empty tag, which means it can contain only a list of attributes and it has no closing tag.

**Syntax:** ``

**Attribute:** Most commonly used tags `<img>` with attributes:

- **src**: It is used to specify the path to the image.
- **alt**: It is used to specify an alternate text for the image. It is useful as it informs the user about what the image means and also due to any network issue if the image cannot be displayed then this alternate text will be displayed.
- **height**: It is used to specify the height of the image.
- **width**: It is used to specify the width of the image.

**src:** The src stands for source. Every image has an src attribute which tells the browser where to find the image you want to display. The URL of the image provided points to the location where the image is stored. When the webpage loads for the first time, then the browser gets the image from a web server and inserts it into the page. If the image is not spotted by the browser, then users will get a broken link icon. It might be possible if the file path is wrong or the image got deleted from that location.

**Example 1:** This simple example illustrates the use of the `<img>` tag in HTML that is used to embed the image into the webpage.

```
<!DOCTYPE html>
<html><head>
    <title>Welcome To CMR</title>
</head>
<body>
    <h2>CMRCollage</h2>
    <p>This is the demo of <img> tag.</p>
    <img src= https://cmrhyderabad.ac.in/logo.png alt="CMR image" />
</body></html>
```

**alt:** If the image cannot be displayed then the alt attribute acts as an alternative description for the image. The value of the alt attribute is a user-defined text.

**Setting width and height of Image:** The width and height attributes are used to specify the height and width of an image. The attribute values are specified in pixels by default. The width and height attributes are always declared in pixels.

**Setting style to the Image:** By using the border attribute, the thickness of the border can be changed. A thickness of “0” means that there will be no border around the picture.

**Adding titles to Image:** Along with the images, titles can also be added to images to provide further information related to the inserted image.

**Aligning an Image:** By default, an image is aligned on the left side of the page, but it can be aligned to the centre or right using the align attribute.

**Adding animated Image:** Animated images in .gif format can also be added using the “img” tag. `inserted image using <img> tag:</p>

```

alt="This is CMR logo" width="300" height="300" border="5" title="Logo of CMRIT" align="right"/> </body>

Common Image Format: Here is the commonly used image file format that is supported by all the browsers.

| <b>SNo</b> | <b>Abbreviation</b> | <b>File Type</b>                       | <b>Extension</b>                 |
|------------|---------------------|----------------------------------------|----------------------------------|
| 1.         | PNG                 | Portable Network Graphics.             | .png                             |
| 2.         | JPEG.               | Joint Photographic Expert Group image. | .jpg, .jpeg, .jfif, .pjpeg, .pjp |
| 3.         | SVG                 | Scalable Vector Graphics.              | .svg.                            |
| 4.         | GIF                 | Graphics Interchange Format.           | .gif                             |
| 5.         | ICO                 | Microsoft Icon.                        | .ico, .cur                       |
| 6.         | APNG                | Animated Portable Network Graphics.    | .apng                            |

### **HTML Emoji's**

Emojis are letters (characters) from the UTF-8 (Unicode) character set. The `<meta charset="UTF-8">` element in HTML defines the character set. Many UTF-8 characters cannot be typed on a keyboard, but they can always be displayed using numbers (called entity numbers). To let the browser understand that you are displaying a character, you must start the entity number with `&#` and end it with; (semicolon).

Syntax:

```
<head>
  <meta charset="UTF-8">
</head>
<body>
  <p>&#number;</p>
  // Take 'number' corresponding// to UTF-8 characters you want // to display.
</body>
```

The screenshot shows a dual-monitor setup. The left monitor displays the Visual Studio Code interface with the following details:

- EXPLORER**: Shows a folder named "ImgFolder" containing files: 4\_June\_2022.png, Baby2.jpg, Baby3.jpg, Flower1.jpg, Flower2.jpg, and Hand-baby.jpg.
- FILES**: Shows files: AllTags.html, app.js, Footer.html, FormControls.html, Frames.html, Lists.html, Middle.html, SelectUploadButton.h..., Tables.html, and TopFrame.html.
- OUTLINE**: Shows the outline of the current file, AllTags.html.
- TIMELINE**: Shows the timeline of the current file, AllTags.html.

The right monitor displays a web browser window with the URL `127.0.0.1:5500/`. The page content is as follows:

```
<html>
  <body>
    <i><h1><font color="Red"> I am from CMRIT</font>
    <b>One:</b> <i> I am from CMRIT</i><br>
    <b>Two:</b> <u>I am from CMRIT</u><br>
    <b>Three:</b> <q>I am from CMRIT</q>
    <b>Four:</b> <strike>CMRIT</strike><br>
    <b>Five:</b> <strong>CMRIT</strong><br>
    <b>Six:</b><h1>CMRIT</h1><br>
    <b>Seven:</b> <b><small>CMRIT</small></b><br>
    <b>Eight:</b> I am from<del>CMRIT</del><br>
    <b>Nine:</b> <abbr title="World Wide Web">WWW</abbr>
    <b>Ten:</b> </b><abbr title="Water">H<sub>2</sub>O</abbr>
    <b>Eleven:</b> X<sup>2</sup><br>
    <b>Twelve:</b> <bdo dir="rtl">CMRIT</bdo>
  </body>
</html>
```

The browser also shows a status bar with the URL `127.0.0.1:5500/`, a search bar with "Search", and system icons for battery, signal, and time (10:19).

## Some more Tags

## HTML Block and Inline Elements:

Every HTML element has a default display value, depending on what type of element it is. There are two display values: block and inline.

## **Block-level Elements:**

A block-level element always starts on a new line.

A block-level element always takes up the full width available.

A block level element has a top and a bottom margin, whereas an inline element does not. The `<div>` element is a block-level element.

**Example:** <div>Hello World</div>

**Inline Elements:** An inline element does not start on a new line.

An inline element only takes up as much width as necessary.

**Example:** <span>Hello World</span>

**The <div> Element:** The <div> element is often used as a container for other HTML elements. The <div> element has no required attributes, but style, class and id are common.

**Example:** <div style="background-color:black;color:white;padding:20px;">

## CMR College

**CMR** is one of the best colleges in Telangana. It is the most populous college in Telangana.

The <span> Element:

## The <span> Element:

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

The `<span>` element has no required attributes, but style, class and id are common.

When used together with CSS, the `<span>` element can be used to style parts of the text:

**Example:**

```
<p>My mother has <span style="color:blue;font-weight:bold">blue</span> eyes and my father has <span style="color:darkolivegreen;font-weight:bold">dark green</span> eyes.</p>
```

**HTML Lists:**

HTML lists are used to present list of information in well-formed and semantic way. There are three different types of lists in HTML and each one has a specific purpose and meaning.

- **Unordered list** — Used to create a list of related items, in no particular order.
- **Ordered list** — Used to create a list of related items, in a specific order.
- **Description list** — Used to create a list of terms and their descriptions.

**HTML Unordered Lists:**

An unordered list created using the `<ul>` element, and each list item starts with the `<li>` element.

The list items in unordered lists are marked with disc, circle or square.

**Example:**

```
<h1><Center> UNORDRED LISTS </center></h1>
<h3> My College CMRIT Branches </h3>
<ul style=list-style-type:circle>
<li> CSE </li>
<li> AIML </li>
<li> DS </li>
<li> AIDS </li>
</ul>
```

**HTML Ordered Lists:** An ordered list created using the `<ol>` element, and each list item starts with the `<li>` element. Ordered lists are used when the order of the list's items is important.

The list items in an ordered list are marked with numbers.

**Example:**

```
<h1><Center> ORDRED L ISTS </center></h1>
<h3> CMR Group Colleges </h3>
<ol type="a">
<li> CMRIT </li>
<li> CMREC </li>
<li> CMRTC </li>
<li> CMR Pharmacy</li>
</ol>
```

**HTML Description Lists:**

A description list is a list of items with a description or definition of each item.

The description list is created using `<dl>` element. The `<dl>` element is used in conjunction with the `<dt>` element which specify a term, and the `<dd>` element which specify the term's definition.

**Example:**

```
<h1><Center> DATA LISTS </center></h1>
<h3> Some words describing on whether </h3>
<div>
<dl>
<dt> Sunny</dt>
<dd> It's a sunny day, Let's go to beach </dd> </dl>
</div>

<div>
<dl>
<b><dt> Cloudy</dt></b>
<dd> The sky is cloudy, It might rain later</dd> </dl>
</div>

<div>
<dl>
<dt> Rainy</dt>
<dd> It's a rainy day, don't forget your umbrella</dd> </dl>
</div>
```

**Program to display output:**

```

1 lists.html > html
2   <html>
3     <head>
4       <title> CMR COLLEGE</title>
5       <h1> Lists </h1>
6     </head>
7     <a href=".\\ClassPrograms\\AllTags.html">CMR</a>
8     <h1><Center> ORDRED LISTS </center></h1>
9     <h3> CMR Group Colleges </h3>
10    <ol type="a">
11      <li> CMRIT </li>
12      <li> CMREC </li>
13      <li> CMRTC </li>
14      <li> CMR Pharmacy</li>
15    </ol>
16    <br>
17
18    <h1><Center> UNORDRED LISTS </center></h1>
19    <h3> My College CMRIT Branches </h3>
20    <ul style=list-style-type:circle>
21      <li> CSE </li>
22      <li> AIML </li>
23      <li> DS </li>
24      <li> AIDS </li>
25    </ul>
26
27    <h1><Center> DATA LISTS </center></h1>
28    <h3> Some words describing on whether </h3>
29    <div>
30      <dl>
31        <dt> Sunny</dt>
32        <dd> It's a sunny day, Let's go to beach </dd> </dl>
33    </div>
34
35    <div>
36      <dl>
37        <b><dt> Cloudy</dt></b>
38        <dd> The sky is cloudy, It might rain later</dd> </dl>
39    </div>
40
41    <div>
42      <dl>
43        <dt> Rainy</dt>
44        <dd> It's a rainy day, don't forget your umbrella</dd> </dl>
45    </div>

```

## Lists

CMR

### ORDRED LISTS

CMR Group Colleges

- a. CMRIT
- b. CMREC
- c. CMRTC
- d. CMR Pharmacy

### UNORDRED LISTS

My College CMRIT Branches

- o CSE
- o AIML
- o DS
- o AIDS

### DATA LISTS

Some words describing on whether

Sunny	It's a sunny day, Let's go to beach
Cloudy	The sky is cloudy, It might rain later
Rainy	It's a rainy day, don't forget your umbrella

## HTML Table

### Define an HTML Table:

Table is a collection of rows and columns.

The <table> tag defines an HTML table.

Each table row is defined with a <tr> tag. Each table header is defined with a <th> tag.  
Each table data/cell is defined with a <td> tag.

By default, the text in <th> elements are bold and centered.

By default, the text in <td> elements are regular and left-aligned.

We can add different properties to get the new view port.

### Example:

```

<!doctype>
<html>
<head>
<title> CMR COLLEGE</title>
<h1> Table </h1>
</head>
<body>
<h1><Center> Table of Contents </center></h1>

<table border=5 bgcolor="pink" style="width:50%">
<caption><h1><center>Student Description</center></h1></caption>

```

```
<tr bgcolor="Red">
<th style=width:20px > Sno</th>
<th style=width:80px> Rno</th>
<th style=width:100> SName</th>
<th style=width:50> CGPA</th>
</tr>

<tr bgcolor="blue">
<td> 1</td>
<td> 67C1</td>
<td> Kumar</td>
<td> 8.0</td>
</tr>

<tr>
<td> 2</td>
<td> 67C2</td>
<td> Kumar</td>
<td> 8.1</td>
</tr>

<tr>
<td> 3</td>
<td> 67C3</td>
<td> Tapaswi</td>
<td> 8.2</td>
</tr>

<tr>
<td> 4</td>
<td> 67C4</td>
<td> Satwik</td>
<td> 8.5</td>
</tr>

</body>
</html>
```

```

<!-- Tables.html -->
<html>
  <head>
    <h1> Table of Contents </h1>
  </head>
  <body>
    <h1><Center> Table of Contents </center></h1>
    <table border=5 bgcolor="pink" style="width:50%">
      <caption><h1><center>Student Description</center></h1></caption>
      <tr bgcolor="Red">
        <th style="width:20px > Sno</th>
        <th style="width:80px> Rno</th>
        <th style="width:100> SName</th>
        <th style="width:50> CGPA</th>
      </tr>
      <tr bgcolor="blue">
        <td> 1</td>
        <td> 67C1</td>
        <td> Kumar</td>
        <td> 8.0</td>
      </tr>
      <tr>
        <td> 2</td>
        <td> 67C2</td>
        <td> Kumar</td>
        <td> 8.1</td>
      </tr>
      <tr>
        <td> 3</td>
        <td> 67C3</td>
        <td> Tapaswi</td>
        <td> 8.2</td>
      </tr>
      <tr>
        <td> 4</td>
        <td> 67C4</td>
        <td> Satwik</td>
        <td> 8.5</td>
      </tr>
    </table>
  </body>
</html>

```

Table

## Table of Contents

### Student Description

Sno	Rno	SName	CGPA
1	67C1	Kumar	8.0
2	67C2	Kumar	8.1
3	67C3	Tapaswi	8.2
4	67C4	Satwik	8.5

**Using Col-Span:** To make a cell span over multiple columns, use the colspan attribute.

```

<!-- AllTags.html -->
<!-- lists.html -->
<!-- Tables.html -->
<!-- Tab-Row-Col.html -->
<!-- Tab-Col-Span.html -->
<html lang="en">
  <head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
  </head>
  <body>
    <table border="5">
      <tr>
        <td colspan="4" bgcolor="Red"><center>PG</center></td>
        <td colspan="4" bgcolor="Green"><center>UG</center></td>
      </tr>
      <tr>
        <td bgcolor="blue">M.tech</td>
        <td>M.B.A</td>
        <td>M.Com</td>
        <td>M.C.A</td>
        <td>B.tech</td>
        <td>B.B.A</td>
        <td>B.Com</td>
        <td>B.C.A</td>
      </tr>
      <tr>
        </tr>
      </table>
    </body>
</html>

```

127.0.0.1:5500/Tab-Col-Span.html

PG	UG
M.tech	M.B.A M.Com M.C.A B.tech B.B.A B.Com B.C.A

**Using Row-Span:** To make a cell span over multiple rows, use the rowspan attribute.

The screenshot shows a browser window with the URL 127.0.0.1:5500/Tab-Row-Col.html. The page title is "Table of Contents". Below it is a section titled "Row-Span and Column-Span" with a table. The table has two columns and four rows. The first column contains "PG" and "UG". The second column contains four rows for each level, labeled "M.tech", "M.com", "M.Sc", and "M.B.A" for PG, and "B.tech", "B.com", "B.Sc", and "B.B.A" for UG. The code for this table is shown in the left panel of the browser.

```
<!DOCTYPE html>
<html>
<head>
<title> CMR COLLEGE</title>
</head>
<body>
<h1><Center> Table of Contents </center></h1>
<table border=5 bgcolor="pink">
<caption><h5><center><b>Row-Span and Column-Span</b></center></h5>
</caption>
<tr>
<td rowspan="4"> PG
<td> M.tech

|
<td> M.com

|  |
|
<td> M.Sc

|
<td> M.B.A


<tr>
<td rowspan="4"> UG
<td> B.tech

|
<td> B.com

|
<td> B.Sc

|
<td> B.B.A


</tr>
</table>
</body>
</html>
```

## Frames

HTML frames are used to divide our browser window into multiple sections where each section can load a separate HTML document. The window is divided into frames in a similar way the tables are organized: into rows and columns.

### Disadvantages of Frames

There are few drawbacks with using frames, so it's never recommended to use frames in your webpages –

Some smaller devices cannot cope with frames often because their screen is not big enough to be divided up.

Sometimes your page will be displayed differently on different computers due to different screen resolution.

The browser's back button might not work as the user hopes.

There are still few browsers that do not support frame technology.

-----Program-----

### **Frames.html**

```
<!DOCTYPE html>
<html lang="en">
<head>
```

```

<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Frames</title>
<frameset rows="50%,50%" >
  <frame src= "TopFrame.html" name= "top">
  <frameset cols="30%,40%,30%" >
    <frame src= "Middle.html" name= "mid"/>
    <frame src= "BottomMiddle.html" name= "top1"></frame>
    <frame src= "Footer.html" name= "foot"/>
  </frameset>
</frameset>
</head>
<body>
</body>
</html>

```

### **TopFrame.html**

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
</head>
<body bgcolor="blue">
  <h1><center>Top Frame</center></h1>
</body>
</html>

```

### **BottomMiddle.html**

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
</head>
<body bgcolor="pink">
  <h1><center>Bottom Middle Frame</center></h1>
</body>
</html>

```

### **Footer.html**

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
</head>

```

```

<body bgcolor="Orange">
  <h1>Right Bottom</h1>

</body>
</html>

```

The screenshot shows the VS Code interface with the 'Frames.html' file open in the editor. The code defines a frameset with three frames: 'TopFrame.html' (top), 'Middle.html' (middle), and 'Footer.html' (bottom). To the right, a browser window displays the rendered frameset. The top frame is blue and labeled 'Top Frame'. The bottom frame is pink and labeled 'Bottom Middle Frame'. The right frame is orange and labeled 'Right Bottom'. Below the browser window, a legend identifies the colors: green for 'Left bottom', pink for 'Bottom Middle Frame', and orange for 'Right Bottom'.

```

<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Frames</title>
  </head>
  <body>
    <frameset rows="50%,50%">
      <frame src= "TopFrame.html" name= "top">
      <frameset cols="30%,40%,30%">
        <frame src= "Middle.html" name= "mid"/>
        <frame src= "BottomMiddle.html" name= "top1"/>
        <frame src= "Footer.html" name= "foot"/>
      </frameset>
    </frameset>
  </body>
</html>

```

## Course.html

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
</head>
<body bgcolor="palebrown">
  <font size="14">
    <ul style="list-style-type:disc">
      <center><li>Full Stack Web Development</li>
      <li>Machine Learning</li>
      <li>Network Security</li>
      <li>Computer Networks</li></center>
    </ul>
  </font>
</body>
</html>

```

The screenshot shows a code editor interface with the following details:

- File Bar:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Explorer:** Shows a tree view of files under "CLASS-3".
- Code Editor:** Displays the content of "CMR-Site-Frames.html". The code uses framesets to divide the page into sections like logo, header, menu, main content, and footer.
- Browser Preview:** A browser window titled "CMR Engineering College" with a red header bar. It shows a yellow sidebar with links: Home, About Us, Courses, and Contact. The main content area says "Hello CMR!". At the bottom right, there is contact information: Name: CMR, Contact No: 9247000000, WhatsApp No: 9200000011, Email: msg2cmr@gmail.com.

## About.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
</head>
<body bgcolor="pink">
    <font size="20">
        <center>About</center>
    </font>
    <center><p> Myself Narendra, studying in CMR Engineering College</p></center>
</body>
</html>
```

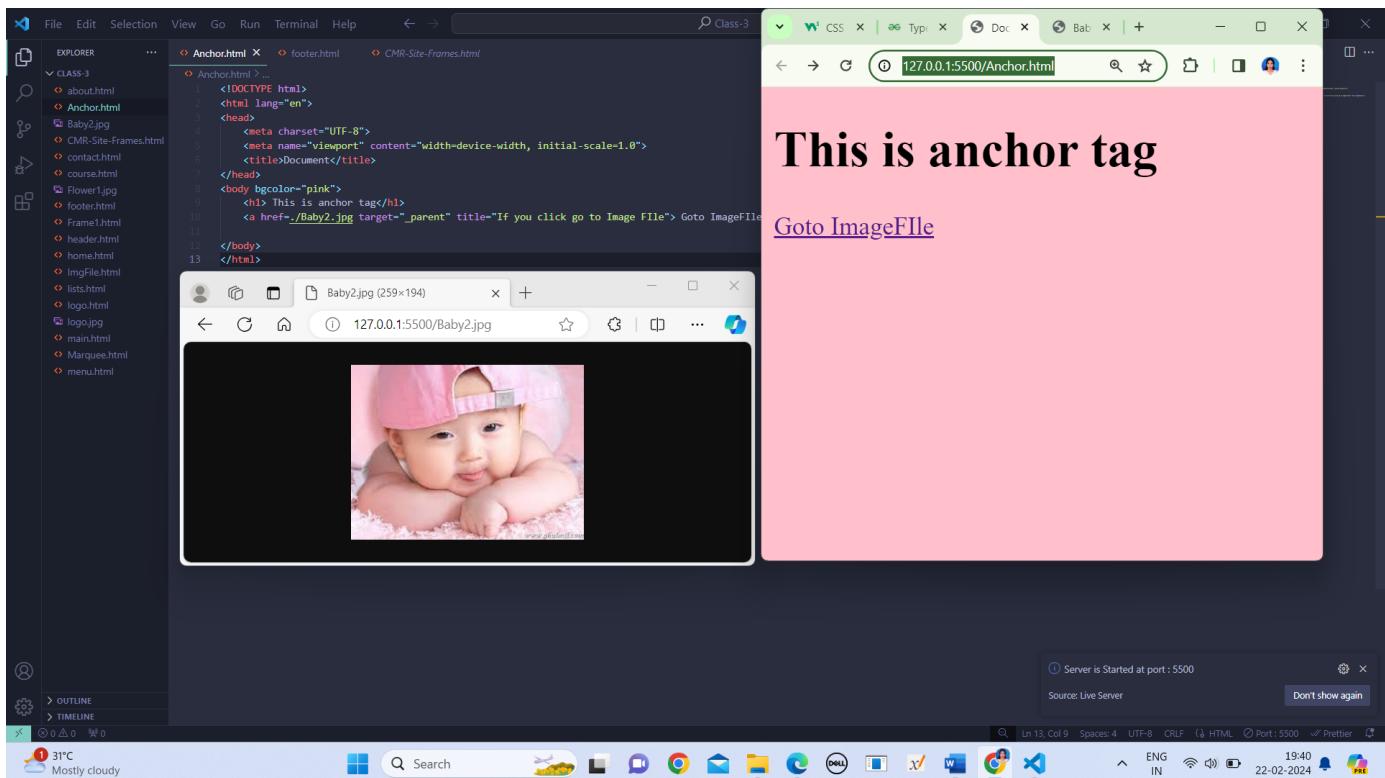
## Anchor.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
</head>
<body bgcolor="pink">
```

```

<h1> This is anchor tag</h1>
<a href= ./Baby2.jpg target=_parent title="If you click go to Image File" > Goto
ImageFile</a>
</body>
</html>

```



### HTML Form:

HTML Form is a collection of elements. Forms are required to collect different kinds of user inputs, such as contact details like name, email address, phone numbers, or details like credit card information, etc.

Forms contain special elements called controls like inputbox, checkboxes, radio-buttons, submit buttons, etc. Users generally complete a form by modifying its controls e.g. Entering text, selecting items, etc. and submitting this form to a web server for further processing.

The `<form>` tag is used to create an HTML form.

```

<form>
  <label>Username: <input type="text"></label>
  <label>Password: <input type="password"></label>
  <input type="submit" value="Submit">
</form>

```

**Input Element:** This is the most commonly used element within HTML forms.

It allows you to specify various types of user input fields, depending on the type attribute. An input element can be of type *text field, password field, checkbox, radio button, submit button, reset button, file select box*.

#### **Text Fields:**

Text fields are one-line areas that allow the user to input text.

Single-line text input controls are created using an `<input>` element, whose type attribute has a value of text.

**Example:**

```
<form>
  <label for="username">Username:</label>
  <input type="text" name="username" id="username">
</form>
```

**Password Field:**

Password fields are similar to text fields. The only difference is; characters in a password field are masked, i.e. they are shown as asterisks or dots. This is to prevent someone else from reading the password on the screen. This is also a single-line text input controls created using an `<input>` element whose type attribute has a value of password.

*Example*

```
<form>
  <label for="user-pwd">Password:</label>
  <input type="password" name="user-password" id="user-pwd">
</form>
```

**Radio Buttons:**

Radio buttons are used to let the user select exactly one option from a pre-defined set of options. It is created using an `<input>` element whose type attribute has a value of radio.

```
<form>
<input type="radio" name="gender" id="male">
<label for="male">Male</label>
<input type="radio" name="gender" id="female">
<label for="female">Female</label>
</form>
```

**Checkboxes:**

Checkboxes allows the user to select one or more option from a pre-defined set of options. It is created using an `<input>` element whose type attribute has a value of checkbox.

**Example:**

```
<form>
  <input type="checkbox" name="sports" id="soccer">
  <label for="soccer">Soccer</label>
  <input type="checkbox" name="sports" id="cricket">
  <label for="cricket">Cricket</label>
  <input type="checkbox" name="sports" id="baseball">
  <label for="baseball">Baseball</label>
</form>
```

## **Textarea:**

Textarea is a multiple-line text input control that allows a user to enter more than one line of text. Multi-line text input controls are created using an `<textarea>` element.

### **Example:**

```
<form>
  <label for="address">Address:</label>
  <textarea rows="3" cols="30" name="address" id="address"></textarea>
</form>
```

## **Submit and Reset Buttons:**

A submit button is used to send the form data to a web server. When submit button is clicked the form data is sent to the file specified in the form's `action` attribute to process the submitted data.

A reset button resets all the forms control to default values.

```
<form action="action.php" method="post">
  <label for="first-name">First Name:</label>
  <input type="text" name="first-name" id="first-name">
  <input type="submit" value="Submit">
  <input type="reset" value="Reset">
</form>
```

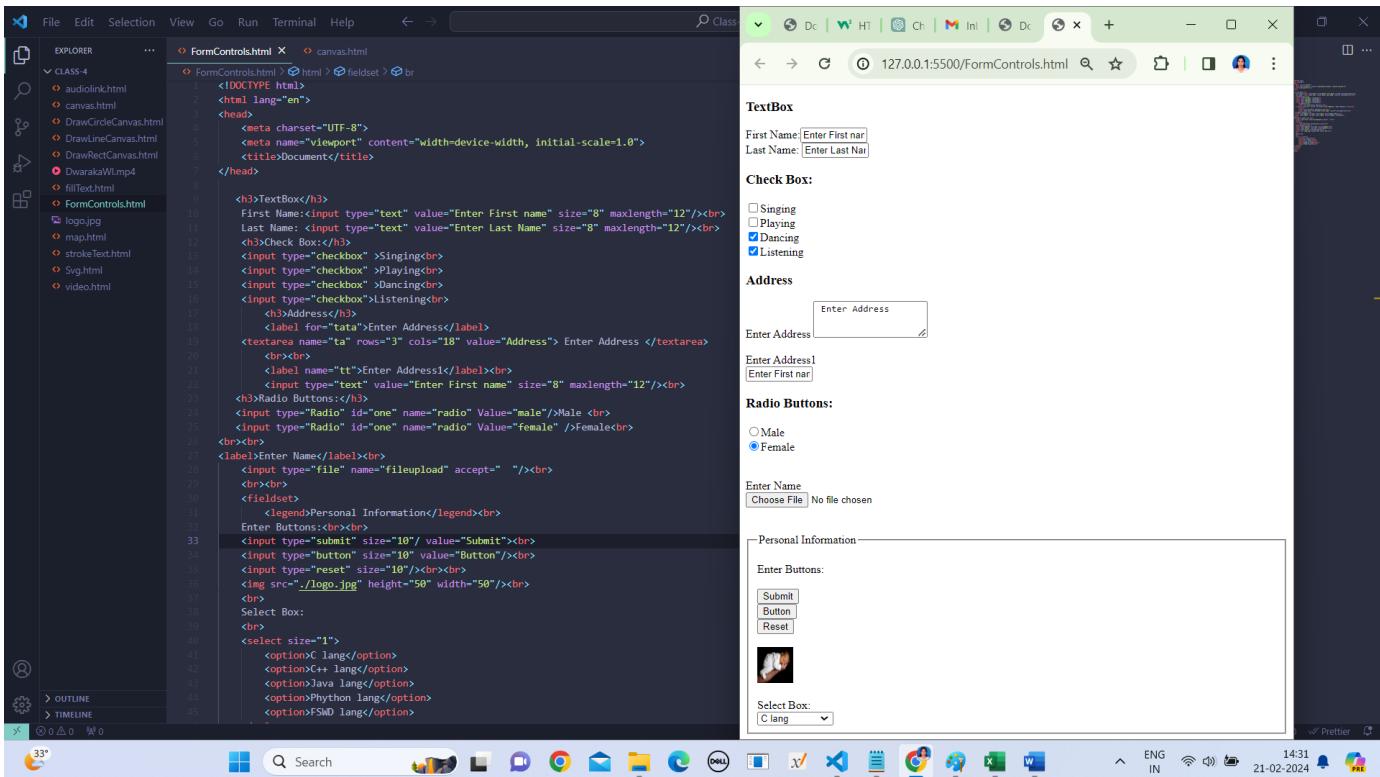
### **Example:**

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
</head>
<h3>TextBox</h3>
  First Name:<input type="text" value="Enter First name" size="8" maxlength="12"/><br>
  Last Name: <input type="text" value="Enter Last Name" size="8" maxlength="12"/><br>
<h3>Check Box:</h3>
  <input type="checkbox" >Singing<br>
  <input type="checkbox" >Playing<br>
  <input type="checkbox" >Dancing<br>
  <input type="checkbox">Listening<br>
<h3>Address</h3>
  <label for="tata">Enter Address</label>
  <textarea name="ta" rows="3" cols="18" value="Address"> Enter Address </textarea>
```

```

<br><br>
<label name="tt">Enter Address1</label><br>
<input type="text" value="Enter First name" size="8" maxlength="12"/><br>
<h3>Radio Buttons:</h3>
<input type="Radio" id="one" name="radio" Value="male"/>Male <br>
<input type="Radio" id="one" name="radio" Value="female" />Female<br>
<br><br>
<label>Enter Name</label><br>
<input type="file" name="fileupload" accept=" " /><br>
<br><br>
<fieldset>
    <legend>Personal Information</legend><br>
Enter Buttons:<br><br>
<input type="submit" size="10" / value="Submit"><br>
<input type="button" size="10" value="Button"/><br>
<input type="reset" size="10"/><br><br>
<br>
<br>
Select Box:
<br>
<select size="1">
    <option>C lang</option>
    <option>C++ lang</option>
    <option>Java lang</option>
    <option>Phyton lang</option>
    <option>FSWD lang</option>
</select>
</fieldset>
</form>
</body>
</html>

```



## DropDownMenu

It will display the element in dropdown items.

Used to create menu items using the tag `<select>`

This select tag has two attributes size and multiple.

If we give `<select size="5">` Five elements will be appeared.

If we give `<select multiple>` We can select multiple options at a time.

-----Program-----

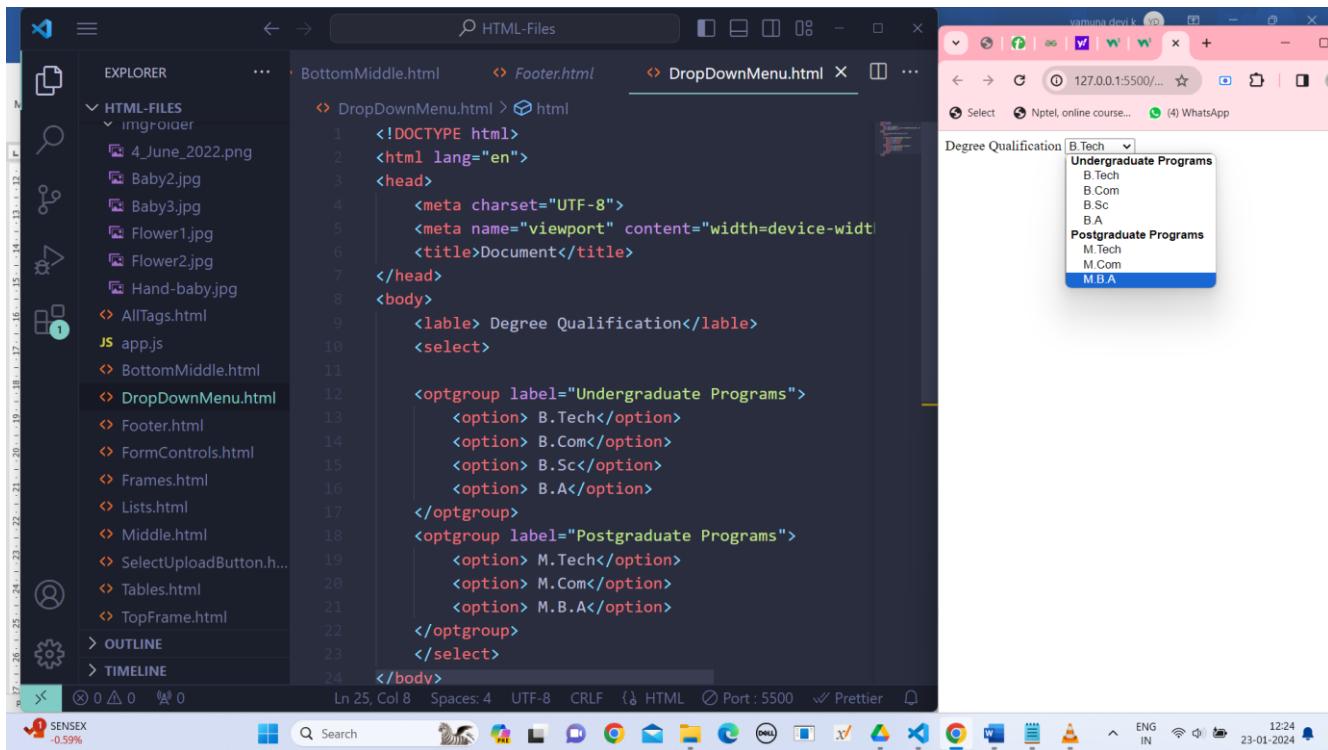
```

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
</head>
<body>
    <label> Degree Qualification</label>
    <select>

        <optgroup label="Undergraduate Programs">
            <option> B.Tech</option>
            <option> B.Com</option>
            <option> B.Sc</option>
            <option> B.A</option>
        </optgroup>
        <optgroup label="Postgraduate Programs">
            <option> M.Tech</option>
            <option> M.Com</option>
            <option> M.B.A</option>
        </optgroup>
    </select>

```

```
</body>
```



## Form( Select, Fileupload, Button)

**Select:** Here we can select any one of the element from the list of elements. It can be visible whenever scroll down the scrollbar. In order to scroll down the list of elements come up.

### Example: Select

```
<select size="3">
<option> C </option>
<option> C++</option>
<option> Java </option>
<option> CSS </option>
</select>
```

**Fileupload:** In some of the websites asking photo or sign. We are supposed to provide certificates or photos as a proof. In such cases it shows the file upload.

### Example: Fileupload

```
<input type="file" name="file1" accept="" >
```

**Button:** Commonly we are using for submit.

### Example: Buttons

```
<input type="submit" name="button" value="Submit">
<input type="reset" name="button" value="Reset">
<input type="Image" name="button" value="Image">
<input type="Button" name="button" value="click">

</head>
<body>
```

```

<h1>Select Tag</h1>
<h2>Interested Courses</h2>
<select size="5" multiple>
    <option>C Language</option><br>
    <option>Java</option><br>
    <option>HTML</option><br>
    <option>CSS</option><br>
    <option>C++</option><br>
</select>
<h1> File Upload</h1>
<input type="file" name="file1" accept=text
<br>
<h1> Button</h1>
<input type="submit" name="submit1"
    value="Submit">&nbsp;
<input type="reset" name="reset1"
    value="Reset">&nbsp;
<input type="Button" name="button1"
    value="Button">&nbsp;
<input type="Image" name="image1"
    src=".ImgFolder/4_June_2022.png"
    height="27" width="50">
</body>
</html>

```

The screenshot shows a browser window displaying the rendered HTML code. The page has a title 'Select Tag'. Below the title, there is a heading 'Interested Courses' followed by a dropdown menu with five options: C Language, Java, HTML, CSS, and C++. Underneath this, there is a 'File Upload' section with a file input field. The 'Choose File' button shows 'No file chosen'. At the bottom, there is a 'Button' section containing three buttons: 'Submit', 'Reset', and a large blue 'Button' button.

## **HTML5**

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).

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

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.

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

- **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.
- **New Semantic Elements** – These are like <header>, <footer>, and <section>.
- **Geolocation** – Now visitors can choose to share their physical location with your web application.
- **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 facilitates 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.

### **HTML5 Syntax:**

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.

**syntax:** <!DOCTYPE html> The 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 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 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 you 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: Some of the useful HTML 5 Tags**

<b>Tag</b>	<b>Description</b>
<audio>	It is used to play audio file in HTML.
<bdi>	The bdi stands for bi-directional isolation.
<canvas>	It is used to draw canvas.
<data>	It provides machine readable version of its data.
<datalist>	It provides auto complete feature for textfield.
<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.
<nav>	It is used to define the navigation link in the document.
<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.

## **HTML5 Form Tags**

<datalist> It represent predefined list for input <option> element.

## **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.

<source> It defines multiple media resources for the media elements.

<video> It defines video content within HTML document.

## **HTML5 New <input> types:**

<b>Type</b>	<b>Description</b>
date	It represents an input field to define a date selector.
Date-time	It defines full date and time display with time zone information.
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 Attributes:**

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

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>

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.

### Some Form Events

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

Attribute	Value	Description
Onchange	script	Fires when the value or state of the element is changed.
Onfocus	script	Fires when the element receives focus.
Oninput user.	script	Fires when the value of an element is changed by the user.
Onreset	script	Fires when the user resets a form.
Onsubmit	script	Fires when a form is submitted.

### Some Mouse Events

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

Attribute	Value	Description
Onclick	script	Fires, user clicks the left mouse button on the element.
Ondblclick	script	Fires when the user double-clicks on the element.
Ondragover	script	Fires, element is being dragged over a valid drop target.
Ondragstart	script	Fires when the user starts to drag a text selection.
Onmousedown	script	Fires when the mouse button is pressed over an element.
Onmousemove	script	Fires, user moves the mouse pointer over an element.
Onmouseout	script	Fires, user moves the mouse pointer outside boundaries.
Onmouseover	script	Fires, user moves the mouse pointer onto an element.

### Some 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.

### HTML5 Audio Tag:

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

**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 buttons.
autoplay	It specifies that the audio will start playing as soon as it is ready.
loop	It specifies, audio file will start over again, 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.

#### **EXAMPLE:**

```
<!DOCTYPE>
<html>
<body>
<audio controls autoplay loop>
  <source src="myaudio.mp3" type="audio/mpeg"></audio>
</body>
</html>
```

## **HTML Video**

#### **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

#### **Supporting browsers are.....**

Browser	<b>mp4</b>	<b>webM</b>	<b>ogg</b>
Internet Explorer	yes	no	no
Google Chrome	yes	yes	yes
Mozilla Firefox	yes	yes	yes
Opera	no	yes	yes
Apple Safari	yes	no	no

#### **Some of the HTML 5 video tag attributes.**

Attribute	Description
controls	It defines the video controls which is displayed with play 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
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.
muted	It is used to mute the video output.
src	It specifies the source URL of the video file.

### **Example:**

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
</head>
<body>
  <video controls>
    <source src="CMR-Anual.mp4" type="video/mp4" width="50" height="50"
    poster="./logo.jpg" >
  </audio>
</body>
</html>
```

### **HTML Canvas Tag:**

The **HTML 5 <canvas> tag** is used to draw graphics using scripting language like JavaScript.

HTML5 element `<canvas>` gives you an easy and powerful way to draw graphics using JavaScript. It can be used to draw graphs, make photo compositions or do simple (and not so simple) animations.

The `<canvas>` element is only a container for graphics, you must need a scripting language to draw the graphics. The `<canvas>` element allows for dynamic and scriptable rendering of 2D shapes and bitmap 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>
```

## Drawing Line on Canvas:

If you want to draw a straight line on the canvas, we can use the following 2 methods.

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

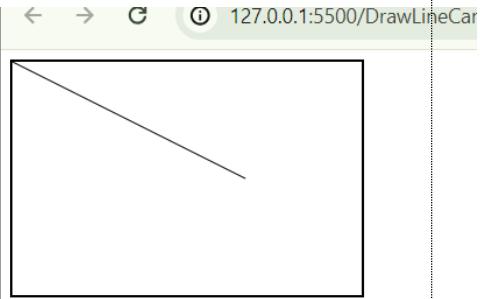
**lineTo(x,y):** It is used to define the ending point of the line.

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.

```
<canvas id="myCanLine" width="200" height="100" style="border:1px solid red;">
```

```
<script>
```

```
var c = document.getElementById("myCanvasLine");
var cctx = c.getContext("2d");
ctx.moveTo(0,0);
ctx.lineTo(200,100);
ctx.stroke();
</script>
```



The screenshot shows a browser window with the URL 127.0.0.1:5500/DrawLineCanvas.html. On the left, the code for the page is visible, including the HTML structure and a script block that draws a line from (0,0) to (200,100). On the right, the browser's content area displays a white canvas with a single black diagonal line segment.

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
</head>
<body>
    <canvas id="myCanvas2" width="300" height="200" style="border:2px solid black;">
        <script>
            var c = document.getElementById("myCanvas2");
            var ctx = c.getContext("2d");
            ctx.moveTo(0,0);
            ctx.lineTo(200,100);
            ctx.stroke();
        </script>
    </body>
</html>
```

## Drawing Circle on Canvas

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

### **Example Program:**

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
</head>
<body>
    <canvas id="myCanvasCircle" width="200" height="100" style="border:1px solid #d3d3d3;">
        Your browser does not support the HTML5 canvas tag.</canvas>
    </body>
</html>
```

```

<script>
var c = document.getElementById("myCanvasCircle");
var ctx = c.getContext("2d");
ctx.beginPath();
ctx.arc(95,50,40,0,2*Math.PI);
ctx.stroke();
</script>
</body>
</html>

```

FormControls.html canvas.html DrawCircleCanvas.html

DrawCircleCanvas.html > html > body

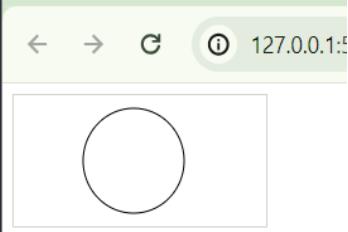
```

1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <title>Document</title>
7  </head>
8  <body>

9
10
11     <canvas id="myCanvasCircle" width="200" height="100" style="border:1px solid #d3d3d3">
12         Your browser does not support the HTML5 canvas tag.</canvas>
13     <script>
14         var c = document.getElementById("myCanvasCircle");
15         var ctx = c.getContext("2d");
16         ctx.beginPath();

17
18         ctx.arc(95,50,40,0,2*Math.PI);
19         ctx.stroke();
20     </script>
21
22 </body>
23 </html>

```



```

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
</head>
<body>
    <canvas id="myCanvas" width="300" height="200" style="border: 3px solid black"></canvas>
    <script>
        var canvas=document.getElementById('myCanvas');
        var ctx=canvas.getContext("2d");
        ctx.fillStyle='green';
        ctx.fillRect(0,0,190,95);
    </script>
</body>
</html>

```

```

FormControls.html    canvas.html    DrawRectCanvas.html
DrawRectCanvas.html > html > body
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <title>Document</title>
7  </head>
8  <body>
9      <canvas id="myCanvas" width="300" height="200" style="border:2px solid;"> </canvas>
10 <script>
11     var c = document.getElementById('myCanvas');
12     var ctx = c.getContext("2d");
13     ctx.fillStyle = 'green';
14     ctx.fillRect(0,0,200,100);
15
16
17 </script>
18
19 </body>
20 </html>
21

```



## SVG

SVG stands for **S**calable **V**ector **G**raphics and it is a language for describing 2D-graphics and graphical applications in XML and the XML is then rendered by an SVG viewer.

SVG is mostly useful for vector type diagrams like Pie charts, Two-dimensional graphs in an X,Y coordinate system etc.

The `<svg>` element defines the SVG graphic.

The width and height attributes specify the dimensions of the SVG canvas.

Inside the `<svg>` element, you can include various SVG elements like `<circle>`, `<rect>`, `<line>`, etc.

In this case, a `<circle>` element is used to draw a red circle with a center at (100, 100) and a radius of 80 pixels.

SVG graphics can be styled using CSS, and they can also contain JavaScript for interactivity.

### HTML SVG Circle Example:

```

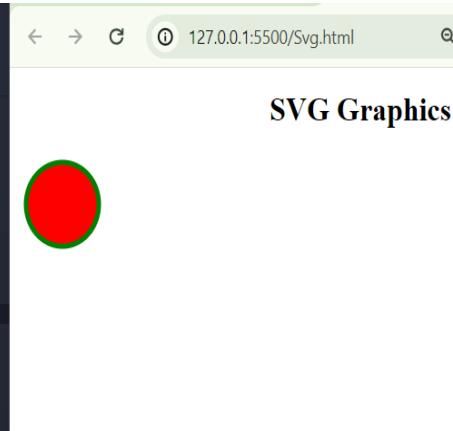
<!DOCTYPE html>
<html>
<body>
<svg width="100" height="100">
    <circle cx="50" cy="50" r="40" stroke="yellow" stroke-width="4" fill="red" />
</svg>
</body>
</html>

```

```

FormControls.html    canvas.html    Svg.html X
Svg.html > ...
1 <html lang="en">
2   <head>
3     ...
4   </head>
5   <body>
6     <h1><center>SVG Graphics</center></h1>
7     <svg width="500" height="500">
8
9       <circle cx="50" cy="50" r="40" stroke="green" fill="red" stroke-width="5">
10
11   </svg>
12 </body>
13 </html>

```



### 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>

```

### HTML SVG polygon Example:

```

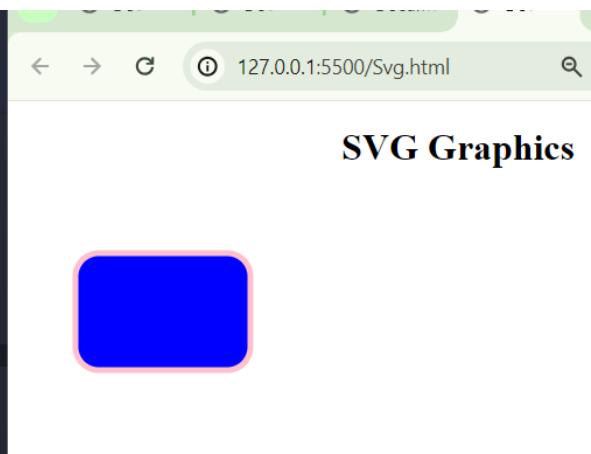
<!DOCTYPE html>
<html>
<body>
<svg height="210" width="500">
<polygon points="100,10 40,198 190,78 10,78 160,198"
  style="fill:red;stroke:yellow;stroke-width:5;fill-rule:nonzero;" />
</svg>
</body>
</html>

```

```

FormControls.html    canvas.html    Svg.html X
Svg.html > ...
1 <html lang="en">
2   <head>
3     ...
4   </head>
5   <body>
6     <h1><center>SVG Graphics</center></h1>
7     <svg width="500" height="500">
8
9       <rect x="50" y="50" width="150" height="100"
10         style="fill:#blue; stroke:#pink; stroke-width:5; rx="20" ry="20"/>
11
12   </svg>
13 </body>
14 </html>

```



### 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.

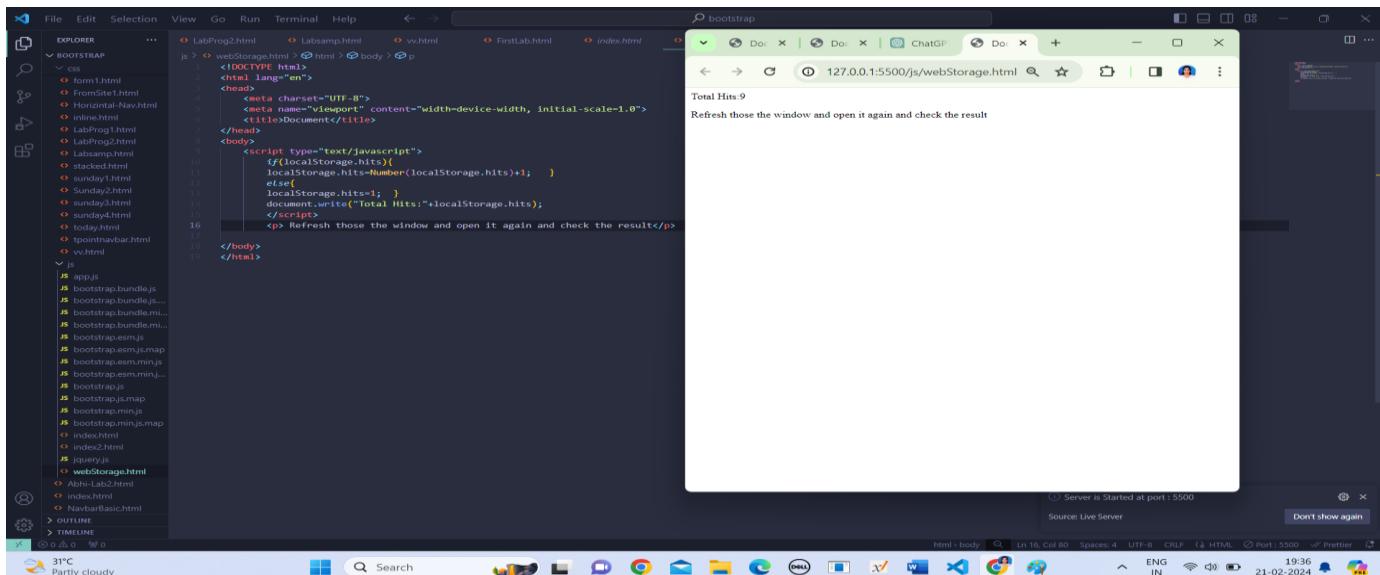
In compared to cookies Web Storage has Following Advantages:

- 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.

**1. Local Storage:** Local Storage uses `Window.localStorage` 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).



### **Example:**

```
<html> <head>
<script type="text/javascript">
if(localStorage.hits){
localStorage.hits=Number(localStorage.hits)+1; }
else{
localStorage.hits=1; }
document.write("Total Hits:"+localStorage.hits);
</script>
<p> Refresh those the window and open it again and check the result</p>
</body> </html>
```

```

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
</head>
<body>
    <script type="text/javascript">
        if(localStorage.hits){
            localStorage.hits=Number(localStorage.hits)+1;
        } else{
            localStorage.hits=1;
        }
        document.write("Total Hits:"+localStorage.hits);
    </script>
    <p> Refresh those the window and open it again and check the result</p>
</body>
</html>

```

**2. 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.

```

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
</head>
<body>
    <script type="text/javascript">
        if(sessionStorage.hits){
            sessionStorage.hits=Number(sessionStorage.hits)+1;
        } else{
            sessionStorage.hits=1;
        }
        document.write("Total Hits:"+sessionStorage.hits);
    </script>
    <p> Refresh those the window and open it again and check the result</p>
</body>
</html>

```

### **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.

## HTML Geolocation

- The HTML Geolocation is used to get the real time geographical position of a user, if they allow it.
- To get the Geolocation we have to use JavaScript to get the latitude and longitude.
- 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.
- The `getCurrentPosition()` method is used to return the user's position.

Syntax: `navigator.geolocation.getCurrentPosition()`

```
<html>
<body>
    <p>Click the button to get your coordinates.</p>
    <button onclick="myLocation() id="CMR">Click Me</button>
    <script>
        var x = document.getElementById("CMR");
        function myLocation() {
            if (navigator.geolocation) {
                navigator.geolocation.getCurrentPosition(showPosition);
            }
            else {
                x.innerHTML = "Geolocation is not supported by this browser.";
            }
        }
        function showPosition(pos1) {
            x.innerHTML = "Latitude: " + pos1.coords.latitude +
                "<br>Longitude: " + pos1.coords.longitude;
        }
    </script> </body>
```

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

Click Me

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

### 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):

### **CSS- Cascading Style Sheet**

Cascading Style Sheets (CSS) is a style sheet language used for describing the presentation of a document written in a markup language like HTML.

CSS works by targeting HTML elements and applying style rules to define how they should be displayed, including properties like color, size, layout, and positioning.

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 has been split into "modules". It contains the "old CSS specification".

Some of the most important CSS3 modules are:

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

### **Features of CSS3:**

#### 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. 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.
- 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.

## **Types of CSS**

CSS can be divided into 3 types,

- **Inline styles** — Using the `<style>` attribute in the HTML start tag.
- **Internal or Embedded styles** — Using the `<style>` element in the header section.
- **External style sheets** — Using the `<link>` 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.

While inline styles can be convenient for quick styling changes or for applying styles to individual elements, they are generally not recommended for larger projects because they can make the HTML markup less readable and harder to maintain.

The `style` attribute includes a series of CSS property and value pairs. Each "property: value" pair is separated by a semicolon (;).

Basic Styling using CSS – Basic Styling, positioning & Back ground Images.

### **Example:**

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
</head>
<body style="background-color:green">
  <h1 style="color:white; text-align:center"> Demo on CSS
  </h1>
  <p1 style="color:red; text-align:left">Hello CMR</p1>
</body>
</html>
```

```

1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>Document</title>
7 </head>
8 <body style="background-color: green">
9   <h1 style="color: white; text-align:center"> Demo on CSS
10  </h1>
11
12  <p1 style="color: red; text-align:left">Hello CMR</p1>
13
14 </body>
15 </html>

```

## ii) Internal Style Sheets or Embedded Style Sheets:

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

Embedded style sheets are defined in the `<head>` section of an HTML document using the `<style>` element. You can define any number of `<style>` elements in an HTML document but they must appear between the `<head>` and `</head>` tags.

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
  <style>
    body{background-color:pink;}
    h1{color:red;text-align: center;}
  </style>
</head>
<body>
  <h1> CMR Engg College</h1>
</body>
</html>

```

```

1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>Document</title>
7   <style>
8     body{background-color: pink;}
9     h1{color: red;text-align: center;}
10  </style>
11 </head>
12 <body>
13   <h1> CMR Engg College</h1>
14 </body>
15 </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, we 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>` tag. The `<link>` tag goes inside the `<head>` section.

## **External.html**

```
!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
    <link rel="stylesheet" href="ext.css">
</head>
<body>
    <h1>This is External CSS</h1>
    <h3>No styles applied</h3>
</body>
</html>
<!--ext.css
body{
    background-color:green;
}
h1{
    color:yellow; text-align:center;
} -->
```

The screenshot shows a code editor on the left and a browser window on the right. The code editor displays the HTML file 'External.html' with its content. The browser window shows the rendered output of the HTML, which includes a green background and yellow text for the heading, demonstrating the effect of the external CSS file.

```
External.html > ...
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
    <link rel="stylesheet" href="ext.css">
</head>
<body>
    <h1>This is External CSS</h1>
    <h3>No styles applied</h3>
</body>
</html>

<!--ext.css
body{
    background-color:green;
}
h1{
    color:yellow; text-align:center;
} -->
```

127.0.0.1:5500/External.html

This is External CSS

No styles applied

## CSS Selector

**CSS selectors** are used to find the HTML element or select the content you want to style. Selectors are the part of CSS rule set. CSS selectors select HTML elements according to its id, class, type, attribute etc.

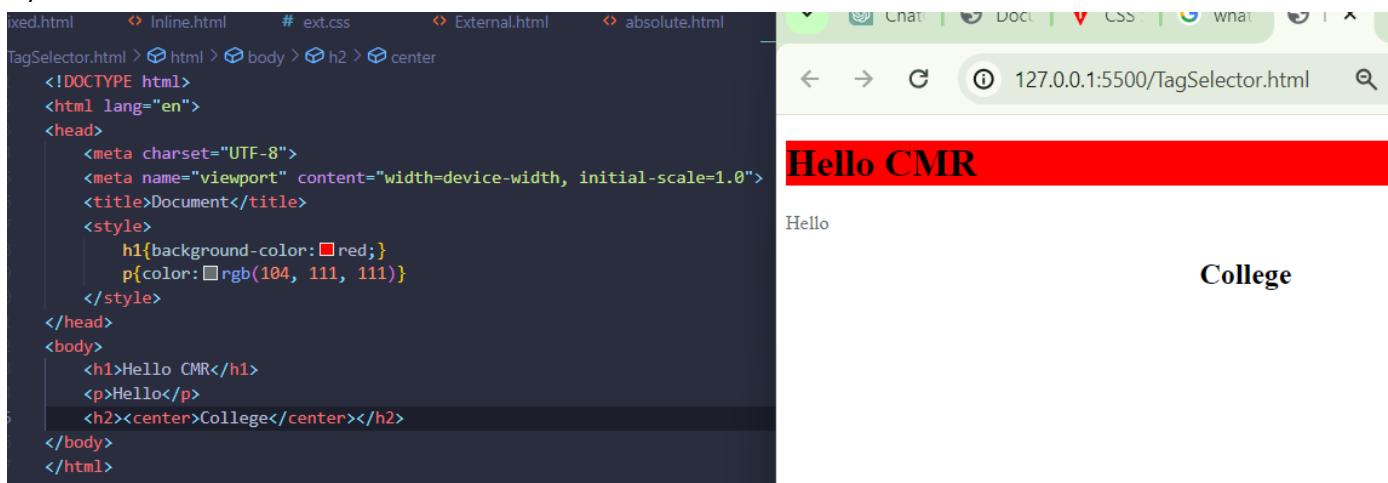
There are several different types of selectors in CSS.

1. Tag Selector or Element Selector
2. ID Selector
3. Class Selector
4. Universal Selector
5. Group Selector.

### **1. Tag Selector or Element Selector**

The element selector selects the HTML element by name.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
  <style>
    h1{background-color:red;}
    p{color:rgb(104, 111, 111)}
  </style>
</head>
<body>
  <h1>Hello CMR</h1>
  <p>Hello</p>
  <h2><center>College</center></h2>
</body>
</html>
```



### **2. ID Based Selectors:**

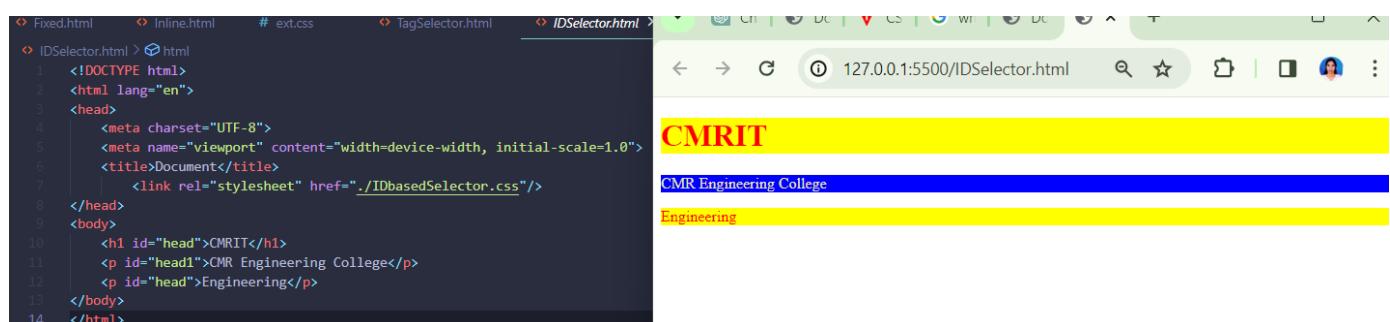
The id selector selects the id attribute of an HTML element to select a specific element. An id is always unique within the page so it is chosen to select a single, unique element.

It is written with the hash character (#), followed by the id of the element.

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
    <link rel="stylesheet" href=".//IDbasedSelector.css"/>
</head>
<body>
    <h1 id="head">CMRIT</h1>
    <p id="head1">CMR Engineering College</p>
    <p id="head">Engineering</p>
</body>
</html>
```

IDbasedselectors.css

```
#head{background-color:yellow;color:red}
#head1{background-color:blue;color:palegoldenrod}
```



### 3. Class Selector

The class selector selects HTML elements with a specific class attribute. It is used with a dot symbol or full stop symbol(.) followed by the class name.

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
    <style>
        #head{background-color:pink; color:blue}
        #head2{background-color:palegoldenrod; color:green}
    </style>

```

```

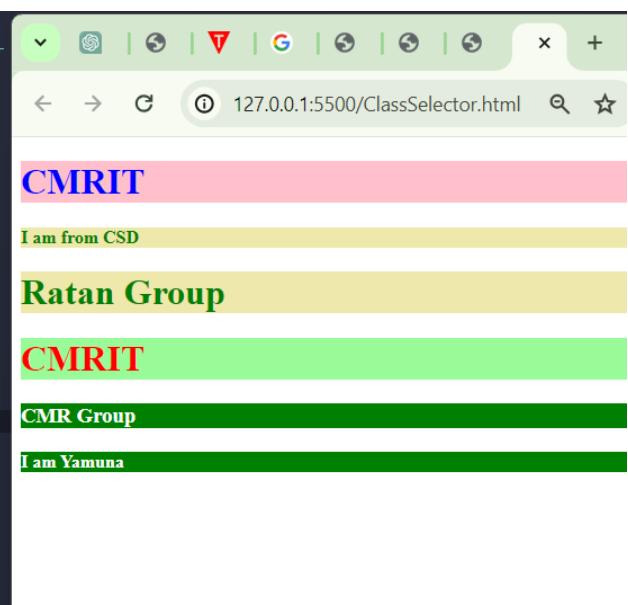
.head1{background-color:palegreen;color:Red;}
.head3{background-color:green; color:white}
</style>
</head>
<body>
<h1 id="head">CMRIT</h1>
<h4 id="head2">I am from CSD</h4>
<h1 id="head2">Ratan Group</h1>
<h1 class="head1">CMRIT</h1>
<h3 class="head3">CMR Group</h3>
<h4 class="head3">I am Yamuna</h4>
</body>
</html>

```

```

Fixed.html      ▷ Inline.html    # ext.css      ▷ TagSelector.html      ▷ ClassSelector.html ×
ClassSelector.html > html > body > h4#head2
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <title>Document</title>
7      <style>
8          #head{background-color: pink; color: blue}
9          #head2{background-color: palegoldenrod; color: green}
10         .head1{background-color: palegreen; color: Red;}
11         .head3{background-color: green; color: white}
12     </style>
13 </head>
14 <body>
15     <h1 id="head">CMRIT</h1>
16     <h4 id="head2">I am from CSD</h4>
17     <h1 id="head2">Ratan Group</h1>
18
19     <h1 class="head1">CMRIT</h1>
20     <h3 class="head3">CMR Group</h3>
21     <h4 class="head3">I am Yamuna</h4>
22
23 </body>
24 </html>

```



## 4.Universal Selector

The universal selector is used as a wildcard (\*) character. It selects all the elements on the pages.

### Example:

```

<!DOCTYPE html>
<html lang="en">
<head>s <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
    <style>
        *{
            background-color:pink;
            color:blue;
        }
        .head2{

```

```

background-color:pink;
color:blue;
}
</style>
</head>
<body>
<p> My College</p>
<h5>Ramana</h5>
<h3> Naidu</h3>
<h1 class="head2">CMRIT</h1>
<h6> CMRIT</h6>
</body>
</html>

```

The screenshot displays a comparison between the source code and its rendered output. On the left, the code editor shows the HTML structure with inline CSS. On the right, the browser window shows the final presentation where every element (p, h5, h3, h1, h6) has a pink background and blue text.

```

UniversalSelector.html > html
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <title>Document</title>
7      <style>
8      *{
9          background-color: pink;
10         color: blue;
11     }
12     .head2{
13         background-color: pink;
14         color: blue;
15     }
16     </style>
17 </head>
18 <body>
19     <p> My College</p>
20     <h5>Ramana</h5>
21     <h3> Naidu</h3>
22     <h1 class="head2">CMRIT</h1>
23     <h6> CMRIT</h6>
24 </body>
25 </html>

```

## 5. Group Selector.

The grouping selector is used to select all the elements with the same style definitions. Grouping selector is used to minimize the code. Commas are used to separate each selector in grouping.

```

<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Document</title>
<style>
h1,h2,p{background-color:pink;
color:red}

```

```

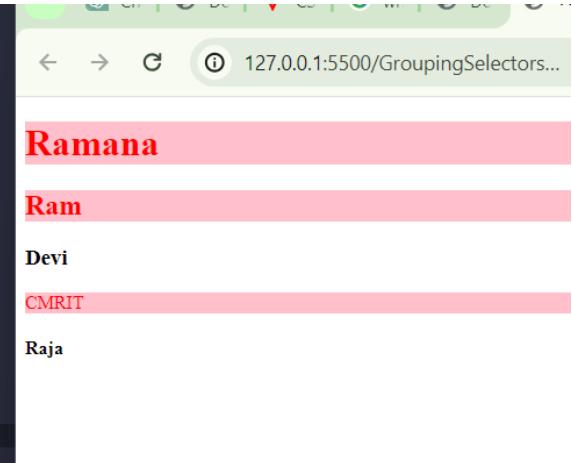
</style>
</head>s
<body>
  <h1>Ramana</h1>
  <h2>Ram</h2>
  <h3>Devi</h3>
  <p>CMRIT</p>
  <h4>Raja</h4>
</body>
</html>

```

```

1  GroupingSelectors.html > html > body
2  <!DOCTYPE html>
3  <html lang="en">
4  <head>
5    <meta charset="UTF-8">
6    <meta name="viewport" content="width=device-width, initial-scale=1.0">
7    <title>Document</title>
8    <style>
9      h1,h2,p{background-color: #pink;
10        color: red}
11    </style>
12  </head>
13  <body>
14    <h1>Ramana</h1>
15    <h2>Ram</h2>
16    <h3>Devi</h3>
17    <p>CMRIT</p>
18    <h4>Raja</h4>
19  </body>
</html>

```



## CSS Borders

The border of an HTML element is simply one or more lines that surround the content and padding of an element. Every border has three aspects: its width, or thickness; its style, or appearance; and its color.

The CSS **border** properties allow you to specify how the border of the box representing an element should look. There are three properties of a border you can change –

- **border-style** - Specifies whether a border should be solid, dashed line, double line, or one of the other possible values.
- **border-color** - Specifies the color of a border. The default value is the foreground color of the element and if element is blank, then color of the parent element
- **border-width** - Specifies the width of a border. The default value is medium.

### CSS Border Style

The **border-style** property specifies what kind of border to display. Following values can be passed to border-style:

Value	Description
none	No border
hidden	A hidden border, same as 'none' except for table elements
dotted	A series of dots
dashes	A series of short dashes

solid	A single solid line
double	Two parallel lines with a small gap between them
groove	A border that appears to be carved into the page
ridge	A border that appears to be slightly raised above the page
inset	A border that appears embedded into the page
outset	A border that appears slightly raised out of the page

The border-style property can have up to four values in one statement where we can specify the border style for top, right, bottom and left border.

```

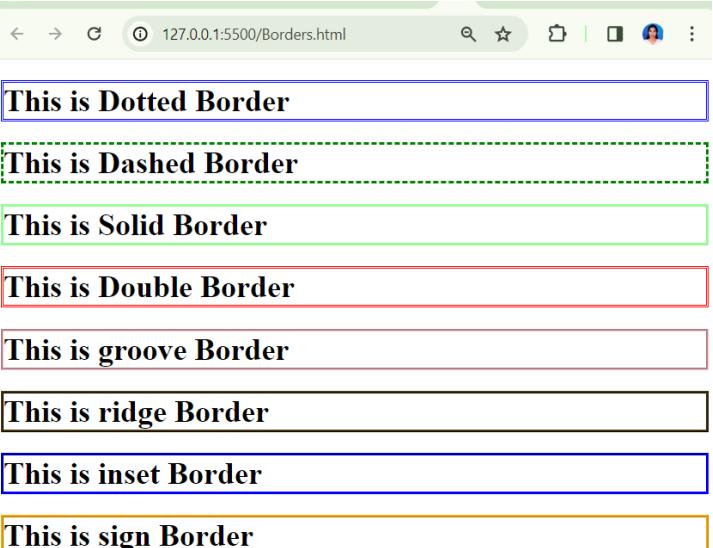
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
  <style>
    .dotted{ border-style: double; border-width: 3px; border-color: blue; }
    .dashed{ border-style: dashed; border-width: 3px; border-color: green; }
    .solid{ border-style: solid; border-width: 3px; border-color: palegreen; }
    .double{ border-style: double; border-width: 3px; border-color: red; }
    .groove{ border-style: groove; border-width: 3px; border-color: pink; }
    .ridge{ border-style: ridge; border-width: 3px; border-color: rgb(72, 51, 12); }
    .inset{ border-style: inset; border-width: 3px; border-color: blue; }
    .sign{ border-style: ridge; border-width: 3px; border-color: orange; }
  </style>
</head>
<body>
  <h1 class="dotted">This is Dotted Border</h1>
  <h1 class="dashed">This is Dashed Border</h1>
  <h1 class="solid">This is Solid Border</h1>
  <h1 class="double">This is Double Border</h1>
  <h1 class="groove">This is groove Border</h1>
  <h1 class="ridge">This is ridge Border</h1>
  <h1 class="inset">This is inset Border</h1>
  <h1 class="sign">This is sign Border</h1>
</body>
</html>

```

```

1 colors.html 2 Borders.html 3 Html-selector.html 4 MarginsPadding.html
  Borders.html > html
<html lang="en">
<head>
<title>Document</title>
<style>
.dotted{ border-style: double; border-width: 3px; border-color: blue; }
.dashed{ border-style: dashed; border-width: 3px; border-color: green; }
.solid{ border-style: solid; border-width: 3px; border-color: palegreen; }
.double{ border-style: double; border-width: 3px; border-color: red; }
.groove{ border-style: groove; border-width: 3px; border-color: pink; }
.ridge{ border-style: ridge; border-width: 3px; border-color: rgb(72, 61, 12); }
.inset{ border-style: inset; border-width: 3px; border-color: blue; }
.sign{ border-style: ridge; border-width: 3px; border-color: orange; }
</style>
</head>
<body>
<h1 class="dotted">This is Dotted Border</h1>
<h1 class="dashed">This is Dashed Border</h1>
<h1 class="solid">This is Solid Border</h1>
<h1 class="double">This is Double Border</h1>
<h1 class="groove">This is groove Border</h1>
<h1 class="ridge">This is ridge Border</h1>
<h1 class="inset">This is inset Border</h1>
<h1 class="sign">This is sign Border</h1>
</body>
</html>

```



## CSS Padding & CSS Margin

**CSS Padding:** The CSS padding property is used to specify the space between the content of an element and its borders.

**CSS Margin:** Setting up a margin around an HTML element is one of the things that sets CSS so far above traditional web markup.

The CSS margin property is a shorthand property to set the margin area around an HTML element.

```

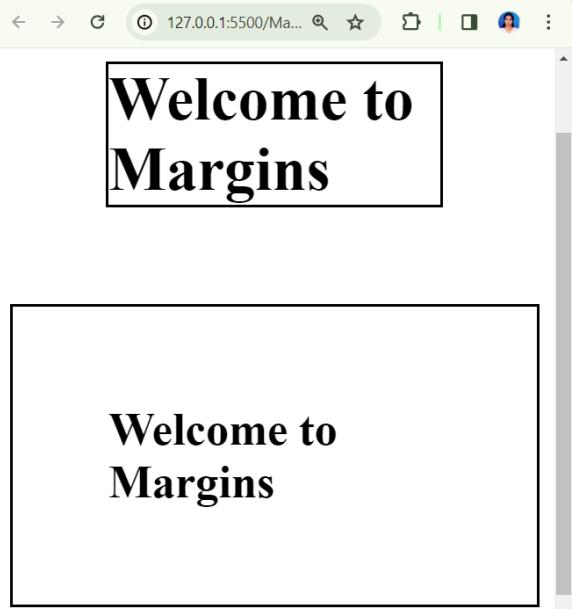
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Document</title>
<style>
h1{
    border-style: solid; margin-top:50px; margin-bottom:50px; margin-left:50px; margin-right:50px;
}
h2{
    border-style: solid; padding-top:50px; padding-bottom:50px; padding-left:50px; padding-right:50px;
}
</style>
</head>
<body>
<h1> Welcome to Margins</h1>
<h2> Welcome to Margins</h2>
</body>
</html>

```

```

MarginsPadding.html > html
  <!DOCTYPE html>
  <html lang="en">
    <head>
      <meta charset="UTF-8">
      <meta name="viewport" content="width=device-width, initial-scale=1.0">
      <title>Document</title>
      <style>
        h1{
          border-style: solid; margin-top:50px; margin-bottom:50px; margin-left:50px; margin-right:50px;
        }
        h2{
          border-style: solid; padding-top:50px; padding-bottom:50px; padding-left:50px; padding-right:50px;
        }
      </style>
    </head>
    <body>
      <h1> Welcome to Margins</h1>
      <h2> Welcome to Margins</h2>
    </body>
  </html>

```



## Positioning

### Set the Background Image Position

set the background image position 100 pixels away from the left side and 200 pixels down from the top.

background-position:100px 200px;

```

<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
    <style>
      body{
        background-image:Url("Baby3.jpg");
        background-color:palegreen;
        background-repeat:no-repeat;
        background-position:left top;
        <!--background-attachment:scroll;-->
        background-attachment:fixed;
      }
    </style>
  </head>
  <h1> This is for Image</h1>
  <h1> This is for Image</h1>
  <h1> This is for Image</h1>

```

```
colors.html > html > head
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <title>Document</title>
7      <style>
8
9          body{
10              background-image:url("Baby3.jpg");
11              background-color:# palegreen;
12              background-repeat:no-repeat;
13              background-position:left top;
14              <!--background-attachment:scroll;-->
15              background-attachment:fixed;
16          }
17      </style>
18  </head>
19  <h1> This is for Image</h1>
20  <h1> This is for Image</h1>
21  <h1> This is for Image</h1>
22  <h1> This is for Image</h1>
23  <h1> This is for Image</h1>
24  <h1> This is for Image</h1>
25  <h1> This is for Image</h1>
26  <h1> This is for Image</h1>
27  <h1> This is for Image</h1>
28  <h1> This is for Image</h1>
29  <h1> This is for Image</h1>
30  <h1> This is for Image</h1>
31  <h1> This is for Image</h1>
32  <h1> This is for Image</h1>
33  <h1> This is for Image</h1>
34  <body>
```



## BootStrap

Bootstrap can be defined as a free and open-source framework that can be used to create responsive 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 5 is also useful because it is compatible with all modern browsers which include Google Chrome, Firefox, Internet Explorer 10+, Edge, Safari, and Opera.
4. Lastly, Bootstrap 5 is a very simple and yet very effective grid system.

#### **Usages of Bootstrap 5:**

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.
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 5 can take a lot of time to create a website. Therefore, it is not a very bright idea to use Bootstrap 5 when there is no investment.
3. A person using Bootstrap 5 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 5 is the latest version of Bootstrap. In this section, it is shown how to download and setup Bootstrap 5.

There are two ways of using Bootstrap 5 in the Website:

- a) The first method that can be used is to start using Bootstrap 5 by inserting it from the CDN which stands for **Content Delivery Network**.
- b) And the second method is to download it from [getbootstrap.com](http://getbootstrap.com).

**Using CDN:** The Bootstrap 5 can be used in the website by inserting it from the CDN which is short for - Content Delivery Network.

<!-- Compiled and Minified Bootstrap CSS -->

```
<!-- jQuery Library -->
<!-- Popper-->
<!-- Compiled and Minified Bootstrap JavaScript -->
```

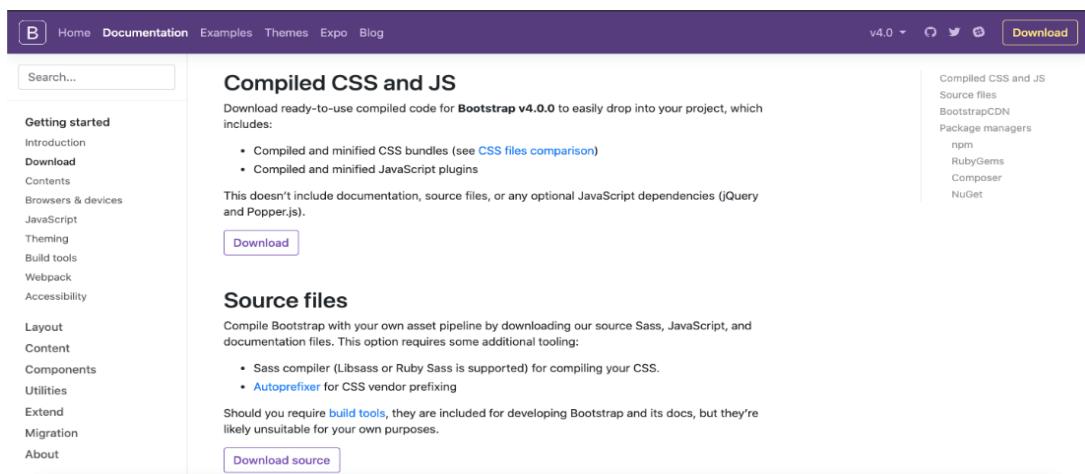
In this process, also insert the CDN versions of **jQuery** and **Popper.js** which is used for the purpose of using the different JavaScript components.

- 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 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 5:

The Bootstrap 5 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 with a person's own asset pipeline by downloading Bootstrap 5's source Sass, JavaScript, and documentation files.

The screenshot shows the Bootstrap documentation homepage. On the left, there's a sidebar with navigation links like 'Getting started', 'Introduction', 'Download', 'Contents', 'Browsers & devices', 'JavaScript', 'Theming', 'Build tools', 'Webpack', 'Accessibility', 'Layout', 'Content', 'Components', 'Utilities', 'Extend', 'Migration', and 'About'. A search bar is at the top. The main content area has two sections: 'Compiled CSS and JS' and 'Source files'. Each section has a 'Download' button, which is circled in red in the screenshot. The 'Compiled CSS and JS' section also has a note about including minified CSS and JavaScript 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 5:**

Once the compiled version Bootstrap 5 is downloaded, extract the ZIP file, and you will see the so many files/directories-

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

Bootstrap 5 Source Code:

The Bootstrap source code download includes the precompiled CSS, JavaScript, and font assets, along with source Less, JavaScript, and documentation.

The files under the js/ and scss/ are the source code responsible for Bootstrap, CSS and JavaScript.

- The dist/ folder consists of everything that is listed in the pre-compiled download section.

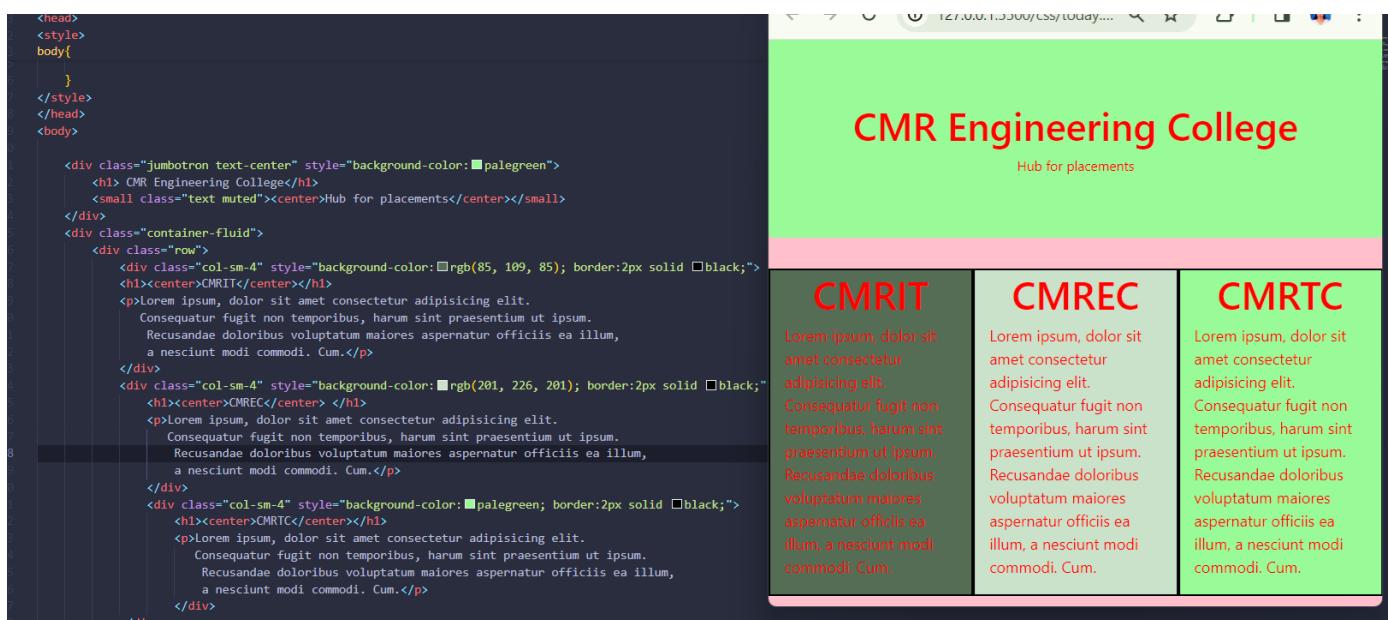
- b. The docs/examples/ only consists of the source code for Bootstrap documentation and examples of Bootstrap usage respectively.
- c. The less/ in the source code given above is basically responsible for the source code for the icon fonts in Bootstrap 5.
- d. Apart from that, the other included files provide support for other important components such as the packages, the information about the license, and anything related to development.

## **Bootstrap Templates**

### **Bootstrap Grid System**

Working of Bootstrap Grid System Grid systems are used for creating page layouts through a series of rows and columns that house your content. Here's how the Bootstrap grid system works:

- Rows must be placed within a .container class for proper alignment and padding.
- Use rows to create horizontal groups of columns. 6 Bootstrap
- Content should be placed within the columns, and only columns may be the immediate children of rows.
- Predefined grid classes like- .row and .col-xs-4 are available for quickly making grid layouts.
- Columns create gutters (gaps between column content) via padding. That padding is offset in rows for the first and the last column via negative margin on- .rows.
- Grid columns are created by specifying the number of twelve available columns you wish to span. For example, three equal columns would use three .col-xs-4.



## **Typography**

Bootstrap provides a strong set of typographic styles and utilities to visually appealing text presentation across different devices and screen sizes.

**Headings:** Bootstrap provides styles for headings (h1-h6) that are designed to be visually appealing and hierarchical. These headings can be easily customized using Bootstrap's utility classes or by overriding default styles.

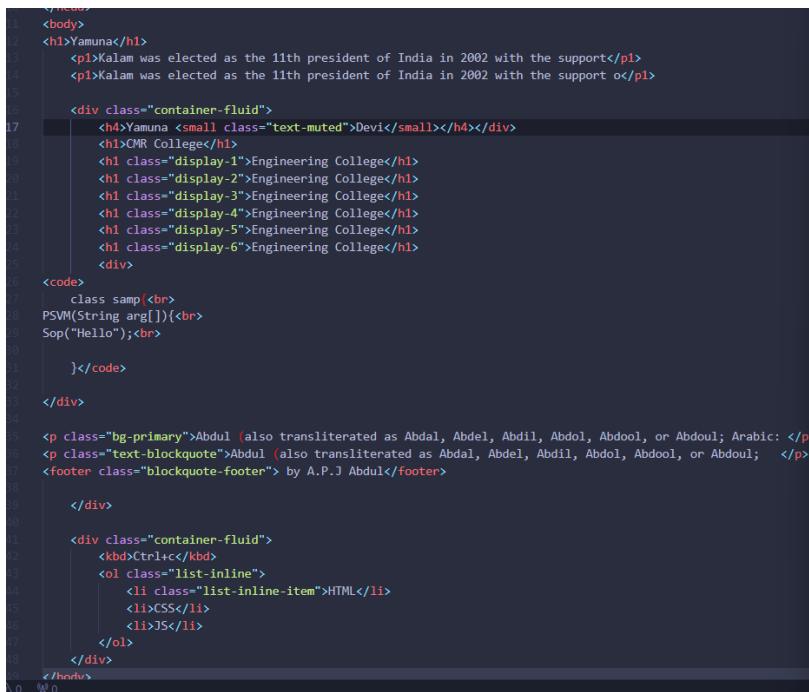
**Body Text:** Bootstrap offers styles for body text.

**Text Utilities:** Bootstrap includes a variety of text utilities that allow you to modify text alignment, transform, weight, color, and more. T

**Responsive Display Heading Classes:** Bootstrap offers classes like display-1, display-2, display-3, display-4, display-5 etc., for creating large, impactful headings that scale well across different screen sizes.

**Customization:** While Bootstrap provides default typography styles, we can easily customize them to match our design requirements. We can modify font sizes, weights, line heights.

**Responsive Font Sizes:** Bootstrap utilizes responsive font sizes to ensure text scales appropriately across various viewport sizes. This helps maintain readability and usability on different devices.



```
<!-- Head -->
<body>
<h1>Yamuna</h1>
<p>Kalam was elected as the 11th president of India in 2002 with the support</p>
<p>Kalam was elected as the 11th president of India in 2002 with the support o</p>

<div class="container-fluid">
<h4>Yamuna <small class="text-muted">Devi</small></h4></div>
<h1>CMR College</h1>
<h1 class="display-1">Engineering College</h1>
<h1 class="display-2">Engineering College</h1>
<h1 class="display-3">Engineering College</h1>
<h1 class="display-4">Engineering College</h1>
<h1 class="display-5">Engineering College</h1>
<h1 class="display-6">Engineering College</h1>
</div>

<code>
    class samp(<br>
PSVM(String arg[]){<br>
Sop("Hello");<br>
}
</code>
</div>

<p class="bg-primary">Abdul (also transliterated as Abdal, Abdel, Abdil, Abdool, or Abdoul; Arabic: </p>
<p class="text-blockquote">Abdul (also transliterated as Abdal, Abdel, Abdil, Abdool, or Abdoul; </p>
<footer class="blockquote-footer"> by A.P.J Abdul</footer>

</div>

<div class="container-fluid">
<kbd>Ctrl+c</kbd>
<ol class="list-inline">
<li class="list-inline-item">HTML</li>
<li>CSS</li>
<li>JS</li>
</ol>
</div>
</body>
```

CMR College  
Engineering College  
Abdul (also transliterated as Abdal, Abdel, Abdil, Abdool, or Abdoul; Arabic:  
Abdul (also transliterated as Abdal, Abdel, Abdil, Abdool, or Abdoul;  
— by A.P.J Abdul  
Ctrl+c  
HTML  
CSS  
JS

## Navigation Bar

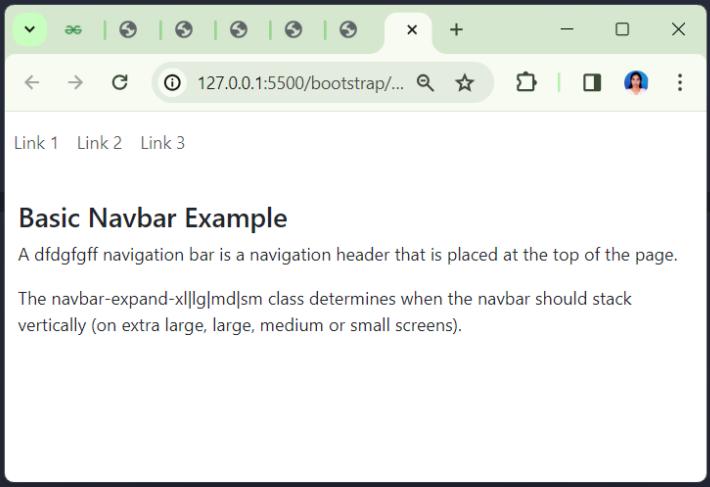
A navigation bar is used in every website to make it more user-friendly so that the navigation through the website becomes easy and the user can directly search for the topic of their interest.

**Navbar:** Bootstrap provide various types of navigation bars:

- Basic Navbar
- Inverted Navbar
- Coloured Navigation Bar
- Right-Aligned Navbar
- Fixed Navigation Bar
- Drop-Down menu Navbar
- Collapsible Navigation Bar.
-

## i). Basic Navbar:

Here, we will initialize it by using a nav tag which is an inbuilt tag just like the paragraph tag and the header tags. Within this tag, write the class="navbar navbar-default" which are the inbuilt classes of the bootstrap that give the top space of your web page to your navigation part and the default part is for the default view of that bar which is white in color. Then going further, we will have two div tags one class="container-fluid" and the other class="navbar-header".



The screenshot shows a browser window with the URL 127.0.0.1:5500/bootstrap/. The page displays a dark blue navigation bar at the top with three items: "Link 1", "Link 2", and "Link 3". Below the navbar, the main content area has a dark background with white text. It contains an h3 tag "Basic Navbar Example" and two paragraphs explaining the navbar's behavior and styling.

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
    <link rel="stylesheet" href="./css/bootstrap.css">
    <script src="./js/bootstrap.js"></script>
    <script src="./js/jquery.js"></script>
</head>
<body>
    <nav class="navbar navbar-expand-sm nav-dark">
        <ul class="navbar-nav navbar-expand-md bg-success">
            <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 dfdfgfff navigation bar is a navigation header that is placed at the top of the page.</p>
        <p>The navbar-expand-x|lg|md|sm class determines when the navbar should stack vertically (on extra large, large, medium or small screens).</p>
    </div>
</body>
</html>
```

## ii. 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="./css/bootstrap.css">
    <script src="./js/bootstrap.js"></script>
    <script src="./js/jquery.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>

```

strap > NavbarBasic.html > html > body > nav.navbar.navbar-expand-sm.nav-dark.bg-primary > ul.navbar-nav > li.nav-item > a.nav-link

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
  <link rel="stylesheet" href="./css/bootstrap.css">
  <script src="./js/bootstrap.js"></script>
  <script src="./js/jquery.js"></script>
</head>
<body>
  <nav class="navbar navbar-expand-sm nav-dark bg-primary">
    <ul class="navbar-nav" navbar-expand-md">

      <li class="nav-item">
        <a class="nav-link" href="#">Colour 1</a>
      </li>

      <li class="nav-item">
        <a class="nav-link" href="#">Colour 2</a>
      </li>

      <li class="nav-item">
        <a class="nav-link" href="#">Colour 3</a>
      </li>
    </ul>
  </nav>
  <br>
  <div class="container-fluid">
    <h3>Coloured 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>

```

### iii. **Collapsible Navigation Bar**

Collapsible Navigation Bar: To hide links on the navigation bar and make them visible after clicking a button, we add a collapsible navigation bar.

Add a navbar-toggle class to a button, then add data-toggle="collapse", then add an id to the list of links and add the name of that id inside data-target="#id". Then finally, wrap all the content of the navbar inside a div element with class navbar-collapse collapse

```

<!DOCTYPE html>
<html lang="en">
<head>
  <link rel="stylesheet" href="./css/bootstrap.css">
  <script src="./js/bootstrap.js"></script>
  <script src="./js/jquery.js"></script>
</head>
<body>
  <nav class="navbar navbar-expand-lg bg-primary">
    <div class="container-fluid">
      <a class="navbar-brand" href="#">CMR College</a>
      <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarSupportedContent"

```

```
aria-controls="navbarSupportedContent" aria-expanded="false" aria-
label="Toggle navigation">
<span class="navbar-toggler-icon"></span>
</button>

<div class="collapse navbar-collapse" id="navbarSupportedContent">
<ul class="navbar-nav me-auto mb-2 mb-lg-0">
<li class="nav-item">
<a class="nav-link active" aria-current="page" href="#">Home</a>
</li>

<li class="nav-item">
<a class="nav-link disabled">Message</a>
</li>

<li class="nav-item">
<a class="nav-link" href="#">About Us</a>
</li>

<li class="nav-item dropdown">
<a class="nav-link dropdown-toggle" href="#" role="button"
data-bs-toggle="dropdown" aria-expanded="false">
Profile
</a>
<ul class="dropdown-menu">
<li><a class="dropdown-item" href="#">User 1</a></li>
<li><a class="dropdown-item" href="#">User 2</a></li>
</ul>
</li>

</ul>
<form class="d-flex" role="search">
<input class="form-control me-2" type="search"
placeholder="Search Here" aria-label="Search">
<button class="btn btn-outline-warning" type="submit">Search</button>
</form>
</div>
</div>
</nav>
</body>
</html>
```

The screenshot shows a web browser window with the URL 127.0.0.1:5500/bootstrap/css/tpointnavbar.html#. The page displays a navigation bar at the top. On the left, the HTML code for the navigation bar is visible. The navigation bar includes a logo for 'CMR College', a 'Home' link (which is active), a 'Message' link (disabled), a 'About Us' link, a 'Profile' dropdown menu containing 'User 1' and 'User 2', and a search bar with placeholder text 'Search Here' and a 'Search' button.

#### iv. Dropdown Navigation Bar

Navigation bars can also hold dropdown menus.

A dropdown menu is a toggleable menu that allows the user to choose one value from a predefined list:

Dropdowns are toggleable, contextual overlays for displaying lists of links and more. They're made interactive with the included Bootstrap dropdown JavaScript plugin. They're toggled by clicking, not by hovering; this is an intentional design decision.

#### Program:

```
<!DOCTYPE html>
<html lang="en">
<head>
<title>My Personal website</title>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1">
<link rel="stylesheet" href="./css/bootstrap.css">
  <script src=".js/bootstrap.js"></script>
  <script src=".js/jquery.js"></script>
<style>
  body{ margin: 20px; background-color:pink; }
</style>
</head> <body>
<ul class="nav nav-pills nav-justified">
  <li class="nav-item">
    <a href="#" class="nav-link active"><i class="fa fa-home"></i>Home</a>
  </li>
  <li class="nav-item dropdown">
    <a href="#" class="nav-link dropdown-toggle" data-toggle="dropdown">
      <i class = "fa fa-user"></i>Profile</a>
      <div class="dropdown-menu">
        <a href="#" class="dropdown-item">User 1</a>
        <a href="#" class="dropdown-item">User 2</a>
      </ul>
    </div>
  </li>
</ul>
<form class="d-flex" role="search">
  <input class="form-control me-2" type="search"
    placeholder="Search Here" aria-label="Search">
  <button class="btn btn-outline-warning" type="submit">Search</button>
</form>
</div>
</div>
```

```

        <a href="#" class="dropdown-item">User 2</a>
        <a href="#" class="dropdown-item">User 3</a>
        <a href="#" class="dropdown-item">User 4</a>
    </div>
</li>
<li class="nav-item">
    <a href="#" class="nav-link"><i class ="fa fa-envelope"></i>Personal
Messages</a>
</li>
<li class="nav-item">
    <a href="#" class="nav-link"><i class ="fa fa-area-chart"></i>Contact Us</a>
</li>
<li class="nav-item">
    <a href="#" class="nav-link disabled"><i class ="fa fa-address-card"></i>About
Us</a>
</li>
<div class="nav nav-pill ">
    <a href="#" class="btn btn-primary btn-lg"><i class="fa fa-search">Search</i></a>
    <a href="#" class="btn btn-secondary btn-lg"><i class="fa fa-users">User</i></a>
</div>
</ul>
</body>
</html>

```

The screenshot shows a web browser window with the URL 127.0.0.1:5500/bootstrap/Abhi-Lab2.html. The page has a pink background. At the top is a navigation bar with the following items: Home (blue button), Profile (dropdown menu), Personal Messages, Contact Us, About Us, a search input field with placeholder 'Search', and a user icon. Below the navigation bar is a list of links. A dropdown menu is visible over the 'Profile' link, containing four items: 'User 1', 'User 2', 'User 3', and 'User 4'. The rest of the links are standard anchor tags.

```

<style>
    body{ margin: 20px; background-color: #pink; }</style>
</head> <body>
    <ul class="nav nav-pills nav-justified">
        <li class="nav-item">
            <a href="#" class="nav-link active"><i class="fa fa-home"></i>Home</a>
        </li>
        <li class="nav-item dropdown">
            <a href="#" class="nav-link dropdown-toggle" data-toggle="dropdown">
                <i class="fa fa-user"></i>Profile
            </a>
            <div class="dropdown-menu">
                <a href="#" class="dropdown-item">User 1</a>
                <a href="#" class="dropdown-item">User 2</a>
                <a href="#" class="dropdown-item">User 3</a>
                <a href="#" class="dropdown-item">User 4</a>
            </div>
        </li>
        <li class="nav-item">
            <a href="#" class="nav-link"><i class ="fa fa-envelope"></i>Personal Messages</a>
        </li>
        <li class="nav-item">
            <a href="#" class="nav-link"><i class ="fa fa-area-chart"></i>Contact Us</a>
        </li>
        <li class="nav-item">
            <a href="#" class="nav-link disabled"><i class ="fa fa-address-card"></i>About Us</a>
        </li>
    <div class="nav nav-pill ">
        <a href="#" class="btn btn-primary btn-lg"><i class="fa fa-search">Search</i></a>
        <a href="#" class="btn btn-secondary btn-lg"><i class="fa fa-users">User</i></a>
    </div>
</ul>
</body>
</html>

```

## Bootstrap Table

Table is a collection of rows and columns.

- Just add the base class .table to any <table>, then extend with custom styles or our various included modifier classes.
- A <caption> functions like a heading for a table.

- o we can also invert the colors—with light text on dark backgrounds—with .table-dark.
- o Similar to tables and dark tables, use the modifier classes .thead-light or .thead-dark to make <thead>s appear light or dark gray.
- o Use .table-striped to add zebra-striping to any table row within the <tbody>.
- o Add .table-bordered for borders on all sides of the table and cells.
- o Add .table-hover to enable a hover state on table rows within a <tbody>.
- o Add .table-sm to make tables more compact by cutting cell padding in half.
- o Responsive tables allow tables to be scrolled horizontally with ease. Make any table responsive across all viewports by wrapping a .table with .table-responsive.

**Example:**

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Bootstrap Table</title>
  <link rel="stylesheet" href="bootstrap/css/bootstrap.css">
  <script src="bootstrap/js/bootstrap.js"></script>
  <script src="bootstrap/js/jQuety.js"></script>
</head>
<body>
  <h1><center>Bootstrap Table</center> </h1>
  <div>
<table class="table table-sm">
  <tr class="table-dark">
    <th>Sno</th>
    <th>FirstName</th>
    <th>LastName</th>
    <th>Email</th>
  </tr>
  <tr class="table-primary">
    <td>111</td>
    <td>Ramana</td>
    <td>Naidu</td>
    <td>123@gmail.com</td>
  </tr>
  <tr class="table-success">
    <td>222</td>
    <td>Harika</td>
    <td>Rai</td>
    <td>546@gmail.com</td>
  </tr>
</table>
```

```

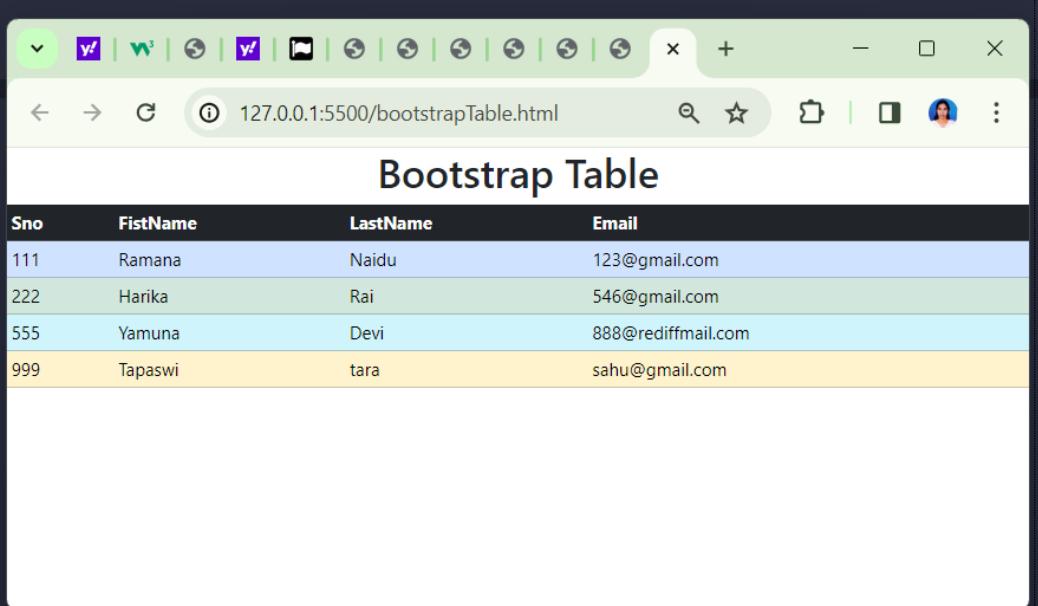
</tr>
<tr class="table-info">
    <td>555</td>
    <td>Yamuna</td>
    <td>Devi</td>
    <td>888@rediffmail.com</td>
</tr>
<tr class="table-warning">
    <td>999</td>
    <td>Tapaswi</td>
    <td>tara</td>
    <td>sahu@gmail.com</td>
</tr>
</table>
</div>
</body>
</html>

```

```

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Bootstrap Table</title>
    <link rel="stylesheet" href="bootstrap/css/bootstrap.css">
    <script src="bootstrap/js/bootstrap.js"></script>
    <script src="bootstrap/js/jquety.js"></script>
</head>
<body>
    <h1><center>Bootstrap Table</center> </h1>
    <div>
        <table class="table table-sm">
            <tr class="table-dark">
                <th>Sno</th>
                <th>FirstName</th>
                <th>LastName</th>
                <th>Email</th>
            </tr>
            <tr class="table-primary">
                <td>111</td>
                <td>Ramana</td>
                <td>Naidu</td>
                <td>123@gmail.com</td>
            </tr>
            <tr class="table-success">
                <td>222</td>
                <td>Harika</td>
                <td>Rai</td>
                <td>546@gmail.com</td>
            </tr>
            <tr class="table-info">
                <td>555</td>
                <td>Yamuna</td>
                <td>Devi</td>
                <td>888@rediffmail.com</td>
            </tr>
            <tr class="table-warning">
                <td>999</td>
                <td>Tapaswi</td>
                <td>tara</td>
                <td>sahu@gmail.com</td>
            </tr>
        </table>
    </div>

```



## Forms controls

Form is a collection of Elements

*Textual form controls:*

like `<input>`s, `<select>`s, and `<textarea>`s—are styled with the `.form-control` class. Included are styles for general appearance, focus state, sizing, and more. *Checkboxes and radios:* Default checkboxes and radios are improved upon with the help of `.form-`

**check.** Checkboxes are for selecting one or several options in a list, while radios are for selecting one option from many.

*Default (stacked):* By default, any number of checkboxes and radios that are immediate sibling will be vertically stacked and appropriately spaced.

```
<div class="form-check">
<input class="form-check-input" type="checkbox" value="" id="defaultCheck1">
<label class="form-check-label" for="defaultCheck1">
  Default checkbox
</label>
</div>
```

**Form groups:** The .form-group class is the easiest way to add some structure to forms. It provides a flexible class that encourages proper grouping of labels, controls, optional help text, and form validation messaging.

**Sizing:** Set heights using classes like .form-control-lg and .form-control-sm.

## Bootstrap Form Layouts

Bootstrap forms provides two types of form layouts:

- Stacked Form or Vertical form
- Inline form

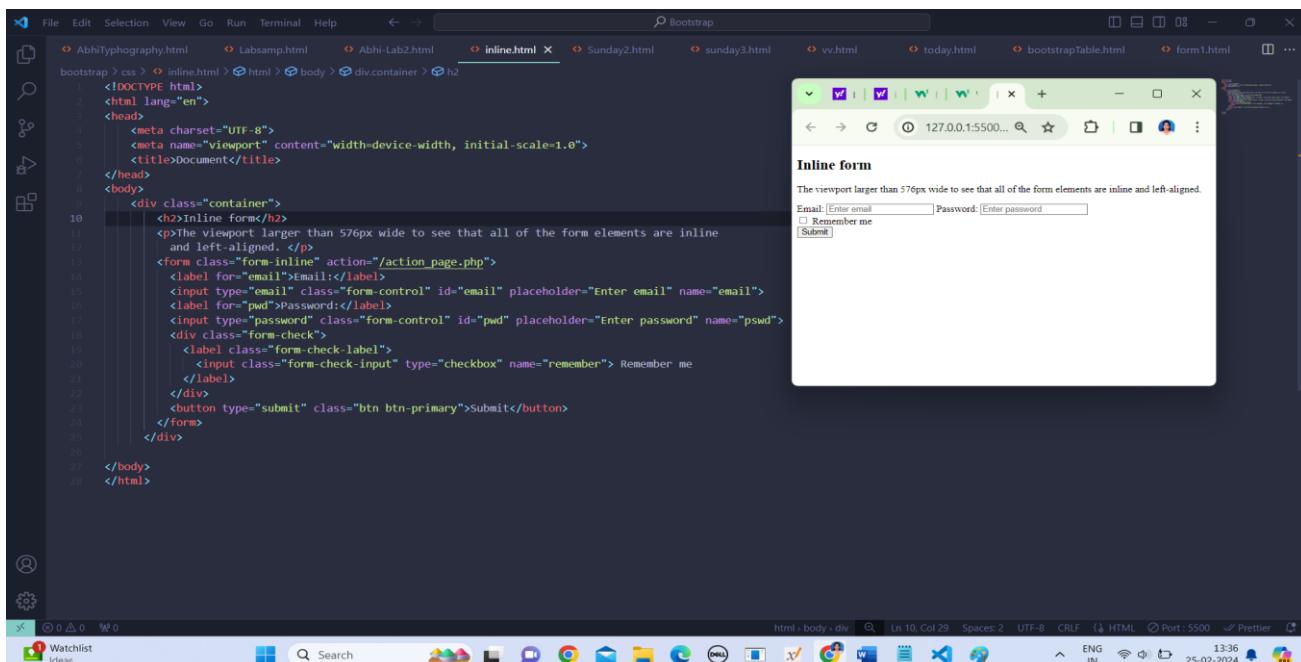
Standard rules for all two form layouts:

- Wrap labels and form controls in <div class="form-group">
- Add class .form-control to all textual <input>, <textarea>, and <select> elements

### Inline forms

Use the .form-inline class to display a series of labels, form controls, and buttons on a single horizontal row. Form controls within inline forms vary slightly from their default states.

Controls only appear inline in viewports that are at least 576px wide to account for narrow viewports on mobile devices.

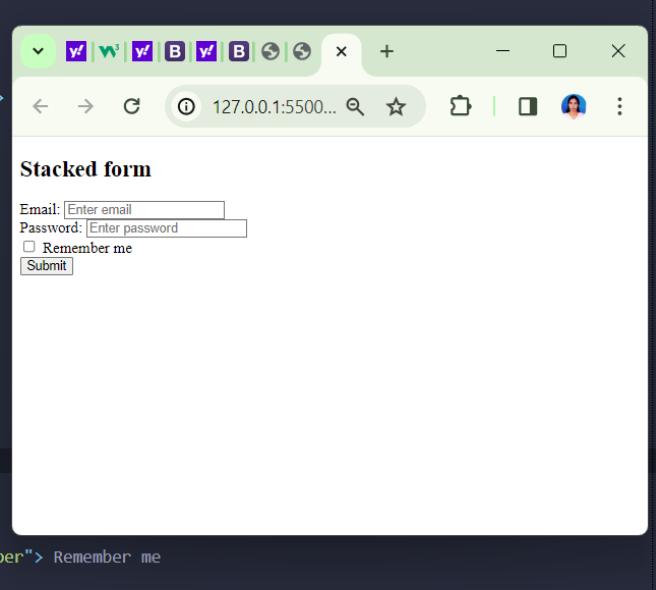


## Stacked forms

Stacked form is a series of labels, form controls, and buttons on a single vertical row. Form controls within inline forms vary slightly from their default states.

By default, any number of checkboxes and radios that are immediate sibling will be vertically stacked and appropriately.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
</head>
<body>
  <div class="container">
    <h2>Stacked form</h2>
    <form action="./myCMR.html">
      <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>
```



## Form with all Elements:

### Example:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
  <link rel="stylesheet" href="./bootstrap.css">
  <script src="./js/jquery.js"></script>
  <script src="./bootstrap.js"></script>
</head>
<body>
  <label for=" " > Name</label>
  <input type="text"> <br>
  <label for=" " > Name</label>
  <input type="text" class="form-control w-50"> <br>
  <label for=" " class="form-label"> Name</label>
  <input type="text" class="form-control w-50"> <br>
```

```

<div class="input-group w-50">
  <label for=" " class="input-group-text"> weight</label>
  <input type="text" class="form-control w-50">
  <span class="input-group-text">Kgs</span>
</div>  <br>
<select name="" class="form-control w-10">
  <option>CMR1</option>
  <option>CMR2</option>
  <option>CMR3</option>  </select>  <br>
<label for="">Remember Me</label>
<input type="checkbox">  <br>  <div>
<label for="" class="form-check-label">Remember Me</label>
<input type="checkbox" class="form-check-input">  <br>
<label for="">Color</label>
<input type="range"/>  <br>
<label for="" class="form-label">Color</label>
<input type="range" class="form-range w-50"/>  <br>
<button>Click me</button><br>
<button class="btn">Click Me</button><br>
<button class="btn btn-primary">Click Me</button><br>
<input type="file" class="form-control">
  </div>
</div>s
</body>
</html>

```

The screenshot displays a comparison between the source HTML code and its visual representation in a browser.

**Left Side (HTML Code):**

```

<link rel="stylesheet" href="./bootstrap.css">
<script src="./js/jquery.js"></script>
<script src="./bootstrap.js"></script>
</head>
<body>
  <label for=" " > Name</label>
  <input type="text">
  <br>
  <label for=" " > Name</label>
  <input type="text" class="form-control w-50">
  <br>
  <label for=" " class="form-label"> Name</label>
  <input type="text" class="form-control w-50">
  <br>
  <div class="input-group w-50">
    <label for=" " class="input-group-text"> weight</label>
    <input type="text" class="form-control w-50">
    <span class="input-group-text">Kgs</span>
  </div>
  <br>
  <select name="" class="form-control w-10">
    <option>CMR1</option>
    <option>CMR2</option>
    <option>CMR3</option>
  </select>
  <br>
  <label for="">Remember Me</label>
  <input type="checkbox">
  <br>
  <div>
    <label for="" class="form-check-label">Remember Me</label>
    <input type="checkbox" class="form-check-input">
  <br>
  <label for="">Color</label>
  <input type="range"/>
  <br>
  <label for="" class="form-label">Color</label>
  <input type="range" class="form-range w-50"/>
  <br>
  <button>Click me</button><br>
  <button class="btn">Click Me</button><br>
  <button class="btn btn-primary">Click Me</button><br>
  <input type="file" class="form-control">
    </div>
  </div>s
</body>
</html>

```

**Right Side (Rendered UI):**

- Name:** A text input field labeled "Name".
- Weight:** A text input field labeled "weight" with a "Kgs" suffix.
- Select:** A dropdown menu with options CMR1, CMR2, and CMR3. The option "CMR1" is currently selected.
- Checkboxes:** Two checkboxes labeled "Remember Me".
- Color:** A color picker and a range slider labeled "Color".
- Buttons:** Three buttons labeled "Click me", "Click Me", and "Click Me".
- File Input:** A file input field labeled "Choose File" with the message "No file chosen".