

**TOPS TECHNOLOGIES**

Training | Outsourcing | Placement | Study Abroad

# Website Designing

# Modules

**Module 1 - [Introduction]**

**Module 2 - [HTML ]**

**Module 3 - [CSS and CSS 3 ]**

**Module 4 - [HTML5 ]**

**Module 5 - [Wordpress]**

**Module 6 - [JAVASCRIPT BASIC & DOM ]**

**Module 7 - [ Advance CSS ]**

**Module 8 - [ JQuery ]**

**Module 9 - [ Framework]**

# **Module - 1**

## **[ Introduction ]**

- ✓ Careers in Web Technologies and Job Roles
- ✓ How the Website Works?
- ✓ Client and Server Scripting Languages
- ✓ Domains and Hosting
- ✓ Types of Websites (Static and Dynamic Websites)
- ✓ Web Standards and W3C recommendations
- ✓ Responsive Web Designing
- ✓ Protocol
- ✓ Basics of SEO
- ✓ Basic of html

[Github Link :https://github.com/TopsCode/WEB\\_DESIGNING/blob/main/Module-1/HTML/01\\_Intro.html](https://github.com/TopsCode/WEB_DESIGNING/blob/main/Module-1/HTML/01_Intro.html)

# Careers in Web Technologies and Job Roles

The average base salary of a web developer in India is around Rs 3,08,000 per annum that includes around Rs 30,000 in bonuses and Rs 20,000 on a profit-sharing basis. This figure can go up to a maximum of 7,80,000 per annum or even beyond that depending on your experience, skillset, certifications, location, and employer.

[Github Link :\[https://github.com/TopsCode/WEB\\\_DESIGNING/blob/main/Module-1/HTML/01\\\_Intro.html\]\(https://github.com/TopsCode/WEB\_DESIGNING/blob/main/Module-1/HTML/01\_Intro.html\)](https://github.com/TopsCode/WEB_DESIGNING/blob/main/Module-1/HTML/01_Intro.html)

## Average Web Developer Salary in India

₹308,040

Avg. Salary

Show Hourly Rate

₹29,700  
BONUS

₹22,500  
COMMISSION

₹20,067  
PROFIT SHARING

What am I worth?

Get pay report

How should I pay?

Price a job

The average salary for a Web Developer in India is ₹308,040.



A Web Developer typically makes between ₹123k - ₹777k.

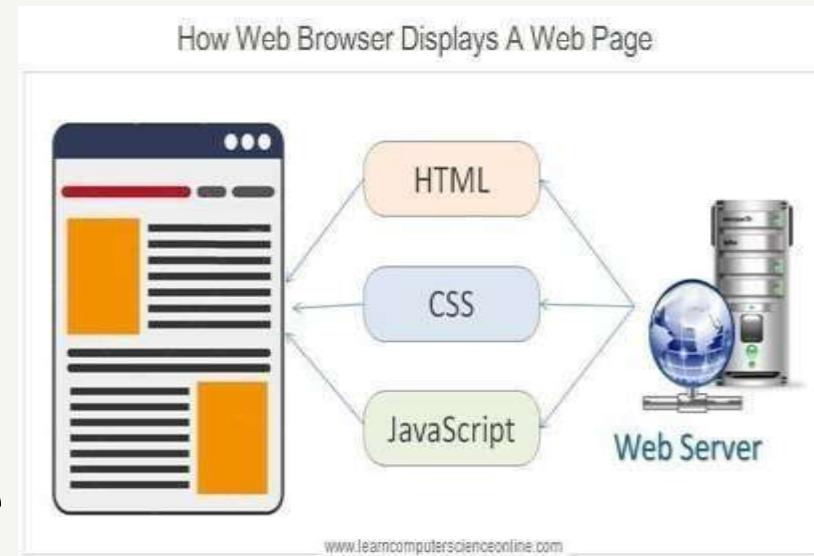


[Github Link :https://github.com/TopsCode/WEB\\_DESIGNING/blob/main/Module-1/HTML/01\\_Intro.html](https://github.com/TopsCode/WEB_DESIGNING/blob/main/Module-1/HTML/01_Intro.html)

# How the Website Works?

**How the web works** provides a simplified view of what happens when you view a webpage in a web browser on your computer or phone.

This theory is not essential to writing web code in the short term, but before long you'll really start to benefit from understanding.



[Github Link :\[https://github.com/TopsCode/WEB\\\_DESIGNING/blob/main/Module-1/HTML/01\\\_Intro.html\]\(https://github.com/TopsCode/WEB\_DESIGNING/blob/main/Module-1/HTML/01\_Intro.html\)](https://github.com/TopsCode/WEB_DESIGNING/blob/main/Module-1/HTML/01_Intro.html)

# What is website?

A website is a collection of many web pages, and web pages are digital files that are written using HTML(Hypertext Markup Language). To make your website available to every person in the world, it must be stored or hosted on a computer connected to the Internet round a clock. Such computers are known as a **Web Server**.



[Github Link :\[https://github.com/TopsCode/WEB\\\_DESIGNING/blob/main/Module-1/HTML/01\\\_Intro.html\]\(https://github.com/TopsCode/WEB\_DESIGNING/blob/main/Module-1/HTML/01\_Intro.html\)](https://github.com/TopsCode/WEB_DESIGNING/blob/main/Module-1/HTML/01_Intro.html)

# Types of websites Static website

In Static Websites, Web pages are returned by the server which are prebuilt source code files built using simple languages such as HTML, CSS, or JavaScript. There is no processing of content on the server (according to the user) in Static Websites. Web pages are returned by the server with no change therefore, static Websites are fast .

[Github Link :https://github.com/TopsCode/WEB\\_DESIGNING/blob/main/Module-1/HTML/01\\_Intro.html](https://github.com/TopsCode/WEB_DESIGNING/blob/main/Module-1/HTML/01_Intro.html)

# Dynamic website

In Dynamic Websites, Web pages are returned by the server which is processed during runtime means they are not prebuilt web pages, but they are built during runtime according to the user's demand with the help of server-side scripting languages such as PHP, Node.js, ASP.NET and many more supported by the server.

**Github Link :**[https://github.com/TopsCode/WEB\\_DESIGNING/blob/main/Module-1/HTML/01\\_Intro.html](https://github.com/TopsCode/WEB_DESIGNING/blob/main/Module-1/HTML/01_Intro.html)

# Difference between Web Designer and Web Developer

In Dynamic Websites, Web pages are returned by the server which is processed during runtime means they are not prebuilt web pages, but they are built during runtime according to the user's demand with the help of server-side scripting languages such as PHP, Node.js, ASP.NET and many more supported by the server.

# Client and Server

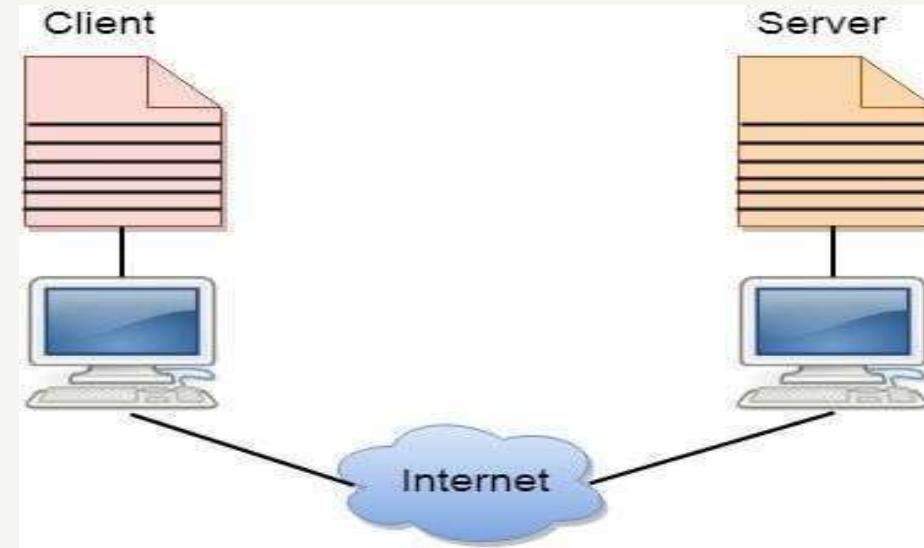
## Client

A client is a program that runs on the local machine requesting service from the server. A client program is a finite program means that the service started by the user and terminates when the service is completed.

**Github Link :**[https://github.com/TopsCode/WEB\\_DESIGNING/blob/main/Module-1/HTML/01\\_Intro.html](https://github.com/TopsCode/WEB_DESIGNING/blob/main/Module-1/HTML/01_Intro.html)

# Server

A server is a program that runs on the remote machine providing services to the clients. When the client requests for a service, then the server opens the door for the incoming requests, but it never initiates the service.



GitHub Link :[https://github.com/TopsCode/WEB\\_DESIGNING/blob/main/Module-1/HTML/01\\_Intro.html](https://github.com/TopsCode/WEB_DESIGNING/blob/main/Module-1/HTML/01_Intro.html)

# Domains and Hosting

## Domain

A Domain name is an address where one can find the website by typing the web address in the browser URL bar to visit a website. When you enter the domain name of the website in the search box, a powerful engine searches the web's largest pool of names and takes us to the website

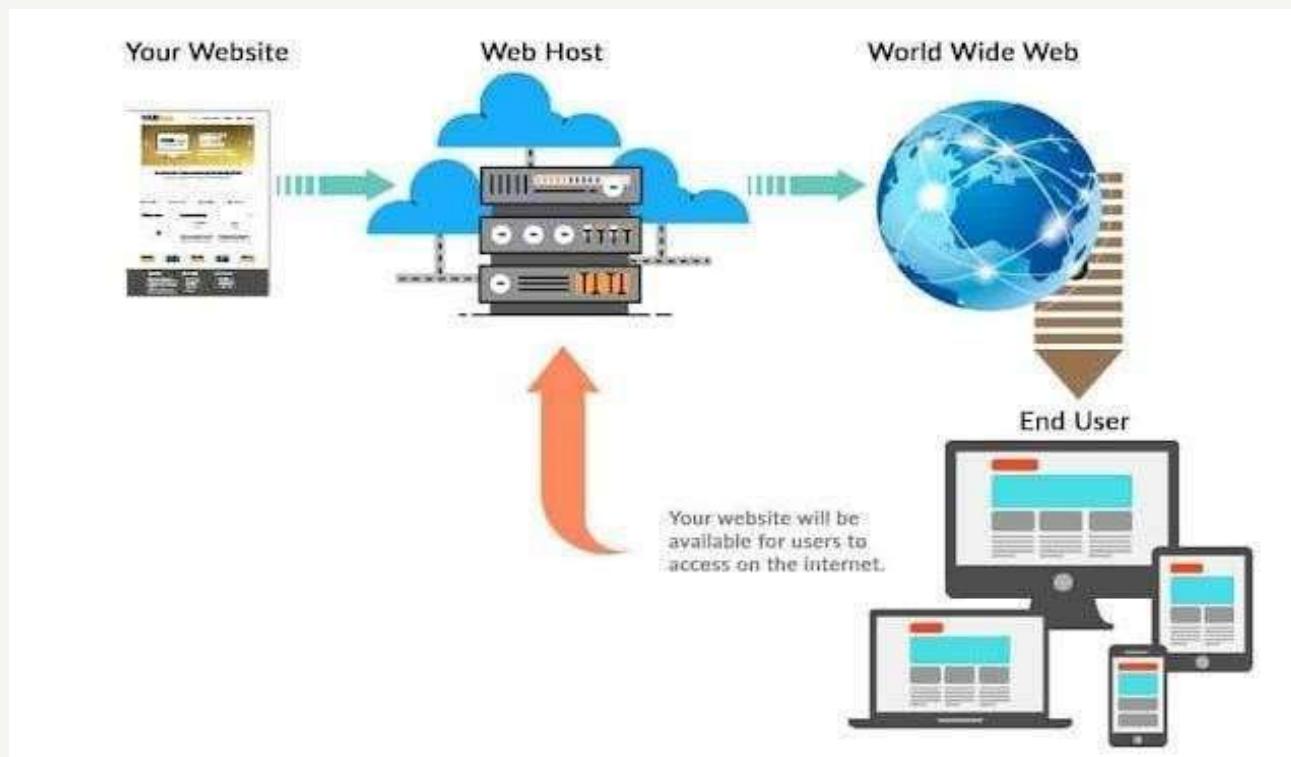


# Hosting

Web hosting is a service that allows organizations and individuals to post a website or web page onto the Internet. A web host, or web hosting service provider, is a business that provides the technologies and services needed for the website or webpage to be viewed in the Internet. Websites are hosted, or stored, on special computers called servers. When Internet users want to view your website, all they need to do is type your website address or domain into their browser.

Github Link :[https://github.com/TopsCode/WEB\\_DESIGNING/blob/main/Module-1/HTML/01\\_Intro.html](https://github.com/TopsCode/WEB_DESIGNING/blob/main/Module-1/HTML/01_Intro.html)

# Hosting



# W3C standard

The World Wide Web Consortium (W3C) develops international Web standards: HTML, CSS, and many more. W3C's Web standards are called *W3C Recommendations*. All W3C standards are reviewed for accessibility support by the Accessible Platform Architectures ([APA](#)) Working Group. The W3C standards and Working Group Notes introduced below are particularly relevant to accessibility.

# W3C Recommendation

The W3C Recommendation Track process is designed to maximize consensus about the content of a technical report, to ensure high technical and editorial quality, and to earn endorsement by W3C and the broader community.

**W3C Recommendation**

**Media Queries**  
W3C Recommendation 19 June 2012

**This Version:**  
<http://www.w3.org/TR/2012/REC-css3-mediaqueries-20120619/>

**Latest Version:**  
<http://www.w3.org/TR/css3-mediaqueries/>

# Responsive web designing

Responsive web design is about creating web pages that look good on all devices!

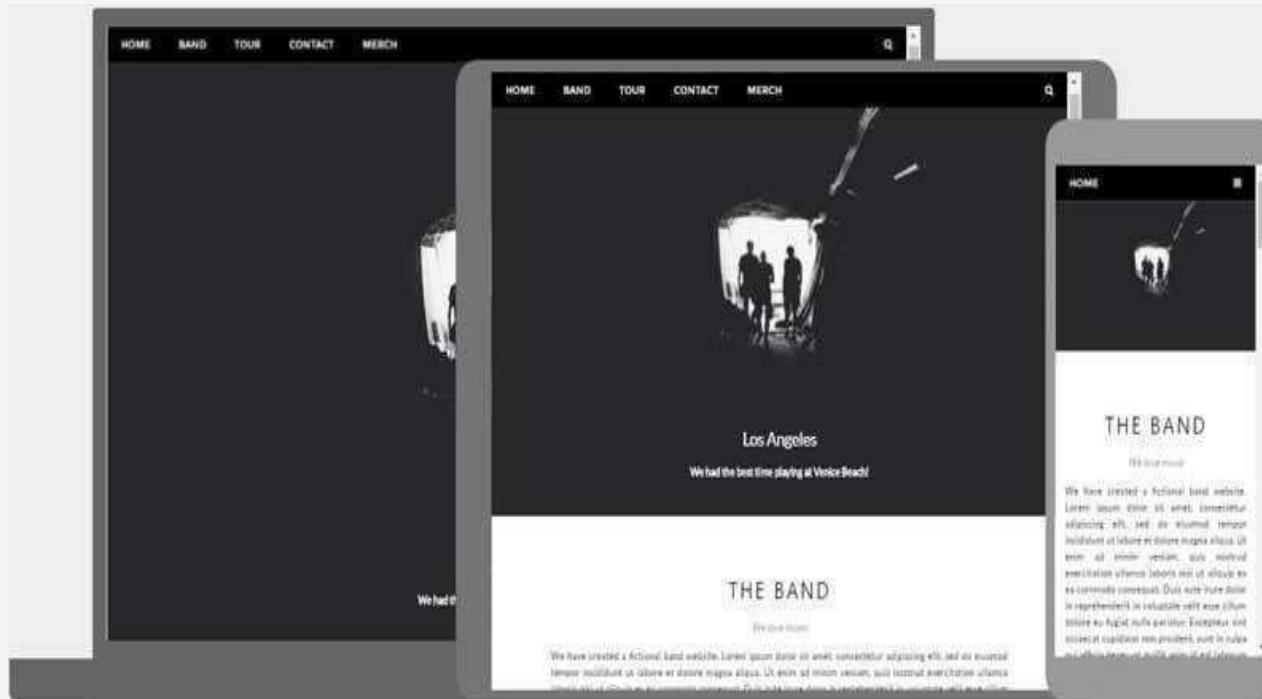
A responsive web design will automatically adjust for different screen sizes and viewports.

## Setting The Viewport

To create a responsive website, add the following <meta> tag to all your web pages:

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

# Responsive web designing



# Protocol

It is a digital language through which we communicate with others on the [Internet](#). protocol meaning is that it a set of mutually accepted and implemented rules at both ends of the communications channel for the proper exchange of [information](#).

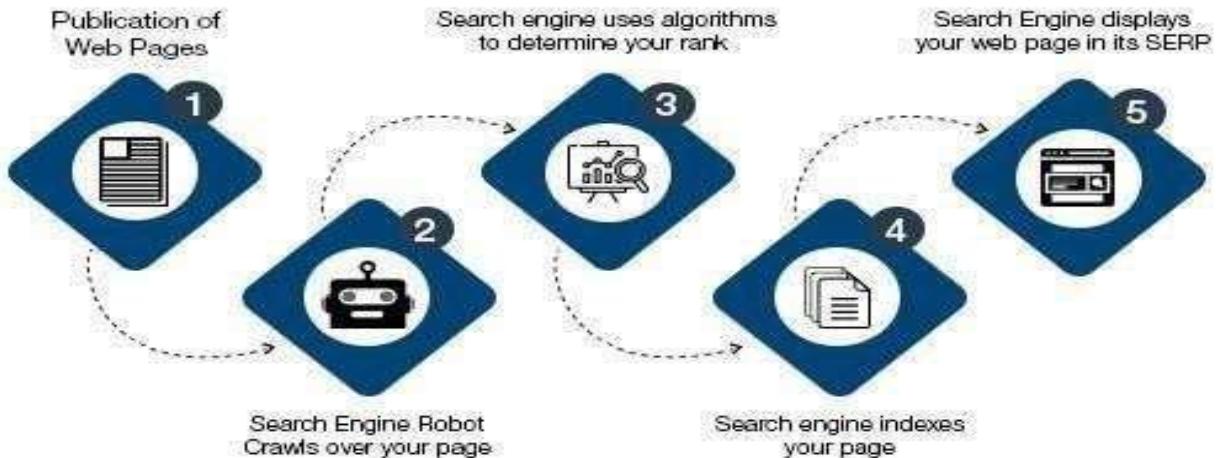
## What Is Internet?

The Internet is a global network of interconnected computers and servers that allows for the exchange of data and communication between devices all over the world.

# What is SEO

Search engine optimization is the process of improving the quality and quantity of website traffic to a website or a web page from search engines. SEO targets unpaid traffic rather than direct traffic or paid traffic.

## How SEO Works



Well, SEO stands for 'Search Engine Optimization', which is **the process of getting traffic from free, organic, editorial, or natural search results in search engines.** It aims to improve your website's position in search results pages. Remember, the higher the website is listed; the more people will see it.



# Web Development Tools

- Git and GITHUB Training
- Web development tools and environments are essential for building, testing, and deploying web applications. Here's an overview of some key tools and environments used in web development:
  - Code Editors and Integrated Development Environments (IDEs) - Visual Studio, sublime, atom, jetbrains
  - Version Control Systems - Git, Github, bitbucket
  - Package Managers - NPM, Yarn, composer
  - Task Runners and Build Tools - Grunt, Gulp, Webpack, Parcel

- Frameworks and Libraries - React, angular, vue.js, bootstrap, tailwind
- Development and Testing Servers - Node.js, Apache, Ngnix, Live Server
- Database management systems - MySQL, PostgreSQL, MongoDB, SQLite
- Front End Build Tools - SaSS, LESS, Babel
- Development Environments - XAMPP, MAMP
- Browser Developer Tools - Chrome DevTools, Firefox Developer Tools, Safari Developer Tools
  - Collaboration and Communication Tools - trello, slack, Asana
  - Continuous Integration and Deployment (CI/CD) Tool - Jenkins, Travis, CirceCI, Github Actions
- Design and Prototyping Tools - Figma, Adobe XD, Sketch

# **Module - 2**

## **[ HTML]**

- ✓ Introduction to HTML
- ✓ Tags in HTML
- ✓ HTML Attribute
- ✓ HTML Elements
- ✓ Text Formatting in HTML
- ✓ IFRAME And File Path in HTML
- ✓ HTML Tables
- ✓ HTML List
- ✓ HTML FORMS
- ✓ HTML Head, ID, Class and Layout
- ✓ HTML Entities
- ✓ HTML Events

- ✓ Advance HTML
- ✓ HTML Audio and Video Tag
- ✓ HTML SVG
- ✓ Scalable Vector Graphics
- ✓ Canvas and URL in HTML
- ✓ URL Encode
- ✓ XHTML
- ✓ API in HTML5

# Browsers

Google Chrome

Internet Explorer

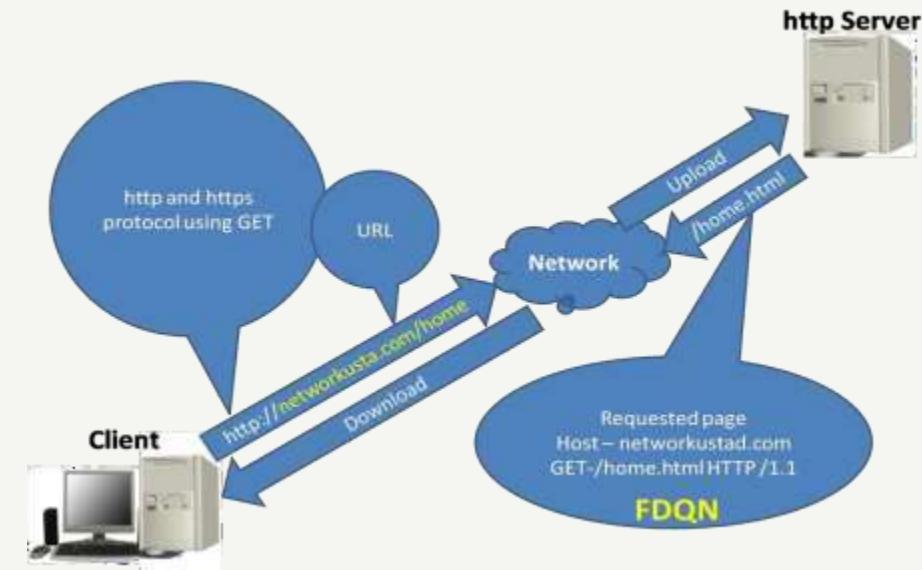
Firefox

Safari

Edge

# Hypertext Transfer Protocol

The Hypertext Transfer Protocol is an application layer protocol in the Internet protocol suite model for distributed, collaborative, hypermedia information systems



**Github link :** [https://github.com/TopsCode/WEB DESIGNING/blob/main/Module-1/HTML/01\\_Intro.html](https://github.com/TopsCode/WEB DESIGNING/blob/main/Module-1/HTML/01_Intro.html)

# What is HTML

HTML stands for Hypertext Markup Language.

HTML is used to create web pages and web applications.

HTML is widely used language on the web.

We can create a static website by HTML only.

Technically, HTML is a Markup language rather than a programming language.

**Github link :**[https://github.com/TopsCode/WEB\\_DESIGNING/blob/main/Module-1/HTML/01\\_Intro.html](https://github.com/TopsCode/WEB_DESIGNING/blob/main/Module-1/HTML/01_Intro.html)

# WWW

The World Wide Web (WWW), often simply referred to as the Web, is a system of interlinked hypertext documents and media that are accessed via the Internet using web browsers. It allows people to access a vast amount of information, services, and resources by navigating web pages that are connected through hyperlinks.

# SEO

SEO (Search Engine Optimization) is the practice of optimizing a website or online content to increase its visibility and ranking on search engine results pages (SERPs) like Google, Bing, and others. The goal of SEO is to make a website more attractive to search engines, helping it appear higher in search results for relevant queries.

# Domain Name

A domain name is a unique address used to identify a website on the Internet. It is essentially the address that people type into a web browser to visit a specific website, like [www.example.com](http://www.example.com). Domain names are easier to remember and use than numeric IP addresses, which are the actual addresses used by computers to communicate over the Internet.

Example of a Domain Name:  In [www.example.com](http://www.example.com):

"www" is the subdomain (optional, typically used to indicate a website).  
"example" is the second-level domain (SLD).  
.com" is the top-level domain (TLD).

# Text-Editor

A text editor is a software application used to create and edit plain text files. These files can contain programming code, configuration settings, or just simple text. Text editors are essential tools for developers, writers, and anyone working with text-based content. They allow you to write, modify, and save text in a variety of formats.

Visual Studio Code (VS Code) is a free, open-source, and powerful text editor developed by Microsoft. It's one of the most popular code editors, widely used by developers for web development, software engineering, and other programming tasks. It provides a lot of features that help developers write, test, and debug code efficiently.

# Text-Editor

1. Download VS Code:  Go to the official Visual Studio Code website.   
The website should automatically detect your operating system and provide a download button for the right version (Windows, macOS, or Linux).
  
2. Choose Your Platform:  Windows: Click the Download for Windows button to download the installer file (e.g., VSCodeSetup.exe).  macOS: Click the Download for macOS button to download the .zip file.  Linux: For Linux, you will have different installation options depending on the distribution (Debian, Ubuntu, Red Hat, etc.). You can download a .deb or .rpm file, or follow instructions for snap.

# Text-Editor

## 3. Install VS Code: For Windows:

1. Run the installer you downloaded (VSCodeSetup.exe).
2. Follow the on-screen instructions. The default installation settings should work for most users.
3. Select whether you want to add VS Code to your PATH for easy access from the command line.
4. After installation, you can open VS Code from the Start Menu or by searching for it in the taskbar search.

# Introduction of html

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

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

# HTML Tags

HTML tags are like keywords which defines that how web browser will format and display the content.

With the help of tags, a web browser can distinguish between an HTML content and a simple content.

HTML tags contain three main parts: opening tag, content and closing tag.

But some HTML tags are unclosed tags.

GitHub link : [https://github.com/TopsCode/WEB\\_DESIGNING/blob/main/Module-1/HTML/02\\_basictags.html](https://github.com/TopsCode/WEB_DESIGNING/blob/main/Module-1/HTML/02_basictags.html)

# Rules for Tags

All HTML tags must enclosed within <> these brackets. Every tag in HTML perform different tasks.

If you have used an open tag <tag>, then you must use a close tag  
</tag> (except some tags)

Ex. <p></p> , <strong></strong>

Github Link :[https://github.com/TopsCode/WEB\\_DESIGNING/blob/main/Module-1/HTML/02\\_basictags.html](https://github.com/TopsCode/WEB_DESIGNING/blob/main/Module-1/HTML/02_basictags.html)

# Example 1

Write an HTML program to print “Hello World”.

## Example

```
<!DOCTYPE html>
<html>
<body>
<h1>My First Heading</h1>
<p>Hello world.</p>
</body>
</html>
```

# HTML Meta Tags

```
<!DOCTYPE html>
<title>
<link>
<meta>
<style>
```

# HTML Text Tags

<p>  
<h1>, <h2>, <h3>, <h4>, <h5>, <h6>  
<strong>  
<em>  
<cite>

# HTML Unclosed Tags

<br>

<hr>

# HTML Element

These elements are responsible for creating web pages and define content in that webpage.

An element is a collection of start tag, attributes, end tag, content between them.

# Void Element

Void element: All the elements in HTML do not require to have start tag and end tag, some elements does not have content and end tag such elements are known as Void elements or empty elements. These elements are also called as unpaired tag.

Ex.

<br>

<hr>

# Block-level element

These are the elements, which structure main part of webpage, by dividing a page into coherent blocks.

A block-level element always start with new line and takes the full width of web page, from left to right.

These elements can contain block-level as well as inline elements

Ex.      <div>  
          <h1>-<h6>  
          <hr>

# Inline elements:

Inline elements are those elements, which differentiate the part of a give text and provide it a particular function.

These elements does not start with new line and take width as per requirement.

The Inline elements are mostly used with other elements.

Ex.      <a>  
          <label>  
          <b>

# Useful HTML Elements

HTML Link

Tags HTML

Image HTML

List HTML

Table HTML

Form

# Attributes

HTML attributes are special words which provide additional information about the elements or attributes are the modifier of the HTML element.

Each element or tag can have attributes, which defines the behaviour of that element.

Attributes should always be applied with start tag.

The Attribute should always be applied with its name and value pair.

The Attributes name and values are case sensitive, and it is recommended by W3C that it should be written in Lowercase only.

You can add multiple attributes in one HTML element, but need to give space between two attributes.

# Syntax

```
<element attribute_name="value">content</element>
```

Ex.

```
<body text="green" bgcolor="orange">
```

Ex.

```
<h1 title="This is heading tag">Example of title  
attribute</h1>
```

# Text Formatting

HTML Formatting is a process of formatting text for better look and feel. HTML provides us ability to format text without using CSS.

## Categories:

- Physical tag: These tags are used to provide the visual appearance to the text.
- Logical tag: These tags are used to add some logical or semantic value to the text.

Git Link : [https://github.com/TopsCode/WEB\\_DESIGNING/blob/main/Module-1/HTML/02\\_basictags.html](https://github.com/TopsCode/WEB_DESIGNING/blob/main/Module-1/HTML/02_basictags.html)

# Formatting Text

1. Bold Text: **<b>** and **<strong>**
2. Italic Text: *<i>* and *<em>*
3. Marked Formatting: **<mark>**
4. Underlined Text: <u> and  
<ins>
5. Strike Text: **<strike>** and **<del>**
6. Monospaced Font: **<tt>**
7. Superscript Text: **<sup>**
8. Subscript Text: **<sub>**
9. Larger Text: **<big>**
10. Smaller Text: **<small>**

# HTML Marquee tag

The Marquee Tag was used to create scrolling or moving text or images across the page. It was a simple way to add dynamic effects to a webpage.

Example of the marquee

direction="Left">This text moves from right to left. marquee  
direction="right">This text moves from right to left. marquee  
direction="up">This text moves from right to left. marquee  
direction="down">This text moves from right to left. marquee  
direction="slide">This text moves from right to left.

# HTML Phrase tag

The HTML phrase tags are special purpose tags, which defines the structural meaning of a block of text or semantics of text.

# Phrase Tags Ex.

1. Abbreviation: <abbr title = "">  
</abbr>
2. Marked: <mark>
3. Strong: <strong>
4. Emphasized: <em>
5. Definition: <dfn>
6. Quoting: <blockquote cite = "">, <cite>
7. Short: <q>
8. Code: <code>
9. Keyboard: <kbd>
10. Address: <address>

# HTML Comments

Syntax:

<!--

Write commented text  
here

-->

```
<html lang="en"> event
  ▶ <head> ... </head>
  ▼ <body>
    ▶ <style> ... </style>
      <!--
        The text in here will be invisible on the website! Here's
        another line of the comment. You can have as many lines as you
        want! 😊
      -->
    ▼ <div class="content">
      And here's that regular HTML content again.
    </div>
  </body>
</html>
```

# Head

The HTML <head> element is used as a container for metadata (data about data). It is used between <html> tag and <body> tag.

The head of an HTML document is a part whose content is not displayed in the browser on page loading. It just contains metadata about the HTML document which specifies data about the HTML document.

# Tags

Following is a list of tags used in metadata:

- <title>
- <link>
- <style>
- <script>
- <meta>
- <base>

# <meta>

```
<meta charset="UTF-8">
<meta name="description" content="Free Web tutorials">
<meta http-equiv="refresh" content="30">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```

# Layout



# Tags

`<header>`: It is used to define a header for a document or a section.

`<nav>`: It is used to define a container for navigation links

`<section>`: It is used to define a section in a document

`<article>`: It is used to define an independent self-contained article

`<aside>`: It is used to define content aside from the content (like a sidebar)

`<footer>`: It is used to define a footer for a document or a section

`<details>`: It is used to define additional details

`<summary>`: It is used to define a heading for the `<details>` element

# HTML Entities

HTML character entities are used as a replacement of reserved characters in HTML. You can also replace characters that are not present on your keyboard by entities.

For example: if you use less than (<) or greater than (>) symbols in your text, the browser can mix them with tags that's why character entities are used in HTML to display reserved characters.

# Syntax

&entity\_name;

OR

&#entity\_number;

# Example

<	non-breaking space	&nbsp;
>	less than	&lt;
&	greater than	&gt;
&	ampersand	&amp;
"	double quotation mark	&quot;
'	single quotation mark (apostrophe)	&apos;
¢	cent	&cent;
£	pound	&pound;
¥	yen	&yen;
€	Euro	&euro;
©	copyright	&copy;
®	registered trademark	&reg;

# Symbols

There are many mathematical, technical and currency symbols which are not present on a normal keyboard. We have to use HTML entity names to add such symbols to an HTML page.

If there no entity name exists, you can use an entity number, a decimal, or hexadecimal reference.

# Ex.

A	&#913;	&Alpha;	GREEK CAPITAL LETTER ALPHA
B	&#914;	&Beta;	GREEK CAPITAL LETTER BETA
Γ	&#915;	&Gamma;	GREEK CAPITAL LETTER GAMMA
Δ	&#916;	&Delta;	GREEK CAPITAL LETTER DELTA
E	&#917;	&Epsilon;	GREEK CAPITAL LETTER EPSILON
Z	&#918;	&Zeta;	GREEK CAPITAL LETTER ZETA
←	&#8592;	&larr;	LEFTWARDS ARROW
↑	&#8593;	&uarr;	UPWARDS ARROW
→	&#8594;	&rarr;	RIGHTWARDS ARROW
↓	&#8595;	&darr;	DOWNWARDS ARROW

# HTML Table

HTML table tag is used to display data in tabular form (row \* column). There can be many columns in a row.

HTML tables are used to manage the layout of the page e.g. header section, navigation bar, body content, footer section etc. But it is recommended to use div tag over table to manage the layout of the page .

GitHub link :[https://github.com/TopsCode/WEB\\_DESIGNING/blob/main/Module-1/HTML/05\\_table\\_tag.html](https://github.com/TopsCode/WEB_DESIGNING/blob/main/Module-1/HTML/05_table_tag.html)

# Example:

```
<table>  
  <tr>  
    <th>1 header</th>  
    <th>1 header</th>  
    <th>1 header</th>  
  </tr>  
  <tr>  
    <td>1data</td>  
    <td>1data</td>  
    <td>1data</td>  
  </tr>  
</table>
```

# Table Supportive tags

<table>  
<tr>  
<th>  
<td>  
<caption>  
<tbody>  
<thead>  
<tfooter>

Attributes:  
1. rowspan  
2. colspan

# HTML Lists

HTML Lists are used to specify lists of information. All lists may contain one or more list elements. There are three different types of HTML lists:

- Ordered List or Numbered List (ol)
- Unordered List or Bulleted List (ul)
- Description List or Definition List (dl)

Github link :[https://github.com/TopsCode/WEB\\_DESIGNING/blob/main/Module-1/HTML/04\\_list\\_tags.html](https://github.com/TopsCode/WEB_DESIGNING/blob/main/Module-1/HTML/04_list_tags.html)

# Ordered List or Numbered List

All the list items are marked with numbers by default.

```
<ol>
<li>Aries</li>
<li>Bingo</li>
<li>Leo</li>
<li>Oracle</li>
</ol>
```

Types:

1, I, i, A, a

GitHub link : [https://github.com/TopsCode/WEB\\_DESIGNING/blob/main/Module-1/HTML/04\\_list\\_tags.html](https://github.com/TopsCode/WEB_DESIGNING/blob/main/Module-1/HTML/04_list_tags.html)

# Unordered List or Bulleted List

All the list items are marked with bullets.

```
<ul>
<li>Aries</li>
<li>Bingo</li>
<li>Leo</li>
<li>Oracle</li>
</ul>
```

Types:  
disc, circle, square, none

Github link :

[https://github.com/TopsCode/WEB DESIGNING/blob/main/Module-1/HTML/04\\_list\\_tags.html](https://github.com/TopsCode/WEB DESIGNING/blob/main/Module-1/HTML/04_list_tags.html)

# Description List or Definition List

Entries are listed like a dictionary or encyclopedia.

```
<dl>
  <dt>HTML</dt>
  <dd>is a markup language</dd>
  <dt>Java</dt>
  <dd>is a programming language and
  platform</dd>
  <dt>JavaScript</dt>
  <dd>is a scripting language</dd>
  <dt>SQL</dt>
  <dd>is a query language</dd>
</dl>
```

**Github link :**

[https://github.com/TopsCode/WEB\\_DESIGNING/blob/main/Module-1/HTML/04\\_list\\_tags.html](https://github.com/TopsCode/WEB_DESIGNING/blob/main/Module-1/HTML/04_list_tags.html)

# File Paths

An HTML file path is used to describe the location of a file in a website folder. Used to link images, file, CSS file, JS file, video, etc.

Attributes:

1. src

Ex.

1. 
2. 
3. 
4. 

# Types of File Paths

## 1. Absolute File Paths:

Absolute file path specifies full URL address. Ex.

```

```

## 2. Relative File Paths

```

```

# Iframes

HTML Iframe is used to display a nested webpage (a webpage within a webpage). Used to embed Webpage or a YouTube video.

Syntax:<iframe src="URL"></iframe>

Attributes:

1. src
2. width
3. height
4. frameborder
5. allowfullscreen

Github link :

[https://github.com/TopsCode/WEB\\_DESIGNING/blob/main/Module-3/HTML5/03embed\\_iframe.html](https://github.com/TopsCode/WEB_DESIGNING/blob/main/Module-3/HTML5/03embed_iframe.html)

# frames

In HTML, frames were used to divide the browser window into multiple sections, each of which could load a different HTML document. The concept was popular for creating multi-pane layouts and for displaying multiple web pages within a single browser window. However, frames are now considered outdated and have been replaced by more modern approaches like CSS Grid, Flexbox, and iframes. The `<frameset>` and `<frame>` tags are now deprecated in HTML5.

# HTML Form

An HTML form is a section of a document which contains controls such as text fields, password fields, checkboxes, radio buttons, submit button, menus etc.

An HTML form facilitates the user to enter data that is to be sent to the server for processing such as name, email address, password, phone number, etc.

HTML forms are required if you want to collect some data from of the site visitor.

Ex. Online Shopping Website

# Syntax

```
<form action="server url"  
method="get/post">  
  
</form>
```

Github link : [https://github.com/TopsCode/WEB\\_DESIGNING/blob/main/Module-1/HTML/08form\\_tags.html](https://github.com/TopsCode/WEB_DESIGNING/blob/main/Module-1/HTML/08form_tags.html)

# Image Mapping

Image Mapping is a technique used in HTML to define specific clickable areas within an image. These areas are called image map hotspots, and each can have an associated hyperlink or action. Image maps are typically used when you want to make different parts of an image interactive, such as creating clickable regions on a map, a chart, or a diagram.

[WEB DESIGNING/Module-1/HTML/Image\\_map.html at main · TopsCode/WEB DESIGNING · GitHub](#)

# Image Mapping

Example Of Image Mapping:

```

<map name="mapname">
  <area shape="circle" coordes="400,300,50" href="link1.html" alt="area1"
        title="Area1">
  <area shape="react" coordes="34,30,50" href="link2.html" alt="area2"
        title="Area2">
  <area shape="poly" coordes="80,60,120" href="link3.html" alt="area3"
        title="Area3">
</map>
```

# HTML - Embed Multimedia

In HTML, you can embed multimedia content like audio, video, and images using specific tags. Here's how to do that:

## 1. Embedding Images :

```

```

## 2. Embedding Audio

```
<audio controls>
```

```
<source src="audio.mp3" type="audio/mp3">  
</audio>
```

# HTML - Embed Multimedia

## 3. Embedding Video:

```
<video width="640" height="360" controls>
<source src="video.mp4" type="video/mp4">
</video>
```

## 4. Embedding YouTube Videos :

```
<frame width="560" height="315"
src="https://www.youtube.com/embed/VIDEO\_ID" frameborder="0"
allow="accelerometer; autoplay; encrypted-media; gyroscope; pictureinpicture" allowfullscreen>
```

# HTML - Embed Multimedia

5. Embedding Other Media (like PDFs) :

```
<embed src="document.pdf" width="600" height="400"  
type="application/pdf">
```

# HTML - ASCII Codes

In HTML, ASCII codes (also known as character entities) represent special characters that cannot be typed directly or are reserved for HTML syntax. These characters are represented by a numeric or named entity in HTML

<p>The letter A in decimal ASCII is: &#65</p>

<p>The letter A in hexadecimal ASCII is: &#41</p>

# Lookup

To help with lookup, I can assist you in finding the ASCII code for a specific character or symbol, or even the character corresponding to a specific ASCII code. Please let me know what you'd like to look up! Here are a few examples of what you can ask for:

1. Lookup a character by its ASCII code:

⇒ "What character corresponds to ASCII code 65?"

2. Lookup the ASCII code for a character:

⇒ "What is the ASCII code for the letter A?"

Feel free to provide a character or code, and I'll look it up for you!

# Ref MIME Media Types

MIME stands for Multipurpose Internet Mail Extensions. It is a standard used to define the type of data (or content type) being sent over the internet, especially for email and web communication.

=> Structure of MIME Types A MIME type consists of two parts:

1. Type: The general category of the content (e.g., text, image, audio).
2. Subtype: The specific format within the type (e.g., plain, html, png, jpeg).

# ISO Codes HTML

ISO codes are standardized codes used to represent languages, countries, currencies, and other entities. These codes are essential for maintaining consistency and internationalization in web development. The most commonly used ISO codes in HTML are ISO 639 (language codes), ISO 3166 (country codes), and ISO 4217 (currency codes).

# HTML - Deprecated Tags

In HTML, deprecated tags are those that are no longer recommended for use in modern web development because they have been replaced by more efficient and accessible alternatives

Example=>

<font> , <center> ,<Marquee>.

# HTML Form Elements

- <form>
- <input>
- <textarea>
- <label>
- <fieldset>
- <legend>

Github link : [https://github.com/TopsCode/WEB\\_DESIGNING/blob/main/Module-1/HTML/08form\\_tags.html](https://github.com/TopsCode/WEB_DESIGNING/blob/main/Module-1/HTML/08form_tags.html)

# Input Types

```
<input type="text" name="username">
<input type="password" name="password">
<input type="email" name="email">
<input type="radio" name="password">
<input type="checkbox" id="cricket" name="cricket" value="cricket"/>
<input type="submit" value="submit">
<input type="button" value="button">
```

# Input Types

- text: Defines a one-line text input field
- password: Defines a one-line password input
- submit : field
- reset: Defines a submit button to submit the form to server
- radio: Defines a reset button to reset all values in
- checkbox: the form. Defines a radio button which allows select one option.
- button: Defines checkboxes which allow select multiple options form.
- file: Defines a simple push button, which can be programmed to perform a task on an event.
- image: Defines to select the file from device storage. Defines a graphical submit button.

# HTML5 Added Input Types

- color: Defines an input field with a specific color.
- date: Defines an input field for selection of date.
- datetime-local: Defines an input field for entering a date without time zone.
- email: Defines an input field for entering an email address.
- month: Defines a control with month and year, without time zone.
- number: Defines an input field to enter a number.
- url: Defines a field for entering URL
- week: Defines a field to enter the date with week-year, without time zone.
- search: Defines a single line text field for entering a search string.
- tel: Defines an input field for entering the telephone number.

# Form Attributes

1. **action:** The action attribute value defines the web page where information proceed. It can be .php, .jsp, .asp, etc. or any URL
  
2. **method:** Defines the HTTP method
  - **post:** We can use the post value of method attribute when we want to process the sensitive data as it does not display the submitted data in URL.
  - **get:** The get value of method attribute is default value while submitting the form. But this is not secure as it displays data in URL after submitting the form.
  
3. **target:** Where to open the response after submitting the form
  - **\_self:** The response will display in current page only.
  - **\_blank:** Load the response in a new page.

# Form Attributes

3. autocomplete: Enables an input field to complete automatically.
3. enctype: Defines the encoding type
  - application/x-www-form-urlencoded: Default
  - multipart/form-data: It does not encode any character. It is used when our form contains file-upload controls.
  - text/plain (HTML5): only space are encoded into + symbol
5. novalidate: Does not perform any type of validation and submit the form.

# Input Attributes

1. name
2. value
3. required
4. autofocus
5. placeholder
6. disabled
7. size
8. form

# Class

The HTML class attribute is used to specify a single or multiple class names for an HTML element.

The class name can be used by CSS and JavaScript to do some tasks for  
HTML elements.

You can use this class in CSS with a specific class, write a period (.) character, followed by the name of the class for selecting elements.

# ID

The id attribute is used to specify the unique ID for an element of the HTML document.

It allocates the unique identifier which is used by the CSS and the JavaScript for performing certain tasks.

# What is an Event?

When a browser reacts on user action, then it is called as an event.

# Form Event Attributes

<b>Attribute</b>	<b>Description</b>
onblur	Executed the script when form element loses the focus.
onchange	Executed the script when the value of the element is changed.
onfocus	Trigger an event when the element gets focused. Executed
oninput	the script when the user enters input to the element.
oninvalid	Executed the script when the element does not satisfy its predefined constraints.
onreset	Triggers the event when user reset the form element values.
onsubmit	Triggers the event when a form is submitted.

**Github link :** [https://github.com/TopsCode/WEB\\_DESIGNING/blob/main/Module-1/HTML/08form\\_tags.html](https://github.com/TopsCode/WEB_DESIGNING/blob/main/Module-1/HTML/08form_tags.html)

# Keyboard Event Attributes

<b>Attribute</b>	<b>Description</b>
onkeydown	Triggers the event when the user presses down a key on the keyboard.
onkeypress	Trigger the event when the user presses the key which displays some character.
onkeyup	Trigger the event when the user releases the currently pressed key.

# Mouse Event Attributes

Attribute	Description
onclick	Trigger the event when the mouse clicks on the element.
ondblclick	Trigger the event when mouse double-click occurs on the element.
onmousedown	Trigger the event when the mouse button is pressed on the element.
onmousemove	Trigger the event when the mouse pointer moves over the element.
onmouseout	Trigger the event when the mouse moves outside the element.
onmouseover	Trigger the event when the mouse moves onto the element.
onmouseup	Trigger the event when the mouse button is released.

# **Module - 3**

## **[ CSS and CSS3 ]**

# CSS

- ✓ Introduction to CSS
- ✓ How to insert CSS
- ✓ Inline
- ✓ Internal
- ✓ External
- ✓ Comments in CSS
- ✓ CSS Selectors
- ✓ CSS Pseudo Selector
- ✓ CSS Specificity
- ✓ CSS Text
- ✓ CSS Fonts
- ✓ CSS Background and Border Properties
- ✓ CSS Display and Position Properties
- ✓ CSS Buttons
- ✓ CSS Positioning

- ✓ CSS Colors
- ✓ CSS Important
- ✓ Line height, Padding and Margin
- ✓ CSS Filters
- ✓ CSS Images
- ✓ CSS Overflow
- ✓ CSS Position Property
- ✓ CSS Vertical Align, White Space and Word Wrap
- ✓ CSS Width and Height
- ✓ CSS Box-shadow and Text-shadow
- ✓ CSS Text Transform
- ✓ CSS Visibility
- ✓ CSS Icons
- ✓ Justify, Text Decoration and Text-Align

**Github link :** [https://github.com/TopsCode/WEB\\_DESIGNING/tree/main/Module-2/CSS-CSS3](https://github.com/TopsCode/WEB_DESIGNING/tree/main/Module-2/CSS-CSS3)

- ✓ CSS List
- ✓ CSS Selectors
- ✓ CSS Specificity
- ✓ Text Indent and Text  
Stroke
- ✓ CSS Calc ()
- ✓ CSS Print Properties
- ✓ CSS Columns
- ✓ CSS hyphens
- ✓ CSS Positions
- ✓ CSS Transform and Resize
- ✓ Transition Delay

# Advance CSS

- ✓ CSS Animation
- ✓ @keyframe
- ✓ CSS Pseudo elements
- ✓ CSS Gradient
- ✓ CSS z-index
- ✓ CSS Combinators
- ✓ Masking
- ✓ CSS Media Query
- ✓ 2D and 3D Transforms
- ✓ CSS Flex
- ✓ CSS Grid

[Github link : https://github.com/TopsCode/WEB\\_DESIGNING/tree/main/Module-2/CSS-CSS3](https://github.com/TopsCode/WEB_DESIGNING/tree/main/Module-2/CSS-CSS3)

# What is CSS

CSS stands for Cascading Style Sheets. It is a style sheet language which is used to describe the look and formatting of a document written in markup language. It provides an additional feature to HTML. It is generally used with HTML to change the style of web pages and user interfaces. It can also be used with any kind of XML documents including plain XML, SVG and XUL.

CSS is used along with HTML and JavaScript in most websites to create user interfaces for web applications and user interfaces for many mobile applications.

# What is CSS

- CSS stands for Cascading Style Sheet.
- CSS is used to design HTML tags.
- CSS is a widely used language on the web.
- HTML, CSS and JavaScript are used for web designing. It helps the web designers to apply style on HTML tags.

# Advantages of CSS

- Solves a big problem
- Saves a lot of time
- Provide more attributes

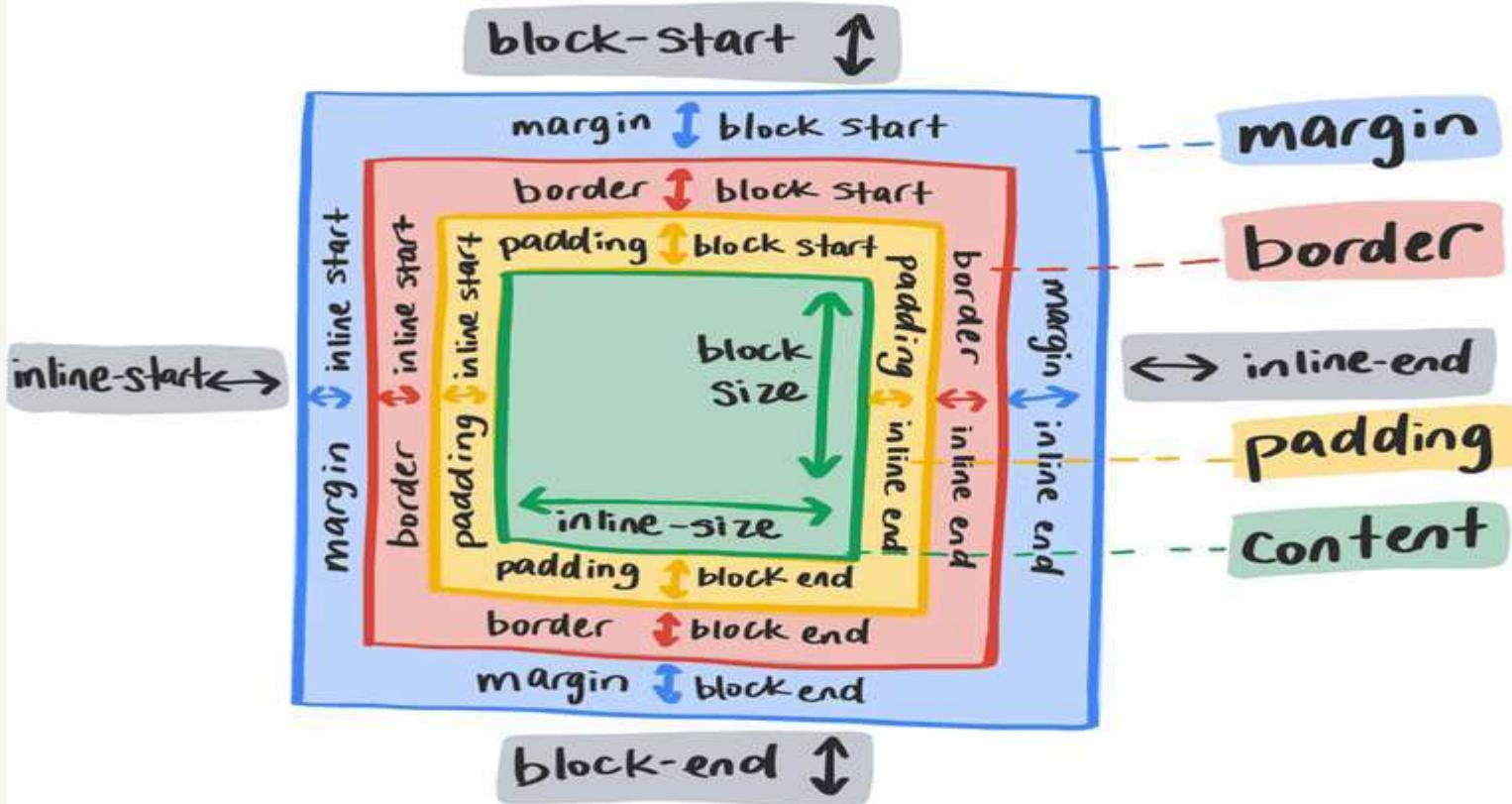
# @import in CSS (Importing Style Sheets)

The @import rule in CSS is used to import external stylesheets into a CSS file. It allows you to keep your styles modular by separating them into different files, making your CSS more organized and easier to maintain

```
@import url("style.css");
```



# Css Layout Model



# CSS Syntax

declaration



selector → p{color: black}

property

value

Selector{Property1: value1; Property2: value2; .....;}

# CSS Syntax

**Selector:** Selector indicates the HTML element you want to style. It could be any tag like `<h1>`, `<title>` etc.

**Declaration Block:** The declaration block can contain one or more declarations separated by a semicolon.

**Property:** A Property is a type of attribute of HTML element. It could be color, border etc.

**Value:** Values are assigned to CSS properties. In the above example, value "yellow" is assigned to color property.

# How to add CSS

```
<body>
<h1>Write Your First CSS Example</h1>
<p>This is Paragraph.</p>
</body>
```

# How to add CSS

```
<head>  
  
<style>h1{ color:white;  
background-color:red;  
padding:5px;  
}  
p{ color:blue;  
}  
</style>  
</head>
```

# CSS Selector

CSS selectors are used to 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.

Types of CSS Selector:

- CSS Universal Selector
- CSS Element Selector
- CSS Id Selector
- CSS Class Selector
- CSS Group Selector

[Github link : https://github.com/TopsCode/WEB\\_DESIGNING/blob/main/Module-2/CSS-CSS3/07\\_CSS%20Selectors.html](https://github.com/TopsCode/WEB_DESIGNING/blob/main/Module-2/CSS-CSS3/07_CSS%20Selectors.html)

# CSS Universal Selector

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

**Ex.**

```
<style>
* {
    color: green; font-
    size: 20px;
}
</style>
<h2>This is heading</h2>
<p>This style will be applied on every paragraph.</p>
<p id="para1">Me too!</p>
<p>And me!</p>
```

# CSS Element Selector

The element selector selects the HTML element by name.

**Ex.**

```
<style> p{  
    text-align: center;  
    color: blue;  
}  
</style>
```

<p>This style will be applied on every paragraph.</p>

**Github link :**

[https://github.com/TopsCode/WEB\\_DESIGNING/blob/main/Module-2/CSS-CSS3/07\\_CSS%20Selectors.html](https://github.com/TopsCode/WEB_DESIGNING/blob/main/Module-2/CSS-CSS3/07_CSS%20Selectors.html)

# CSS Id Selector

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.

**Ex.**

```
<style>
#para1 {
    text-align: center;
    color: blue;
}
</style>
<p id="para1">Hello Javatpoint.com</p>
```

**Github link :**

[https://github.com/TopsCode/WEB\\_DESIGNING/blob/main/Module-2/CSS-](https://github.com/TopsCode/WEB_DESIGNING/blob/main/Module-2/CSS-)

# CSS Class Selector for specific element

If you want to specify that only one specific HTML element should be affected then you should use the element name with class selector.

**Ex.**

```
<style>
p.center {
    text-align: center;
    color: blue;
}
</style>
<h1 class="center">This heading is not affected</h1>
<p class="center">This paragraph is blue and center-aligned.</p>
```

# CSS 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.

**Ex.**

```
<style> h1, h2, p  
{  
    text-align: center; color:  
    blue;  
}  
</style>
```

GitHub link :

[https://github.com/TopsCode/WEB\\_DESIGNING/blob/main/Module-2/CSS-CSS3/07\\_CSS%20Selectors.html](https://github.com/TopsCode/WEB_DESIGNING/blob/main/Module-2/CSS-CSS3/07_CSS%20Selectors.html)

# CSS Combinators

CSS Combinators clarifies the relationship between two selectors, whereas the selectors in CSS are used to select the content for styling.

There can be more than one simple selector in a CSS selector, and between these selectors, we can include a combinator. Combinators combine the selectors to provide them a useful relationship and the position of content in the document.

There are four types of combinators in CSS that are listed as follows:

1. General sibling selector (~)
2. Adjacent sibling selector (+)
3. Child selector (>)
4. Descendant selector (space)

# General Sibling Selector

uses the tilde (~) sign as the separator between the elements. It selects the elements that follow the elements of first selector, and both of them are the children of the same parent. It can be used for selecting the group of elements that share the common parent element.

It is useful when we have to select the siblings of an element even if they are not adjacent directly.

## Syntax:

```
element ~ element {  
    /*style properties*/  
}
```

# Adjacent Sibling Selector( + )

It uses the plus (+) sign as the separator between the elements. It matches the second element only when the element immediately follows the first element, and both of them are the children of the same parent. This sibling selector selects the adjacent element, or we can say that the element which is next to the specified tag.

It only selects the element which is just next to the specified first element.

## Syntax:

```
element + element {  
    /*style properties*/  
}
```

# Child Selector (>)

It uses the greater than (>) sign as the separator between the elements. It selects the direct descendant of the parent. This combinator only matches the elements that are the immediate child in the document tree. It is stricter as compared to the descendant selector because it selects the second selector only when the first selector is its parent.

The parent element must always be placed at the left of the ">". If we remove the greater than (>) symbol that designates this as a child combinator, then it will become the descendant selector.

## Syntax:

```
element > element {  
    /*style properties*/  
}
```

# Descendant Selector (space)

It combines two selectors in which the first selector represents an ancestor (parent, parent's parent, etc.), and the second selector represents descendants. The elements matched by the second selector are selected if they have an ancestor element that matches the first selector.

## Syntax:

```
element element {  
    /*style properties*/  
}
```

# CSS pseudo-classes

A pseudo-class can be defined as a keyword which is combined to a selector that defines the special state of the selected elements. It is added to the selector for adding an effect to the existing elements based on their states. For example, The ":hover" is used for adding special effects to an element when the user moves the cursor over the element.

## Syntax:

selector: pseudo-class

```
{ property: value;  
    Github link :  
}
```

[https://github.com/TopsCode/WEB\\_DESIGNING/blob/main/Module-2/CSS-](https://github.com/TopsCode/WEB_DESIGNING/blob/main/Module-2/CSS-)

# :hover

This pseudo-class adds a special style to an element when the user moves the cursor over it.

Ex.

```
h1:hover  
{  
    color:red;  
}
```

Github link :

[https://github.com/TopsCode/WEB\\_DESIGNING/blob/main/Module-2/CSS-CSS3/08 %20Pseudo%20Classes%20and%20Elements.html](https://github.com/TopsCode/WEB_DESIGNING/blob/main/Module-2/CSS-CSS3/08 %20Pseudo%20Classes%20and%20Elements.html)

# :active

It applies when the elements are clicked or activated. It selects the activated element.

**Ex.**

```
a:active{  
    color: yellow;  
}
```

**GitHub link :**

[https://github.com/TopsCode/WEB\\_DESIGNING/blob/main/Module-2/CSS-CSS3/08 %20Pseudo%20Classes%20and%20Elements.html](https://github.com/TopsCode/WEB_DESIGNING/blob/main/Module-2/CSS-CSS3/08 %20Pseudo%20Classes%20and%20Elements.html)

# :visited

It selects the visited links and adds special styles to them.

**Ex.**

```
a:visited{  
    color: red;  
}
```

**Github link :**

<https://github.com/TopsCode/WEB DESIGNING/blob/main/Module-2/CSS-CSS3/08 %20Pseudo%20Classes%20and%20Elements.html>

# :lang

It is helpful in documents that require multiple languages.

**Ex.**

```
p:lang(fr)
{
    font-family:Verdana;
    color:blue;

}
```

<p lang="fr">With :lang pseudo class with the value  
fr</p>

**Github link :**

[https://github.com/TopsCode/WEB\\_DESIGNING/blob/main/Module-2/CSS-CSS3/08\\_%20Pseudo%20Classes%20and%20Elements.html](https://github.com/TopsCode/WEB_DESIGNING/blob/main/Module-2/CSS-CSS3/08_%20Pseudo%20Classes%20and%20Elements.html)

# :focus

It selects the elements that are currently focused on by the user.

**Ex.**

```
input:focus{  
    border:5px solid lightblue;  
    box-shadow:10px 10px 10px  
    black; color: blue;  
    width:300px;  
}
```

**Github link :**

[https://github.com/TopsCode/WEB DESIGNING/blob/main/Module-2/CSS-CSS3/08\\_%20Pseudo%20Classes%20and%20Elements.html](https://github.com/TopsCode/WEB DESIGNING/blob/main/Module-2/CSS-CSS3/08_%20Pseudo%20Classes%20and%20Elements.html)

# :first-child

It matches a particular element, which is the first child of another element and adds a special effect to the corresponding element.

**Ex.**

```
h1:first-child {  
    text-indent:  
    200px; color:blue;  
}
```

**Github link :**

[https://github.com/TopsCode/WEB\\_DESIGNING/blob/main/Module-2/CSS-CSS3/08 %20Pseudo%20Classes%20and%20Elements.html](https://github.com/TopsCode/WEB_DESIGNING/blob/main/Module-2/CSS-CSS3/08 %20Pseudo%20Classes%20and%20Elements.html)

# :nth-child(n)

This selector is used for matching the elements based on their position regardless of the type of its parent. The n can either be a keyword, formula, or a number. It is used to match the elements based on their position within a group of siblings. It matches each element, which is the nth-child.

**Ex.**

```
p:nth-child(2n+1) {  
    background:  
        yellow; color: black;  
    font-size:30px;  
}
```

**Github link :**

[https://github.com/TopsCode/WEB\\_DESIGNING/blob/main/Module-2/CSS-](https://github.com/TopsCode/WEB_DESIGNING/blob/main/Module-2/CSS-)

# CSS Pseudo-elements

A pseudo-class can be defined as a keyword which is combined to a selector that defines the special state of the selected elements. Unlike the pseudo-classes, the pseudo-elements are used to style the specific part of an element, whereas the pseudo-classes are used to style the element.

## Syntax:

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

We have used the double colon notation (::pseudo-element) in the syntax. In CSS3, the double colon replaced the single colon notation for pseudo-elements.

[Github](#)

link :[https://github.com/TopsCode/WEB\\_DESIGNING/blob/main/Modul](https://github.com/TopsCode/WEB_DESIGNING/blob/main/Modul)

# Psuedo Elements

Psuedo Element	Description
::first-letter (:first-letter)	It selects the first letter of the text.
::first-line (:first-line)	It styles the first line of the text.
::before (:before)	It is used to add something before the element's content.
::after (:after)	It is used to add something after the element's content.
::selection	It is used to select the area of an element that is selected by the user.

# ::first-letter

it affects the first letter of the text. It can be applied only to block-level elements. Instead of supporting all CSS properties, it supports some of the CSS properties that are given below.

- Color properties (such as color)
- Font properties (such as font-style, font-family, font-size, font-color, and many more).
- Margin properties (such as margin-top, margin-right, margin-bottom, and margin-left).
- Border properties (like border-top, border-right, border-bottom, border-left, border-color, border-width, and many more).
- Padding properties (such as padding-top, padding-right, padding-bottom, and padding-left).
- Background properties (such as background-color, background-repeat, background-image, and background-position).
- Text related properties (such as text-shadow, text-transform, text-decoration, etc.).
- Other properties are vertical-align (only when the float is 'none') word-spacing, line-height, line-spacing, etc.

Github

link :<https://github.com/TopsCode/WEB DESIGNING/blob/main>

# ::first-line

It is similar to the ::first-letter pseudo-element, but it affects the entire line. It adds the special effects to the first line of the text. It supports the following CSS properties:

- Color properties (such as color)
- Font properties (such as font-style, font-family, font-size, font-color, and many more).
- Background properties (such as background-color, background-repeat, background-image, and background-position).
- Other properties are word-spacing, letter-spacing, line-height, vertical-align, text-transform, text-decoration.

Github link :[https://github.com/TopsCode/WEB DESIGNING/blob/main/Module-2/CSS-CSS3/27pseudo\\_element.html](https://github.com/TopsCode/WEB DESIGNING/blob/main/Module-2/CSS-CSS3/27pseudo_element.html)

# ::before

It allows us to add something before the element's content. It is used to add something before the specific part of an element. Generally, it is used with the content property.

**Ex:**

```
h1::before {  
    content: "Hello World.";  
}
```

**Github link :**[https://github.com/TopsCode/WEB DESIGNING/blob/main/Module-2/CSS-CSS3/27pseudo\\_element.html](https://github.com/TopsCode/WEB DESIGNING/blob/main/Module-2/CSS-CSS3/27pseudo_element.html)

# ::after

It works similar to ::before pseudo-element, but it inserts the content after the content of the element. It is used to add something after the specific part of an element. Generally, it is used with the content property.

**Ex:**

```
h1::after {  
    content: "Welcome to the LearnVern";  
}
```

**Github link :**[https://github.com/TopsCode/WEB\\_DESIGNING/blob/main/Module-2/CSS-CSS3/27pseudo\\_element.html](https://github.com/TopsCode/WEB_DESIGNING/blob/main/Module-2/CSS-CSS3/27pseudo_element.html)

# ::selection

It is used to style the part of an element that is selected by the user. We can use the following CSS properties with it:

- color.
- background-color.
- Other properties include cursor, outline, etc.

**Ex.**

```
h1::selection  
{ color: red;  
}
```

**Github**

link :[https://github.com/TopsCode/WEB DESIGNING/blob/main/Module-2/CSS-CSS3/27pseudo\\_element.html](https://github.com/TopsCode/WEB DESIGNING/blob/main/Module-2/CSS-CSS3/27pseudo_element.html)

# How to add CSS

There are three ways to insert CSS in HTML documents.

1. Inline CSS
2. Internal CSS
3. External CSS

GitHub link [https://github.com/TopsCode/WEB\\_DESIGNING/blob/main/Module-2/CSS-CSS3/01CSS.html](https://github.com/TopsCode/WEB_DESIGNING/blob/main/Module-2/CSS-CSS3/01CSS.html)

# Inline CSS

We can apply CSS in a single element by inline CSS technique.

The inline CSS is also a method to insert style sheets in HTML document.

If you want to use inline CSS, you should use the style attribute to the relevant tag.

**Github link** [https://github.com/TopsCode/WEB\\_DESIGNING/blob/main/Module-2/CSS-CSS3/02inline\\_css.html](https://github.com/TopsCode/WEB_DESIGNING/blob/main/Module-2/CSS-CSS3/02inline_css.html)

# Inline CSS

We can apply CSS in a single element by inline CSS technique.

The inline CSS is also a method to insert style sheets in HTML document.

If you want to use inline CSS, you should use the style attribute to the relevant tag.

**Github link** [https://github.com/TopsCode/WEB\\_DESIGNING/blob/main/Module-2/CSS-CSS3/02inline\\_css.html](https://github.com/TopsCode/WEB_DESIGNING/blob/main/Module-2/CSS-CSS3/02inline_css.html)

# Syntax

```
<htmltag style="cssproperty1:value; cssproperty2:value;"> </htmltag>
```

**Ex.**

```
<h2 style="color:red;margin-left:40px;">Inline CSS is applied on this heading.</h2>
<p>This paragraph is not affected.</p>
```

# Disadvantages of Inline CSS

- You cannot use quotations within inline CSS. If you use quotations the browser will interpret this as an end of your style value.
- These styles cannot be reused anywhere else.
- These styles are tough to be edited because they are not stored at a single place.
- It is not possible to style pseudo-codes and pseudo-classes with inline CSS.
- Inline CSS does not provide browser cache advantages.

# Internal CSS

The internal style sheet is used to add a unique style for a single document.  
It is defined in <head> section of the HTML page inside the <style> tag.

Ex.

```
<style>  
  
body {  
  
    background-color: linen;  
  
}  
  
h1 {  
    color: red; margin-left: 80px;  
  
}  
  
</style>
```

**Github link**[https://github.com/TopsCode/WEB DESIGNING/blob/main/Module-2/CSS-CSS3/03internal\\_css.html](https://github.com/TopsCode/WEB DESIGNING/blob/main/Module-2/CSS-CSS3/03internal_css.html)

# External CSS

The external style sheet is generally used when you want to make changes on multiple pages. It is ideal for this condition because it facilitates you to change the look of the entire web site by changing just one file.

It uses the `<link>` tag on every pages and the `<link>` tag should be put inside the head section.

**Ex.**

```
<head>
<link rel="stylesheet" type="text/css" href="mystyle.css">
</head>
```

The external style sheet may be written in any text editor but must be saved with a .css extension. This file should not contain HTML elements.

**Githublink**

[https://github.com/TopsCode/WEB\\_DESIGNING/blob/main/Module-](https://github.com/TopsCode/WEB_DESIGNING/blob/main/Module-)

# CSS comments

CSS comments are generally written to explain your code. It is very helpful for the users who reads your code so that they can easily understand the code.

Comments are ignored by browsers.

# Comments

Comments are single or multiple lines statement  
and written within /\*.....\*/.

**Ex.**

```
<style>  
p {  
    color: blue;  
    /* This is a single-line comment  
     */ text-align: center;  
}
```

/\* This is  
a multi-line  
comment \*/  
</style>

# Specificity

Specificity is the way which helps the browsers to decide which property value is most relevant for the element. It determines which style declaration is applied to an element.

- The CSS specificity is important only when various selectors are affecting the same element. In this case, the browser needs a way to identify the style to be applied to the matching element, and CSS specificity is the way of doing it.
- When two or more selectors have equal specificity value, then the latest one considers.
- Universal selectors (\*) and the inherited values have lower specificity, i.e., 0 specificity.
- The style property has a greater specificity value compare to the selectors (except the !important in the stylesheet selector).
- The !important alter the selector specificity. When two selectors have equal specificity, then the selector having !important

# hierarchy

**Inline styles:** It is directly attached to the element which is to be styled.

For example: <p style="color: red;">. It has the highest priority.

**IDs:** It is a unique identifier for the elements of a page that has the second-highest priority. For example: #para.

**Classes, attributes, and pseudo-classes:** It includes classes, attributes, and pseudo-classes (like :focus, :hover, etc.).

**Elements and pseudo-elements:** It includes the name of elements (div, h1) and pseudo-elements (like :after and :before). They have the lowest priority.

# Rules

- The specificity of ID selectors is higher than attribute selectors
- In equal specificity, the latest rule will count
- The specificity of class selector is greater than the element selectors

# CSS Text Transform

This CSS property allows us to change the case of the text. It is used to control the text capitalization. This CSS property can be used to make the appearance of text in all-lowercase or all-uppercase or can convert the first character of each word to uppercase.

## Syntax:

`text-transform: capitalize | uppercase | lowercase | none | initial | inherit;`

# 1. capitalize

It transforms the first character of each word to uppercase. It will not capitalize the first letter after the number. It only affects the first letters of the words instead of changing the rest of the letters in the word.

If we apply the capitalize property on a word that already has capital letters, then the letters of that word will not switch to lowercase.

## Syntax:

`text-transform: capitalize;`

## 2. uppercase

As its name implies, it transforms all characters of the word into uppercase.

### Syntax:

```
text-transform: uppercase;
```

# 3. lowercase

It transforms all characters of the word into lowercase.

**Syntax:**

text-transform: lowercase;

## 4. none

It is the default value that has no capitalization. It renders the text as it is.

**Syntax:**

```
text-transform: none;
```

# CSS text-overflow

This property specifies the representation of overflowed text, which is not visible to the user. It signals the user about the content that is not visible. This property helps us to decide whether the text should be clipped, show some dots (ellipsis), or display a custom string.

This property does not work on its own. We have to use white-space: nowrap; and overflow: hidden; with this property

## Syntax:

text-overflow: clip | ellipsis | string

Githublink

[https://github.com/TopsCode/WEB\\_DESIGNING/blob/main/Module-](https://github.com/TopsCode/WEB_DESIGNING/blob/main/Module-)

# 1. clip

**clip:** It is the default value that clips the overflowed text. It truncates the text at the limit of the content area, so that it can truncate the text in the middle of the character.

**Syntax:**

text-overflow: clip;

Githublink : [https://github.com/TopsCode/WEB\\_DESIGNING/blob/main/Module-2/CSS-CSS3/34\\_Clip.html](https://github.com/TopsCode/WEB_DESIGNING/blob/main/Module-2/CSS-CSS3/34_Clip.html)

## 2. ellipsis

**ellipsis:** This value displays an ellipsis (?) or three dots to show the clipped text. It is displayed within the area, decreasing the amount of text.

**Syntax:**

text-overflow: ellipsis;

# CSS text-orientation

This CSS property specifies the orientation of characters in the line of content. It only applies to the vertical mode of content. This property does not affect elements with horizontal writing mode.

It helps us to control the display of languages that use a vertical script. This property has five values: **mixed**, **sideways**, **upright**, **sideways-right**, and **use-glyph-orientation**. Its default value is mixed. In latest browsers only mixed and upright are the working property values.

This property depends upon the writing-mode property. It works only when the writing-mode is not set to horizontal-tb.

## Syntax:

text-orientation: mixed | upright

[WEB DESIGNING/Module-2/CSS-CSS3/text\\_orientation.html at main · TopsCode/WEB DESIGNING · GitHub](#)

# Ex.

```
#lr {  
writing-mode: vertical-lr;  
text-orientation: mixed;  
}
```

```
#rl {  
writing-mode: vertical-rl;  
text-orientation:  
upright;  
}
```

# Text Indent

This CSS property sets the indentation of the first line in a block of text. It specifies the amount of horizontal space that puts before the lines of text.

It allows the negative values, and if any negative value is defined, then the indentation of the first line will be towards left.

## Syntax:

text-indent: length

# Values

**length:** This value sets the fix indentation with the units cm, pt, em, px, and others. Its default value is 0. It allows negative values. The indentation of the first line is on the left when its value is negative.

**percentage:** It specifies the amount in space in the percentage of the width of the containing block.

# Text Stroke

This CSS property adds a stroke to the text and also provides decoration options for them. It defines the color and width of strokes for text characters.

This CSS property is the shorthand of the following two properties:

**text-stroke-width:** It describes the thickness of the stroke effect and takes the unit value.

**text-stroke-color:** It takes the value of a color.

**-webkit-text-fill-color:** It fills color inside the text.

The text-stroke can only be used with the -webkit- prefix.

[WEB DESIGNING/Module-2/CSS-CSS3/Text Stroke.html at main · TopsCode/WEB DESIGNING · GitHub](#)

# CSS Fonts

CSS Font property is used to control the look of texts. By the use of CSS font property you can change the text size, color, style and more.

- CSS Font color
- CSS Font family
- CSS Font size
- CSS Font style
- CSS Font variant
- CSS Font weight

# CSS Font Color

It is used to change the color of the text. There are three different formats to define a color:

By a color name    By hexadecimal value    By RGB

**Ex.**

```
h1 { color: red; }  
h2 { color: #9000A1; }  
p { color:rgb(0, 220, 98); }
```

Githublink :[https://github.com/TopsCode/WEB\\_DESIGNING/blob/main/Module-2/CSS-CSS3/08color.html](https://github.com/TopsCode/WEB_DESIGNING/blob/main/Module-2/CSS-CSS3/08color.html)

# CSS Font Family

Generic family: It includes Serif, Sans-serif, and Monospace.

Font family: It specifies the font family name like Arial, New Times Roman etc.

Serif: Serif fonts include small lines at the end of characters. Example of serif: Times new roman, Georgia etc.

Sans-serif: A sans-serif font doesn't include the small lines at the end of characters. Example of Sans-serif: Arial, Verdana etc.

F F

Sans-serif

Serif

# CSS Font Family

Ex.

```
h1 { font-family: sans-serif;  
} h2 { font-family: serif; }  
p { font-family: monospace;  
}
```

Githublink : [https://github.com/TopsCode/WEB\\_DESIGNING/blob/main/Module-2/CSS-CSS3/10\\_Font%20Styling.html](https://github.com/TopsCode/WEB_DESIGNING/blob/main/Module-2/CSS-CSS3/10_Font%20Styling.html)

# Font Family Through Google Font:

Step 1: Choose a Font from Google Fonts

1. Go to Google Fonts.
2. Browse or search for the font you like.
3. Once you find a font you like, click on it. You'll see options for different weights and styles (e.g., Regular, Bold).
4. Click the "+ Select this style" button to add it to your collection.

# Font Family Through Google Font:

## Step 2: Embed the Font in Your HTML

Once you've selected the font, you will be provided with a `<link>` tag to embed the font into your HTML file. You need to include this `<link>` in the `<head>` section of your HTML document.

# import fontface rule

The @font-face rule in CSS allows you to define custom fonts that can be loaded from a local or external source and applied to elements on a webpage. This rule is especially useful when you want to use a custom font that isn't available on Google Fonts or other online font services.

# CSS Font Size

CSS font size property is used to change the size of the font.

font-size:xx-small;  
font-size:x-small;  
font-size:small;  
font-size:medium;  
font-size:large;  
font-size:x-large;

font-size:xx-large; font-size:smaller;  
font-size:larger;  
font-size:200%;  
font-size:20px;

Githublink :

[https://github.com/TopsCode/WEB\\_DESIGNING/blob/main/Module-2/CSS-CSS3/10\\_Font%20Styling.html](https://github.com/TopsCode/WEB_DESIGNING/blob/main/Module-2/CSS-CSS3/10_Font%20Styling.html)

# CSS Font Size

## 1. Absolute-size:

It is used to set the text to a definite size. Using absolute-size, it is not possible to change the size of the text in all browsers. It is advantageous when we know the physical size of the output.

Ex. Font-size with em

## 2. Relative-size:

It is used to set the size of the text relative to its neighboring elements.

With relative-size, it is possible to change the size of the text in browsers.

Githublink :

[https://github.com/TopsCode/WEB\\_DESIGNING/blob/main/Module-2/CSS-](https://github.com/TopsCode/WEB_DESIGNING/blob/main/Module-2/CSS-)

# CSS Font Size

## 3. Responsive font size:

We can set the size of the text by using a vw unit, which stands for the 'viewport width'. The viewport is the size of the browser window.

**Ex.** 1vw = 1% of viewport width.

## 4. Font-size with the length property:

It is used to set the size of the font in length. The length can be in cm, px, pt, etc.

**Ex.** font-size: 5cm;

Githublink :

[https://github.com/TopsCode/WEB\\_DESIGNING/blob/main/Module-2/CSS-CSS3/10\\_Font%20Styling.html](https://github.com/TopsCode/WEB_DESIGNING/blob/main/Module-2/CSS-CSS3/10_Font%20Styling.html)

# CSS Font Style

CSS Font style property defines what type of font you want to display. It may be italic, oblique, or normal.

**Ex.**

```
h2 { font-style: italic; }  
h3 { font-style: oblique;  
}  
h4 { font-style: normal; }
```

Githublink :

[https://github.com/TopsCode/WEB\\_DESIGNING/blob/main/Module-2/CSS-CSS3/10\\_Font%20Styling.html](https://github.com/TopsCode/WEB_DESIGNING/blob/main/Module-2/CSS-CSS3/10_Font%20Styling.html)

# CSS Font Variant

CSS font variant property specifies how to set font variant of an element. It may be normal and small-caps.

**Ex.**

```
p { font-variant: small-caps;  
} h3 { font-variant: normal; }
```

# CSS Font Weight

CSS font weight property defines the weight of the font and specify that how bold a font is. The possible values of font weight may be normal, bold, bolder, lighter or number (100, 200..... upto 900).

**Ex.**

```
font-weight:bold;  
font-  
weight:bolder;  
font-weight:lighter;  
font-weight:100;
```

```
font-weight:200;  
font-weight:300;  
font-weight:900;
```

Githublink :

[https://github.com/TopsCode/WEB DESIGNING/blob/main/Module-2/CSS-CSS3/10\\_Font%20Styling.html](https://github.com/TopsCode/WEB DESIGNING/blob/main/Module-2/CSS-CSS3/10_Font%20Styling.html)

# CSS Font Stretch

The font-stretch property in CSS allows us to select a normal, expanded, or condensed face from the font's family. This property sets the text wider or narrower compare to the default width of the font. It will not work on any font but only works on the font-family that has a width-variant face.

**Ex.**

font-stretch: normal / semi-condensed / condensed / extra-condensed / ultra-condensed / semi-expanded/ expanded / extra-expanded / ultra-expanded

# CSS Element Size

In CSS, controlling the size of elements is a fundamental aspect of layout and design. You can define the size of elements using various properties, including width, height, max-width, max-height, minwidth, min-height, and box-sizing;

# **caption-side**

The `caption-side` property in CSS is used to control the position of the `<caption>` element in a `<table>`. The `<caption>` tag is used to provide a title or description for the table, and by default, it appears at the top of the table. The `caption-side` property allows you to move it to the bottom if needed.

# counter-increment

The counter-increment property in CSS is used to increment (or increase) a counter value for an element. This property is often used in conjunction with counter-reset and content to create custom counters for things like numbered lists, sections, or any element that needs automatic counting.

counter-increment: counter-name [increment-value];

# counter-reset

The counter-reset property in CSS is used to initialize or reset a counter to a specific value. It is commonly used in conjunction with counter-increment to create custom counters for elements such as lists, sections, or any other content that requires numbering or automatic counting.

counter-reset: counter-name [initial-value];

# empty-cells

The empty-cells property in CSS is used to control the visibility of borders in empty cells within a table. Specifically, it determines whether or not the borders of empty table cells are displayed. This property is useful for styling tables, especially when you want to manage the appearance of empty cells.

# letter-spacing

The letter-spacing property in CSS is used to control the space between characters (letters) in a text element. This property allows you to adjust the spacing between characters to improve the appearance, readability, or to achieve a particular design effect.

[WEB DESIGNING/Module-2/CSS-CSS3/Latter spacing.html at main · TopsCode/WEB DESIGNING · GitHub](#)

# CSS Background

CSS background property is used to define the background effects on element. There are 5 CSS background properties that affects the HTML elements:

1. background-color
2. background-image
3. background-repeat
4. background-attachment
5. background-position

Githublink : [https://github.com/TopsCode/WEB\\_DESIGNING/blob/main/Module-2/CSS-CSS3/05background.html](https://github.com/TopsCode/WEB_DESIGNING/blob/main/Module-2/CSS-CSS3/05background.html)

# 1. CSS background-color

The background-color property is used to specify the background color of the element.

**Ex.**

```
<style>  
p{  
    background-color: green;  
}  
</style>
```

## 2. CSS background-image

The background-image property is used to set an image as a background of an element. By default the image covers the entire element

**Ex.**

```
<style>
body {
background-image:
url("image.jpg"); margin-left:100px;
}
</style>
```

Githublink

[https://github.com/TopsCode/WEB\\_DESIGNING/blob/main/Module-2/CSS-CSS3/07background-image.html](https://github.com/TopsCode/WEB_DESIGNING/blob/main/Module-2/CSS-CSS3/07background-image.html)

# 3. CSS background-repeat

By default, the background-image property repeats the background image horizontally and vertically. Some images are repeated only horizontally or vertically.

**Ex.**

```
<style> body {  
    background-image: url("gradient_bg.png");  
    background-repeat: repeat-x;      or      background-repeat: repeat-y;  
}  
</style>
```

Githublink

[https://github.com/TopsCode/WEB\\_DESIGNING/blob/main/Module-2/CSS-CSS3/07background-image.html](https://github.com/TopsCode/WEB_DESIGNING/blob/main/Module-2/CSS-CSS3/07background-image.html)

# 4. CSS background-attachment

The background-attachment property is used to specify if the background image is fixed or scroll with the rest of the page in browser window. If you set fixed the background image then the image will not move during scrolling in the browser.

**Ex.**

```
<style>  
background: white  
url('image.jpg'); background-  
repeat: no-repeat;  
  
background-attachment: fixed; WEB DESIGNING/Module-2/CSS-CSS3/Background attachment.html at main · TopsCode/WEB DESIGNING · GitHub  
</style>
```

# 5. CSS background-position

The background-position property is used to define the initial position of the background image. By default, the background image is placed on the top-left of the webpage.

You can set the positions as:center,top,bottom,left,right

**Ex.**

```
background: white url('good-morning.jpg');  
background-repeat: no-repeat;  
background-attachment: fixed;  
background-position: center;
```

Githublink

[https://github.com/TopsCode/WEB\\_DESIGNING/blob/main/Module-2/CSS-CSS3/25%20Position.html](https://github.com/TopsCode/WEB_DESIGNING/blob/main/Module-2/CSS-CSS3/25%20Position.html)

# CSS Border

The CSS border is a shorthand property used to set the border on an element.

The CSS border properties are used to specify the style, color and size of the border of an element. The CSS border properties are given below

- border-style
- border-color
- border-width
- border-radius

Githublink

<https://github.com/TopsCode/WEB DESIGNING/blob/main/Module-2/CSS-CSS3/12border.html>

# 1. CSS Border Style

The Border style property is used to specify the border type which you want to display on the web page.

**Ex.**

border-style: none;	border-style: groove; border-
border-style: dotted;	style: ridge; border- style:
border-style: dashed;	inset; border- style: outset;
border- style: solid;	border-style: hidden
border- style: double;	

**Githublink** [https://github.com/TopsCode/WEB\\_DESIGNING/blob/main/Module-2/CSS-CSS3/12border.html](https://github.com/TopsCode/WEB_DESIGNING/blob/main/Module-2/CSS-CSS3/12border.html)

## 2. CSS Border Width

The border-width property is used to set the border's width. It is set in pixels. You can also use the one of the three pre-defined values, thin, medium or thick to set the width of the border.

**Ex.**

```
<style>  
p.one {  
    border-style: solid;  
    border-width: 5px; / border-width: medium; / border-width: 1px;  
}  
</style>
```

# 3. CSS Border Color

There are three methods to set the color of the border.

Name: It specifies the color name. For example: "red".

RGB: It specifies the RGB value of the color. For example: "rgb(255,0,0)". Hex: It specifies the hex value of the color. For example: "#ff0000".

**Ex.**

```
p.one {  
    border-style: solid; border-  
    color: red;  
}  
  
p.two {
```

# 4. CSS Border Radius

This CSS property sets the rounded borders and provides the rounded corners around an element, tags, or div. It defines the radius of the corners of an element. The values of this property can be defined in percentage or length units. It is shorthand for border-top-left-radius, border-top-right-radius, border-bottom-right-radius and border-bottom-left-radius.

## Ex.

Variations are as below:

Single Value - border-radius: 30px; Two

Value - border-radius: 20% 10%

# 5. CSS Border Collapse

This CSS property is used to set the border of the table cells and specifies whether the table cells share the separate or common border.

This property has two main values that are separate and collapse. When it is set to the value separate, the distance between the cells can be defined using the border-spacing property. When the border-collapse is set to the value collapse, then the inset value of border-style property behaves like groove, and the outset value behaves like ridge.

**Ex.**

```
table {  
border-collapse: separate;           / border-collapse: collapse;  
}  
WEB DESIGNING/Module-2/CSS-CSS3/Border collapse.html at main . TopsCode/WEB DESIGNING - GitHub
```

# 6. CSS Border Spacing

This CSS property is used to set the distance between the borders of the adjacent cells in the table. It applies only when the border-collapse property is set to separate. There will not be any space between the borders if the border-collapse is set to collapse.

When only one value is specified, then it sets both horizontal and vertical spacing. When we use the two-value syntax, then the first one is used to set the horizontal spacing (i.e., the space between the adjacent columns), and the second value sets the vertical spacing (i.e., the space between the adjacent rows).

**Ex.**

```
table {  
border-collapse: separate; border-spacing :
```

# 7. CSS Outline

CSS outline is just like CSS border property. It facilitates you to draw an extra border around an element to get visual attention.

**Ex.**

```
.Outline {  
    background-color: #eee;  
    border: 3px solid  
    Lightgreen; padding: 5px  
    10px;  
    outline-width: 3px;  
    outline-style: solid;  
    outline-color: red;
```

Outline offset: The outline offset is used to create a distance between outline and border.

# 7. CSS Outline...

Outline offset: The outline offset is used to create a distance between outline and border.

.Outline-

```
Offset{ background-  
color: #eee; outline:  
3px solid  
red; outline-offset: 6px;  
border: 3px solid  
Lightgreen; padding: 5px  
10px; }
```

[WEB DESIGNING/Module-2/CSS-CSS3/CSS\\_outline.html at  
main · TopsCode/WEB DESIGNING · GitHub](#)

# 8. CSS Border Image

This CSS property defines an image to be used as the element's border. It draws an image outside the element and replaces the element's border with the corresponding image. It is an interesting task to replace the border of an element with the image.

It is the shorthand property for border-image-source, border-image-slice, border- image-width, border-image-outset, and border-image-repeat.

**Ex.**

```
border-image: url('border.png') 60 / 20px 20px round;
```

**Githublink**

<https://github.com/TopsCode/WEB DESIGNING/blob/main/Module-2/CSS-CSS3/55 %20Border%20Image.html>

# 8. CSS Border Image...

Value	Description
border-image-source	It specifies the source of the border-image.
border-image-slice	It is used to divide or slice the image, which is specified by the border- image-source property.
border-image-width	It sets the width of the border-image.
border-image-outset	It sets the amount of space by which the border image is set out from its border box.
border-image-repeat	It controls the repetition of the image to fill the area of the border. stretch  repeat  round  space

# Lists

There are various CSS properties that can be used to control lists. Lists can be classified as ordered lists and unordered lists. In ordered lists, marking of the list items is with alphabet and numbers, whereas in unordered lists, the list items are marked using bullets.

We can style the lists using CSS. CSS list properties allow us to:

1. Set the distance between the text and the marker in the list.
2. Specify an image for the marker instead of using the number or bullet point.
3. Control the marker appearance and shape.
4. Place the marker outside or inside the box that contains the list items.
5. Set the background colors to list items and lists.

# Properties

**list-style-type:** This property is responsible for controlling the appearance and shape of the marker.

**list-style-image:** It sets an image for the marker instead of the number or a bullet point.

**list-style-position:** It specifies the position of the marker.

**list-style:** It is the shorthand property of the above properties.

**marker-offset:** It is used to specify the distance between the text and the marker.  
It is unsupported in IE6 or Netscape 7.

# list-style-type

It allows us to change the default list type of marker to any other type such as square, circle, roman numerals, Latin letters, and many more. By default, the ordered list items are numbered with Arabic numerals (1, 2, 3, etc.), and the items in an unordered list are marked with round bullets (•).

If we set its value to none, it will remove the markers/bullets.

**Possible values:**

1. decimal
2. lower-alpha
3. lower-roman
4. circle
5. square
6. disc

Githublink

<https://github.com/TopsCode/WEB DESIGNING/blob/main/Module-2/CSS-CSS3/17 %20CSS%20Lists.html>

# list-style-position

It represents whether the appearing of the marker is inside or outside of the box containing the bullet points. It includes two values.

inside: It means that the bullet points will be in the list item. In this, if the text goes on the second line, then the text will be wrap under the marker.

outside: It represents that the bullet points will be outside the list item. It is the default value.

## Syntax:

list-style-position:inside; [WEB DESIGNING/Module-2/CSS-CSS3/list\\_style\\_position.html at main · TopsCode/WEB DESIGNING · GitHub](#)

list-style-position:outside;

# list-style-image

It specifies an image as the marker. Using this property, we can set the image bullets. Its syntax is similar to the background-image property. If it does not find the corresponding image, the default bullets will be used.

## Syntax:

```
list-style-image: url(img.png);
```

[WEB DESIGNING/Module-2/CSS-CSS3/List style image.html at main · TopsCode/WEB DESIGNING · GitHub](#)

# list-style

It is the shorthand property that is used to set all list properties in one expression. The order of the values of this property is type, position, and image. But if any property value is missing, then the default value will be inserted.

## Syntax:

```
list-style: lower-alpha inside url(img.png);
```

## Github Link

[https://github.com/TopsCode/WEB\\_DESIGNING/blob/main/Module-2/CSS-CSS3/17 %20CSS%20Lists.html](https://github.com/TopsCode/WEB_DESIGNING/blob/main/Module-2/CSS-CSS3/17 %20CSS%20Lists.html)

# Style-type

It seems like you might be referring to the `list-style-type` property in CSS, which is used to define the style of the list item markers (such as bullets or numbers) in unordered (`ul`) and ordered (`li`) lists.

## Syntax:

=>none: Removes the list markers (bullets or numbers).

=>disc: The default marker for unordered lists, a filled circle (•).

=>circle: A hollow circle (○) for unordered lists.

=>square: A filled square (▪) for unordered lists.

# Style-type

- =>decimal: The default marker for ordered lists, numbers (1, 2, 3, ...).
- =>lower-alpha: Uses lowercase letters (a, b, c, ...).
- => upper-alpha: Uses uppercase letters (A, B, C, ...).
- =>lower-roman: Uses lowercase Roman numerals (i, ii, iii, ...).
- =>upper-roman: Uses uppercase Roman numerals (I, II, III, ...).
- =>decimal-leading-zero: Uses numbers with leading zeros for ordered lists (01, 02, 03, ...).

# CSS Display

CSS display is the most important property of CSS which is used to control the layout of the element. It specifies how the element is displayed.

CSS Display Value:

- display: inline;
- display: inline-block;
- display: block;
- display: none;
- flex

Github Link

[https://github.com/TopsCode/WEB\\_DESIGNING/blob/main/Module-2/CSS-CSS3/37display flex.html](https://github.com/TopsCode/WEB_DESIGNING/blob/main/Module-2/CSS-CSS3/37display%20flex.html)

# 1. Inline

The inline element takes the required width only. It doesn't force the line break so the flow of text doesn't break in inline example.

The inline elements are:

<span>

<a>

<em>

<b> etc.

**Ex.**

p {

display: inline;

}

[Github Link](#)

[https://github.com/TopsCode/WEB\\_DESIGNING/blob/main/Module-2/CSS-CSS3/37display\\_flex.html](https://github.com/TopsCode/WEB_DESIGNING/blob/main/Module-2/CSS-CSS3/37display_flex.html)

## 2. inline-block

The CSS display inline-block element is very similar to inline element but the difference is that you are able to set the width and height.

**Ex.**

```
p {  
display: inline-block;  
}
```

**Github Link**

[https://github.com/TopsCode/WEB\\_DESIGNING/blob/main/Module-2/CSS-CSS3/37display\\_flex.html](https://github.com/TopsCode/WEB_DESIGNING/blob/main/Module-2/CSS-CSS3/37display_flex.html)

# 3. block

The CSS display block element takes as much as horizontal space as they can. Means the block element takes the full available width. They make a line break before and after them.

**Ex.**

```
p {  
display: block;  
}
```

**Github Link**

[https://github.com/TopsCode/WEB\\_DESIGNING/blob/main/Module-2/CSS-CSS3/37display\\_flex.html](https://github.com/TopsCode/WEB_DESIGNING/blob/main/Module-2/CSS-CSS3/37display_flex.html)

# 4. none

The "none" value totally removes the element from the page. It will not take any space.

**Ex.**

hidden

```
{ display:  
  none;  
}
```

[Github Link](#)

[https://github.com/TopsCode/WEB\\_DESIGNING/blob/main/Module-2/CSS-CSS3/37display flex.html](https://github.com/TopsCode/WEB_DESIGNING/blob/main/Module-2/CSS-CSS3/37display%20flex.html)

# 5. flex

It is used to display an element as an block-level flex container. It is new in css3.

**Ex.**

```
p {  
display: flex;  
}
```

**Github Link**

[https://github.com/TopsCode/WEB\\_DESIGNING/blob/main/Module-2/CSS-CSS3/37display\\_flex.html](https://github.com/TopsCode/WEB_DESIGNING/blob/main/Module-2/CSS-CSS3/37display_flex.html)

# CSS Flex

## CSS Flexbox Layout Module

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

- ✓ Block, for sections in a webpage
- ✓ Inline, for text
- ✓ Table, for two-dimensional table data
- ✓ Positioned, for explicit position of an element
- ✓ The Flexible Box Layout Module, makes it easier to design flexible

responsive layout structure without using float or positioning.

[Github Link](#)

[https://github.com/TopsCode/WEB\\_DESIGNING/blob/main/Module-2/CSS-CSS3/37display\\_flex.html](https://github.com/TopsCode/WEB_DESIGNING/blob/main/Module-2/CSS-CSS3/37display_flex.html)

# CSS Flex Items

The direct child elements of a flex container automatically becomes flexible (flex) items.

```
<style>  
.flex-container  
{ display: flex;  
background-color: #f1f1f1;  
}  
.flex-container > div  
{ background-color:  
DodgerBlue; color: white;  
width: 100px;  
margin: 10px;  
text-align: center;  
line-height: 75px;  
font-size: 30px;  
}  
</style>
```



```
<!DOCTYPE html>
<html>
<head>
<style>
.flex-container { display:
  flex;
  background-color: DodgerBlue;}
.flex-container > div { background-
  color: #f1f1f1; margin: 10px;
  padding: 20px; font-size:
  30px;}
</style>
</head>
<body>
<h1>Create a Flex Container</h1>
<div class="flex-container">
  <div>1</div>
  <div>2</div>
  <div>3</div>
</div>
</body>
</html>
```

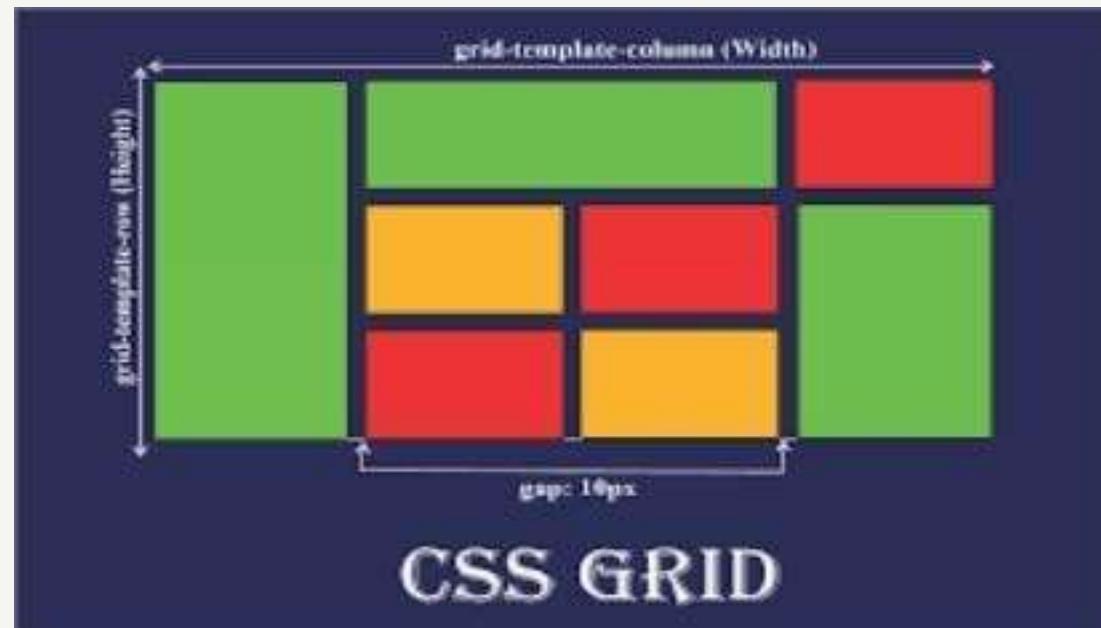


# CSS Grid

## Grid Layout

The CSS Grid Layout Module offers a grid-based layout system, with rows and columns, making it easier to design web pages without having to use floats and positioning.

```
<div class="grid-container">  
  <div class="grid-item">1</div>  
  <div class="grid-item">2</div>  
  <div class="grid-item">3</div>  
  <div class="grid-item">4</div>  
  <div class="grid-item">5</div>  
  <div class="grid-item">6</div>  
  <div class="grid-item">7</div>  
</div>
```



**GitHub Link**

[https://github.com/TopsCode/WEB\\_DESIGNING/blob/main/Module-](https://github.com/TopsCode/WEB_DESIGNING/blob/main/Module-)

# CSS Grid Container

To make an HTML element behave as a grid container, you have to set the display property to grid or inline-grid.

```
<style>
.grid-container
{ display: grid;
grid-template-columns: auto auto auto auto;
gap: 10px;
background-color: #2196F3;
padding: 10px;
}

.grid-container > div {
background-color: rgba(255, 255, 255, 0.8);
text-align: center;
padding: 20px 0;
font-size: 30px;
}
</style>
```

# CSS Grid Item

A grid *container* contains grid *items*.

By default, a container has one grid item for each column, in each row, but you can style the grid items so that they will span multiple columns and/or rows.

**Note:** The grid-column property is a shorthand property for the grid-column-start and the grid-column-end properties.

## Example

Make "item1" start on column 1 and end before column 5:

```
.item1 {  
  grid-column: 1 / 5;  
}
```

# CSS Position

The CSS position property is used to set position for an element. it is also used to place an element behind another and also useful for scripted animation effect.

You can position an element using the top, bottom, left and right properties. These properties can be used only after position property is set first. A position element's computed position property is relative, absolute, fixed or sticky.

1. CSS Static Positioning
2. CSS Fixed Positioning
3. CSS Relative Positioning
4. CSS Absolute Positioning

**Github Link**

[https://github.com/TopsCode/WEB DESIGNING/blob/main/Module-2/CSS-CSS3/32css\\_positions.html](https://github.com/TopsCode/WEB DESIGNING/blob/main/Module-2/CSS-CSS3/32css_positions.html)

# 1. CSS Static Positioning

This is a by default position for HTML elements. It always positions an element according to the normal flow of the page. It is not affected by the top, bottom, left and right properties.

**Ex.**

```
position: static;
```

Github Link

[https://github.com/TopsCode/WEB\\_DESIGNING/blob/main/Module-2/CSS-CSS3/32css\\_positions.html](https://github.com/TopsCode/WEB_DESIGNING/blob/main/Module-2/CSS-CSS3/32css_positions.html)

## 2. CSS Fixed Positioning

The fixed positioning property helps to put the text fixed on the browser. This fixed test is positioned relative to the browser window, and doesn't move even you scroll the window.

**Ex.**

```
position: fixed;
```

**Github Link**

[https://github.com/TopsCode/WEB\\_DESIGNING/blob/main/Module-2/CSS-CSS3/32css\\_positions.html](https://github.com/TopsCode/WEB_DESIGNING/blob/main/Module-2/CSS-CSS3/32css_positions.html)

# 3. CSS Relative Positioning

The relative positioning property is used to set the element relative to its normal position.

**Ex.**

```
position: relative;
```

**Github Link**

[https://github.com/TopsCode/WEB\\_DESIGNING/blob/main/Module-2/CSS-CSS3/35relative\\_position.html](https://github.com/TopsCode/WEB_DESIGNING/blob/main/Module-2/CSS-CSS3/35relative_position.html)

# 4. CSS Absolute Positioning

The absolute positioning is used to position an element relative to the first parent element that has a position other than static. If no such element is found, the containing block is HTML.

**Ex.**

position: absolute;

**Github Link**

[https://github.com/TopsCode/WEB\\_DESIGNING/blob/main/Module-2/CSS-CSS3/36absolute\\_z-index.html](https://github.com/TopsCode/WEB_DESIGNING/blob/main/Module-2/CSS-CSS3/36absolute_z-index.html)

# 5. CSS Sticky Property

The CSS position property is used to set the position for an element. It is also used to place an item behind another element and also useful for the scripted animation effect. The "position: sticky;" is used to position the element based on the scroll position of the user.

This CSS property allows the elements to stick when the scroll reaches to a certain point. Depends upon the scroll position, a sticky element toggles in between fixed and relative. The element will be positioned relative until the specified position of offset is met in the viewport. Then, similar to position: fixed, the element sticks in one place.

**Ex.**

position: sticky; [Github Link](#)

---

[https://github.com/TopsCode/WEB DESIGNING/blob/main/Module-2/CSS-CSS3/32css\\_positions.html](https://github.com/TopsCode/WEB DESIGNING/blob/main/Module-2/CSS-CSS3/32css_positions.html)

# 5. CSS Sticky Ex.

```
.stick{  
    position: sticky;  
    top:50px;  
    padding: 10px;  
    font-size:20px;  
    font-weight:bold;  
    background-color:  
    lightblue; border: 1px solid  
    blue;  
}
```

# CSS Positions

CSS Positioning property provides the possible ways

we can set the element in different positions as shown below:

which

1. top
2. left
3. right
4. bottom

# How the property works

The effects of this property on positioned elements other than the value static are listed as follows:

- When the element is absolutely or fixed positioned (i.e., position: absolute; and position: fixed;), the left property specifies the distance between the element's left edge and the left edge of its containing block (ancestor to which the element is relatively positioned).
- If the element is relatively positioned (i.e., position: relative;), the left property sets the element's left edge to the left/right from its normal position.
- If the position is set to sticky, e., position: sticky; then the positioning context is the viewport. When the element is inside the viewport, the left property behaves like its position is relative. When the element is outside, the left property behaves like its position is fixed.

# left, right, top and bottom

This CSS property specifies the left, right, top and bottom offset for the horizontal positioned elements and does not affect the non-positioned elements.

When both left and right properties are defined, the right value has a preference if the container is right-to-left, and the left value has preference if the container is left-to-right.

## Syntax:

left/right/top/bottom : auto | length | percentage

# Ex.

```
.len{  
    left/right/top/bottom : length in px;  
}  
.per{  
    left/right/top/bottom : length in %;  
}  
.auto{  
    left/right/top/bottom : auto;  
}
```

# Values

Value	Description
auto	This is the default value.
length	This value defines the position of the property in px, cm, pt, etc. It allows negative values.
percentage	This value defines the position of the property in percentage (%). It is calculated to the width of the element's containing block. It also allows negative values.

# CSS Cursor

It is used to define the type of mouse cursor when the mouse pointer is on the element.

Ex.

1. cursor:alias
2. cursor:auto
3. cursor:all-scroll
4. cursor:col-resize
5. cursor:crosshair
6. cursor:default
7. cursor:copy
8. cursor:pointer
9. cursor:move

Github Link

[https://github.com/TopsCode/WEB\\_DESIGNING/blob/main/Module-](https://github.com/TopsCode/WEB_DESIGNING/blob/main/Module-)

# CSS Cursor...

- 10. cursor:e-resize
- 11. cursor:ew-resize
- 12. cursor:ne-resize
- 13. cursor:nw-resize
- 14. cursor:n-resize
- 15. cursor:se-resize
- 16. cursor:sw-resize
- 17. cursor:s-resize
- 18. cursor:w-resize
- 19. cursor:text
- 20. cursor:wait
- 21. cursor:help
- 22. cursor:progress
- 23. cursor:no-drop
- 24. cursor:not-allowed
- 25. cursor:vertical-text
- 26. cursor:zoom-in
- 27. cursor:zoom-out

# CSS Button

In HTML, we use the button tag to create a button, but by using CSS properties, we can style the buttons. Buttons help us to create user interaction and event processing.

# background-color

This property is used for setting the background color of the button element.

**Ex.**

```
button {  
    color:lightgoldenrodyellow;  
}
```

[WEB DESIGNING/Module-2/CSS-CSS3/Background\\_color.html at  
main · TopsCode/WEB DESIGNING · GitHub](#)

# border

It is used to set the border of the button. It is the shorthand property for border-width, border-color, and border-style.

**Ex.**

```
button {  
    border:none; / border:5px brown solid; / border: 5px red dashed;  
/ border: 5px black dotted; / border:5px blue double;  
}
```

**Github Link**

[https://github.com/TopsCode/WEB\\_DESIGNING/blob/main/Module-2/CSS-CSS3/12border.html](https://github.com/TopsCode/WEB_DESIGNING/blob/main/Module-2/CSS-CSS3/12border.html)

# border-radius

It is used to make the rounded corners of the button. It sets the border radius of the button.

**Ex.**

```
button {  
    border-radius: value in px or %;  
}
```

# box-shadow

As its name implies, it is used to create the shadow of the button box. It is used to add the shadow to the button. We can also create a shadow during the hover on the button.

box-shadow: [horizontal offset] [vertical offset] [blur radius] [optional spread radius] [color];

**Ex.**

```
button { box-shadow : 0 8px 16px 0 black, 0 6px 20px 0 rgba(0, 0, 0, 0.19);}
```

```
button :hover{
```

```
    box-shadow : 0 8px 16px 0 black, 0 6px 20px 0 rgba(0, 0, 0, 0.19);
```

```
}
```

[Github Link](#)

<https://github.com/TopsCode/WEB DESIGNING/blob/main/Module 8 Tai lwind/30-box-shadow.html>

# padding

It is used to set the button padding.

box-shadow: [horizontal offset] [vertical offset] [blur radius] [optional spread radius] [color];

**Ex.**

```
button {  
    padding: value in px;  
}
```

**Github Link**

[https://github.com/TopsCode/WEB\\_DESIGNING/blob/main/Module-2/CSS-CSS3/10margin\\_padding\\_box-sizing.html](https://github.com/TopsCode/WEB_DESIGNING/blob/main/Module-2/CSS-CSS3/10margin_padding_box-sizing.html)

# CSS Float

The CSS float property is a positioning property. It is used to push an element to the left or right, allowing other elements to wrap around it. It is generally used with images and layouts.

Center Aligned



Left Aligned



Right Aligned



**Github Link**

<https://github.com/TopsCode/WEB DESIGNING/blob/main/Module-2/CSS->

# Concept of Float

A floated element may be moved as far to the left or the right as possible. Simply, it means that a floated element can display at extreme left or extreme right.

The elements after the floating element will flow around it.

The elements before the floating element will not be affected.

If the image floated to the right, the texts flow around it, to the left and if the image floated to the left, the text flows around it, to the right.

# Float Example

```
<style>
img {
    float: right;
}
</style>
```

**Github Link**

<https://github.com/TopsCode/WEB DESIGNING/blob/main/Module-2/CSS-CSS3/09 %20Float%20and%20Clear%20and%20Alignment.html>

# CSS clearfix

A clear float (or clearfix) is a way for an element to fix or clear the child elements so that we do not require to add additional markup. It resolves the error which occurs when more than one floated elements are stacked next to each other.

**Ex.**

```
p{  
    clear:right;  
}
```

[WEB DESIGNING/Module-2/CSS-CSS3/Clearfix.html](#)  
[at main · TopsCode/WEB DESIGNING · GitHub](#)

# CSS Colors

The color property in CSS is used to set the color of HTML elements. Typically, this property is used to set the background color or the font color of an element.

Ways to show colors in CSS:

1. RGB format.
2. RGBA format.
3. Hexadecimal notation.
4. HSL.
5. HSLA.
6. Built-in color.

---

**Github Link**

[https://github.com/TopsCode/WEB\\_DESIGNING/blob/main/Module-](https://github.com/TopsCode/WEB_DESIGNING/blob/main/Module-)

# RGB Format

RGB format is the short form of 'RED GREEN and BLUE' that is used for defining the color of an HTML element simply by specifying the values of R, G, B that are in the range of 0 to 255.

The color values in this format are specified by using the `rgb()` property. This property allows three values that can either be in percentage or integer (range from 0 to 255).

This property is not supported in all browsers;

## Syntax:

`color: rgb(R, G, B);`

---

**Github Link**

[https://github.com/TopsCode/WEB\\_DESIGNING/blob/main/Module-2/CSS-CSS3/08color.html](https://github.com/TopsCode/WEB_DESIGNING/blob/main/Module-2/CSS-CSS3/08color.html)

# RGBA Format

It is almost similar to RGB format except that RGBA contains A (Alpha) that specifies the element's transparency. The value of alpha is in the range 0.0 to 1.0, in which 0.0 is for fully transparent, and 1.0 is for not transparent.

## Syntax:

```
color:rgba(R, G, B, A);
```

## Github Link

[https://github.com/TopsCode/WEB\\_DESIGNING/blob/main/Module-2/CSS-CSS3/08color.html](https://github.com/TopsCode/WEB_DESIGNING/blob/main/Module-2/CSS-CSS3/08color.html)

# Hexadecimal notation

Hexadecimal can be defined as a six-digit color representation. This notation starts with the # symbol followed by six characters ranges from 0 to F. In hexadecimal notation, the first two digits represent the red (RR) color value, the next two digits represent the green (GG) color value, and the last two digits represent the blue (BB) color value.

## Syntax:

color:#(0-F)(0-F)(0-F)(0-F)(0-F)(0-F);

## Shorthand Hex-Code:

It is a short form of hexadecimal notation in which every digit is recreated to arrive at an equivalent hexadecimal value.

For example, #7B6 becomes #77BB66 in hexadecimal.

### Github Link

[https://github.com/TopsCode/WEB\\_DESIGNING/blob/main/Module-2/CSS-CSS3/08color.html](https://github.com/TopsCode/WEB_DESIGNING/blob/main/Module-2/CSS-CSS3/08color.html)

# HSL

It is a short form of Hue, Saturation, and Lightness.

Let's understand them individually.

**Hue:** It can be defined as the degree on the color wheel from 0 to 360. 0 represents red, 120 represents green, 240 represents blue.

**Saturation:** It takes value in percentage in which 100% represents fully saturated, i.e., no shades of gray, 50% represent 50% gray, but the color is still visible, and 0% represents fully unsaturated, i.e., completely gray, and the color is invisible.

**Lightness:** The lightness of the color can be defined as the light that we want to provide the color in which 0% represents black (there is no light), 50% represents neither dark nor light, and 100% represents white (full lightness).

# HSLA

It is entirely similar to HSL property, except that it contains A (alpha) that specifies the element's transparency. The value of alpha is in the range 0.0 to 1.0, in which

0.0 indicates fully transparent, and 1.0 indicates not transparent.

## Syntax:

color:hsla(H, S, L, A);

## Github Link

[https://github.com/TopsCode/WEB\\_DESIGNING/blob/main/Module-2/CSS-CSS3/08color.html](https://github.com/TopsCode/WEB_DESIGNING/blob/main/Module-2/CSS-CSS3/08color.html)

# Built-in Color

As its name implies, built-in color means the collection of previously defined colors that are used by using a name such as red, blue, green, etc.

## Syntax:

```
color: color-name;
```

# CSS Important

This property in CSS is used to give more importance compare to normal property. The !important means 'this is important'. This rule provides a way of making the Cascade in CSS.

If a rule is defined with this attribute, it will reject the normal concern in which the later used rule overrides the previous ones. If we use more than one declaration marked !important, then the normal cascade takes it over again. That means the new marked !important will replace the previous one.

# Ex.

```
element {  
    font-size: 14px  
    !important; color: blue  
        !important;  
    ...  
}
```

# Ex.

```
h1 {  
    border-color: red;  
    border: 5px green  
    solid; border-color:  
    black;  
}  
  
h1 {  
    border-color: red  
    !important;  
    border: 5px green solid;
```

# Line Height

The CSS line height property is used to define the minimal height of line boxes within the element. It sets the differences between two lines of your content.

It defines the amount of space above and below inline elements. It allows you to set the height of a line of independently from the font size.

**Ex.**

```
line-height: value in  
numbers; line-height: value  
in %;  
line-height: value in px, pt, cm;
```

[Github Link](#)

[https://github.com/TopsCode/WEB\\_DESIGNING/blob/main/Module-2/CSS-CSS3/12\\_Line%20Height.html](https://github.com/TopsCode/WEB_DESIGNING/blob/main/Module-2/CSS-CSS3/12_Line%20Height.html)

# CSS Margin

CSS Margin property is used to define the space around elements. It is completely transparent and doesn't have any background color. It clears an area around the element.

Top, bottom, left and right margin can be changed independently using separate properties. You can also change all properties at once by using shorthand margin property.

## Margin Properties:

margin

margin-left

margin-

right

margin-top

margin-bottom

**Github Link**

[https://github.com/TopsCode/WEB DESIGNING  
/blob/main/Module-2/CSS-  
CSS3/10margin\\_padding\\_box-sizing.html](https://github.com/TopsCode/WEB DESIGNING/blob/main/Module-2/CSS-CSS3/10margin_padding_box-sizing.html)

# CSS Margin

**Ex.**

```
margin: value auto;  
margin: value length in px,pt  
cm; margin: value %;
```

## Margin Shorthand

**properties:** margin: 50px  
100px 150px 200px; margin:  
50px 100px 150px; margin:  
50px 100px; margin 50px;

[Github Link](#)

[https://github.com/TopsCode/WEB DESIGNING/blob/main/Module-2/CSS-CSS3/10margin\\_padding\\_box-sizing.html](https://github.com/TopsCode/WEB DESIGNING/blob/main/Module-2/CSS-CSS3/10margin_padding_box-sizing.html)

# CSS Padding

CSS Padding property is used to define the space between the element content and the element border.

It is different from CSS margin in the way that CSS margin defines the space around elements. CSS padding is affected by the background colors. It clears an area around the content.

Top, bottom, left and right padding can be changed independently separate properties. You can also change all properties at once by shorthand padding property.

using  
using

**Github Link**

[https://github.com/TopsCode/WEB DESIGNING/blob/main/Module-2/CSS-CSS3/10margin\\_padding\\_box-sizing.html](https://github.com/TopsCode/WEB DESIGNING/blob/main/Module-2/CSS-CSS3/10margin_padding_box-sizing.html)

# CSS Padding Properties

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

**Values:**

padding: value in %;  
padding: value in length as in px, pt,  
cm;

## Margin Shorthand

**properties:** padding: 50px  
100px 150px 200px; padding:  
50px 100px 150px; padding:  
50px 100px; padding: 50px;

**Github Link**

[https://github.com/TopsCode/WEB\\_DESIGNING/blob/main/Module-2/CSS-CSS3/10margin\\_padding\\_box-sizing.html](https://github.com/TopsCode/WEB_DESIGNING/blob/main/Module-2/CSS-CSS3/10margin_padding_box-sizing.html)

# What is CSS Filters

CSS filters are used to set visual effects to text, images, and other aspects of a webpage. The CSS filter property allows us to access the effects such as color or blur, shifting on the rendering of an element before the element gets displayed.

## Syntax:

filter: none

filter: invert()

filter: drop-shadow()

filter: brightness()

filter: saturate()

filter: blur()

filter: hue-rotate() [Github Link](#)

filter: contrast()

filter: opacity()

filter:

grayscale()

filter: sepia()

filter: url();

# brightness()

As its name implies, it is used to set the brightness of an element. If the brightness is 0%, then it represents completely black, whereas 100% brightness represents the original one. It can also accept values above 100% that provide brighter results.

## Syntax:

```
filter: brightness( value in %);
```

# blur()

It is used to apply the blur effect to the element. If the blur value is not specified, then the value 0 is used as a default value. The parameter in blur() property does not accept the percentage values. A larger value of it creates more blur.

## Syntax:

`filter: blur( value in px);`

[https://github.com/TopsCode/WEB\\_DESIGNING/blob/main/Module-2/CSS-CSS3/Blur.html](https://github.com/TopsCode/WEB_DESIGNING/blob/main/Module-2/CSS-CSS3/Blur.html)

# invert()

It is used to invert the samples in the input image. Its 100% value represents completely inverted, and 0% values leave the unchanged input. Negative values are not allowed in it.

## Syntax:

`filter: invert(value);`

[WEB DESIGNING/Module-2/CSS-CSS3/Invert.html at  
main · TopsCode/WEB DESIGNING · GitHub](#)

# saturate()

It sets the saturation of an element. The 0% saturation represents the completely unsaturated element, whereas the 100% saturation represents the original one. The values greater than 100% are allowed that provides super-saturated results. We cannot use negative values with this property.

## Syntax:

```
filter: saturate(40);
```

# drop-shadow()

It applies the drop-shadow effect to the input image.

The values it accepts are h-shadow, v-shadow, blur, spread, and color.

## Syntax:

```
filter: drop-shadow(10px 20px 30px yellow);
```

# contrast()

It adjusts the contrast of the input. Its 0% value will create a completely black image, whereas the 100% values leave the unchanged input, i.e., represents the original one. Values greater than 100% are allowed that provides results with less contrast.

## Syntax:

```
filter: contrast(50%);
```

# opacity()

It is used to apply transparency to the input image. Its 0% value indicates completely transparent, whereas the 100% value represents the original image, i.e., fully opaque.

## Syntax:

```
filter: opacity(40%);
```

[WEB DESIGNING/Module-2/CSS-CSS3/Opacity.html](#)  
[at main · TopsCode/WEB DESIGNING · GitHub](#)

# hue-rotate()

It applies a hue-rotation on the input image. Its perimeter value defines the number of degrees around the color circle; the image will be adjusted. Its default value is 0 degree, which represents the original image. Its maximum value is 360 degrees.

## Syntax:

```
filter: hue-rotate(240deg);
```

# grayscale()

It converts the input image into black and white. 0% grayscale represents the original one, whereas 100% represents completely grayscale. It converts the object colors into 256 shades of gray.

## Syntax:

```
filter: grayscale(80%);
```

# sepia()

It is used to transform the image into a sepia image. 0% value represents the original image, whereas the 100% value indicates the completely sepia.

## Syntax:

```
filter: sepia(90%);
```

[WEB DESIGNING/Module-2/CSS-CSS3/Sepia.html at  
main · TopsCode/WEB DESIGNING · GitHub](#)

# CSS Images

Images are an important part of any web application. Including a lot of images in a web application is generally not recommended, but it is important to use the images wherever they required. CSS helps us to control the display of images in web applications.

The styling of an image in CSS is similar to the styling of an element by using the borders and margins. There are multiple CSS properties such as border property, height property, width property, etc. that helps us to style an image.

---

**Github Link**

[https://github.com/TopsCode/WEB\\_DESIGNING/blob/main/Module-2/CSS-CSS3/07background-image.html](https://github.com/TopsCode/WEB_DESIGNING/blob/main/Module-2/CSS-CSS3/07background-image.html)

# Thumbnail Image

The border property is used to make a thumbnail image.

**Ex.**

```
img{  
border: 2px solid  
red; border-  
radius:5px;  
padding:10px;  
}
```

[WEB DESIGNING/Module-2/CSS-CSS3/Thumbnail.html at  
main · TopsCode/WEB DESIGNING · GitHub](#)

# Transparent image

To make an image transparent, we have to use the opacity property. The value of this property lies between 0.0 to 1.0, respectively.

**Ex.**

```
img{  
border: 2px solid red;  
border-radius:5px;  
padding:10px;  
opacity:0.3;  
}
```

[WEB DESIGNING/Module-2/CSS-CSS3/Transperent.html at  
main · TopsCode/WEB DESIGNING · GitHub](#)

# Rounded image

The border-radius property sets the radius of the bordered image. It is used to create the rounded images.

**Ex.**

```
#img1{  
border-radius:10px;  
}
```

```
#img2{  
border-radius:50%;  
}
```

[WEB DESIGNING/Module-2/CSS-CSS3/Rounded\\_img.htm at main · TopsCode/WEB DESIGNING · GitHub](#)

# Responsive Image

It automatically adjusts to fit on the screen size. It is used to adjust the image to the specified box automatically.

**Ex.**

```
#img1{  
max-width:100%;  
height:auto;  
}
```

[WEB DESIGNING/Module-2/CSS-CSS3/Responsive\\_img.html at  
main · TopsCode/WEB DESIGNING · GitHub](#)

# Center an Image

We can center an image by using the left-margin and right-margin property. We have to set these properties to auto in order to make a block element.

**Ex.**

```
#img1{  
margin-left:auto;  
margin-right:auto;  
display:block;  
}
```

[WEB DESIGNING/Module-2/CSS-CSS3/Center\\_img.html at main · TopsCode/WEB DESIGNING · GitHub](#)

# CSS Overflow

The CSS overflow property specifies how to handle the content when it overflows its block level container.

We know that every single element on a page is a rectangular box and the size, positioning and behavior of these boxes are controlled via CSS.

Let's take an example: If you don't set the height of the box, it will grow as large as the content. But if you set a specific height or width of the box and the content inside cannot fit then what will happen. The CSS overflow property is used to overcome this problem. It specifies whether to clip content, render scroll bars, or just display content.

[GitHub Link](#)

<https://github.com/TopsCode/WEB DESIGNING/blob/main/Module-2/CSS-CSS3/16overflow.html>

# Overflow Values

1. visible: It specifies that overflow is not clipped. it renders outside the element's box. this is a default value.
1. hidden: It specifies that the overflow is clipped, and rest of the content will be invisible.
1. scroll: It specifies that the overflow is clipped, and a scroll bar is used to see the rest of the content.
1. auto: It specifies that if overflow is clipped, a scroll bar is needed to see the rest of the content.

# CSS Overflow Ex

```
<style>
div.scroll
{
    background-
    color:
    #00ffff; width: 100px;
    height: 100px;
    overflow: scroll / visible / hidden /
    auto;
}
</style>
```

---

**Github Link**

[https://github.com/TopsCode/WEB\\_DESIGNING/blob/main/Module-2/CSS-CSS3/16overflow.html](https://github.com/TopsCode/WEB_DESIGNING/blob/main/Module-2/CSS-CSS3/16overflow.html)

# Vertical Align

The CSS vertical align property is used to define the vertical alignment of an inline or table-cell box.

1. It is applied to inline or inline-block elements.
2. It affects the alignment of the element, not its content. (except table cells)
3. When it applied to the table cells, it affect the cell contents, not the cell itself.

[WEB DESIGNING/Module-2/CSS-CSS3/Vertical\\_align.html at  
main · TopsCode/WEB DESIGNING · GitHub](#)

# Vertical Align

## Ex. Values

vertical-align:  
baseline; vertical-  
align: length; vertical-  
align: %; vertical-align:  
sub; vertical-align:  
super;

vertical-align: top;  
vertical-align: bottom;  
vertical-align: text-top;  
vertical-align: middle;  
vertical-align: text-bottom;

# White Space

The CSS white space property is used to specify how to display the content within an element. It is used to handle the white spaces inside an element.

## Values:

white-space: normal;  
white-space: nowrap;  
white-space: pre;  
white-space: pre-line;  
white-space: pre-wrap;

[Github Link](#)

[https://github.com/TopsCode/WEB\\_DESIGNING/blob/main/Module-2/CSS-CSS3/45\\_White-space.html](https://github.com/TopsCode/WEB_DESIGNING/blob/main/Module-2/CSS-CSS3/45_White-space.html)

# Word Wrap

CSS word wrap property is used to break the long words and wrap onto the next line. This property is used to prevent overflow when an unbreakable string is too long to fit in the containing box.

## Values:

word-wrap: normal;  
word-wrap: break-word;  
word-wrap: break-all;

[WEB DESIGNING/Module-2/CSS-CSS3/Word Wrap.html at main · TopsCode/WEB DESIGNING · GitHub](#)

# CSS Width Property

The CSS width property is used to set the width of the content area of an element.

It does not include padding borders or margins. It sets width of the area inside the padding, border, and margin of the element.

## Values:

width: auto;

width:

length;

width: %;

We can use **max-width** and **min-width** properties to set the maximum and minimum width of

---

Github Link

[https://github.com/TopsCode/WEB\\_DESIGNING/blob/main/Module-2/CSS-](https://github.com/TopsCode/WEB_DESIGNING/blob/main/Module-2/CSS-)

# CSS Height Property

This CSS property sets the height of an element. It is used to set the height of content area of an element.

It does not include padding borders or margins, whereas it sets the height of the area inside the padding, border, and margin of the element. It can accept the length and percentage values. But it does not allow negative values.

If we set the height to a numeric value (like in px, %, etc.), the content can be overflow if it does not fit in the given height. We can manage the overflowing content by defining the overflow property.

[WEB DESIGNING/Module-2/CSS-CSS3/Height\\_property.html at main · TopsCode/WEB DESIGNING · GitHub](#)

# CSS Height Values

## Values:

width: auto;

width: length;

We can use **max-height** and **min-height** properties to set the maximum and minimum height of the element

# Box-shadow CSS

It is used to add shadow-like effects around the frame of an element.

## Syntax:

```
box-shadow: h-offset v-offset blur spread color  
| inset | inherit | initial | none;
```

# Text-shadow CSS

As its name implies, this CSS property adds shadows to the text. It accepts the comma-separated list of shadows that applied to the text. Its default property is none. It applies one or more than one text-shadow effect on the element's text content.

## Syntax:

text-shadow: h-shadow v-shadow blur-radius color | none | initial | inherit;

[WEB DESIGNING/Module-2/CSS-CSS3/Text\\_shadow.html at main · TopsCode/WEB DESIGNING · GitHub](https://main - TopsCode/WEB DESIGNING - GitHub)

# Box-shadow/Text Shadow Values

**h-offset:** It horizontally sets the shadow position. Its positive value will set the shadow to the right side of the box. Its negative value is used to set the shadow on the left side of the box.

**v-offset:** Unlike the h-offset, it is used to set the shadow position vertically. The positive value in it sets the shadow below the box, and the negative value sets the shadow above of the box.

**blur:** As its name implies, it is used to blur the box-shadow. This attribute is optional.

**spread:** It sets the shadow size. The spread size depends upon the spread value.

**color:** As its name implies, this attribute is used to set the color of the shadow. It is an optional attribute.

# Box-shadow/Text Shadow Values

inset: Normally, the shadow generates outside of the box, but by using inset, the shadow can be created within the box.

initial: It is used to set the property of the box-shadow to its default

value. inherit: it is inherited from its parent.

none: It is the default value that does not include any shadow property.

# Box-shadow Ex.

Ex.

h-offset, v-offset and blur  
attributes box-shadow: 5px 10px  
10px;

spread attribute  
box-shadow: 5px 10px 10px 10px;

color attribute  
box-shadow: 5px 10px 10px 10px  
orange;

inset attribute  
box-shadow: 5px 10px 10px 10px orange  
inset;

initial attribute  
box-shadow: initial;

default attribute  
box-shadow:  
none;

# Box-shadow Ex.

**Ex.**

initial attribute

```
box-shadow: initial;
```

default attribute

```
box-shadow:  
none;
```

# Text-shadow

**Ex.**

Simple Shadow

`text-shadow: 3px 3px red;`

Fuzzy Shadow

`text-shadow: 3px 3px 3px violet;`

Multiple Shadows

`text-shadow: -3px -3px 3px blue, 3px 3px 3px red;`

Glow Effect

`text-shadow: 0 0 .1em cyan;`

# CSS Visibility

The CSS visibility property is used to specify whether an element is visible or not.

Note: An invisible element also take up the space on the page. By using display property you can create invisible elements that don't take up space.

## Syntax:

visibility: visible | hidden

## Github Link

[https://github.com/TopsCode/WEB\\_DESIGNING/blob/main/Module-2/CSS-CSS3/11\\_Opacity%20and%20Visibility.html](https://github.com/TopsCode/WEB_DESIGNING/blob/main/Module-2/CSS-CSS3/11_Opacity%20and%20Visibility.html)

# Ex. with CSS

```
h1.visible  
{ visibility:  
  visible  
}  
  
h1.hidden  
{ visibility:  
  hidden  
}
```

# Ex. with javascript

```
function myFunction()
{ if(document.getElementById("myDIV").style.visibility ==
"visible"){ document.getElementById("myDIV").style.visibility =
"hidden";
}
else{
document.getElementById("myDIV").style.visibility = "visible";
}
```

# CSS Icons

Icons can be defined as the images or symbols used in any computer interface refer to an element. It is a graphical representation of a file or program that helps the user to identify about the type of file quickly.

Using the icon library is the easiest way to add icons to our HTML page. It is possible to format the library icons by using CSS. We can customize the icons according to their color, shadow, size, etc.

There are given some of the icon libraries such as Bootstrap icons, Font Awesome icons, and Google icons that can be used in CSS easily. There is no need to install or download the libraries mentioned above.

[Github Link](#)

<https://github.com/TopsCode/WEB DESIGNING/blob/main/Module-1/CSS%20Icons.md>

# Font Awesome Icons

To use this library in our HTML page, we need to add the following link within the

<head></head> section.

```
<link rel="stylesheet"  
      href="https://cdnjs.cloudflare.com/ajax/libs/font-  
      awesome/4.7.0/css/font-awesome.min.css">  
  
    <i class="fa fa-cloud"></i>  
    <i class="fa fa-file"></i>  
    <i class="fa fa-heart"></i>  
    <i class="fa fa-bars"></i>  
    <i class="fa fa-car"></i>
```

[WEB DESIGNING/Module-2/CSS-CSS3/Font awesome.html at main · TopsCode/WEB DESIGNING · GitHub](#)

# Bootstrap Icons

As similar to the font awesome library, we can add the bootstrap icons in our HTML page using the link given below in the `<head></head>` section.

```
<link rel="stylesheet"
      href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css"
>
<i class="glyphicon glyphicon-cloud"></i>
  <i class="glyphicon glyphicon-file"></i>
  <i class="glyphicon glyphicon-heart"></i>
  <i class="glyphicon glyphicon-user"></i>
  <i class="glyphicon glyphicon-thumbs-up"></i>
  <i class="glyphicon glyphicon-remove"></i>
  <i class="glyphicon glyphicon-envelope"></i>
```

# Google Icons

As similar to the above libraries, we can add it in our HTML page simply by adding the link given below in the `<head></head>` section.

```
<link rel="stylesheet"
      href="https://fonts.googleapis.com/icon?family=Material+Icons"
>
  <i class="material-icons">cloud</i>
  <i class="material-icons">attachment</i>
  <i class="material-icons">computer</i>
  <i class="material-icons">favorite</i>
  <i class="material-icons">traffic</i>
```

# Justify

This CSS property is used to align the items of the flexible box container when the items do not use all available space on the main-axis (horizontally). It defines how the browser distributes the space around and between the content items.

This CSS property can't be used to describe containers or items along the vertical axis. To align the items vertically, we have to use the align-items property.

## Syntax:

justify-content: center | flex-start | flex-end | space-around | space-evenly  
| space-between

[WEB DESIGNING/Module-2/CSS-CSS3/Justify.html at  
main · TopsCode/WEB DESIGNING · GitHub](#)

# Property Values

**center:** As its name implies, it set the position of items at the center of the container.

**flex-start:** It is the default value that positioned the items at the beginning of the container.

**flex-end:** It set the position of items at the end of the container.

**space-around:** It positioned the items with equal spacing from each other. It evenly distributes the items in the line along with the same space around them.

**space-between:** Items are evenly spaced in which the first item is at the beginning, and the last item is at the end.

**space-evenly:** It also positioned the items with equal space, but the spacing from the corners differs.

# Text Decoration

It is a CSS property that decorates the content of the text. It adds lines under, above, and through the text. It sets the appearance of decorative lines on text. This CSS property decorates the text with several kinds of lines. This is shorthand for text-decoration-line, text-decoration-color, and text-decoration-style.

Text Decoration can be done in below mentioned ways:

1. text-decoration-line
2. text-decoration-color
3. text-decoration-style

---

**Github Link**

<https://github.com/TopsCode/WEB DESIGNING/blob/main/Module->

# text-decoration-line

It sets the kind of text-decoration like overline, underline, or line-through. It can be used to add a combination of lines to the selected text.

## Syntax:

```
text-decoration: underline | line-through |  
overline  
text-decoration: overline underline line-  
through;
```

# text-decoration-style

This property is used to set the style of the line. Its values are solid, dotted, wavy, double, and dashed.

## Syntax:

text-decoration: underline double | line-through dashed | overline

overline text-decoration: lightblue wavy overline underline line-through;

# text-decoration-color

This property is used to provide color to the decoration. Its value is any color in a valid format.

## Syntax:

text-decoration: underline double cyan | line-through dashed green |  
dotted overline blue

text-decoration: lightblue wavy overline underline line-through;

# Text Align

This CSS property is used to set the horizontal alignment of a table-cell box or the block element. It is similar to the vertical-align property but in the horizontal direction.

## Syntax:

text-align: justify | center | right | left

[WEB DESIGNING/Module-2/CSS-CSS3/Text\\_align.html at main · TopsCode/WEB DESIGNING · GitHub](#)

# word-spacing

The word-spacing property in CSS is used to control the space between words in a text element. It helps in adjusting the distance between words for better readability or for achieving a particular design effect.

=>normal: This is the default value, which applies the normal space between words.

=>length: Specifies the amount of space between words. You can use units like px, em, rem, %, etc. A positive value increases the space, while a negative value decreases the space between words.

# Property Values

**justify:** It is generally used in newspapers and magazines. It stretches the element's content in order to display the equal width of every line.

**center:** It centers the inline text.

**right:** It is used to align the text to the right.

**left:** It is used to align the text to the left.

# **nth-child(n) selector**

This selector is used for matching the elements based on their position regardless of the type of its parent. The n can either be a keyword, formula, or a number. It is used to match the elements based on their position within a group of siblings. It matches each element, which is the nth-child.

## **Syntax:**

```
:nth-child(n) {  
}
```

The "n" in the parentheses is the argument that represents the pattern for matching elements. It can be a functional notation, even or odd.

## **Github Link**

[https://github.com/TopsCode/WEB\\_DESIGNING/blob/main/Module-2/CSS-CSS3/08 %20Pseudo%20Classes%20and%20Elements.html](https://github.com/TopsCode/WEB_DESIGNING/blob/main/Module-2/CSS-CSS3/08 %20Pseudo%20Classes%20and%20Elements.html)

# Ex.

1. Using Function:

```
<style>  
    p:nth-child(2n+1) {  
    }  
</style>
```

2. Using even and odd.

```
p:nth-child(even) {  
}  
  
p:nth-child(odd) { }
```

# first-child and last-child Selector

The :first-child selector is used to select the specified selector, only if it is the first child of its parent.

**Syntax:**

```
:first-child {  
}
```

The :last-child selector matches every element that is the last child of its parent.

**Syntax:**

```
p:last-child {
```

}      **Github Link**  
[https://github.com/TopsCode/WEB\\_DESIGNING/blob/main/Module-2/CSS-CSS3/08 %20Pseudo%20Classes%20and%20Elements.html](https://github.com/TopsCode/WEB_DESIGNING/blob/main/Module-2/CSS-CSS3/08 %20Pseudo%20Classes%20and%20Elements.html)

# calc()

It is an inbuilt CSS function which allows us to perform calculations. It can be used to calculate length, percentage, time, number, integer frequency, or angle. It uses the four simple arithmetic operators add (+), multiply (\*), subtract (-), and divide (/).

It is a powerful CSS concept because it allows us to mix any units, such as percentage and pixel.

## Syntax:

calc( Expression );

[WEB DESIGNING/Module-2/CSS-CSS3/Calc.html at  
main · TopsCode/WEB DESIGNING · GitHub](#)

# Values

This CSS function takes a single parameter expression, and the result of this mathematical expression will be used as a value. It can be any simple expression using the four arithmetic operators (+, -, \*, /). The expression is mandatory to define.

Ex.

```
.div {  
    width: calc(150% - 75%);  
    height: calc(350px -  
    75px);  
}
```

```
.div {  
    width: calc(40% + 10em);  
    height: calc(350px +  
    75px);  
}
```

# CSS Print

While we need to print the current webpage, css prints page related properties comes into the picture.

There are total of 3 types of print properties available.

1. page-break-before
2. page-break-inside
3. page-break-after

# page-break-before

As the name implies, this CSS property is used to add the page break before the element, when printing the document. It inserts a page break before the specified element during printing the document. We cannot use this CSS property on absolutely positioned elements or an empty <div> element that does not generate a box.

## Syntax:

page-break-before: auto | always | left | right | avoid

## Github Link

[https://github.com/TopsCode/WEB\\_DESIGNING/blob/main/Module-2/CSS-CSS3/37\\_Page-break.html](https://github.com/TopsCode/WEB_DESIGNING/blob/main/Module-2/CSS-CSS3/37_Page-break.html)

# Values

Value	Description
auto	It is the default value that inserts a page break before the <code>&lt;element&gt;</code> , if necessary.
always	This value always forces a page break before the specified element.
avoid	It is used to avoid a page break before the element whenever possible.
left	This value forces either one or two page breaks before the <code>&lt;element&gt;</code> so that the next page will be depicted as the left-hand page.
right	The right value forces either one or two page breaks before the <code>&lt;element&gt;</code> so that the next page will be depicted as the right-hand page.

# page-break-inside

this CSS property is used to specify the page break inside the element, when printing the document. This CSS property cannot be used on absolutely positioned elements or an empty <div> element that does not generate a box. It inserts or avoids the page break inside the specified element during printing the document.

**Syntax:**

page-break-inside: auto | avoid

# Values

Value	Description
auto	It is the default value that inserts a page break inside the element, if necessary.
always	It is used to avoid a page break inside the element whenever possible.

# page-break-after

This CSS property is used to adjust the page break after the element when printing the document. It inserts a page break after the specified element during printing.

## Syntax:

page-break-after: auto | always | left | right | avoid

# Values

Value	Description
auto	It is the default value that inserts a page break after the element, if necessary.
always	It always forces a page break after the specified element.
avoid	It is used to avoid a page break after the element whenever possible.
left	It forces either one or two page breaks after the specified element so that the next page will be depicted as the left-hand page.
right	It forces either one or two page breaks after the specified element so that the next page will be depicted as the right-hand page.

# CSS Columns

The columns property in CSS sets the number and width of columns in a single declaration. It is a shorthand property that can take several values at a time.

It is used to set both column-count and column-width properties at the same time. Both of these properties are used to control how many columns will appear. The column-count property is used to set the number of columns, whereas the column-width property specifies the width of the columns.

Together using column-count and column-width properties creates a multi-column layout that breaks automatically into a single column at narrow browser widths without using the media queries. It is not always helpful to use both of them because it can restrict the responsiveness and flexibility of the layout.

**Github Link**

[https://github.com/TopsCode/WEB\\_DESIGNING/blob/main/Module-2/CSS-CSS3/51\\_Column.html](https://github.com/TopsCode/WEB_DESIGNING/blob/main/Module-2/CSS-CSS3/51_Column.html)

# Values

## Syntax:

columns: auto | column-width column-count |

Value	Description
Auto	It is the default value which sets the values of column-count and column-width to the default browser values.
column-width	It is used to set the minimum width for columns. However, the actual width of the column may be narrower or wider based on the available space.
column-count	It specifies the maximum number of columns.

# Ex.

```
element name/selector{  
    columns: (length) (column number);  
}
```

# CSS Hyphens

This CSS property is used to control the hyphenation of the text in the block-level elements. It defines how the word is hyphenated if it is too long or when the text wraps across multiple lines.

This property allows us to split the word into two lines to improve the text layout.

## Syntax:

hyphens: none | manual | auto

[WEB DESIGNING/Module-2/CSS-CSS3/Hyphens.html](#)  
[at main · TopsCode/WEB\\_DESIGNING · GitHub](#)

# Values

Value	Description
none	This value does not hyphenate the words. It never hyphenates the words at line breaks or even if the word is too long.
manual	It is the default value that hyphenates the word only when the characters in the word suggest hyphenation opportunities. The two Unicode characters are defined below, which can be used manually to specify the possible line breakpoints in the text.
auto	In this value, the algorithm decides where the words are hyphenated.

# Ex.

```
.none{ hyphens: none;  
}
```

```
.manual{ hyphens: manual;  
}
```

```
.auto{ hyphens: auto;  
}
```

# CSS transform-origin property

This CSS property is used to change the position of transformed elements. It is a point around which the transformation is applied. It establishes the origin of the transformation of an element. It can be applied to both 2D and 3D rotations.

The transform-origin property must be used with the transform property. **The 2d transformation can change the x-axis and y-axis of the element, whereas the 3D transformation can change the z-axis along with the x-axis and y-axis.**

This property can be specified by using one, two, or three values. The first value affects the horizontal position, the second value affects the vertical position, and the third value indicates the position of the z-axis. The third value can also work on 3D transformations and cannot be defined using a percentage.

---

**Github Link**

[https://github.com/TopsCode/WEB\\_DESIGNING/blob/main/Module-2/CSS-CSS3/28\\_Transform.html](https://github.com/TopsCode/WEB_DESIGNING/blob/main/Module-2/CSS-CSS3/28_Transform.html)

# Ex.

## Syntax:

transform-origin: x-axis y-axis z-axis

Ex:

for 2D Transformation: transform: rotate(35deg); transform-origin: 5% 2%;

for 3D Transformation:

transform: rotate3d(3, 2, 1, 75deg);  
transform-origin: 70% 10% 150px;

# CSS Resize property

This CSS property allows the user to control the resizing of an element just by clicking or dragging the bottom right corner of the element

This CSS property is used to define how an element is resizable by the user. It doesn't apply on the block or inline elements where overflow is set to visible. So, to control the resizing of an element, we have to set the overflow other than visible like (overflow: hidden or overflow: scroll).

## Resize:

resize: none | horizontal | vertical | both

[WEB DESIGNING/Module-2/CSS-CSS3/Resize.html at  
main - TopsCode/WEB DESIGNING - GitHub](#)

# Values

Value	Descriptions
none	It is the default value of this property, which does not allow resizing the element.
horizontal	This value allows the user to resize the element's width. It resizes the element in a horizontal direction. There is a unidirectional horizontal mechanism for controlling the width of an element.
vertical	It allows the user to resize the height of an element. It resizes the element in a vertical direction. There is a unidirectional vertical mechanism for controlling the height of an element.
both	It allows the user to resize the width and height of an element. It resizes the element in both horizontal and vertical

# Ex.

```
div{  
    resize: none | horizontal | vertical |  
    both;  
}
```

# CSS Transition Delay Property

This CSS property specifies the duration to start the transition effect. The value of this property is defined as either **seconds (s)** or **milliseconds (ms)**. The CSS transitions are effects that are added to change the element gradually from one style to another, without using flash or JavaScript.

Without using the **transition-delay**, the animation will start with the hover on the element, but using this CSS property, we can delay the animation by an amount of time.

The default value of the transition-delay property is 0, which means that the transition will start to occur immediately without any delay.

## Syntax:

transition-delay: time

[WEB DESIGNING/Module-2/CSS-CSS3/Transitiondelay.html at main · TopsCode/WEB DESIGNING - GitHub](#)

# Values

Value	Description
time	It specifies the amount of time (in seconds or milliseconds) to wait before the transition starts.
initial	It sets this property to its default value.
inherit	It inherits this property from its parent element.

# Note

The negative value of the transition-delay property will immediately start the transition effect i.e., the effect will be animated as though it had already begun. The positive value of this property causes the transition effect to start for the given time.

# Ex. 1

- Change the color of the div on hover

```
div{  
width: 100px;  
height:  
100px;  
background: lightblue;  
transition-property: background-  
color; transition-duration: 1s;  
transition-delay: 0.5s;
```

```
/* For Safari browser */  
-webkit-transition-property:  
background-color;  
-webkit-transition-duration: 1s;  
-webkit-transition-delay: 0.5s;  
}  
div:hover{  
background-color: brown;
```

# Ex. 2

- Change the width of div on hover

```
.first{  
width: 150px;  
height: 150px;  
background-color: lightblue;  
transition-property: width;  
transition-duration: 1s;  
transition-delay: initial;
```

```
.first:hover  
{ width:  
300px;  
}
```

# Ex. 3

- Rotate the div on hover

```
padding:15px;  
width: 200px;  
height: 200px;  
  
background: lightgreen;  
transition: background-color  
    1s, width 2s, height 2s,  
transform 2s; transition-delay: 1.5ms;
```

```
div:hover  
{ width:  
300px;  
height:  
300px;  
  
-webkit-transform:  
rotate(360deg);  
/* Chrome, Safari, Opera  
*/ transform:  
rotate(360deg);  
background-color: orange;
```

# Table-layout

The table-layout property in CSS is used to control the layout of a table, specifically how its columns and rows are sized. It defines the algorithm the browser uses to calculate the table's width and the width of its columns.

table-layout: auto | fixed;

[WEB DESIGNING/Module-2/CSS-CSS3/Table layout.html at main · TopsCode/WEB DESIGNING · GitHub](#)

# CSS Animation

CSS Animation property is used to create animation on the webpage. It can be used as a replacement of animation created by Flash and JavaScript.

An animation makes an element change gradually from one style to another. You can add as many as properties you want to add. You can also specify the changes in percentage. 0% specify the start of the animation and 100% specify its completion.

## **CSS3 @keyframes Rule:**

The animation is created in the @keyframe rule. It is used to control the intermediate steps in a CSS animation sequence.

[https://github.com/TopsCode/WEB\\_DESIGNING/blob/main/Module-2/CSS-CSS3/30animation.html](https://github.com/TopsCode/WEB_DESIGNING/blob/main/Module-2/CSS-CSS3/30animation.html)

# How does it work?

When the animation is created in the @keyframe rule, it must be bound with selector; otherwise the animation will have no effect.

The animation could be bound to the selector by specifying at least these two properties:

- The name of the animation
- The duration of the animation

**Github Link**

[https://github.com/TopsCode/WEB\\_DESIGNING/blob/main/Module-2/CSS-CSS3/30animation.html](https://github.com/TopsCode/WEB_DESIGNING/blob/main/Module-2/CSS-CSS3/30animation.html)

# Animation

Property	Description
@keyframes	It is used to specify the animation.
animation	This is a shorthand property, used for setting all the properties, except the animation-play-state and the animation-fill-mode property.
animation-delay	It specifies when the animation will start.
animation-direction	It specifies if or not the animation should play in reserve on alternate cycle.
animation-duration	It specifies the time duration taken by the animation to complete one cycle.
animation-fill-mode	<b>it</b> specifies the static style of the element. (when the animation is not playing)
animation-iteration-count	<b>It</b> specifies the number of times the animation should be played.
animation-play-state	It specifies if the animation is running or paused.
animation-name	It specifies the name of @keyframes animation.
animation-timing-function	It specifies the speed curve of the animation.

# Ex. 1 with ‘from’ and ‘to’

- change background color of rectangle from RED to BLACK.

```
div {  
    width: 100px;  
    height: 100px;  
    background:  
        red;  
    -webkit-animation: myfirst 6s; /* Chrome, Safari, Opera  
    */ animation: myfirst 5s;  
}
```

```
/* Chrome, Safari, Opera */  
@-webkit-keyframes myfirst  
{  
    from {background:  
        red;} to {background:  
        green;}  
}  
/* Standard syntax  
 */ @keyframes  
myfirst {  
    from {background:  
        red;} to {background:  
        green;}  
}
```

# Ex. 2 with percentage values

- Moving Rectangle

```
div {  
    width: 100px;  
    height: 100px;  
    background: red;  
    position:  
    relative;  
    -webkit-animation: myfirst  
    5s;  
/* Chrome, Safari, Opera  
 */ animation: myfirst 5s;  
}
```

```
@keyframes myfirst {  
    0% {background:red; left:0px;  
    top:0px;} 25% {background:yellow;  
    left:300px;  
    top:0px;}  
    50% {background:blue; left:300px;  
    top:200px;}  
    75% {background:green; left:0px;  
    top:200px;}  
    100% {background:red; left:0px;  
    top:0px;}  
}
```

# CSS @keyframes rule

The CSS @keyframe specifies the animation rule that defines the CSS properties for the elements at each state with a timeline.

We can create complex animation properties by using the @keyframe. An animation is created with the changeable CSS styles that can be repeated indefinitely or a finite number of times. A simple animation can just have two keyframes, while the complex animation has several keyframes.

To use keyframes, we need to create a @keyframes rule along with the animation- name property for matching an animation with its keyframe declaration.

**Github Link**

[https://github.com/TopsCode/WEB\\_DESIGNING/blob/main/Module-2/CSS-CSS3/29\\_Animation.html](https://github.com/TopsCode/WEB_DESIGNING/blob/main/Module-2/CSS-CSS3/29_Animation.html)

# Property Values

Values	Description
animation-name	It is the required value that defines the name of the animation.
keyframes-selector	It specifies the percentage along with the animation when the keyframe occurs. Its value lies between 0% to 100%, from (same as 0%), to (same as 100%). Keyframe percentages can be written in any order because they will be handled in the order they occur.
css-styles	It defines one or more than one CSS style properties.

# @keyframe timings

Values	Description
linear	It means that the transition rate will be constant from start to end.
ease	It means that the animation starts slowly, and then after a time period, the speed increases, and before end speed will again slow down.
ease-in	It is similar to ease, but the animation ends quickly.
ease-out	It is also similar to ease, but the animation starts fast.

# Ex.

```
div
{
width:200px;
height:200px;
animation:demo 5s ease-in
infinite, trans 5s ease-in-out
infinite; border-radius:40px;
}
```

```
@keyframes demo
{
  0% {background:red;}
  50% {background:yellow;
        width:100px; height:100px;}
  100% {background:green; width:300px;
         height:300px;}
}

@keyframes trans
{
  0%{transform:translate(0px)scale(1.4)
      rotate(80deg);}
  50%{transform:translate(250px)scale(1.2)
       rotate(40deg);}
}
```

# CSS Gradient and its uses

CSS gradient is used to display smooth transition within two or more specified colors.

- You don't have to use images to display transition effects.
- The download time and bandwidth usage can also be reduced.
- It provides better look to the element when zoomed, because the gradient is generated by the browser.

There are two types of gradients in CSS3

1. Linear gradients
2. Radial gradients

[Github Link](#)

[https://github.com/TopsCode/WEB\\_DESIGNING/blob/main/  
Module-2/CSS-CSS3/09background-gradient-image.html](https://github.com/TopsCode/WEB_DESIGNING/blob/main/Module-2/CSS-CSS3/09background-gradient-image.html)

# CSS Linear Gradient

The CSS3 linear gradient goes up/down/left/right and diagonally. To create a CSS3 linear gradient, you must have to define two or more color stops. The color stops are the colors which are used to create a smooth transition. Starting point and direction can also be added along with the gradient effect.

## Syntax:

```
background: linear-gradient (direction, color-stop1, color-stop2. ... );
```

## Github Link

[https://github.com/TopsCode/WEB\\_DESIGNING/blob/main/Module-2/CSS-CSS3/09background-gradient-image.html](https://github.com/TopsCode/WEB_DESIGNING/blob/main/Module-2/CSS-CSS3/09background-gradient-image.html)

# CSS Linear Gradient: (Top to Bottom)

Top to Bottom Linear Gradient is the default linear gradient. Let's take an example of linear gradient that starts from top. It starts red and transitions to green.

**Ex.**

```
#grad1 {  
    height: 100px;  
  
    background: -webkit-linear-gradient(red, green); /* For Safari 5.1 to 6.0  
   */ background: -o-linear-gradient(red, green); /* For Opera 11.1 to 12.0  
   */ background: -moz-linear-gradient(red, green); /* For Firefox 3.6 to 15  
   */ background: linear-gradient(red, green); /* Standard syntax (must be  
last) */}
```

[WEB DESIGNING/Module-2/CSS-CSS3/linear\\_gradient\\_top\\_to\\_bottom.html at main · TopsCode/WEB DESIGNING · GitHub](#)

# CSS Linear Gradient: (Left to Right)

The following example shows the linear gradient that starts from left and goes to right. It starts red from left side and transitioning to green.

**Ex.**

```
#grad1 {  
    height: 100px;  
    background: -webkit-linear-gradient(left, red, green); /* For Safari 5.1 to 6.0 */  
    /*  
    background: -o-linear-gradient(right, red, green); /* For Opera 11.1 to 12.0 */  
    background: -moz-linear-gradient(right, red, green); /* For Firefox 3.6 to 15 */  
    background: linear-gradient(to right, red , green); /* Standard syntax (must be last) */  
}
```

[WEB DESIGNING/Module-2/CSS-CSS3/linear\\_gradient\\_left\\_to\\_right.html](#)  
[at main · TopsCode/WEB DESIGNING · GitHub](#)

# CSS Linear Gradient: (Diagonal)

If you specify both the horizontal and vertical starting positions, you can make a linear gradient diagonal.

**Ex.**

```
#grad1 {  
    height: 100px;  
    background: -webkit-linear-gradient(left top, red , green); /* For Safari 5.1 to 6.0 */  
    background: -o-linear-gradient(bottom right, red, green); /* For Opera 11.1 to 12.0 */  
    background: -moz-linear-gradient(bottom right, red, green); /* For Firefox 3.6 to 15 */  
    background: linear-gradient(to bottom right, red , green); /* Standard syntax (must be last)  
*/  
}
```

[WEB DESIGNING/Module-2/CSS-CSS3/linear\\_gradient\\_diagonal.html at  
main · TopsCode/WEB DESIGNING · GitHub](#)

# CSS Radial Gradient

You must have to define at least two color stops to create a radial gradient.  
It is defined by its center.

## Syntax:

```
background: radial-gradient(shape size at position, start-color, ..., last-color);
```

## [Github Link](#)

[https://github.com/TopsCode/WEB\\_DESIGNING/blob/main/Module-2/CSS-CSS3/06background-gradient.html](https://github.com/TopsCode/WEB_DESIGNING/blob/main/Module-2/CSS-CSS3/06background-gradient.html)

# CSS Radial Gradient: (Evenly Spaced Color Stops)

Evenly spaced color stops is a by default radial gradient. Its by default shape is eclipse, size is farthest- carner, and position is center.

**Ex.**

```
#grad1 {  
    height: 150px;  
    width: 200px;  
    background: -webkit-radial-gradient(blue, green, red); /* Safari 5.1 to 6.0 */  
    background: -o-radial-gradient(blue, green, red); /* For Opera 11.6 to 12.0 */  
    background: -moz-radial-gradient(blue, green, red); /* For Firefox 3.6 to 15 */  
    background: radial-gradient(blue, green, red); /* Standard syntax (must be  
    last) */
```

[WEB DESIGNING/Module-2/CSS-](#)

[CSS3/Radial Gradient Evenly Spaced Color Stops.html at main .](#)

[TopsCode/WEB DESIGNING - GitHub](#)

# Radial Gradient: (Differently Spaced Color Stops)

Ex.

```
#grad1 {  
    height: 150px;  
    width: 200px;  
    background: -webkit-radial-gradient(blue 5%, green 15%, red 60%); /* Safari 5.1 to 6.0 */  
    background: -o-radial-gradient(blue 5%, green 15%, red 60%); /* For Opera 11.6 to 12.0 */  
    background: -moz-radial-gradient(blue 5%, green 15%, red 60%); /* For Firefox 3.6 to 15 */  
    background: radial-gradient(blue 5%, green 15%, red 60%); /* Standard syntax (must be last) */  
}
```

[WEB DESIGNING/Module-2/CSS-CSS3/Radial gradient Differently Spaced Color Stops.html at main · TopsCode/WEB DESIGNING · GitHub](#)

# Radial Gradient: (From Corners)

Ex.

```
#grad1 {  
    background-image: radial-gradient(farthest-side at left bottom, red, yellow, cyan );  
}  
  
#grad2{  
    background-image: radial-gradient(farthest-corner at right bottom ,blue, yellow,green);  
}  
  
#grad3{  
    background-image: radial-gradient(closest-side , red, yellow, lightgreen);  
}  
  
#grad4{  
    background-image: radial-gradient(closest-corner , blue, yellow, green);  
}
```

# z-index

The z-index in CSS allows us to define a visual hierarchy on the 3-dimensional plane, i.e., the z-axis. It is used to specify the stacking order of the positioned elements (elements whose position value is either fixed, absolute, relative, or sticky). The stacking order means that the element's position along the z-axis, which is perpendicular to the screen.

Syntax:

z-index: number | auto

**Github Link**

[https://github.com/TopsCode/WEB DESIGNING/blob/main/Module-2/CSS-CSS3/36absolute\\_z-index.html](https://github.com/TopsCode/WEB DESIGNING/blob/main/Module-2/CSS-CSS3/36absolute_z-index.html)

# Ex.

```
div{  
    position:fixed;  
    z-index:20;  
}  
  
img{  
    position:relative;  
    z-index:22;  
}  
  
h1{  
    position:relative;  
    z-index:28;  
}
```

# CSS Masking

The mask property in CSS is used to hide an element using the clipping or masking the image at specific points. Masking defines how to use an image or the graphical element as a luminance or alpha mask. It is a graphical operation that can fully or partially hide the portions of an element or object.

Using masking, it is possible to show or hide the parts of an image with different opacity levels. In CSS, the masking is achieved by using the mask-image property, and we have to provide an image as the mask.

[WEB DESIGNING/Module-2/CSS-CSS3/Masking.html](#)  
[at main · TopsCode/WEB DESIGNING · GitHub](#)

# Ex.

## Ex. 1: Mask

```
#mask{  
-webkit-mask-box-image: url(box.png);  
}
```

## Ex. 2: Gradient

```
#mask{  
-webkit-mask-image: -webkit-gradient(linear, right top, right  
bottom, from(rgba(0,0,0,0)), to(rgba(0,0,0,0.9))); |  
-webkit-mask-image: radial-gradient(circle at 50% 50%, blue 40%,  
rgba(0,0,0,0.3) 70%);  
border: 9px ridge red;  
}
```

# What is Media Query?

CSS Media query is a W3C recommendation and a CSS3 module which is used to adapt to conditions such as screen resolution (e.g. Smartphone screen vs. computer screen).

- The media query technique first used in CSS3.
- It became a W3C recommendation in June 2012.
- It is an extension of media dependent stylesheets used in different media types (i.e. screen and print) found in CSS2.
- The most commonly used media feature is "width".
- It uses the @media rule to include a block of CSS properties only if a certain condition is true.[Github Link](#)

# What is Responsive Web Design?

It facilitates you to use fluid grids, flexible images, and media queries to progressively enhance a web page for different viewing contexts i.e. Desktop, Smartphone, Tablet etc.



# different screen resolutions

Smartphone: 320px

Tablet: 768px

Netbook: 1024px

Desktop: 1600px

Large Desktop: 1920px

**Github Link**

[https://github.com/TopsCode/WEB\\_DESIGNING/blob/main/Module-2/CSS-CSS3/45basic\\_media\\_query.html](https://github.com/TopsCode/WEB_DESIGNING/blob/main/Module-2/CSS-CSS3/45basic_media_query.html)

# Ex.

```
<meta name="viewport" content="width=device-width, initial-scale=1.0"/>
<style>
body {
    background-color:yellow;
}

@media only screen and (max-width: 500px)
{
    body {
        background-color:green;
    }
}
</style>
```

# Clip Path

In CSS, the `clip-path` property is used to create a clipping region for an element. This allows you to visually "cut out" a part of the element and hide the rest. You can clip an element to any shape, including basic geometric shapes (like circles, polygons, and rectangles) or even custom paths.

# CSS Transforms

CSS3 supports transform property. This transform property facilitates you to translate, rotate, scale, and skews elements.

Transformation is an effect that is used to change shape, size and position.

There are two type of transformation i.e. 2D and 3D transformation supported in CSS3.

**Ex.:translate() rotate() scale() skewX() skewY() skew() matrix()**

**GitHub Link**

[https://github.com/TopsCode/WEB DESIGNING/blob/main/Module-2/CSS-CSS3/23transform\\_translate.html](https://github.com/TopsCode/WEB DESIGNING/blob/main/Module-2/CSS-CSS3/23transform_translate.html)

# translate()

The CSS translate() method is used to move an element from its current position according to the given parameters i.e. X-axis and Y-axis.

**Ex:**

```
transform: translate(100px,80px);
```

**Github Link**

[https://github.com/TopsCode/WEB\\_DESIGNING/blob/main/Module-2/CSS-CSS3/23transform\\_translate.html](https://github.com/TopsCode/WEB_DESIGNING/blob/main/Module-2/CSS-CSS3/23transform_translate.html)

# rotate()

The CSS rotate() method is used to rotate an element clockwise or anti-clockwise according to the given degree.

**Ex:**

```
-ms-transform: rotate(30deg);  
-webkit-transform:  
rotate(30deg); transform:  
rotate(30deg);
```

**Github Link**

[https://github.com/TopsCode/WEB\\_DESIGNING/blob/main/Module-2/CSS-CSS3/24transform\\_rotate.html](https://github.com/TopsCode/WEB_DESIGNING/blob/main/Module-2/CSS-CSS3/24transform_rotate.html)

# scale()

The CSS scale() method is used to increase or decrease the size of an element according to the given width and height.

**Ex:**

```
transform: scale(2.5,2);
```

**GitHub Link**

[https://github.com/TopsCode/WEB\\_DESIGNING/blob/main/Module-2/CSS-CSS3/25transform\\_scale.html](https://github.com/TopsCode/WEB_DESIGNING/blob/main/Module-2/CSS-CSS3/25transform_scale.html)

# skewX()

The CSS skewX() method is used to skew an element along the X axis according to the given angle.

**Ex:**

```
transform: skewX(30deg);
```

# skewY()

The CSS skewY() method is used to skew an element along the Y axis according to the given angle.

**Ex:**

```
transform: skewY(30deg);
```

[WEB DESIGNING/Module-2/CSS-CSS3/SkewY.html at  
main · TopsCode/WEB\\_DESIGNING · GitHub](#)

# skew()

The CSS skew() method is used to skew an element along with X-axis and Y- axis according to the given angle.

**Ex:**

```
transform: skew(30deg,20deg);
```

# matrix()

The CSS matrix() method combines all the six 2D transform methods altogether.

## Syntax:

matrix(scaleX(),skewY(),skewX(),scaleY(),translateX(),translateY())

## Ex:

transform: matrix(1, -0.3, 0, 1, 0, 0);

[WEB DESIGNING/Module-2/CSS-CSS3/Matrix.html at  
main · TopsCode/WEB DESIGNING · GitHub](#)

# The 3D rotateX() method (X-axis rotation)

The CSS 3D rotateX() method is used to rotate an element around its X- axis according to the given degree.

**Ex.:**

```
transform: rotateX(150deg); /* Standard syntax */
```

# The 3D rotateY() method (Y-axis rotation)

The CSS 3D rotateY() method is used to rotate an element around its Y-axis according to the given degree.

**Ex.:**

```
transform: rotateY(150deg); /* Standard syntax */
```

# The 3D rotateZ() method (Z-axis rotation)

The CSS 3D rotateZ() method is used to rotate an element around its Z- axis according to the given degree.

**Ex.:**

```
transform: rotateZ(90deg); /* Standard syntax */
```

# Hosting a website with free domain name

Hosting a website with a free domain name is a good way to get started with web development or to create a simple website without incurring any costs. While free hosting services may have limitations (like limited storage, bandwidth, or advertisements), they are perfect for personal projects, portfolios, or learning purposes

- a. GitHub
- b. Netlify
- c. InfinityFree
- d. Webhost

# **Module - 4**

## **[ HTML- 5 ]**

# HTML Audio Tag

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

- mp3
- wav
- ogg

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

# Ex.

```
<audio controls>
  <source src="koyal.mp3"
  type="audio/mpeg"> Your browser does
not support the html audio tag.
</audio>
```

GitHub link : [https://github.com/TopsCode/WEB\\_DESIGNING/blob/main/Module-3/HTML5/01audio.html](https://github.com/TopsCode/WEB_DESIGNING/blob/main/Module-3/HTML5/01audio.html)

# HTML Video Tag

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:

- mp4
- webM
- ogg

# Ex.

```
<video controls>
```

```
  <source src="movie.mp4"
```



Your browser does not support the html video  
tag.

```
</video>
```

GitHub link : [https://github.com/TopsCode/WEB\\_DESIGNING/blob/main/Module-3/HTML5/02video.html](https://github.com/TopsCode/WEB_DESIGNING/blob/main/Module-3/HTML5/02video.html)

# Video Attributes

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

# What is SVG?

The HTML SVG is an acronym which stands for Scalable Vector Graphics.

HTML SVG is a modularized language which is used to describe graphics in XML. It describes two-dimensional vector and mixed vector/raster graphics in XML.

SVG is mostly used for vector type diagrams like pie charts, 2-Dimensional graphs in an X,Y coordinate system etc.

**Github link :** [https://github.com/TopsCode/WEB\\_DESIGNING/blob/main/Module-3/HTML5/06\\_SVG%20\(Scalable%20Vector%20Graphics\).html](https://github.com/TopsCode/WEB_DESIGNING/blob/main/Module-3/HTML5/06_SVG%20(Scalable%20Vector%20Graphics).html)

# Ex. Circle

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

cx, cy and r are attributes of circle tag. These attributes can't be used with  
svg other tag.

# Ex. Ellipse

```
<svg height="140" width="500">  
  <ellipse cx="200" cy="80" rx="70" ry="50"  
  style="fill:yellow;stroke:purple;stroke-width:2" />  
Sorry, your browser does not support inline SVG.  
</svg>
```

The cx attribute defines the x coordinate of the center of the ellipse  
The cy attribute defines the y coordinate of the center of the ellipse  
The rx attribute defines the horizontal radius  
The ry attribute defines the vertical radius

# Ex. Rectangle

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

cx, cy and r are attributes of circle tag. These attributes can't be used with  
svg other tag.

[WEB DESIGNING/Module-3/HTML5/Rectangle.html at  
main · TopsCode/WEB DESIGNING · GitHub](#)

# SVG Line

**Ex.**

```
<svg height="210" width="500">
  <line x1="0" y1="0" x2="200" y2="200" style="stroke:blue;stroke-width:2" />
</svg>
```

[WEB DESIGNING/Module-3/HTML5/Svg\\_line.html at  
main · TopsCode/WEB DESIGNING · GitHub](#)

# SVG Polygon

The <polygon> element is used to create a graphic that contains at least three sides.

Polygons are made of straight lines, and the shape is "closed" (all the lines connect up).

**Ex.**

```
<svg height="210" width="500">  
  <polygon points="200,10 250,190 160,210"  
  style="fill:lime;stroke:purple;stroke-width:1" />  
</svg>
```

# SVG Polyline

The <polyline> element is used to create any shape that consists of only straight lines (that is connected at several points)

**Ex.**

```
<svg height="200" width="500">
  <polyline points="20,20 40,25 60,40 80,120 120,140 200,180"
    style="fill:none;stroke:black;stroke-width:3" />
</svg>
```

[WEB DESIGNING/Module-3/HTML5/Svg\\_polygon.html](#)  
[at main · TopsCode/WEB DESIGNING · GitHub](#)

# SVG Path

The <path> element is used to define a path.

The following commands are available for path

data: M = moveto L = lineto

H = horizontal lineto V = vertical lineto

Z = closepath C = curveto

S = smooth curveto

**Github link : [https://github.com/TopsCode/WEB\\_DESIGNING/blob/main/Module-3/HTML5/06\\_SVG%20\(Scalable%20Vector%20Graphics\).html](https://github.com/TopsCode/WEB_DESIGNING/blob/main/Module-3/HTML5/06_SVG%20(Scalable%20Vector%20Graphics).html)**

# Ex. Path

Ex.

```
<svg height="210" width="400">
  <path d="M150 0 L75 200 L225 200 Z" />
</svg>
```

[https://github.com/TopsCode/WEB\\_DESIGNING/blob/main/Module-3/HTML5/Svg\\_path.html](https://github.com/TopsCode/WEB_DESIGNING/blob/main/Module-3/HTML5/Svg_path.html)

# SVG Text

The <text> element is used to define a text.

**Ex.**

```
<svg height="30" width="200">
  <text x="0" y="15" fill="red">I love SVG!</text>
</svg>
```

```
<svg height="60" width="200">
  <text x="0" y="15" fill="red" transform="rotate(30 20,40)">
    I love SVG</text>
</svg>
```

[WEB DESIGNING/Module-3/HTML5/Svg\\_text.html at  
main · TopsCode/WEB DESIGNING · GitHub](#)

# SVG Stroke

SVG offers a wide range of stroke properties. In this chapter we will look at the following:

- stroke
- stroke-width
- stroke-linecap
- stroke-dasharray

[WEB DESIGNING/Module-3/HTML5/Svg\\_Stroke.html](#)  
[at main · TopsCode/WEB DESIGNING · GitHub](#)

# SVG Stroke Ex.

```
<svg height="80" width="300">
  <g fill="none">
    <path stroke="red" d="M5 20 l215 0" />
    <path stroke="black" d="M5 40 l215 0" />
    <path stroke="blue" d="M5 60 l215 0" />
  </g>
</svg>
```

# SVG Gradient

A gradient is a smooth transition from one color to another. In addition, several color transitions can be applied to the same element.

There are two main types of gradients in SVG:

- Linear
- Radial

# Linear Gradient

The <linearGradient> element is used to define a linear gradient.

The <linearGradient> element must be nested within a <defs> tag. The <defs> tag is short for definitions and contains definition of special elements (such as gradients).

Linear gradients can be defined as horizontal, vertical or angular gradients:

Horizontal gradients are created when y1 and y2 are equal and x1 and x2 differ  
Vertical gradients are created when x1 and x2 are equal and y1 and y2 differ  
Angular gradients are created when x1 and x2 differ and y1 and y2 differ

[WEB DESIGNING/Module-3/HTML5/Svg\\_linear\\_Gradient.html at main · TopsCode/WEB DESIGNING - GitHub](#)

# Ex. Linear Gradient

```
<svg height="150" width="400">
<defs>
  <linearGradient id="grad1" x1="0%" y1="0%" x2="100%" y2="0%">
    <stop offset="0%" style="stop-color:rgb(255,255,0);stop-opacity:1" />
    <stop offset="100%" style="stop-color:rgb(255,0,0);stop-opacity:1" />
  </linearGradient>
</defs>
<ellipse cx="200" cy="70" rx="85" ry="55" fill="url(#grad1)" />
</svg>
```

# Radial Gradient

The <radialGradient> element is used to define a radial gradient.

**Ex.**

```
<svg height="150" width="500">
  <defs>
    <radialGradient id="grad1" cx="50%" cy="50%" r="50%" fx="50%"
      fy="50%">
      <stop offset="0%" style="stop-
color:rgb(255,255,255); stop-opacity:0" />
      <stop offset="100%" style="stop-color:rgb(0,0,255);stop-opacity:1" />
    </radialGradient>
  </defs>
  <ellipse cx="200" cy="70" rx="85" ry="55" fill="url(#grad1)" />
</svg>
```

[WEB DESIGNING/Module-3/HTML5/Svg\\_radialgradient.html at main · TopsCode/WEB DESIGNING · GitHub](#)

# HTML Canvas

The HTML canvas element provides HTML a bitmapped surface to work with. It is used to draw graphics on the web page.

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

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.

[Github link :https://github.com/TopsCode/WEB\\_DESIGNING/blob/main/Module-3/HTML5/05\\_Canvas.html](https://github.com/TopsCode/WEB_DESIGNING/blob/main/Module-3/HTML5/05_Canvas.html)

# Create a Canvas

```
<canvas id="myCanvas1" width="300" height="100"  
style="border:2px solid;">  
Your browser does not support the HTML5 canvas tag.  
</canvas>
```

# Canvas Tag with Javascript

```
<script>  
var c =  
document.getElementById("myCanvas"); var  
ctx = c.getContext("2d");  
ctx.fillStyle = "#FF0000";  
ctx.fillRect(0,0,200,100);  
</script>
```

# Line on Canvas

```
<canvas id="myCanvasLine" width="200" height="100"  
style="border:1px solid #d3d3d3;">  
Your browser does not support the HTML5 canvas tag.</canvas>  
<script>  
var c = document.getElementById("myCanvasLine"); var  
cctx = c.getContext("2d");  
cctx.moveTo(0,0);  
cctx.lineTo(200,100); cctx.stroke();  
</script>
```

[WEB DESIGNING/Module-3/HTML5/Canvas\\_line.html](#)  
[at main · TopsCode/WEB DESIGNING · GitHub](#)

# Circle on Canvas

```
<canvas id="myCanvasCircle" width="200" height="100"  
style="border:1px solid #d3d3d3;">  
Your browser does not support the HTML5 canvas tag.</canvas>  
<script>  
var c =  
document.getElementById("myCanvasCircle"); var  
cctx = c.getContext("2d");  
cctx.beginPath();  
cctx.arc(95,50,40,0,2*Math.PI)  
; cctx.stroke(); WEB DESIGNING/Module-3/HTML5/Canvas\_circle.html at main · TopsCode/WEB DESIGNING · GitHub
```

# Text on Canvas

## 1. Fill Text

```
<canvas id="myCanvasText1" width="300" height="100"  
style="border:1px solid #d3d3d3;">  
Sorry! Your browser does not support the HTML5 canvas tag.</canvas>  
<script>  
var c =  
document.getElementById("myCanvasText1"); var  
cctx = c.getContext("2d");  
cctx.font = "30px Arial";  
cctx.fillText("Hello
```

[WEB DESIGNING/Module-3/HTML5/Canvas\\_text.html](#)  
[at main · TopsCode/WEB DESIGNING · GitHub](#)

# Text on Canvas

## 2. Stroke Text

```
<canvas id="myCanvasText2" width="300" height="100" style="border:1px solid #d3d3d3;">  
    Sorry! Upgrade your browser. It does not support the HTML5 canvas tag.</canvas>  
<script>  
    var c = document.getElementById("myCanvasText2");  
  
    var cctx = c.getContext("2d");  
  
    ctx.font = "30px Arial";  
  
    ctx.strokeText("Hello JavaTpoint",10,50);  
</script>
```

# What is URL

URL stands for Uniform Resource Locator. It is actually a web address. A URL can contain words i.e. (learnvern.com) or an Internet Protocol (IP) address i.e. 195.201.68.81. But most of the user use URL in the form of words because it is easy to remember than numbers.

## Syntax:

scheme://prefix.domain:port/path/filename

# What is URL

- scheme        is used to define the type of Internet service (most common is http or https).
- prefix        is used to define a domain prefix (default for http is domain www). is used to define the Internet domain name (like lernvern.com).
- port        is used to define the port number at the host (default for http is 80).
- path        is used to define a path at the server (If omitted: the root directory of the site).
- filename        is used to define the name of a document or resource.

# URL Encode

URL encoding is used to convert non-ASCII characters into a format that can be used over the Internet because a URL is sent over the Internet by using the ASCII character-set only. If a URL contains characters outside the ASCII set, the URL has to be converted.

In URL encoding, the non-ASCII characters are replaced with a "%" followed by hexadecimal digits.

URLs cannot contain spaces. URL encoding normally replaces a space with a plus (+) sign, or %20.

Ex. © will be replaced with %C2%A9

# What is XHTML?

XHTML is a stricter, more XML-based version of HTML.

- XHTML stands for EXtensible HyperText Markup Language
- XHTML is a stricter, more XML-based version of HTML
- XHTML is HTML defined as an XML application
- XHTML is supported by all major browsers

XML is a markup language where all documents must be marked up correctly (be "well-formed").

# Difference between HTML and XHTML

1. <!DOCTYPE> is mandatory:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"  
"http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
```

2. The xmlns attribute in <html> is mandatory

```
<html xmlns="http://www.w3.org/1999/xhtml">
```

3. <html>, <head>, <title>, and <body> are mandatory

3. Elements must always be properly nested

3. Elements must always be closed

# Difference between HTML and XHTML

6. Elements must always be in lowercase

```
<body>  
<p>This is a paragraph</p>  
</body>
```

7. Attribute names must always be in lowercase

```
<a href="https://learnvern.com">Visit our HTML tutorial</a>
```

8. Attribute values must always be quoted

```
<a href="https://learnvern.com">Visit our HTML tutorial</a>
```

9. Attribute minimization is forbidden

```
<input type="checkbox" name="vehicle" value="car" checked="checked" />
```

# Module - 5 Wordpress ]

# What is Wordpress

- ✓ WordPress is an open-source content management system (CMS) primarily used for creating websites and blogs.
- ✓ It was first released in 2003 by Matt Mullenweg and Mike Little as a blogging platform but has since evolved into a versatile CMS.
- ✓ WordPress is written in PHP and uses a MySQL database.
- ✓ WordPress powers over 40% of all websites on the internet.

# Wordpress.org vs Wordpress.com

## 1) WordPress.org:

- **Self-hosted:** WordPress.org provides free, open-source software that you can download and install on your own web server.
- **Full control:** With WordPress.org, you have complete control over your website. You can customize it however you like, install any themes and plugins, and even modify the underlying code if you have the technical expertise.
- **Customization:** You can use any theme or plugin available, including free and premium options from third-party developers or ones you create yourself.

# Wordpress.org vs Wordpress.com

- **Monetization:** You have the freedom to monetize your website in any way you choose, such as displaying ads, selling products or services, or accepting donations.
- **Maintenance:** You are responsible for maintaining your website, including updates, backups, security, and performance optimization.
- **Cost:** While the WordPress software itself is free, you'll need to pay for web hosting and domain registration. Costs can vary depending on your hosting provider and the features you require.

# Wordpress.org vs Wordpress.com

## 2) WordPress.com:

- **Hosted service:** WordPress.com is a commercial platform that offers hosting and website building services.
- **Limited control:** While WordPress.com allows some customization options, such as selecting from a range of pre-designed themes and installing certain plugins (depending on your plan), you have less control over your website compared to WordPress.org.
- **Customization limitations:** The ability to install third-party themes and plugins is restricted on WordPress.com. You're limited to the themes and plugins available in the WordPress.com repository or those offered in premium plans.

# Wordpress.org vs Wordpress.com

- **Monetization limitations:** Some monetization options, such as displaying ads or using e-commerce features, may require a paid plan on WordPress.com. Additionally, you may be subject to revenue-sharing agreements for ads displayed on your site.
- **Maintenance:** WordPress.com takes care of most maintenance tasks, including updates, backups, security, and performance optimization, although certain advanced features may require manual intervention.
- **Cost:** It offers various plans, including free and paid options. Paid plans provide additional features and customization options, with prices ranging from affordable to more expensive depending on the plan's features and your needs.

# Setting up a local development environment

- **Install a Local Server Environment:**
  - a. You can use software like XAMPP, WampServer, MAMP, or Local by Flywheel.
  - b. These software packages provide everything you need to run a server environment locally, including Apache, MySQL, and PHP.
- **Download WordPress:**
  - a. Go to the official WordPress website ([wordpress.org](https://wordpress.org)) and download the latest version of WordPress.
  - b. Extract the WordPress files into the "htdocs" or "www" directory of your local server environment.

# Setting up a local development environment

- **Create a Database:**
  - a. Access the phpMyAdmin. Create a new MySQL database for your WordPress installation.
  - b. Take note of the database name, username, and password as you'll need them during the WordPress installation process.
- **Configure wp-config.php:**
  - a. Extracted the WordPress files, locate the "wp-config-sample.php" file.
  - b. Rename this file to "wp-config.php".
  - c. Open "wp-config.php" in a text editor and fill in the database details (database name, username, password) you created in the previous step.

# Setting up a local development environment

- **Install WordPress:**

- a. <http://localhost/your-folder-name> (replace "your-folder-name" with the directory where you installed WordPress).
- b. Follow the instructions & enter details such as site title, username, password, and email address.

- **Access Your Local WordPress Site:**

- a. Once the installation is complete, you can log in to the WordPress dashboard by going to <http://localhost/your-folder-name/wp-admin> and entering the username and password you set during installation.

# Domains, Hostings and Security

# Advantages of Wordpress

- **Ease of Use:**
  - a. **User-Friendly Interface:** WordPress offers an intuitive dashboard that allows users to easily manage their website content, themes, and plugins without needing technical expertise.
  - b. **WYSIWYG Editor:** The built-in visual editor (Gutenberg) enables users to create and format content with ease, similar to using a word processor.
- **Flexibility and Customization:**
  - a. **Themes:** WordPress provides a vast library of themes, both free and premium, allowing users to choose a design that suits their brand or style.

# Advantages of Wordpress

- **b. Plugins:** With thousands of plugins available, users can extend WordPress functionality to add features such as contact forms, SEO optimization, e-commerce, and more.
- **Scalability :**
  - WordPress can scale from small blogs to large enterprise websites seamlessly. Its modular architecture and extensive ecosystem of themes and plugins make it adaptable to various website sizes and needs.
- **SEO-Friendly :**
  - WordPress is inherently optimized for search engines, with clean code, semantic markup, and SEO-friendly URLs.

# Domains and ITS Types

1. Educational Domains
2. Internet Domains
3. Business Domains
4. Science and Technology Domains
5. Mathematical Domains
6. Information Technology Service Management (ITSM) Domains
7. Security Domains

# Domain Registration Providers

1. GoDaddy
2. Namecheap
3. Google Domains
4. Bluehost
5. HostGator
6. Domain.com
7. Hover
8. DreamHost

# • Things to Consider Before Buying a Domain Name

1. Relevance and Branding
2. Domain Extension
3. Length and Simplicity
4. Spelling and Pronunciation
5. Keywords
6. Availability on Social Media
7. Legal Considerations
8. Pricing and Renewal Fees

- **Web hosting and Its Service Providers**

- **Types of Web Hosting**

Shared Hosting

Virtual Private Server (VPS) Hosting

Dedicated Hosting

Cloud Hosting

Managed Hosting

WordPress Hosting

Reseller Hosting

- **Web hosting and Its Service Providers**

- **Top Web Hosting Service Providers**

Bluehost

HostGator

SiteGround

A2 Hosting

InMotion Hosting

DreamHost

GoDaddy

# • Choosing Hosting And Its Service Providers

## • Steps

1. Understand Your Hosting Needs
2. Types of Hosting Services
3. Key Features to Look For
4. Research and Compare Providers

# • Connecting Domain Name with Web Hosting Using NameServer

## • Steps

1. Obtain Nameserver Information from Your Hosting Provider
2. Log In to Your Domain Registrar Account
3. Access Domain Management Settings
4. Update the Nameserver Settings
5. Wait for DNS Propagation
6. Verify the Connection

- **Secure Sockets Layer**

Secure Sockets Layer (SSL) is a standard security protocol that establishes an encrypted link between a web server and a web browser.

- **Key Concepts of SSL**

1. Encryption
2. Authentication
3. Data Integrity

- **Secure Sockets Layer**

- **Benefits of SSL**

1. Enhanced Security
2. Improved SEO Rankings
3. Increased Trust and Credibility
4. Compliance with Regulations

- **Creating Email Account with Same Domain**

1. **Choose an Email Hosting Provider**

**Integrated with Web Hosting:** If your web hosting provider offers email services (like Bluehost, SiteGround, etc.), you can use their built-in tools.

**Dedicated Email Hosting:** Providers like Google Workspace (formerly G Suite), Microsoft 365 and others offer robust email solutions.

2. **Domain Verification and Configuration**

**Verification:** Most email hosting services will require you to verify that you own the domain by adding specific DNS records.

# Wordpress Installation and Theme

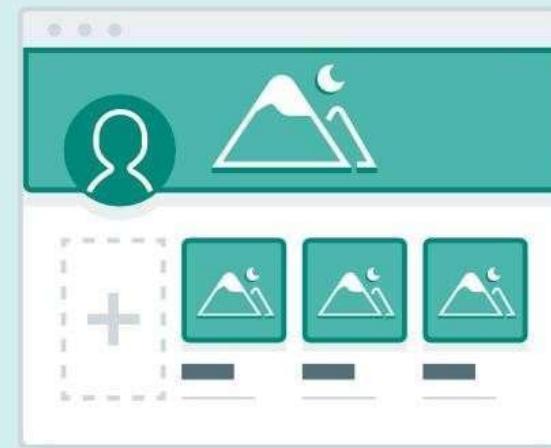
# Wordpress Free and Paid Themes

Free

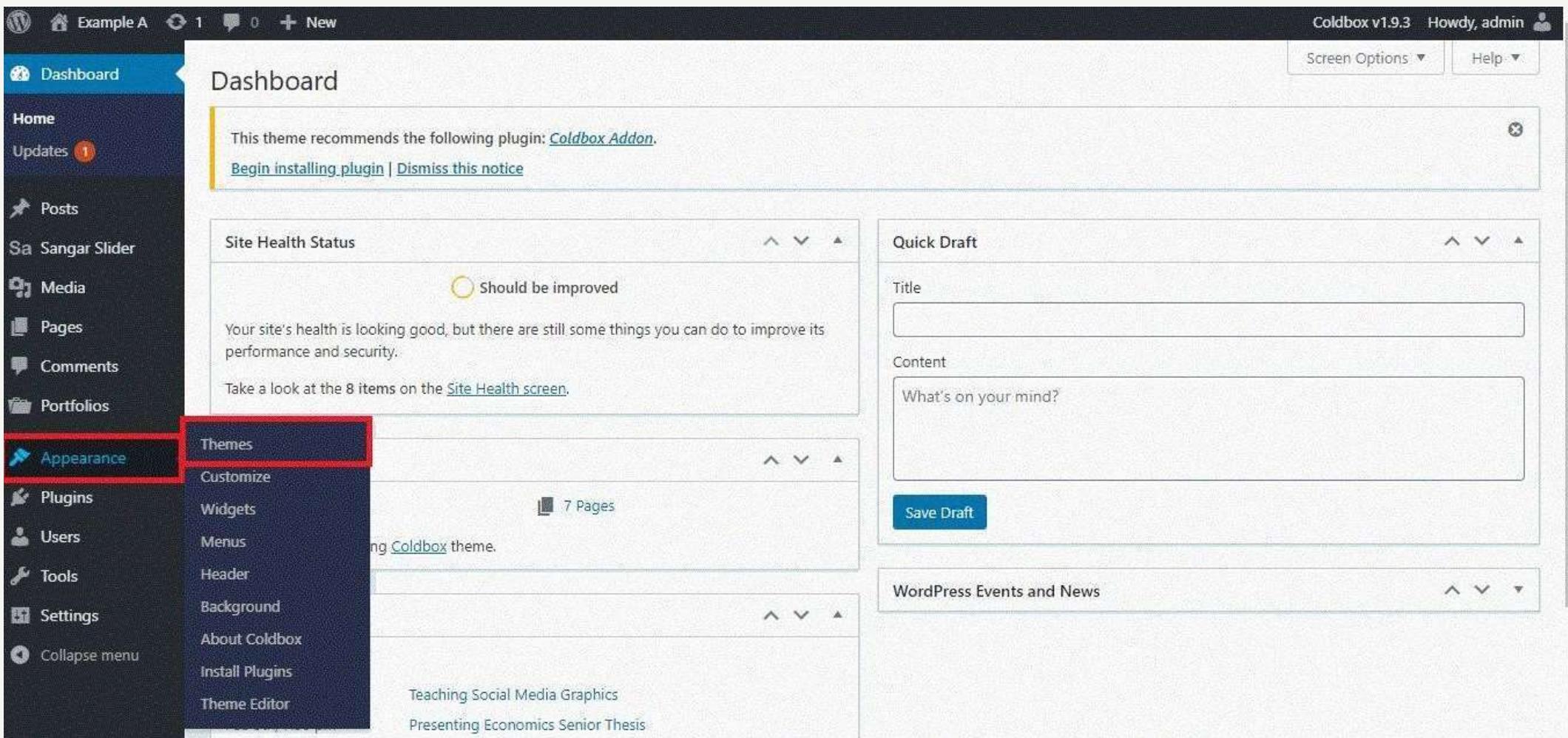


VS

Paid



# How to search and choose a theme



The screenshot shows the WordPress dashboard with a dark blue sidebar and a light gray main content area. The sidebar on the left has a red box around the 'Appearance' section, which is currently selected. Under 'Appearance', a red box highlights the 'Themes' option. Other options in the sidebar include Home, Updates (with 1 update), Posts, Sangar Slider, Media, Pages, Comments, Portfolios, Plugins, Users, Tools, Settings, and Collapse menu.

The main content area displays a 'Dashboard' header with a notice about a recommended plugin: 'This theme recommends the following plugin: [Coldbox Addon](#)'. Below this is the 'Site Health Status' section, which says 'Should be improved' and provides a link to the Site Health screen. To the right is a 'Quick Draft' section with fields for 'Title' and 'Content', and a 'Save Draft' button. At the bottom, there's a 'WordPress Events and News' section.

# Overview of Theme Templates

Example A    1    0    + New

Coldbox v1.9.3    Howdy, admin    Help ▾

Themes 6    Add New    Search installed themes...

This theme recommends the following plugin: [Coldbox Addon](#).  
[Begin installing plugin](#) | [Dismiss this notice](#)

**Coldbox**    Active

**Perfect Portfolio**

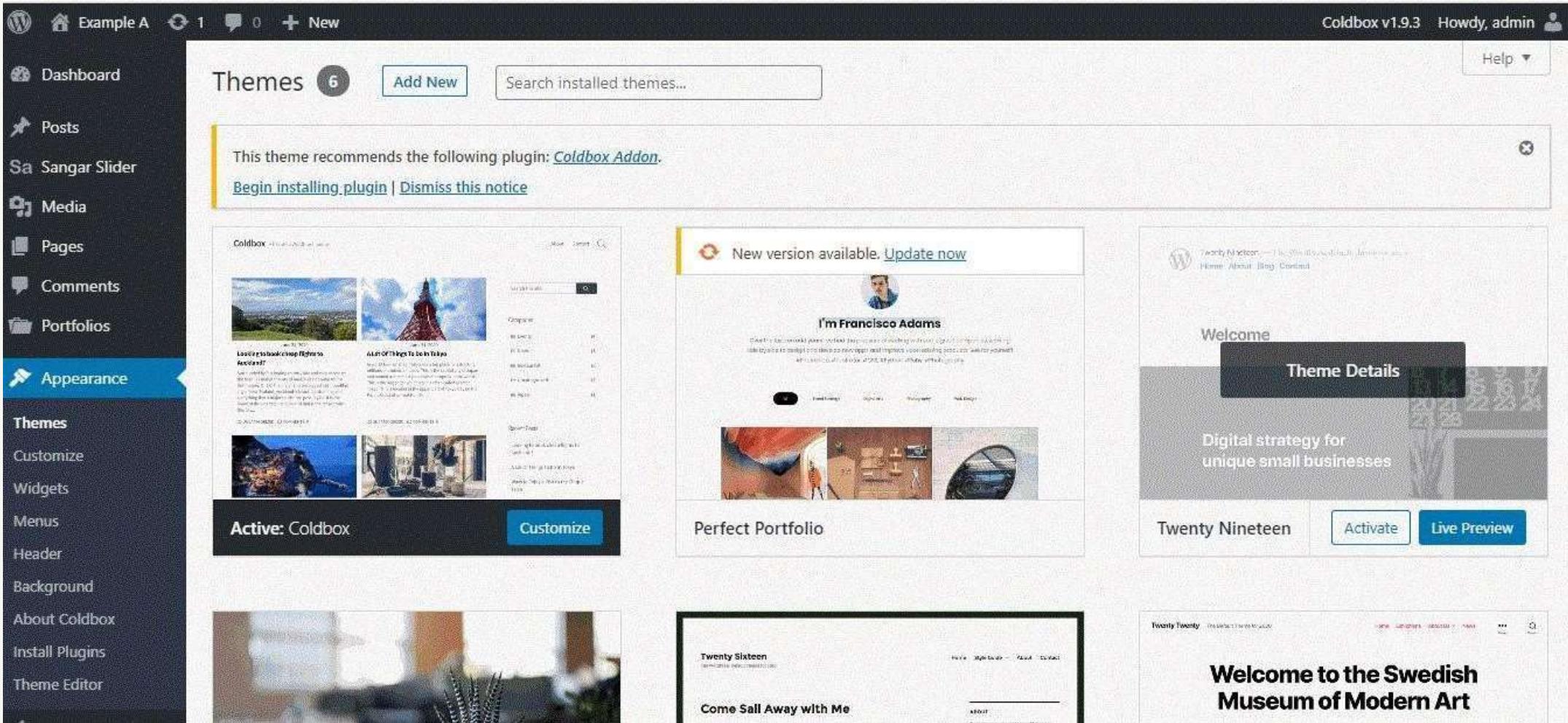
**Twenty Nineteen**    Activate    Live Preview

**Twenty Sixteen**

Come Sell Away with Me

**Twenty Twenty**

Welcome to the Swedish Museum of Modern Art



# Installation of Wordpress

- Download XAMPP



## XAMPP Apache + MariaDB + PHP + Perl

---

### What is XAMPP?

XAMPP is the most popular PHP development environment

XAMPP is a completely free, easy to install Apache distribution containing MariaDB, PHP, and Perl. The XAMPP open source package has been set up to be incredibly easy to install and to use.



**Download**  
Click here for other versions

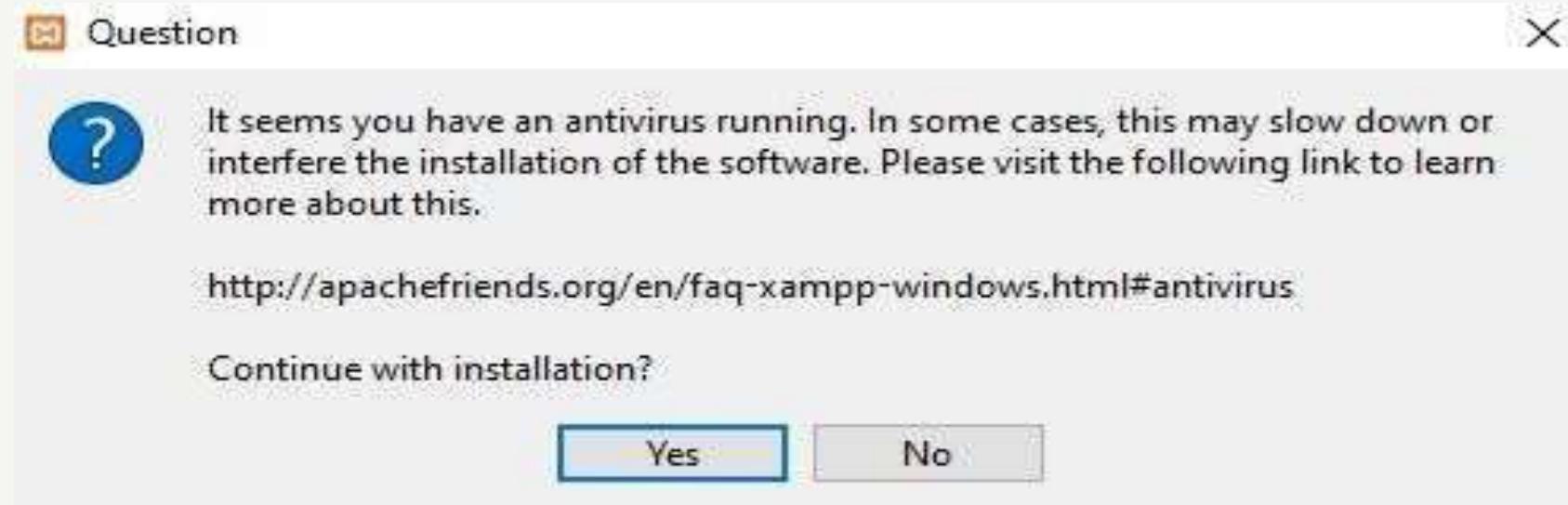
 XAMPP for Windows  
8.1.6 (PHP 8.1.6)

 XAMPP for Linux  
8.1.6 (PHP 8.1.6)

 XAMPP for OS X  
8.1.6 (PHP 8.1.6)

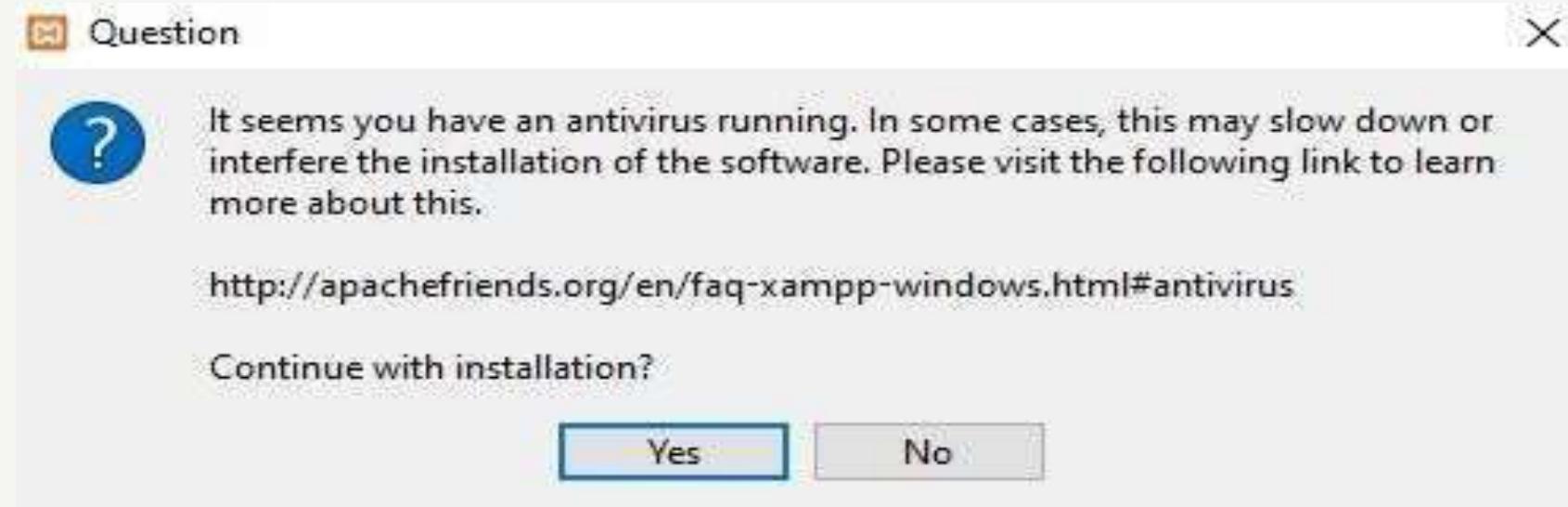
# Installation of Wordpress

- Install and Run XAMPP on Your Computer



# Installation of Wordpress

- **Install and Run XAMPP on Your Computer**



- After installing xampp, run the application and configure the environment. Start Apache and MySQL modules to perform wordpress installation.

# Create a Local Database

phpMyAdmin

Server: 127.0.0.1

Databases SQL Status User accounts Export Import Settings Replication VariablesCharsets Engines More

New information\_schema mysql performance\_schema phpmyadmin test

Create database

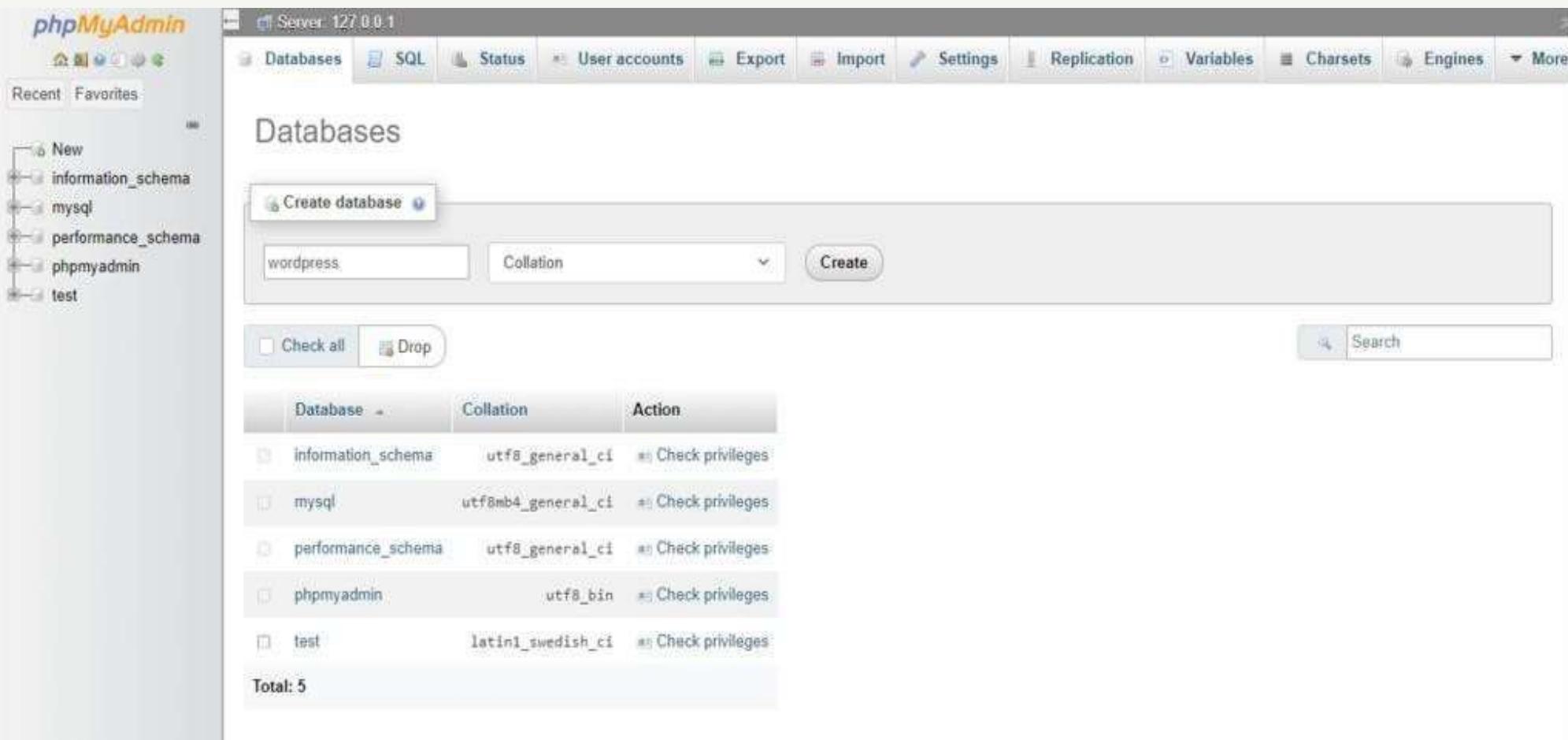
wordpress Collation Create

Check all  Drop

Search

Database	Collation	Action
information_schema	utf8_general_ci	<a href="#">Check privileges</a>
mysql	utf8mb4_general_ci	<a href="#">Check privileges</a>
performance_schema	utf8_general_ci	<a href="#">Check privileges</a>
phpmyadmin	utf8_bin	<a href="#">Check privileges</a>
test	latin1_swedish_ci	<a href="#">Check privileges</a>

Total: 5



## Install WordPress on Localhost

- **Database name** –the name of the database you created in phpMyAdmin.
- **Username** –enter “root” as the default username.
- **Password** –leave the MySQL database password field blank.
- **Database host** –keep the default “localhost.”
- **Table prefix** –keep the default “wp\_.”

## Install WordPress on Localhost



Below you should enter your database connection details. If you are not sure about these, contact your host.

**Database Name**

The name of the database you want to use with WordPress.

**Username**

Your database username.

**Password**

Your database password.

**Database Host**

You should be able to get this info from your web host; if localhost does not work.

**Table Prefix**

If you want to run multiple WordPress installations in a single database, change this.

**Submit**

# Install WordPress on Localhost

## Welcome

Welcome to the famous five-minute WordPress installation process! Just fill in the information below and you'll be on your way to using the most extendable and powerful personal publishing platform in the world.

## Information needed

Please provide the following information. Do not worry, you can always change these settings later.

Site Title

Username

Usernames can have only alphanumeric characters, spaces, underscores, hyphens, periods, and the @ symbol.

Password

 O0j)!D&8wg\$Wx)tXYa  
Strong

 Hide

**Important:** You will need this password to log in. Please store it in a secure location.

Your Email

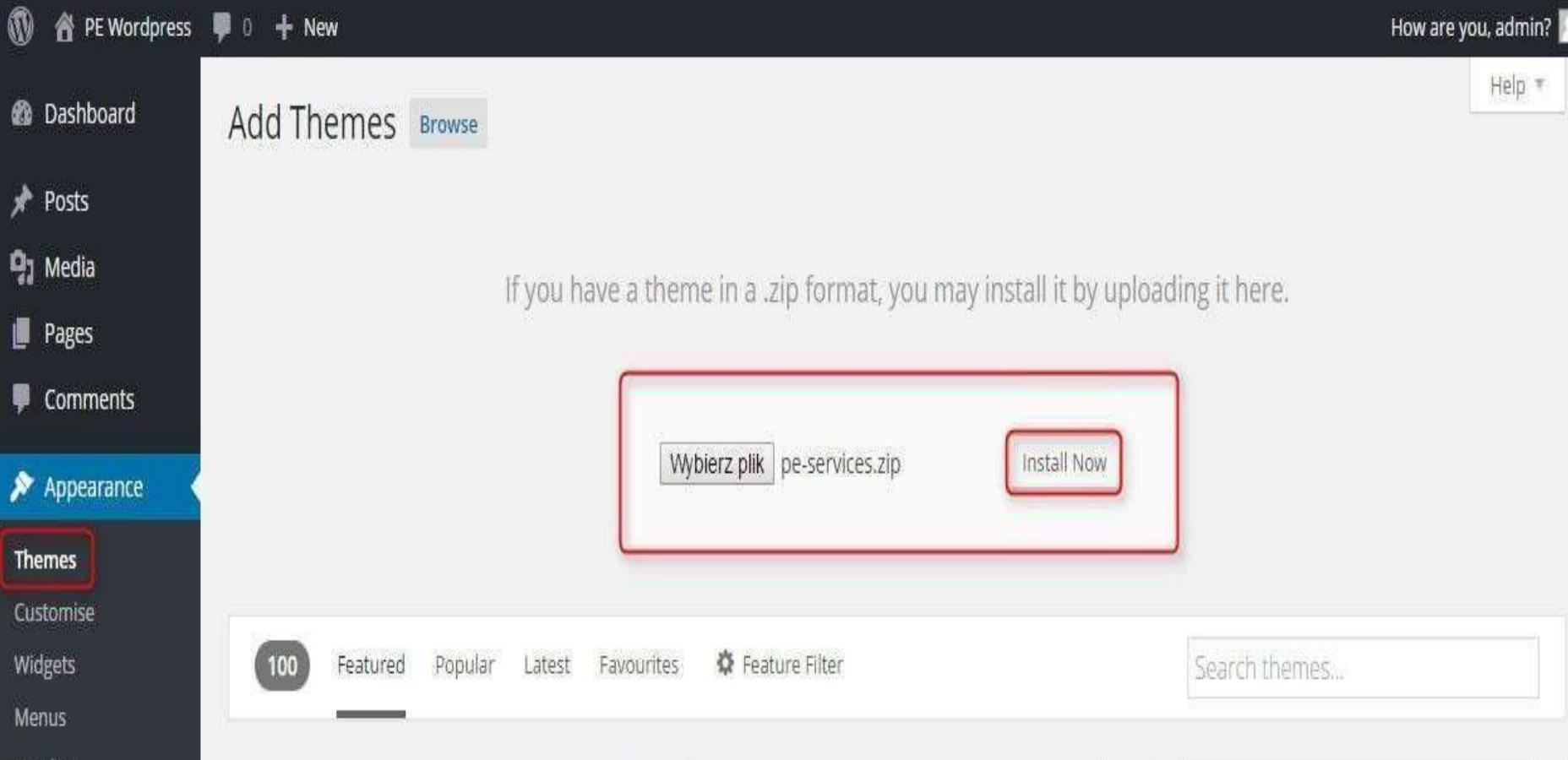
Double-check your email address before continuing.

Search engine visibility

Discourage search engines from indexing this site  
It is up to search engines to honor this request.

[Install WordPress](#)

## Theme download and Admin Panel



The screenshot shows the WordPress Admin Panel with a dark sidebar and a light main content area.

**Top Bar:**  
- Icons: PE Wordpress, 0 notifications, + New.  
- User: How are you, admin? (with a user icon).  
- Help: Help icon.

**Left Sidebar (Dark Mode):**

- Dashboard
- Posts
- Media
- Pages
- Comments
- Appearance** (selected)
- Themes** (highlighted with a red border)
- Customise
- Widgets
- Menus

**Main Content Area:**

### Add Themes

If you have a theme in a .zip format, you may install it by uploading it here.

**Upload Form:**  
- Input field: Wybierz plik pe-services.zip  
- Button: Install Now

**Footer:**

- 100 themes listed
- Filter buttons: Featured, Popular, Latest, Favourites, Feature Filter
- Search bar: Search themes...

## Theme download and Admin Panel

### 1)Installing Theme

1. Go to Appearance > Themes section
2. Click Add New and select the Upload Theme option
3. Choose the theme\_name.zip file and press Install Now
4. Once the theme is uploaded you need to activate it.

### 2)Install Plugins

## Install Required Plugins

All (9) | To Install (9)

Bulk Actions ▾ Apply

<input type="checkbox"/>	Plugin	Source	Type
<input type="checkbox"/>	CMB2 <a href="#">Install</a>	WordPress Repository	Required
<input type="checkbox"/>	Display Widgets <a href="#">Install</a>	WordPress Repository	Required
<input type="checkbox"/>	Ninja Forms <a href="#">Install</a>	WordPress Repository	Required
<input type="checkbox"/>	PE Hotel Plugin <a href="#">Install</a>	Pre-Packaged	Required
<input type="checkbox"/>	Redux Framework <a href="#">Install</a>	WordPress Repository	Required
<input type="checkbox"/>	Slider Revolution <a href="#">Install</a>	Pre-Packaged	Required
<input type="checkbox"/>	WP LESS <a href="#">Install</a>	WordPress Repository	Required
<input type="checkbox"/>	Newsletter <a href="#">Install</a>	WordPress Repository	Recommended
<input type="checkbox"/>	W3 Total Cache <a href="#">Install</a>	WordPress Repository	Recommended
<input type="checkbox"/>	Plugin	Source	Type

Bulk Actions ▾ Apply

# Wordpress Dashboard and Plugins

[https://github.com/TopsCode/WEB\\_DESIGNING/tree/main/Module-4Javascript](https://github.com/TopsCode/WEB_DESIGNING/tree/main/Module-4Javascript)

# Understanding Website Sections

## 1. Header

- **Location:** Top section of the website.
- **Content:** Typically includes the website logo, navigation bar, contact information, and sometimes a search bar.
- **Purpose:** Provides a consistent branding element and easy access to key site areas.

# Understanding Website Sections

## 2. Navigation Bar

- **Location:** Usually below or within the header.
- **Content:** Contains links to the main sections of the website (e.g., Home, About, Services, Contact).
- **Purpose:** Facilitates easy navigation and helps users find information quickly.

# Understanding Website Sections

## 3. Home Page

- **Content:** Overview of what the website offers, including key messages, featured content, and primary calls to action.
- **Purpose:** Serves as the main entry point and provides an introduction to the site.

## 4. About Us Page

- **Content:** Information about the organization, its history, mission, vision, team members, and values.
- **Purpose:** Builds trust and credibility by providing background information.

# Understanding Website Sections

## 5. Services/Products Page

- **Content:** Detailed descriptions of the services or products offered, including features, benefits, pricing, and images.
- **Purpose:** Provides potential customers with information needed to make purchasing decisions.

## 6. Blog/News Section

- **Content:** Articles, news updates, and other regularly updated content.
- **Purpose:** Engages visitors with fresh content, improves SEO, and establishes the site as a thought leader.

# Wordpress Dashboard and Directories

- **WordPress Dashboard Overview**
  - a. Admin Bar
  - b. Navigation Menu
  - c. Main Dashboard Screen
  - d. Screen Options
  - e. Help Section

# Wordpress Dashboard and Directories

- **WordPress Dashboard Sections**
  - a. Posts
  - b. Media
  - c. Pages
  - d. Comments
  - e. Appearance
  - f. Plugins
  - g. Users

# Wordpress Dashboard and Directories

- **WordPress Directory Structure**
  - a. Root Directory
  - b. wp-admin Directory
  - c. wp-content Directory
  - d. wp-includes Directory
  - e. Other Core Files and Directories

# • Understanding the Plugins

## • What are WordPress Plugins?

1. Definition
2. Importance
3. Types of Plugins

- **Understanding the Plugins**

## **1.What are WordPress Plugins?**

Definition

Importance

Types of Plugins

## **2.Finding and Installing Plugins**

Using the WordPress Plugin Directory

Installing from a .zip File

Installing via FTP

- **Understanding the Plugins**

### **3. Activating and Deactivating Plugins**

Activating a Plugin

Deactivating a Plugin

Bulk Actions

### **4. Configuring Plugin Settings**

Accessing Plugin Settings

Common Configuration Options

Best Practices for Plugin Configuration

- **Understanding the Plugins**

## **5. Updating Plugins**

Importance of Keeping Plugins Updated

Automatic vs. Manual Updates

Checking for Compatibility

## **6. Managing Plugins**

Viewing Installed Plugins

Deleting Unused Plugins

Troubleshooting Plugin Issues

- **Understanding the Plugins**

## **7. Popular WordPress Plugins and Their Uses**

SEO Plugins

Security Plugins

Performance Optimization Plugins

## **8. Best Practices for Using Plugins**

Choosing Reliable Plugins

Limiting the Number of Plugins

Backup Before Installing or Updating Plugins

# • Creating Subdomains

## 1. What is a Subdomain?

### Definition

- **Subdomain:** A prefix added to your main domain to create a separate URL. For example, if your main domain is **example.com**, a subdomain could be **blog.example.com** or **store.example.com**.
- **Common Uses**

**Blog:** **blog.example.com**

**Store:** **store.example.com**

**Support:** **support.example.com**

- **Creating Subdomains**

## 2. Steps to Create a Subdomain

### Accessing Your Hosting Control Panel

- **Log in to your hosting account:** Most hosting providers use control panels like cPanel, Plesk, or a custom dashboard.
- **Navigate to the subdomain section:** Look for options labeled “Subdomains” or “Domain Management.”

## • Creating Subdomains

### 2. Steps to Create a Subdomain

#### Creating the Subdomain

- 1. Locate the Subdomains section:** In cPanel, it's usually found under the “Domains” section.
- 2. Enter the subdomain name:** Type the prefix you want to add (e.g., **blog, store**).
- 3. Select the domain:** Choose the main domain you want to add the subdomain to.
- 4. Specify the document root:** This is the directory where the subdomain’s files will be stored. By default, it’s often something like **public\_html/blog**.

- **Creating Subdomains**

## 2. Steps to Create a Subdomain

### Configuring DNS Records

- **Automatic Configuration:** Most hosting control panels will automatically configure DNS records for you.
- **Manual Configuration:** If needed, go to the DNS settings for your domain and add an A record pointing the subdomain to your server's IP address.

- **Creating Subdomains**

## 2. Steps to Create a Subdomain

### Installing Software or Content on the Subdomain

- **Upload Files:** Use an FTP client or file manager to upload content to the subdomain's document root.
- **Install CMS:** If you're using a content management system (CMS) like WordPress, install it in the subdomain directory.

- **Creating Subdomains**

### 3. Managing Subdomains

#### Accessing Subdomain Files

- **File Manager:** Use your hosting control panel's file manager to navigate to the subdomain's document root.
- **FTP/SFTP:** Connect to your server using FTP/SFTP and navigate to the appropriate directory

- **Creating Subdomains**

### 3. Managing Subdomains

#### Setting Up SSL for Subdomains

- **Free SSL Certificates:** Use tools like Let's Encrypt to generate SSL certificates for your subdomains.
- **Paid SSL Certificates:** Purchase and install SSL certificates through your hosting provider or a third-party SSL provider.

- **Creating Subdomains**

### 3. Managing Subdomains

#### Redirecting Subdomains

- **301 Redirects:** Set up permanent redirects using `.htaccess` or server configurations.

#### Deleting Subdomains

1. **Go to the Subdomains section:** In your hosting control panel.
2. **Select the subdomain to delete:** There will be an option to remove or delete the subdomain.
3. **Confirm deletion:** This will typically delete the subdomain's DNS records and document root.

## • Creating Subdomains

### 4. Examples and Best Practices

#### Examples of Subdomain Uses

- **Corporate Blog:** blog.corporate.com
- **Online Store:** shop.corporate.com
- **Customer Support:** support.corporate.com
- **Regional Content:** us.corporate.com, uk.corporate.com

# Customization of Website

- Customization of Web Page
- Slider Editing And Importing
- Wordpress home page
- Creating a Contact Form in Wordpress
- Website Page Setting using Elementor and unyson
- Edit Menu in Wordpress
- Customization of Themes and Widgets
- Creation and Posting of a Blog
- Adding Social Share Buttons
- Adding Live Chat and Call Now Plugin
- Clearing Plugin in Wordpress
- Creating Website Backups in Wordpress

# E-commerce Website Creation

[https://github.com/TopsCode/WEB\\_DESIGNING/tree/main/Module-4Javascript](https://github.com/TopsCode/WEB_DESIGNING/tree/main/Module-4Javascript)

- Woocommerce for shopping Website
- Customization of Shopping Website
- Adding New Product and Shipping Cost
- WooCOmmerce Plugin and Razorpay integration
- Tas, shipping and email setting
- Generative invoice and process and order

# Monetization Through Website using Google Adsense

- Website Scores
- Creating a Sitemap and Robots.txt
- Google Analytics and Page Speed Insight
- SEO Plugin
- Understanding Google Adsense
- Google Ads Designing and Tricks to increase earnings
- Adding Google Adsense to website Demo

# **Module - 6**

# **[JavaScript]**

# JavaScript

- ✓ What is JavaScript, Creating First JavaScript Program, Way to apply JavaScript , Event in JavaScript ,
- ✓ How to select Tag Classes and Id
- ✓ JS Introduction
- ✓ JS Getting Started
- ✓ JS Syntax
- ✓ JS Variables
- ✓ JS Generating Output
- ✓ JS Data Types
- ✓ JS Operators
- ✓ JS Events
- ✓ JS If, Else JS Switch Case

- ✓ JAVASCRIPT & DOM
- ✓ JS DOM Manipulation
- ✓ JS DOM Navigation
- ✓ Practical Example: 1) Create program for input color and output that code 2)  
Create program for pattern using loop
- ✓ Functions, Alert ,Confirm , Prompt , Addition of Two Number  
, Hide and Show Password
- ✓ JS Loops, JS Functions
- ✓ Practical Example: 1) Slider • If Else Statement, JavaScript  
Form Validation • Practical Examples: 1) Get input data and perform different  
operations
- ✓ 2) Make dynamic CSS by click

# Javascript

- JavaScript is an object-based scripting language which is lightweight and cross-platform.
- JavaScript is not a compiled language, but it is a translated language. The JavaScript Translator (embedded in the browser) is responsible for translating the JavaScript code for the web browser.



# What is JavaScript

- JavaScript (js) is a light-weight object-oriented programming language which is used by several websites for scripting the webpages.
- It is an interpreted, full-fledged programming language that enables dynamic interactivity on websites when applied to an HTML document.
- It was introduced in the year 1995 for adding programs to the webpages in the Netscape Navigator browser.

# What is JavaScript

- With JavaScript, users can build modern web applications to interact directly without reloading the page every time.
- The traditional website uses js to provide several forms of interactivity and simplicity.
- Although, JavaScript has no connectivity with Java programming language. In addition to web browsers, databases such as CouchDB and MongoDB uses JavaScript as their scripting and query language.

# Features of JavaScript

- All popular web browsers support JavaScript as they provide built-in execution environments.
- JavaScript follows the syntax and structure of the C programming language. Thus, it is a structured programming language.
- JavaScript is a weakly typed language, where certain types are implicitly cast (depending on the operation).

# Features of JavaScript

- JavaScript is an object-oriented programming language that uses prototypes rather than using classes for inheritance.
- It is a light-weighted and interpreted language.
- It is a case-sensitive language.
- JavaScript is supportable in several operating systems including, Windows, macOS, etc.
- It provides good control to the users over the web browsers.

# Application of JavaScript

JavaScript is used to create interactive websites.

- Client-side validation,
- Dynamic drop-down menus,
- Displaying date and time,
- Displaying pop-up windows and dialog boxes (like an alert dialog box, confirm dialog box and prompt dialog box),
- Displaying clocks etc.

# Example of Javascript

```
<script type="text/javascript">  
document.write("Welcome to  
LearnVern");  
</script>
```

- > The script tag specifies that we are using JavaScript.
- > The text/javascript is the content type that provides information to the browser about the data.

# Ways to add JS code

1. Between the body tag of html

```
<body><script>...</script></body>
```

2. Between the head tag of html

```
<head><script>...</script></head>
```

3. In .js file (external javaScript)

```
<script src=""></script>
```

**GitHub link :**

[https://github.com/TopsCode/WEB\\_DESIGNING/blob/main/Module-4Javascript/01FirstJs.html](https://github.com/TopsCode/WEB_DESIGNING/blob/main/Module-4Javascript/01FirstJs.html)

# 1. code between the body tag

Ex.:

```
<html>
<body>
<script
type="text/javascript">
alert("Hello Javatpoint");
</script>
</body>
</html>
```

## 2. code between the head tag

Ex.:

```
<script type="text/javascript">  
function msg(){  
    alert("Hello Javatpoint");  
}  
</script>  
  
<form>  
<input type="button" value="click" onclick="msg()" />  
</form>
```

### 3. In .js file (External JavaScript)

- It provides code re usability because single JavaScript file can be used in several html pages.
- An external JavaScript file must be saved by .js extension. It is recommended to embed all JavaScript files into a single file. It increases the speed of the webpage.
- In the current file, add the .js file through the <script> tag

**Ex.:**

```
<head>
<script type="text/javascript" src="message.js"></script>
</head>
```

# Advantages of External JavaScript

There will be following benefits if a user creates an external javascript:

1. It helps in the reusability of code in more than one HTML file.
2. It allows easy code readability.
3. It is time-efficient as web browsers cache the external js files, which further reduces the page loading time.
4. It enables both web designers and coders to work with html and js files parallelly and separately, i.e., without facing any code conflicts.
5. The length of the code reduces as only we need to specify the location of the js file.

# Disadvantages of External JavaScript

There are the following disadvantages of external files:

- The stealer may download the coder's code using the url of the js file.
- If two js files are dependent on one another, then a failure in one file may affect the execution of the other dependent file.
- The web browser needs to make an additional http request to get the js code.

# Disadvantages of External JavaScript

- A tiny to a large change in the js code may cause unexpected results in all its dependent files.
- We need to check each file that depends on the commonly created external javascript file.
- If it is a few lines of code, then better to implement the internal javascript code.

# What is Comment

- The JavaScript comments are meaningful way to deliver message.
- It is used to add information about the code, warnings or suggestions so that end user can easily interpret the code.
- The JavaScript comment is ignored by the JavaScript engine i.e. embedded in the browser.

# Advantages of Comments

Advantages of JavaScript comments

- There are mainly two advantages of JavaScript comments.
- To make code easy to understand It can be used to elaborate the code so that end user can easily understand the code.
- To avoid the unnecessary code It can also be used to avoid the code being executed. Sometimes, we add the code to perform some action. But after sometime, there may be need to disable the code. In such case, it is better to use comments.

# JavaScript Single line Comment

- It is represented by double forward slashes (//). It can be used before and after the statement.

**Ex.**

```
<script>
// It is single line comment
document.write("hello
javascript");
</script>
```

[WEB DESIGNING/Module-4Javascript/Comments\\_in\\_Js.html at main · TopsCode/WEB DESIGNING · GitHub](#)

# JavaScript Multiline Comment

It can be used to add single as well as multi line comments. So, it is more convenient.

**Ex.**

```
<script>
/* It is multi line comment.
It will not be displayed */
document.write("example of javascript multiline comment");
</script>
```

[WEB DESIGNING/Module-4Javascript/Comments in Js.html at main - TopsCode/WEB DESIGNING - GitHub](#)

# Variables in JavaScript

- A JavaScript variable is simply a name of storage location. There are two types of variables in JavaScript : local variable and global variable.
- There are some rules while declaring a JavaScript variable (also known as identifiers).
  1. Name must start with a letter (a to z or A to Z), underscore( \_ ), or dollar( \$ ) sign.
  2. After first letter we can use digits (0 to 9), for example value1.
  3. JavaScript variables are case sensitive, for example x and X are different variables.

GitHub link :

[https://github.com/TopsCode/WEB\\_DESIGNING/blob/main/Module-4Javascript/05variable.js](https://github.com/TopsCode/WEB_DESIGNING/blob/main/Module-4Javascript/05variable.js)

# Ex.

- Correct way to define javascript variable:

```
var x = 10;
```

```
var _value="sonoo";
```

- Incorrect way to define Javascript variable:

```
var 123=30;
```

```
var *aa=320;
```

# Ex.

```
<script>  
    var x = 10;  
    var y = 20;  
    var z=x+y;  
    document.write(z);  
</script>
```

# Using let and const

- Before 2015, using the var keyword was the only way to declare a JavaScript variable.
- The 2015 version of JavaScript (ES6) allows the use of the const keyword to define a variable that cannot be reassigned, and the let keyword to define a variable with restricted scope.

Github link :

[https://github.com/TopsCode/WEB\\_DESIGNING/blob/main/Module-4Javascript/05variable.js](https://github.com/TopsCode/WEB_DESIGNING/blob/main/Module-4Javascript/05variable.js)

# JavaScript Local Variable

- A JavaScript local variable is declared inside block or function. It is accessible within the function or block only.

**Ex.1**

```
<script>
function
abc(){
    var x=10;//local variable
}
</script>
```

**Ex.2**

```
<script>
if(10<13){
    var y=20;//JavaScript local
    variable
}
</script>
```

# JavaScript Global Variable

- A JavaScript global variable is accessible from any function. A variable i.e. declared outside the function or declared with window object is known as global variable.

```
<script>
    var data=200;//global variable
    function a(){ document.writeln(data);
    }
    function
    b()
    {
        document.writeln(data);
    }
    a();//calling JavaScript function b();
</script>
```

# Global Variable

- A JavaScript global variable is declared outside the function or declared with window object. It can be accessed from any function.

**Ex.**

```
<script>  
var value=50;//global variable function  
a(){alert(value);}  
function  
b()  
{  
    alert(value);}  
</script>
```

# Declaring Global Variable in a Function

- To declare JavaScript global variables inside function, you need to use window object.

**Ex.**

```
function m(){  
window.value=100;//declaring global variable by window object  
}  
function n(){  
alert(window.value);//accessing global variable from other function  
}
```

# Internals of global variable in JavaScript

- When you declare a variable outside the function, it is added in the window object internally. You can access it through window object also.

**Ex.**

```
var  
value=50;  
function a(){  
alert(window.value);//accessing global variable  
}
```

# Data Types

- JavaScript provides different data types to hold different types of values.  
There are two types of data types in JavaScript.
  1. Primitive data type
  2. Non-primitive (reference) data type
- JavaScript is a dynamic type language, means you don't need to specify type of the variable because it is dynamically used by JavaScript engine. You need to use var here to specify the data type. It can hold any type of values such as numbers, strings etc.

```
var a=40; // Variable with numeric value
```

```
var b="Rahul"; // Variable with String
```

```
value
```

Github link :

[https://github.com/TopsCode/WEB\\_DESIGNING/blob/main/Module-4Javascript/06datatype.js](https://github.com/TopsCode/WEB_DESIGNING/blob/main/Module-4Javascript/06datatype.js)

[https://github.com/TopsCode/WEB\\_DESIGNING/blob/main/Module-4Javascript/06datatype.html](https://github.com/TopsCode/WEB_DESIGNING/blob/main/Module-4Javascript/06datatype.html)

# JavaScript primitive data types

Data Type	Description
String	represents sequence of characters e.g. "hello"
Number	represents numeric values e.g. 100
Boolean	represents boolean value either false or true
Undefined	represents undefined value
Null	represents null i.e. no value at all

# The Concept of Data Types

- var x = 15 + "Hello";
- var x = “15” + "Hello";
- var x = “Hello” + 15;
- var x = 50 + 50 + "Hello";
- var x = “Hello” + 50 + 50;

# The Concept of Data Types

- **Assigning value to variable:**

```
var x;      // Now x is undefined  
x = 50;    // Now x is a Number  
x = "Joel"; // Now x is a String
```

- **String:**

```
var answer1 = "His "; // Single quote inside double quotes  
var answer2 = "name is 'Joel"'; // Single quotes inside double  
var answer3 = 'name is "Joel"'; quotes  
                           // Double quotes inside single  
                           quotes
```

# The Concept of Data Types

- **Boolean:**

```
var x = 5;  
var y = 5;  
var z = 6;  
(x == y) // Returns true  
(x == z) // Returns false
```

- **Null:**

In JavaScript null is "nothing". It is supposed to be something that doesn't exist.

Unfortunately, in JavaScript, the data type of null is an object.

# The Concept of Data Types

**Ex.**

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

```
<script>
document.getElementById("demo").innerHTML
= typeof undefined + "<br>" +
typeof null + "<br><br>" +
(null === undefined) + "<br>" +
(null == undefined);
</script>
```

# JavaScript non-primitive data types

Data Type	Description
Object	represents instance through which we can access members
Array	represents group of similar values

# Operators

- JavaScript operators are symbols that are used to perform operations on operands.

There are following types of operators in JavaScript.

1. Arithmetic Operators
2. Comparison (Relational) Operators
3. Bitwise Operators
4. Logical Operators
5. Assignment Operators
6. Special Operators

Github Link

[https://github.com/TopsCode/WEB\\_DESIGNING/blob/main/Module-2/CSS-CSS3/36absolute\\_z-index.html](https://github.com/TopsCode/WEB_DESIGNING/blob/main/Module-2/CSS-CSS3/36absolute_z-index.html)

# Arithmetic Operator

- Arithmetic operators are used to perform arithmetic operations on the operands. The following operators are known as JavaScript arithmetic operators.

Operator	Description	Example
+	Addition	$1 + 1$
-	Subtraction	$1 - 1$
*	Multiplication	$1 * 1$
/	Division	$1 / 1$
%	Modulus	$1 \% 1$
++	Increment	$++i$
--	Decrement	$--i$

[WEB DESIGNING/Module-4Javascript/Arithmatic oprator.html at main · TopsCode/WEB DESIGNING · GitHub](#)

# Comparison Operator

- The JavaScript comparison operator compares the two operands.

Operator	Description	Example
<code>==</code>	Is equal to	<code>a == b</code>
<code>===</code>	Is equal and of same type	<code>a === b</code>
<code>!=</code>	Not equal to	<code>a != b</code>
<code>!==</code>	Not equal and of not of same type	<code>a !== b</code>
<code>&gt;</code>	Greater than	<code>a &gt; b</code>
<code>&gt;=</code>	Greater than or equal to	<code>a &lt;= b</code>
<code>&lt;</code>	Less than	<code>a &lt; b</code>
<code>&lt;=</code>	Less than or equal to	<code>a &lt;= b</code>

# Bitwise Operator

- The bitwise operators perform bitwise operations on operands.

Operator	Description	Example
&	Bitwise AND	a & b
	Bitwise OR	a   b
^	Bitwise XOR	a ^ b
~	Bitwise NOT	~a
<<	Bitwise Left Shift	a << b
>>	Bitwise Right Shift	a >> b
>>>	Bitwise Right Shift with zero	a >>> b

# Logical Operator

Operator	Description	Example
<code>&amp;&amp;</code>	Logical AND	<code>a &amp;&amp; b</code>
<code>  </code>	Logical OR	<code>a    b</code>
<code>!</code>	Logical Not	<code>!a</code>

# Assignment Operator

Operator	Description	Example
=	Assign	a = b
+=	Add and Assign	a += b
-=	Subtract and Assign	a -= b
*=	Multiply and Assign	a *= b
/=	Divide and Assign	a /= b
%=	Modulus and Assign	a %= b

[WEB DESIGNING/Module-4Javascript/Assignment Operator.html at main · TopsCode/WEB DESIGNING · GitHub](#)

# Special Operator

Operator	Description	Example
? :	Conditional Operator	a ==b ? x=true : x=false;
,	Comma operator allows multiple expressions to be evaluated as single statement	let a = (1,2,3); a=3
delete	Delete property deletes an entry from the object	delete x
in	in property checks if the given property exist in the object	prop in object
instanctof	checks if the object is an instance of given type	object instanctof object_type
new	creates a new object	var a = new a();
typeof	checks the type of object	typeof 5 // "number"
void	it discards the expressions return value	void(x)

[WEB DESIGNING/Module-4Javascript/Special oprator.html at main · TopsCode/WEB DESIGNING · GitHub](#)

# Conditional Statements

- The JavaScript Conditional statement is used to execute the code whether condition is true or false. There are three forms of if statement in JavaScript.
  1. If Statement
  2. If else statement
  3. if else if statement

**Github link :**

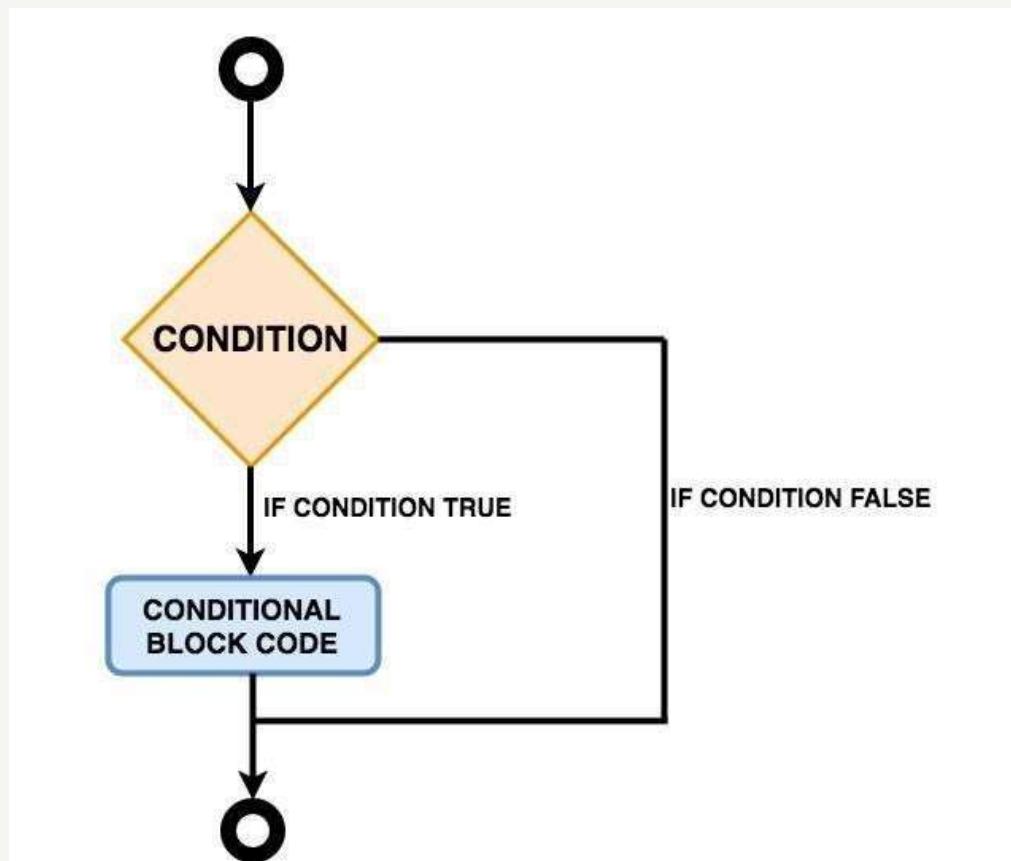
[https://github.com/TopsCode/WEB\\_DESIGNING/blob/main/Module-4Javascript/19conditional.html](https://github.com/TopsCode/WEB_DESIGNING/blob/main/Module-4Javascript/19conditional.html)

# if statement

- It evaluates the content only if expression is true. The signature of JavaScript if statement is given below.

## Syntax:

```
if(expression){  
    //content to be  
    evaluated  
}
```



# Example of 'if'

Ex.

```
<script>  
var a=true;  
if(a){  
document.write("LearnVern is a wonderful place to learn  
Javascript");  
}  
</script>
```

Github link :

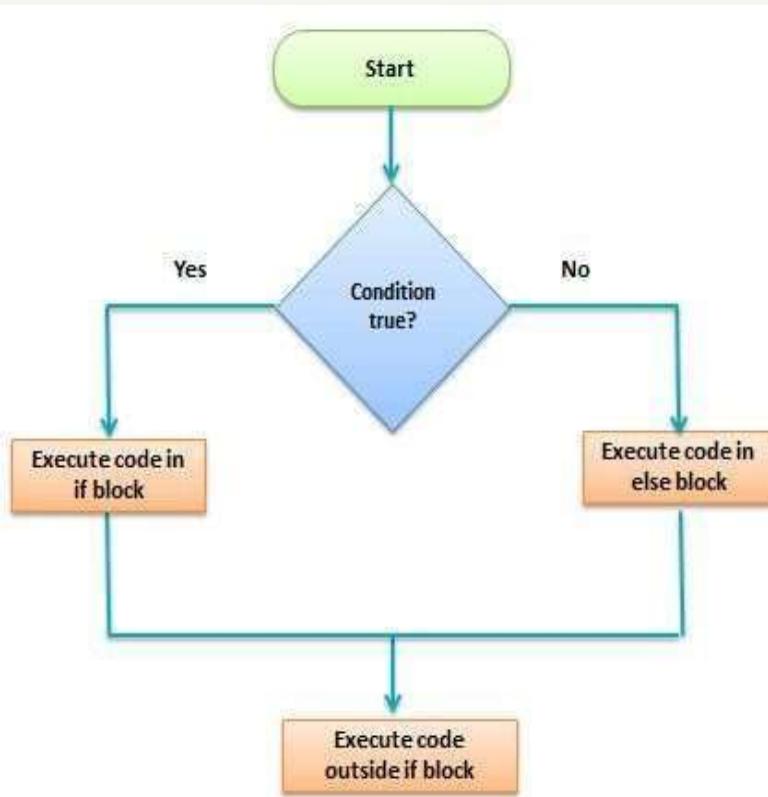
[https://github.com/TopsCode/WEB\\_DESIGNING/blob/main/Module-4Javascript/20if.html](https://github.com/TopsCode/WEB_DESIGNING/blob/main/Module-4Javascript/20if.html)

# if...else statement

- It evaluates the content whether condition is true or false. The syntax of JavaScript if-else statement is given below.

## Syntax:

```
if(expression){  
    //evaluated if condition is true  
}  
else{  
    //evaluated if condition is  
    false  
}
```



# if...else...if statement

- It evaluates the content only if expression is true from several expressions.

**Syntax:**

```
if(expression1){  
    //content to be evaluated if expression1 is true  
}  
  
else if(expression2){  
    //content to be evaluated if expression2 is true  
}  
  
else{  
    //content to be evaluated if no expression is true  
}
```

**Github link :**

[https://github.com/TopsCode/WEB\\_DESIGNING/blob/main/Module-4Javascript/21if-else.html](https://github.com/TopsCode/WEB_DESIGNING/blob/main/Module-4Javascript/21if-else.html)

# Switch Statement

- The JavaScript switch statement is used to execute one code from multiple expressions. It is just like else if statement that we have learned in previous page.
- It is convenient than if..else..if because it can be used with numbers, characters etc.

**Github link :**

[https://github.com/TopsCode/WEB\\_DESIGNING/blob/main/Module-4Javascript/26switch.html](https://github.com/TopsCode/WEB_DESIGNING/blob/main/Module-4Javascript/26switch.html)

# Switch Statement

## Syntax:

```
switch(expression){ case value1:  
    //code to be executed;  
    break;  
    case value2:  
        //code to be executed;  
        break;  
        .....  
    default:  
        //code to be executed if above values are not matched;  
}
```

# The break and default Keyword

- When JavaScript reaches a break keyword, it breaks out of the switch block.
- This will stop the execution inside the switch block.
- It is not necessary to break the last case in a switch block. The block breaks (ends) there anyway.
- The default keyword specifies the code to run if there is no case match:

# Strict Comparison

- Switch cases use strict comparison (====).
- The values must be of the same type to match.
- A strict comparison can only be true if the operands are of the same type.
- In the example explained in next slide there will be no match for x:

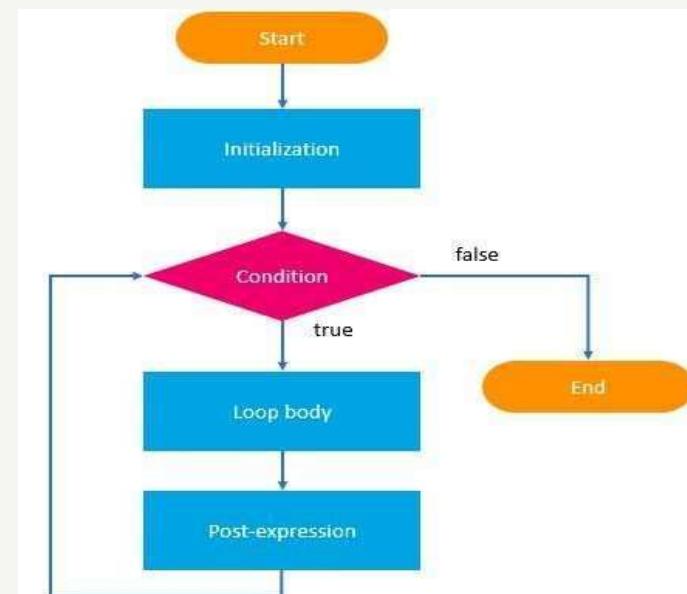
[WEB DESIGNING/Module-4Javascript/Strick comparision.html at main · TopsCode/WEB DESIGNING · GitHub](#)

# Ex. Strict Comparison

```
var x= "0" switch  
(x) { case 0:  
  
    text =  
    "Off";  
  
    break; case  
    1:  
    text = "On";  
    break; default:  
    text = "No value found"; }
```

# Loops

- The JavaScript loops are used to iterate the piece of code using for, while, do while or for-in loops. It makes the code compact. It is mostly used in array.
- There are four types of loops in JavaScript.
  1. for loop
  2. while loop
  3. do-while loop
  4. for-in loop
  5. for-of loop



GitHub link :

[https://github.com/TopsCode/WEB\\_DESIGNING/blob/main/Module-4Javascript/48Loops.html](https://github.com/TopsCode/WEB_DESIGNING/blob/main/Module-4Javascript/48Loops.html)

# for Loop

- The JavaScript for loop iterates the elements for the fixed number of times.  
It should be used if number of iteration is known.

## Syntax:

```
for (initialization; condition; increment)
{
    code to be executed
}
```

# for Loop Example

```
<script>
for (i=1; i<=5; i++)
{
document.write(i +
"<br/>")
}
</script>
```

Github link :

[https://github.com/TopsCode/WEB\\_DESIGNING/blob/main/Module-4Javascript/48Loops.html](https://github.com/TopsCode/WEB_DESIGNING/blob/main/Module-4Javascript/48Loops.html)

# while Loop

- The JavaScript while loop iterates the elements for the infinite number of times. It should be used if number of iteration is not known.

## Syntax:

```
while (condition)
{
    code to be executed
}
```

Github link :

[https://github.com/TopsCode/WEB\\_DESIGNING/blob/main/Module-4Javascript/48Loops.html](https://github.com/TopsCode/WEB_DESIGNING/blob/main/Module-4Javascript/48Loops.html)

# while Loop Example

```
<script>
var i=11;
while (i<=15)
{
document.write(i +
"<br/>"); i++;
}
</script>
```

# do while Loop

- The JavaScript do while loop iterates the elements for the infinite number of times like while loop. But, code is executed at least once whether condition is true or false.

## Syntax:

```
do{  
    //code to be executed  
}while (condition);
```

**Github link :**

[https://github.com/TopsCode/WEB\\_DESIGNING/blob/main/Module-4Javascript/48Loops.html](https://github.com/TopsCode/WEB_DESIGNING/blob/main/Module-4Javascript/48Loops.html)

# do...while Loop Example

```
<script>
var i=21;
do{
document.write(i +
"<br/>"); i++;
}while (i<=25);
</script>
```

# for...in Loop

- for in loops through the properties of an object

## Syntax:

```
for (var in object) {  
    code block to be executed  
}
```

# for...in Loop Example

```
var person = {fname:"John", lname:"Doe",  
age:25};
```

```
var text = "";  
var x;  
for (x in person) {  
    text += person[x] + " ";  
}
```

# for...of Loop

- for of loops through the values of an iterable object

## Syntax:

```
for (variable of iterable)
  { code block to be
    executed
  }
```

# What is Function

- JavaScript functions are used to perform operations. We can call JavaScript function many times to reuse the code.
- There are mainly two **advantages** of JavaScript functions.
  1. Code reusability: We can call a function several times so it save coding.
  2. Less coding: It makes our program compact. We don't need to write many lines of code each time to perform a common task.

GitHub link :

[https://github.com/TopsCode/WEB\\_DESIGNING/blob/main/Module-4Javascript/27functions.html](https://github.com/TopsCode/WEB_DESIGNING/blob/main/Module-4Javascript/27functions.html)

# Syntax and Example

```
function functionName([arg1, arg2, ...argN]){
    //code to be executed
}
```

**Note:** JavaScript Functions can have 0 or more arguments.

# JavaScript Function Object

- In JavaScript, the purpose of Function constructor is to create a new Function object. It executes the code globally. However, if we call the constructor directly, a function is created dynamically but in an unsecured way.

## Syntax:

```
new Function ([arg1[, arg2[, ...argn]]],] funcBody)
```

## Parameter:

arg1, arg2, .... , argn - It represents the argument used by function.

funcBody - It represents the function definition.

Github link :

[https://github.com/TopsCode/WEB\\_DESIGNING/blob/main/Module-4Javascrip/27functions.html](https://github.com/TopsCode/WEB_DESIGNING/blob/main/Module-4Javascrip/27functions.html)

# JavaScript Function Methods

Method	Description
call()	It is used to call a function contains this value and an argument list.
apply()	It is used to call a function contains this value and a single array of arguments.
bind()	It is used to create a new function.
toString()	It returns the result in a form of a string.

# call()

- The JavaScript Function call() method is used to call a function contains this value and an argument provided individually.

## Syntax:

```
function.call(thisArg, arg1,arg2,. . .,argn)
```

## Parameter:

thisArg - It is optional. The this value is given for the call to function.

arg1,arg2,...,argn - It is optional. It represents the arguments for the function.

# apply()

- The JavaScript Function apply() method is used to call a function contains this value and an argument contains elements of an array. Unlike call() method, it contains the single array of arguments.

## Syntax:

```
function.apply(thisArg, [array])
```

## Parameter:

thisArg - It is optional. The this value is given for the call to a function.

array - It is optional. It is an array-like object.

Github link :

[https://github.com/TopsCode/WEB\\_DESIGNING/blob/main/Module-4Javascript/28apply-remove-css.html](https://github.com/TopsCode/WEB_DESIGNING/blob/main/Module-4Javascript/28apply-remove-css.html)

# bind()

- The JavaScript Function bind() method is used to create a new function. When a function is called, it has its own this keyword set to the provided value, with a given sequence of arguments.

## Syntax:

```
function.bind(thisArg [, arg1[, arg2[, ...]]])
```

## Parameter:

thisArg - The this value passed to the target function.

arg1,arg2,...,argn - It represents the arguments for the function.

[WEB DESIGNING/Module-4Javascript/Bind\\_function.html at main · TopsCode/WEB DESIGNING · GitHub](#)

# toString()

- The JavaScript Function `toString()` method returns a string. Here, string represents the source code of the function.

## Syntax:

```
function.toString()
```

[WEB DESIGNING/Module-4Javascript/ToString.html](#)  
[at main · TopsCode/WEB DESIGNING · GitHub](#)

# What is object in Javascript

- A JavaScript object is an entity having state and behavior (properties and method). For example: car, pen, bike, chair, glass, keyboard, monitor etc.
- JavaScript is an object-based language. Everything is an object in JavaScript.
- JavaScript is template based not class based. Here, we don't create class to get the object. But, we direct create objects.

# What is object in Javascript

- There are 3 ways to create objects.
  1. By object literal
  2. By creating instance of Object directly (using new keyword)
  3. By using an object constructor (using new keyword)

Github link :

[https://github.com/TopsCode/WEB\\_DESIGNING/blob/main/Module-4Javascript/66object.html](https://github.com/TopsCode/WEB_DESIGNING/blob/main/Module-4Javascript/66object.html)

# 1. Object by object literal

- The syntax of creating object using object literal is given below:

```
object={  
    property1:value1,  
    property2:value2  
    ....  
    propertyN:valueN  
}
```

[WEB DESIGNING/Module-4Javascript/Obj\\_Literal.html at main · TopsCode/WEB DESIGNING · GitHub](#)

# Object by object literal Example

```
<script>
emp={id:102,name:"Shyam Kumar",salary:40000}
document.write(emp.id+" "+emp.name+
"+emp.salary);
</script>
```

## 2. By creating instance of Object

**Syntax:**

```
var objectname=new Object();
```

Here, new keyword is used to create object.

# By creating instance of Object Example

```
<script>  
var emp=new Object();  
emp.id=101;  
emp.name="Ravi Malik";  
emp.salary=50000;  
document.write(emp.id+" "+emp.name+"  
"+emp.salary);  
</script>
```

[WEB DESIGNING/Module-4Javascript/By Instanceof object.html at main · TopsCode/WEB DESIGNING · GitHub](#)

# 3. By using an Object constructor

- you need to create function with arguments. Each argument value can be assigned in the current object by using this keyword.
- The **this** keyword refers to the current object.

[WEB DESIGNING/Module-4Javascript/Object Constructor.html at main · TopsCode/WEB DESIGNING · GitHub](#)

# How to define method in javascript object

```
e=new emp(103,"Sonoo Jaiswal",30000);
document.write(e.id+" "+e.name+
"+e.salary); e.changeSalary(45000);
document.write("<br>"+e.id+" "+e.name+
"+e.salary);
</script>
```

# JavaScript Object

Methods	Description
Object.assign()	This method is used to copy enumerable and own properties from a source object to a target object
Object.create()	This method is used to create a new object with the specified prototype object and properties.
Object.freeze()	This method prevents existing properties from being removed.
Object.is()	This method determines whether two values are the same value.
Object.isExtensible()	This method determines if an object is extensible
Object.isFrozen()	This method determines if an object was frozen.
Object.isSealed()	This method determines if an object is sealed.
Object.seal()	This method prevents new properties from being added and marks all existing properties as non-configurable.
Object.values()	This method returns an array of values.

[WEB DESIGNING/Module-4Javascript/JS\\_Object.html](#)  
[at main - TopsCode/WEB DESIGNING - GitHub](#)

# Browser Object Model

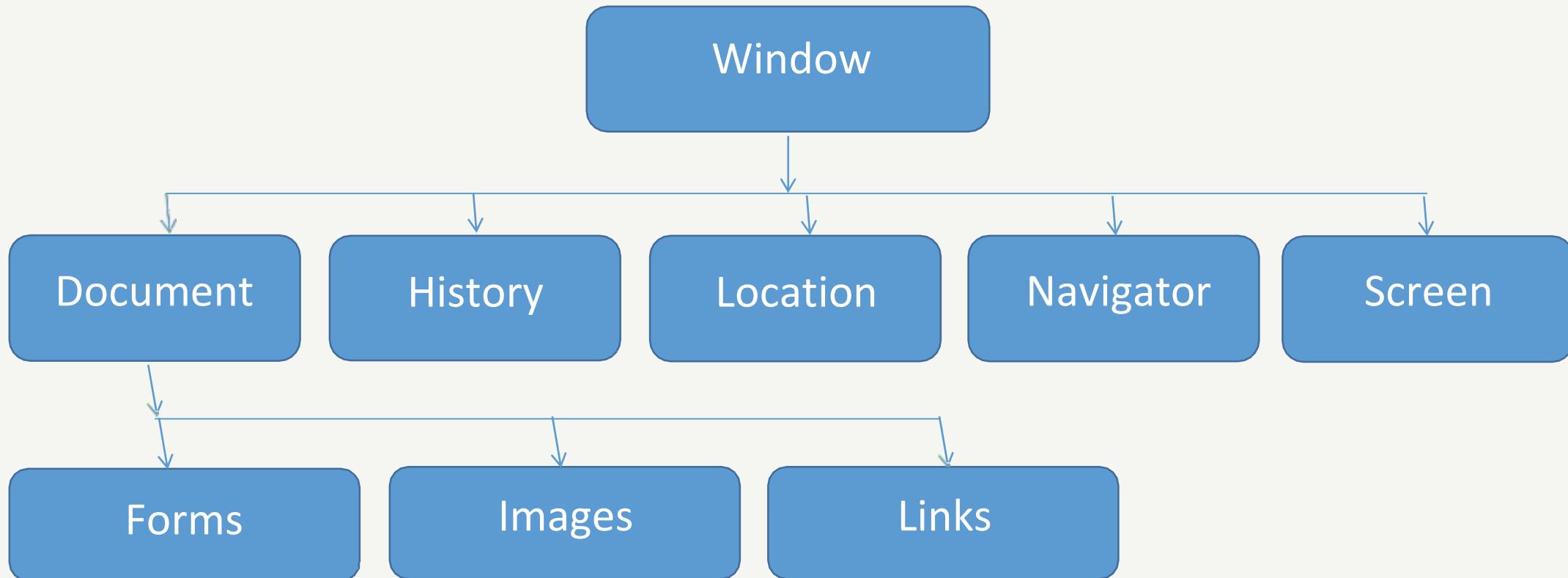
- The Browser Object Model (BOM) is used to interact with the browser.
- The default object of browser is window means you can call all the functions of window by specifying window or directly.

For example:

```
window.alert("hello  
LearnVern");
```

Above is same as : alert("hello LearnVern");

# Browser Object Model



# Window Object

- The window object represents a window in browser. An object of window is created automatically by the browser.
- Window is the object of browser, it is not the object of javascript. The javascript objects are string, array, date etc.
- There are several Window Methods Available in through Window Object

[WEB DESIGNING/Module-4Javascript/Window\\_object.html at main · TopsCode/WEB DESIGNING · GitHub](#)

# Methods of Window Object

Method	Description
alert()	displays the alert box containing message with ok button.
confirm()	displays the confirm dialog box containing message with ok and cancel button.
prompt()	displays a dialog box to get input from the user.
open()	opens the new window.
close()	closes the current window.
setTimeout()	performs action after specified time like calling function, evaluating expressions etc.

# History Object in Javascript

- The JavaScript history object represents an array of URLs visited by the user.  
By using this object, you can load previous, forward or any particular page.

## Syntax:

window.history  
or  
history

# Property

- There is only one Property in History Object  
**history.length;**
- The length property returns the number of URLs in the history list of the current browser window.
- The property returns at least 1, because the list includes the currently loaded page.

## Example:

```
var x = history.length;
```

# History back() Method

- The back() method loads the previous URL in the history list. This is the same as clicking the "Back button" in your browser.

**Example:**

```
<button onclick="goBack()">Go Back</button>
<script>
function goBack() {
    window.history.back()
}
;
```

# History forward() Method

- The forward() method loads the next URL in the history list. This is the same as clicking the "Forward button" in your browser, or history.go(1).

## Example:

```
<button onclick="goForward()">Go Forward</button>
<script>
function goForward()
{
    window.history.forward();
}
</script>
```

[WEB DESIGNING/Module-4Javascript/History Back.html at main · TopsCode/WEB DESIGNING - GitHub](#)

# History go() Method

- The go() method loads a specific URL from the history list.

## Example:

```
<button onclick="goBack()">Go Back 2 Pages</button>
```

```
<script>
function goBack() {
    window.history.go(-2);
}
</script>
```

[WEB DESIGNING/Module-4Javascript/History\\_Go.html at main · TopsCode/WEB DESIGNING · GitHub](#)

# Navigator Object

- The JavaScript navigator object is used for browser detection. It can be used to get browser information such as appName, appCodeName, userAgent etc.

**Syntax:**

window.navigator or

navigator

[WEB DESIGNING/Module-4Javascript/Navigator\\_object.html at main · TopsCode/WEB DESIGNING · GitHub](#)

# Navigator Object

Property	Description
appName	returns the name
appVersion	returns the version
appCodeName	returns the code name
cookieEnabled	returns true if cookie is enabled otherwise false
userAgent	returns the user agent
language	returns the language. It is supported in Netscape and Firefox only.
userLanguage	returns the user language. It is supported in IE only.
plugins	returns the plugins. It is supported in Netscape and Firefox only.
systemLanguage	returns the system language. It is supported in IE only.
mimeTypes[]	returns the array of mime type. It is supported in Netscape and Firefox only.

# Navigator Object Properties

Property	Description
platform	returns the platform e.g. Win32.
online	returns true if browser is online otherwise false.

[WEB DESIGNING/Module-4Javascript/Navigator object Example.html at main · TopsCode/WEB DESIGNING · GitHub](#)

# Screen Object

- The JavaScript screen object holds information of browser screen. It can be used to display screen width, height, colorDepth, pixelDepth etc.

## Syntax:

window.screen  
or  
screen

# Screen Object Property

Method	Description
width	returns the width of the screen
height	returns the height of the screen
availWidth	returns the available width
availHeight	returns the available height
colorDepth	returns the color depth
pixelDepth	returns the pixel depth.

[WEB DESIGNING/Module-4Javascript/Screen\\_object.html at main · TopsCode/WEB DESIGNING · GitHub](#)

# Document Object Model

- The document object represents the whole html document.
- When html document is loaded in the browser, it becomes a document object. It is the root element that represents the html document.
- It has properties and methods. By the help of document object, we can add dynamic content to our web page.

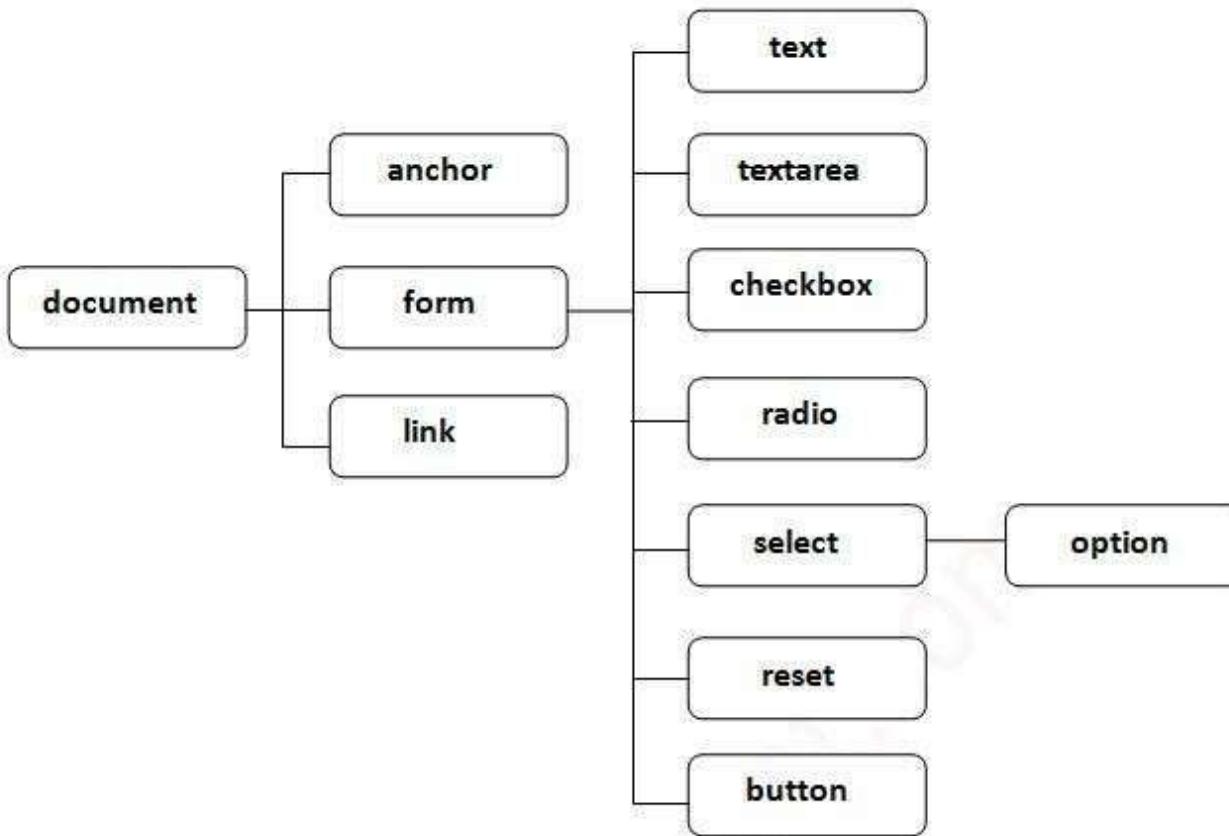
## Syntax:

window.document or  
document

Github link :

[https://github.com/TopsCode/WEB\\_DESIGNING/blob/main/Module-4Javascript/58dom-method.html](https://github.com/TopsCode/WEB_DESIGNING/blob/main/Module-4Javascript/58dom-method.html)

# Properties of document object



# Methods of Document Object

Method	Description
write("string")	writes the given string on the document.
writeln("string")	writes the given string on the document with newline character at the end.
getElementById()	returns the element having the given id value.
getElementsByName()	returns all the elements having the given name value.
getElementsByTagName()	returns all the elements having the given tag name.
getElementsByClassName()	returns all the elements having the given class name.

# How to select Tag Classes and Id

Method	Description	Returns
<code>getElementsByTagName("tagName")</code>	Select elements by <b>tag name</b>	<b>HTMLCollection (live)</b>
<code>getElementsByClassName("class")</code>	Select elements by <b>class name</b>	<b>HTMLCollection (live)</b>
<code>getElementById("id")</code>	Select a <b>single element</b> by ID	<b>Single element (or null )</b>
<code>querySelector("selector")</code>	Select the <b>first element</b> matching the selector	<b>Single element (or null )</b>
<code>querySelectorAll("selector")</code>	Select <b>all matching elements</b>	<b>NodeList (static)</b>

# document.getElementById()

- The document.getElementById() method returns the element of specified id.
- In the previous page, we have used document.form1.name.value to get the value of the input value. Instead of this, we can use document.getElementById() method to get value of the input text. But we need to define id for the input field.

# How to run Javascript

Running JavaScript on a webpage is straightforward, and there are several ways to do it, depending on how you structure your code. Below are the most common methods to run JavaScript, whether it's for simple testing or more complex applications

- =>Browser Console – Quick testing and debugging.
- =>Inline JavaScript – Adding JavaScript directly in HTML using `<script>`
- =>External JavaScript Files – Using .js files for structured, reusable code.
- => Node.js – Running JavaScript on the server-side.

# How to run Javascript

- =>Online Code Editors – Using platforms like JSFiddle or CodePen for experimenting.
- =>Frameworks – Using React, Angular, or Vue for more advanced web apps.

# Form Validation

- It is important to validate the form submitted by the user because it can have inappropriate values. So, validation is must to authenticate user.
- JavaScript provides facility to validate the form on the client-side so data processing will be faster than server-side validation. Most of the web developers prefer JavaScript form validation.
- Through JavaScript, we can validate name, password, email, date, mobile numbers and more fields.

Github link :

[https://github.com/TopsCode/WEB\\_DESIGNING/blob/main/Module-4Javascript/39Form-validate.html](https://github.com/TopsCode/WEB_DESIGNING/blob/main/Module-4Javascript/39Form-validate.html)

# Email Validation

- We can validate the email by the help of JavaScript.
- There are many criteria that need to be follow to validate the email id such as:
  1. email id must contain the @ and . character
  2. There must be at least one character before and after the @.
  3. There must be at least two characters after . (dot).

# JS DOM Nodes

In JavaScript, DOM nodes are represented as objects, and each object corresponds to a part of the HTML document. Here's a breakdown of DOM nodes and how to work with them in JavaScript: In html, there are various events which represents that some activity is performed by the user or by the browser.

## Types of DOM Nodes

1. Element Nodes: These are the nodes that represent HTML elements like `<div>`, `<p>` , `<a>` , etc.

# JS DOM Nodes

2. Text Nodes: These represent the textual content inside an element like `<p>`
3. Attribute Nodes: These represent the attributes of HTML elements like class, id, href, etc.
4. Comment Nodes: These represent comments in the HTML code.

# JS DOM Styling

## 1. Accessing the style Property:

```
=> element.style.propertyName = "value";
```

## 2 Setting Multiple Styles :

```
=>document.getElementById("myElement").setAttribute("style", "color: blue; font-size: 20px;");
```

## 3. Changing CSS Classes:

```
=>document.getElementById("myElement").classList.add("newClass");
```

```
=>document.getElementById("myElement").classList.remove("oldClass");
```

# JS DOM Styling

Other Useful Styling Methods :

**element.style.property**: Allows direct access to specific style properties.

**element.setAttribute('style', value)**: Overrides all inline styles at once.

**element.removeAttribute('style')**: Removes inline styles.

# JS DOM Styling

## 4. Getting Computed Styles:

```
=>const element = document.getElementById("myElement");
=>const computedStyle = window.getComputedStyle(element);
console.log(computedStyle.color);
```

## 5. Manipulating CSS Variables:

```
=>document.documentElement.style.setProperty('--main-color', 'green');
let color =
getComputedStyle(document.documentElement).getPropertyValue('--main-color');
```

# JS DOM Get Set Attributes

In JavaScript, you can use the DOM methods to get and set attributes of HTML elements. This is done through methods like `getAttribute()` and `setAttribute()`. Additionally, the `dataset` property can be used for working with data attributes. Here's a breakdown of how you can get and set attributes in the DOM:

1. Getting an Attribute : To retrieve the value of an attribute of an element, you can use the `getAttribute()` method. This method takes the attribute name as an argument and returns the current value of the attribute.

```
const attributeValue = element.getAttribute("attributeName");
```

# JS DOM Get Set Attributes

2. Setting an Attribute: You can modify the value of an attribute using the `setAttribute()` method. This method takes two arguments: the name of the attribute and the new value you want to assign to it.

```
element.setAttribute("attributeName", "newValue");
```

[WEB DESIGNING/Module-4Javascript/Get Attribute Set Attribute.html at main · TopsCode/WEB DESIGNING · GitHub](#)

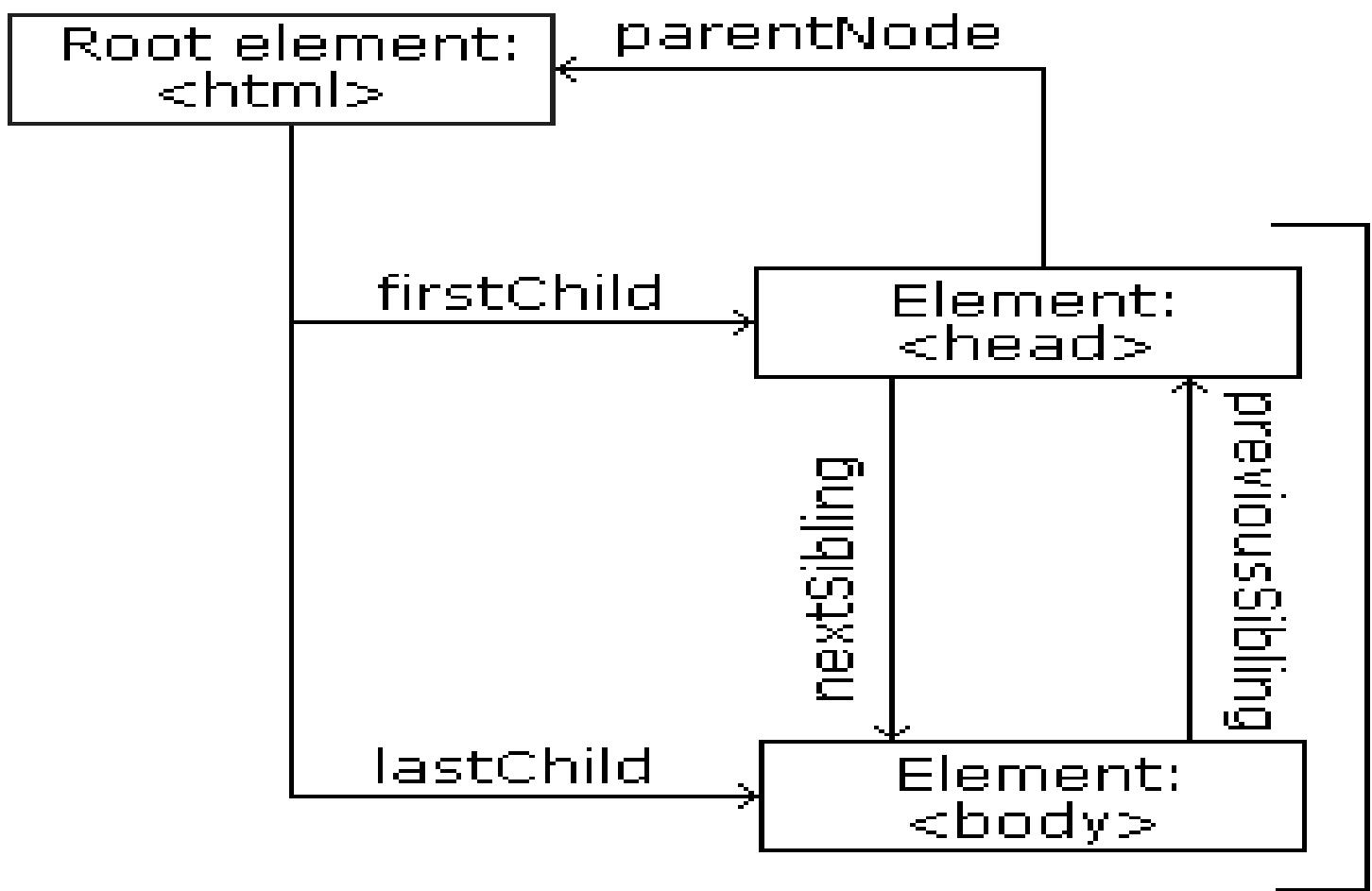
# JS DOM Navigation

DOM Navigation in JavaScript refers to the ability to traverse and navigate through the DOM tree to select or manipulate elements, move between parent and child nodes, siblings, and more. This is crucial for interacting with HTML elements dynamically in web development.

Key Concepts of DOM Navigation:

1. Parent Node: The element that contains the current element.  
`const parentElement = element.parentNode;`
2. Child Nodes: Elements that are nested inside the current element.  
`const childNodes = parentElement.childNodes;`
3. Sibling Nodes: Elements that share the same parent element.

# JS DOM Navigation



childNodes  
to <html>  
and siblings  
to each other

# JS Cookies

.Cookies are small pieces of data that are stored in the user's browser and are used to remember information across requests. In JavaScript, you can use cookies to store and retrieve data on the client-side.

## What is a Cookie?

A cookie is a small text file that is stored on the user's computer by the web browser. It contains information like session data, user preferences, authentication tokens, etc. Cookies have a set expiration time, and once set, they are sent back to the server with every HTTP request for the associated domain.

# JS Cookies

## **.Common Uses of Cookies:**

- ⇒ Remember user sessions: Keeping users logged in when they navigate to a different page or return to the website.
- ⇒ Store user preferences: Such as theme settings, language choices, etc.
- ⇒ Tracking and analytics: Cookies are often used for tracking user behavior on a website.

# JS Cookies

## ·Limitations of Cookies

- ⇒ Size Limit: Each cookie can store only a limited amount of data (usually around 4 KB). □
- ⇒ Security: Cookies can be vulnerable to cross-site scripting (XSS) and cross-site request forgery (CSRF) attacks if not handled properly.
- ⇒ Performance: Cookies are sent with every HTTP request to the domain, which can increase the load time and bandwidth usage.

# JavaScript Events

- The change in the state of an object is known as an Event.
- In html, there are various events which represents that some activity is performed by the user or by the browser.
- When javascript code is included in HTML, js react over these events and allow the execution.
- This process of reacting over the events is called Event Handling. Thus, js handles the HTML events via Event Handlers.

Github link :

[https://github.com/TopsCode/WEB\\_DESIGNING/blob/main/Module-4Javascript/61events.html](https://github.com/TopsCode/WEB_DESIGNING/blob/main/Module-4Javascript/61events.html)

# Mouse Events

Event Performed	Event Handler	Description
Click	onclick	When mouse click on an element
mouseover	onmouseover	When the cursor of the mouse comes over the element
mouseout	onmouseout	When the cursor of the mouse leaves an element
mousedown	onmousedown	When the mouse button is pressed over the element
mouseup	onmouseup	When the mouse button is released over the element
mousemove	onmousemove	When the mouse movement takes place.

[WEB DESIGNING/Module-4Javascript/Mouse Keyboard Events.html at main · TopsCode/WEB DESIGNING · GitHub](#)

# Keyboard Events

Event Performed	Event Handler	Description
keydown	onkeydown	When the user press and then release the key
keyup	onkeyup	When the user press and then release the key

[WEB DESIGNING/Module-4Javascript/Mouse\\_ Keyboard\\_ Events.html at main · TopsCode/WEB\\_DESIGNING - GitHub](#)

# Form Events

Event Performed	Event Handler	Description
Focus	onfocus	When the user focuses on an element
submit	onsubmit	When the user submits the form
Blur	onblur	When the focus is away from a form element
change	onchange	When the user modifies or changes the value of a form element

# Window/Document Events

Event Performed	Event Handler	Description
load	onload	When the browser finishes the loading of the page
unload	onunload	When the visitor leaves the current webpage, the browser unloads it
resize	onresize	When the visitor resizes the window of the browser

# addEventListener()

- The addEventListener() method is used to attach an event handler to a particular element. It does not override the existing event handlers.
- Events are said to be an essential part of the JavaScript. A web page responds according to the event that occurred.

# addEventListener()

- Events can be user-generated or generated by API's. An event listener is a JavaScript's procedure that waits for the occurrence of an event.
- The addEventListener() method is an inbuilt function of JavaScript. We can add multiple event handlers to a particular element without overwriting the existing event handlers.

# addEventListener()

**Syntax:** element.addEventListener(event, function, useCapture);

- event: It is a required parameter. It can be defined as a string that specifies the event's name.
- function: It is also a required parameter. It is a JavaScript function which responds to the event occur.
- useCapture: It is an optional parameter. It is a Boolean type value that specifies whether the event is executed in the bubbling or capturing phase. Its possible values are true and false. When it is set to true, the event handler executes in the capturing phase. When it is set to false, the handler executes in the bubbling phase. Its default value is false.

[WEB DESIGNING/Module-4Javascript/AddEventlister.html at main .](#)

[TopsCode/WEB DESIGNING - GitHub](#)

# Event Bubbling or Event Capturing

- **Bubbling:** in bubbling, the inner element's event is handled first, and then the outermost element's event will be handled.
- **Capturing:** in capturing the outer element's event is handled first, and then the innermost element's event will be handled.

```
addEventListener(event, function, useCapture);
```

# onclick event in Javascript

- The onclick event generally occurs when the user clicks on an element. It allows the programmer to execute a JavaScript's function when an element gets clicked.
- This event can be used for validating a form, warning messages and many more.
- Using JavaScript, this event can be dynamically added to any element.
- It supports all HTML elements except <html>, <head>, <title>, <style>, <script>, <base>, <iframe>, <bdo>, <br>, <meta>, and <param>. It means we cannot apply the onclick event on the given tags.

[WEB DESIGNING/Module-4Javascript/OnClick.html at  
main · TopsCode/WEB DESIGNING · GitHub](#)

# onclick event in JavaScript

- In HTML:

```
<element onclick = "fun()">
```

- In JavaScript:

```
object.onclick = function() { myScript };
```

- In JavaScript by using the **addEventListener()** method:

```
object.addEventListener("click", myScript);
```

# dblclick event

- The dblclick event generates an event on double click the element.
- The event fires when an element is clicked twice in a very short span of time.
- We can also use the JavaScript's addEventListener() method to fire the double click event.

[WEB DESIGNING/Module-4Javascript/Db Click.html](#)  
[at main · TopsCode/WEB DESIGNING · GitHub](#)

# dblclick event

- In HTML:

```
<element ondblclick = "fun()">
```

- In JavaScript:

```
object.ondblclick = function() { myScript };
```

- In JavaScript by using the addEventListener()  
method:

```
object.addEventListener("dblclick", myScript);
```

# onload in javascript

- In JavaScript, this event can apply to launch a particular function when the page is fully displayed. It can also be used to verify the type and version of the visitor's browser. We can check what cookies a page uses by using the `onload` attribute.
- In HTML, the `onload` attribute fires when an object has been loaded. The purpose of this attribute is to execute a script when the associated element loads.

[WEB DESIGNING/Module-4Javascript/Onload.html at  
main · TopsCode/WEB DESIGNING · GitHub](#)

# onload in javascript

- In HTML, the onload attribute is generally used with the <body> element to execute a script once the content (including CSS files, images, scripts, etc.) of the webpage is completely loaded. It is not necessary to use it only with <body> tag, as it can be used with other HTML elements.
- The difference between the document.onload and window.onload is: document.onload triggers before the loading of images and other external content. It is fired before the window.onload. While the window.onload triggers when the entire page loads, including CSS files, script files, images, etc.

# onload in javascript

## Syntax:

```
window.onload =  
fun()
```

# onresize event in javascript

- The onresize event in JavaScript generally occurs when the window has been resized. To get the size of the window, we can use the Java Script's `window.outerWidth` and `window.outerHeight` events.
- We can also use the JavaScript's properties such as `innerWidth`, `innerHeight`, `clientWidth`, `ClientHeight`, `offsetWidth`, `offsetHeight` to get the size of an element.

# onresize event in javascript

- In HTML, we can use the onresize attribute and assign a JavaScript function to it.
- We can also use the JavaScript's addEventListener() method and pass a resize event to it for greater flexibility.

# Syntax

- **In HTML:**

```
<element onresize = "fun()">
```

- **In JavaScript:**

```
object.onresize = function() { myScript };
```

- **In JavaScript by using the addEventListener()  
method:**

```
object.addEventListener("resize", myScript);
```

# JS Arrays

- In JavaScript, arrays are used to store multiple values in a single variable. Arrays can hold a collection of data types, such as numbers, strings, objects, or even other arrays, and allow easy manipulation and access to this data.

[WEB DESIGNING/Module-4Javascript/Array.html at  
main · TopsCode/WEB DESIGNING · GitHub](#)

# JS Sorting Arrays

- In JavaScript, you can sort arrays using the `sort()` method. This method can be used to sort arrays of strings, numbers, or other objects. By default, the `sort()` method sorts elements as strings in lexicographical (alphabetical) order. However, you can also customize the sorting behavior by passing a comparison function to the `sort()` method.

```
const fruits = ['Banana', 'Apple', 'Mango', 'Cherry']; fruits.sort();
console.log(fruits);
```

# How to see Traffic in Our Website

To monitor traffic on your website, there are several tools and techniques you can use to track user activity, visits, page views, and other important metrics. Here's how you can see the traffic on your website:

To track traffic on your website, you can choose from a variety of tools depending on the depth of reporting you need:

# How to see Traffic in Our Website

1. Google Analytics: The most comprehensive tool for monitoring traffic, user behavior, and conversions.
2. Hosting Analytics Tools: Basic traffic data provided by your web host.
3. Alternative Analytics Tools: Matomo, Clicky, Mixpanel, etc
4. Social Media Insights: Useful if your traffic comes from social media platforms.
5. Server Logs: If you need more detailed technical insights.
6. Real-Time Analytics: Tools like Hotjar, Clicky, and Google Analytics Real-Time reports.

# Module - 7

## [ CSS Preprocessors ]

# SASS | Introduction

Sass is the short form of Syntactically Awesome Style Sheet. It is an upgrade to Cascading Style Sheets (CSS). Sass is a CSS pre-processor. It is fully compatible with every version of CSS. Sass reduces the repetition of CSS and therefore saves time. It was designed by Hampton Catlin and developed by Natalie Weizenbaum in 2006. Sass is free to download and use.

## Pre-Requisites:

HTML

CSS

**Working:** A web browser does not understand the Sass code itself. That's why you will require a Sass pre-processor to change Sass codes into simple standard CSS.

The above process is known as **transpiling**. So, you will be required to give the transpiler some Sass codes and then in return get some CSS codes back.

**Note:** Transpiling is the term used for taking a source code input of one language and translating it into an output of another language.

# SASS | Introduction

**File Type:** All Sass files must have the “.scss” file extension.

**Comments:** Sass supports the standard CSS comments “/\* comment \*/”, and along with that it also supports in-line comments “// comment”

## **System Requirements and Installation of SASS:**

**Operating system:** Sass is platform-independent.

**Browser support:** We can't use SASS natively in the browsers. To use SASS:  
Install tools like sass.js on the client side.

Use transpiler like node-sass, dart-sass to transpile .scss files into .css file.

**Programming language:** Sass is based on Ruby.

For more info regarding Sass and installation visit the official Sass website <https://sass-lang.com/>

**A basic example of Sass.** Let us suppose a website has basically 3 fonts throughout. It will be a mess if it has to write the same font style again and again. Instead of writing font values, again and again, you can use the below Sass.

# SASS | Introduction

**Example:**

```
$font_1: Algerian;  
$font_2: Times New Roman;  
$font_3: Serif;
```

```
.main {  
    font: $font_1;  
}
```

```
.menu-1 {  
    font: $font_2;  
}
```

```
.menu-2 {  
    font: $font_3;  
}
```

# SASS | Introduction

This would result in the following CSS Output:

```
.main {  
    font: Algerian;  
}
```

```
.menu-1 {  
    font: Times New Roman;  
}
```

```
.menu-2 {  
    font: Serif;  
}
```

[https://github.com/TopsCode/WEB\\_DESIGNING/tree/main/Module-4%20CSS%20Preprocessors\(%20SASS-LESS%20\)/Scss/Scss](https://github.com/TopsCode/WEB_DESIGNING/tree/main/Module-4%20CSS%20Preprocessors(%20SASS-LESS%20)/Scss/Scss)

# SASS | Interpolation

Interpolation is basically an insertion. Interpolation allows us to interpolate sass expressions into a simple SASS or CSS code. Means, you can define ( some part or the whole ) selector name, property name, CSS at-rules, quoted or unquoted strings etc, as a variable. Interpolation is a new principle and it is widely used in SASS.

To interpolate an expression we need to wrap the expression using #{}.

## **Syntax:**

.....#\${variable\_name}..... where ..... represents some text.

See the example below to get more understanding

# SASS | Interpolation

## SASS file:

```
@mixininterpolation($changeable, $val, $val2, $prop1, $prop2)
{
    background-#{$changeable}: $val;
    position: $val2;
    #{$prop1}: 0px;
    #{$prop2}: 0px;
}
.blockarea{
    @include interpolation("image", url("img.png"), absolute, top, right);
}
.anotherbloakarea{
    @include interpolation("color", lightgray, absolute, top, left);
}
```

# SASS | Interpolation

## Compiled CSS file:

```
.blockarea {  
background-image: url("img.png");  
position: absolute;  
top: 0px;  
right: 0px;  
}  
.anotherbloakarea { background-color: lightgray; position: absolute; top: 0px; left: 0px;}
```

**Interpolation in SASS expressions always returns an unquoted string, no matter whether the string is quoted or not.**

## Uses of Interpolation:

To use dynamically created names as a property name, a variable name or for any other same purposes.

To create a very reusable code; where you can define a property name, strings, selector names etc, as a variable.

# SASS | Nesting

SASS Nesting makes our work very efficient, we don't have to repeat outer selectors again and again. We can write nested selectors in the order they appear in the HTML file (i.e., Parent-Outer and Child-Inner Selector form ). SASS will automatically do the work of combining.

See the example below:

## SASS file:

```
ul {  
    list-style-type: none;  
    li {  
        display: inline-block;  
        a {  
            text-decoration: none;  
            display: inline-block;  
            padding: 10px 15px;  
            color: blue;  
        }  
    }  
}
```

[https://github.com/TopsCode/WEB\\_DESIGNING/blob/main/Module-4%20CSS%20Preprocessors\(%20SASS-LESS%20\)/Scss/Scss/5.nestedrules.scss](https://github.com/TopsCode/WEB_DESIGNING/blob/main/Module-4%20CSS%20Preprocessors(%20SASS-LESS%20)/Scss/Scss/5.nestedrules.scss)

# SASS | Nesting

**Compiled CSS file:**

```
ul {  
    list-style-type: none;  
}  
ul li {  
    display: inline-block;  
}  
ul li a  
{  
    text-decoration: none;  
    display: inline-block;  
    padding: 10px 15px;  
    color: blue;  
}
```

# SASS | Nesting

SASS also allows nesting of selectors with different combinators. You can put combinators either at the start of the inner selector or at the end of the outer selector or in the middle of two.

See the example below:

**SASS file:**

```
ul {  
    + li {  
        display: inline-block;  
    }  
}  
li {  
    > {  
        a {  
            text-decoration: none;  
        }  
    }  
}
```

# SASS | Nesting

```
p ~ {  
    span  
    {  
        text-decoration-line: underline;  
        text-decoration-style: double;  
    }  
}
```

## Compiled CSS file:

```
ul + li  
{  display: inline-block;  
}  
li > a  
{  text-decoration: none;  
} p ~ span  
{  text-decoration-line: underline;  
text-decoration-style: double;}
```

# SASS | @mixin and @include

Mixins can be used to group styles that can be assigned different values and can be used multiple times like function. We can also pass arguments in mixin like function, which allows us to write reusable code.

Mixins are defined using @mixin at-rule and it can be used in the current context using @include at-rule.

Mixins can be used in two ways: **Without Arguments and With Arguments.**

**Without Arguments:** This type of mixin can be used, when we do not need to change the values of the properties i.e., we want to use a group of properties again and again with the same values of properties.

## Syntax:

To define mixin: @mixinname\_of\_mixin {...}

To use mixin in the current block: @include name\_of\_mixin;

# SASS | @mixin and @include

```
@mixin block-name{  
    property-1: value;  
    property-2: value;  
    ...    property-n: value;  
} block-name2{  
    @include block-name;  
}
```

[https://github.com/TopsCode/WEB\\_DESIGNING/blob/main/Module-4%20CSS%20Preprocessors\(%20SASS-LESS%20\)/04-include-extend.scss](https://github.com/TopsCode/WEB_DESIGNING/blob/main/Module-4%20CSS%20Preprocessors(%20SASS-LESS%20)/04-include-extend.scss)

# SASS | @mixin and @include

## SCSS file:

```
@mixin first-mixin {  
    width: 100%;  
    overflow: hidden;  
color: gray;  
}  
@mixin second-mixin {  
    @include first-mixin;  
ul {  
    list-style-type: none;  
}  
li {  
    display: inline-block;  
    width: 100px;  
padding: 5px 10px;  }
```

# SASS | @mixin and @include

```
li a {  
    text-decoration: none;  
}  
}  
  
navigationbar {  
    @include second-mixin;  
}
```

# SASS | @mixin and @include

**Compiled CSS file:**

```
navigationbar {  
    width: 100%;  
    overflow: hidden;  
    color: gray;}  
navigationbar ul  
{  list-style-type: none;  
}  
navigationbar li {  
    display: inline-block;  
    width: 100px;  
    padding: 5px 10px;  
}  
navigationbar li a  
{  text-decoration: none;}
```

# SASS | @mixin and @include

**With Arguments:** This type of mixin can be used, when we want to use a group of properties again and again with the different values and we can pass the different values using arguments like function. Here the concept of the default value of argument is same as any other programming language.

## Syntax:

To define mixin: @mixinname\_of\_mixin (arguments...) {...}

To use mixin in the current block: @include name\_of\_mixin (arguments...);

// Here default values are optional

```
@mixin block-name($parameter1, $parameter2: default, ...)
```

```
{   property-1: $parameter1;
```

```
  property-2: $parameter2;
```

```
  // You can use all the parameters // same as variables}
```

```
block-name2 {
```

```
  @include block-name($argument1, $argument2, ...);}
```

# SASS | @mixin and @include

## SASS file:

```
// Here blue is default value of $three
@mixin first-mixin($one, $two, $three: blue)
{
  width: $one;
  overflow: $two;
  color: $three;
} @mixin second-mixin($one, $two, $three, $four)
{
  // Don't need to pass the third argument if
  // the default value is same as your argument.
@include first-mixin($one, $two /*, Default*/);
ul {
  list-style-type: none;
}
li {
  display: inline-block;
  width: $four;
  padding: 5px 10px; }
```

# SASS | @mixin and @include

```
li a {  
    text-decoration: none;  
}  
}  
navigationbar {  
    @include second-mixin(100%, hidden, blue, 100px);  
}
```

# SASS | @mixin and @include

## Compiled CSS file:

```
navigationbar {  
    width: 100%;  
    overflow: hidden;  
    color: blue  
}  
navigationbar ul  
{  list-style-type: none;  
}  
navigationbar li  
{  display: inline-block;  
    width: 100px;  
    padding: 5px 10px;  
} navigationbar li a {  
    text-decoration: none;}
```

# SASS | Map Functions

The SASS Map data-type is used to display one or more key-value pairs. Along with the map functions shown in the below lists, you can also use any of the SASS list functions with maps as well.

The following lists contains all map functions in SASS:

**1) map-has-key(\$map, \$key) function:** This function returns a Boolean value that checks whether the given map contains the given key or not.

**Example:**

```
map-has-key(("red": #ff0000, "yellow": #ffff00), blue)
```

**Output:**

```
false
```

**2) map-merge(\$map1, \$map2) function:** This function returns a map containing \$map2 joined to the end of \$map1.

**Example:**

```
map-merge(("red": #ff0000, "yellow": #ffff00), ("blue": #0000ff))
```

**Output:**

```
("red": #ff0000, "yellow": #ffff00, "blue": #0000ff)
```

# SASS | Map Functions

**3) map-keys(\$map) function:** This function returns the list of the keys in the given map.

**Example:**

```
map-keys("red": #ff0000, "yellow": #ffff00)
```

**Output:**

```
("red", "yellow")
```

**4) map-remove(\$map, \$keys) function:** This function returns a map without the given keys.

**Example:**

```
map-remove(("red": #ff0000, "yellow": #ffff00), "red")
```

**Output:**

```
("yellow": #ffff00)
```

**5) map-values(\$map) function:** This function returns the list of the values in the given map.

**Example:**

```
map-values(("red": #ff0000, "yellow": #ffff00))
```

**Output:**

```
(#ff0000, #ffff00)
```

[https://github.com/TopsCode/WEB\\_DESIGNING/blob/main/Module-4%20CSS%20Preprocessors\(%20SASS-LESS%20\)/14.MapFunctions.scss](https://github.com/TopsCode/WEB_DESIGNING/blob/main/Module-4%20CSS%20Preprocessors(%20SASS-LESS%20)/14.MapFunctions.scss)

# SASS | Map Functions

6) **map-get(\$map, \$key)** function: This function returns the value associated with the given key.

**Example:**

```
map-get(("blue": #0000ff, "yellow": #ffff00), "blue")
```

**Output:**

```
#0000ff
```

# SASS | List Functions

The SASS list functions are used to modify new lists. The lists need to be created inside circular brackets in order to differentiate from others. SASS list can not be changed hence in some cases new lists are created.

Just like the string functions, SASS lists are one-based (not zero-based) indexed meaning the first element of a string is present at index 1 (not at index 0).

The following lists represents the all list functions in SASS:

**1. list-separator(\$list) function:** This function returns the name of the separator used in the list as a string.

**Example:**

CSS

```
list-separator((1, 2, 3))
```

**Output:**

"comma"

[https://github.com/TopsCode/WEB\\_DESIGNING/blob/main/Module-4%20CSS%20Preprocessors\(%20SASS-LESS%20\)/13.ListFunctions.scss](https://github.com/TopsCode/WEB_DESIGNING/blob/main/Module-4%20CSS%20Preprocessors(%20SASS-LESS%20)/13.ListFunctions.scss)

# SASS | List Functions

**2. zip(\$lists) function:** This function divides the values of the lists given into a single multi-dimensional list.

**Example:**

CSS

```
zip(1px2px3px, soliddasheddotted, redgreenblue)
```

**Output:**

1px solid red, 2px dashed green, 3px dotted blue

**3. join(\$list1, \$list2, [\$separator]) function:** This function appends \$list2 to the end of \$list1. The separator argument could contain the values “comma, space or auto”. The auto value, which is also the default value, will use the separator in the first list.

**Example 1:**

CSS

```
join(123, 456)
```

**Output:**

(1 2 3 4 5 6)

# SASS | List Functions

## Example 2:

CSS

```
join((1, 2, 3), (456), comma)
```

## Output:

```
(1, 2, 3, 4, 5, 6)
```

## Example 3:

CSS

```
join((1, 2, 3), (456), space)
```

## Output:

```
(1 2 3 4 5 6)
```

# SASS | List Functions

**4. nth(\$list, \$n) function:** This function returns the  $n^{\text{th}}$  element of the list.

**Example:**

CSS

```
nth(5467891, 4)
```

**Output:**

7

**5. set-nth(\$list, \$n, \$value) function:** This function sets the  $n^{\text{th}}$  element of the list to the value provided.

**Example:**

CSS

```
set-nth(5467891, 3, 5)
```

**Output:**

(5 4 5 7 8 9 1)

# SASS | List Functions

**6. index(\$list, \$value) function:** This function returns the element at the specific index position called by value.

**Example:**

CSS

```
index((5642397), 4)
```

**Output:**

2

**7. length(\$list) function:** This function returns the number of elements in the list.

**Example:**

CSS

```
length(5467891)
```

**Output:**

7

# SASS | List Functions

**8. is-bracketed(\$list) function:** This function checks whether the list has any square brackets or not.

**Example 1:**

CSS

```
is-bracketed([a b c])
```

**Output:**

true

**Example 2:**

CSS

```
is-bracketed(a b c)
```

**Output:**

false

# SASS | List Functions

**9. append(\$list1, \$val, [\$separator]) function:** This function appends a single value provided to the end of the list. If the separator argument is provided (“auto” being the default separator), the complete list will follow the separator function.

**Example 1:**

CSS

```
append((1, 2, 3), 4, comma)
```

**Output:**

```
(1, 2, 3, 4)
```

**Example 2:**

CSS

```
append((1, 2, 3), 4, space)
```

**Output:**

```
(1 2 3 4)
```

# Less | Leaner Style Sheets

Less (which stands for Leaner Style Sheets) is a backwards-compatible language extension for CSS. This is the official documentation for Less, the language and Less.js, the JavaScript tool that converts your Less styles to CSS styles.

Because Less looks just like CSS, learning it is a breeze. Less only makes a few convenient additions to the CSS language, which is one of the reasons it can be learned so quickly.

*For detailed documentation on Less language features, see [Features](#)*

*For a list of Less Built-in functions, see [Functions](#)*

*For detailed usage instructions, see [Using Less.js](#)*

*For third-party tools for Less, see [Tools](#)*

What does Less add to CSS? Here's a quick overview of features

[\*\*WEB DESIGNING/Module-4 CSS Preprocessors\( SASS-LESS \)/Less/Less/Less\\_Lernal\\_StyleSheet.zip at main · TopsCode/WEB DESIGNING - GitHub\*\*](#)

# Less | Variables

These are pretty self-explanatory:

```
@width:10px;  
@height:@width + 10px;
```

```
#header {  
width: @width;  
height: @height;  
}
```

Outputs:

```
#header {  
width: 10px;  
height: 20px;  
}
```

**Github link :**

[https://github.com/TopsCode/WEB\\_DESIGNING/blob/main/Module-4%20CSS%20Preprocessors\(%20SASS-LESS%20\)/05-less-variable.less](https://github.com/TopsCode/WEB_DESIGNING/blob/main/Module-4%20CSS%20Preprocessors(%20SASS-LESS%20)/05-less-variable.less)

# Less | Mixins

Mixins are a way of including ("mixing in") a bunch of properties from one rule-set into another rule-set. So say we have the following class:

```
.bordered {  
border-top: dotted 1px black;  
border-bottom: solid 2px black;  
}
```

And we want to use these properties inside other rule-sets. Well, we just have to drop in the name of the class where we want the properties, like so:

```
#menua {  
color: #111;  
.bordered();  
}
```

GitHub link :

[https://github.com/TopsCode/WEB\\_DESIGNING/blob/main/Module-4%20CSS%20Preprocessors\(%20SASS-LESS%20\)/06-.less-mixins.less](https://github.com/TopsCode/WEB_DESIGNING/blob/main/Module-4%20CSS%20Preprocessors(%20SASS-LESS%20)/06-.less-mixins.less)

# Less | Mixins

```
.posta {  
color: red;  
.bordered();  
}
```

The properties of the .bordered class will now appear in both #menu a and .posta. (Note that you can also use #ids as mixins.)

[Learn More About Mixins](#)

# Less | Nesting

Less gives you the ability to use nesting instead of, or in combination with cascading. Let's say we have the following CSS:

```
#header {  
color: black;  
}  
  
#header.navigation {  
font-size: 12px;  
}  
  
#header.logo {  
width: 300px;  
}
```

**GitHub link :**

[https://github.com/TopsCode/WEB\\_DESIGNING/blob/main/Module-4%20CSS%20Preprocessors\(%20SASS-LESS%20\)/07-nesting.less](https://github.com/TopsCode/WEB_DESIGNING/blob/main/Module-4%20CSS%20Preprocessors(%20SASS-LESS%20)/07-nesting.less)

# Less | Nesting

In Less, we can also write it this way:

```
#header {  
color: black;  
.navigation {  
font-size: 12px;  
}  
.logo {  
width: 300px;  
}  
}
```

# Less | Nesting

The resulting code is more concise, and mimics the structure of your HTML.

You can also bundle pseudo-selectors with your mixins using this method. Here's the classic clearfix hack, rewritten as a mixin (& represents the current selector parent):

```
.clearfix {  
display: block;  
zoom: 1;  
&:after {  
content: " ";  
display: block;  
font-size: 0;  
height: 0;  
clear: both;  
visibility: hidden;  
}  
}
```

# Nested At-Rules and Bubbling

At-rules such as @media or @supports can be nested in the same way as selectors. The at-rule is placed on top and relative order against other elements inside the same ruleset remains unchanged. This is called bubbling.

```
.component {  
    width: 300px;  
    @media (min-width: 768px) {  
        width: 600px;  
        @media (min-resolution: 192dpi) {  
            background-image: url(/img/retina2x.png);  
        }  
    }  
    @media (min-width: 1280px) {  
        width: 800px;  
    }  
}
```

# Nested At-Rules and Bubbling

outputs:

```
.component {  
width: 300px;  
}@media (min-width: 768px) {  
.component {  
width: 600px;  
}  
}@media (min-width: 768px) and (min-resolution: 192dpi) {  
.component {  
background-image: url(/img/retina2x.png);  
}  
}@media (min-width: 1280px) {  
.component {  
width: 800px;}}
```

# Operations

Arithmetical operations +, -, \*, / can operate on any number, color or variable. If it is possible, mathematical operations take units into account and convert numbers before adding, subtracting or comparing them. The result has leftmost explicitly stated unit type. If the conversion is impossible or not meaningful, units are ignored. Example of impossible conversion: px to cm or rad to %.

*// numbers are converted into the same units*

```
@conversion-1:5cm + 10mm; // result is 6cm
```

```
@conversion-2:2 - 3cm - 5mm; // result is -1.5cm
```

*// conversion is impossible*

```
@incompatible-units:2 + 5px - 3cm; // result is 4px
```

*// example with variables*

```
@base:5%;
```

```
@filler:@base * 2; // result is 10%
```

```
@other:@base + @filler; // result is 15%
```

**Github link :**

[https://github.com/TopsCode/WEB\\_DESIGNING/blob/main/Module-4%20CSS%20Preprocessors\(%20SASS-LESS%20\)/Less/Less/11.Math%20Operation.less](https://github.com/TopsCode/WEB_DESIGNING/blob/main/Module-4%20CSS%20Preprocessors(%20SASS-LESS%20)/Less/Less/11.Math%20Operation.less)

# Operations

Multiplication and division do not convert numbers. It would not be meaningful in most cases - a length multiplied by a length gives an area and css does not support specifying areas. Less will operate on numbers as they are and assign explicitly stated unit type to the result.

```
@base:2cm * 3mm; // result is 6cm
```

You can also do arithmetic on colors:

```
@color: (#224488 / 2); // result is #112244
```

```
background-color: #112244 + #111; // result is #223355
```

However, you may find Less's [Color Functions](#) more useful.

From 4.0, No division is performed outside of parens using / operator.

```
@color:#222 / 2; // results in `#222 / 2`, not #111
```

```
background-color: (#FFFFFF / 16); // results is #101010
```

You can change [Math](#) setting, if you want to make it always work, but not recommended

Github link :

[https://github.com/TopsCode/WEB\\_DESIGNING/blob/main/Module-4%20CSS%20Preprocessors\(%20SASS-LESS%20\)/Less/Less/11.Math%20OPeration.less](https://github.com/TopsCode/WEB_DESIGNING/blob/main/Module-4%20CSS%20Preprocessors(%20SASS-LESS%20)/Less/Less/11.Math%20OPeration.less)

# calc() exception

*Released [v3.0.0](#)*

For CSS compatibility, calc() does not evaluate math expressions, but will evaluate variables and math in nested functions.

```
@var:50vh/2;
```

```
width: calc(50% + (@var - 20px)); // result is calc(50% + (25vh - 20px))
```

# Escaping

Escaping allows you to use any arbitrary string as property or variable value. Anything inside ~"anything" or ~'anything' is used as is with no changes except [interpolation](#).

```
@min768:~"(min-width: 768px)";  
.element {  
@media@min768 {  
font-size: 1.2rem;  
}  
}
```

results in:

```
@media (min-width: 768px) {  
.element {  
font-size: 1.2rem;  
}  
}
```

[WEB DESIGNING/Module-4 CSS Preprocessors\( SASS-LESS \)/Less/Less/Escaping.zip at main · TopsCode/WEB DESIGNING - GitHub](#)

# Escaping

Note, as of Less 3.5, you can simply write:

```
@min768: (min-width: 768px);  
.element {  
@media@min768 {  
font-size: 1.2rem;  
}  
}
```

In 3.5+, many cases previously requiring "quote-escaping" are not needed.

# Functions

Less provides a variety of functions which transform colors, manipulate strings and do maths. They are documented fully in the [function reference](#).

Using them is pretty straightforward. The following example uses percentage to convert 0.5 to 50%, increases the saturation of a base color by 5% and then sets the background color to one that is lightened by 25% and spun by 8 degrees:

```
@base:#f04615;  
@width:0.5;
```

```
.class {  
  width: percentage(@width); // returns '50%'  
  color: saturate(@base, 5%);  
  background-color: spin(lighten(@base, 25%), 8);  
}
```

[\*\*See: Function Reference\*\*](#)

# Namespaces and Accessors

(Not to be confused with [CSS @namespace](#) or [namespace selectors](#)).

Sometimes, you may want to group your mixins, for organizational purposes, or just to offer some encapsulation. You can do this pretty intuitively in Less. Say you want to bundle some mixins and variables under #bundle, for later reuse or distributing:

```
#bundle() {  
  .button {  
    display: block;  
    border: 1px solid black;  
    background-color: grey;  
    &:hover {  
      background-color: white;  
    } }  
  .tab{ ... }  
  .citation { ... }  
}
```

[WEB DESIGNING/Module-4 CSS Preprocessors\( SASS-LESS \)/Less/Less/NameSpaces And Accessors.zip at main . TopsCode/WEB DESIGNING - GitHub](#)

# Namespaces and Accessors

Now if we want to mixinthe .button class in our #header a, we can do:

```
#header a {  
color: orange;  
#bundle.button(); // can also be written as #bundle > .button  
}
```

Note: append () to your namespace (e.g. #bundle()) if you don't want it to appear in your CSS output i.e. #bundle .tab.

[WEB DESIGNING/Module-4 CSS Preprocessors\( SASS-LESS \)/Less/Less/NameSpaces And Accessors.zip at main - TopsCode/WEB DESIGNING · GitHub](#)

# Maps

As of Less 3.5, you can also use mixins and rulesets as maps of values.

```
#colors() {  
    primary: blue;  
    secondary: green;  
}  
.button {  
    color: #colors[primary];  
    border: 1px solid #colors[secondary];  
}
```

This outputs, as expected:

```
.button {  
    color: blue;  
    border: 1px solid green;  
}
```

**Github link :**

[See also: Maps](#)

[https://github.com/TopsCode/WEB\\_DESIGNING/blob/main/Module-4%20CSS%20Preprocessors\(%20SASS-LESS%20\)/Less/Less/14.MapFunctions.less](https://github.com/TopsCode/WEB_DESIGNING/blob/main/Module-4%20CSS%20Preprocessors(%20SASS-LESS%20)/Less/Less/14.MapFunctions.less)

# Scope

Scope in Less is very similar to that of CSS. Variables and mixins are first looked for locally, and if they aren't found, it's inherited from the "parent" scope.

```
@var: red;  
  
#page {  
  @var: white;  
  #header {  
    color: @var; // white  
  }  
}
```

# Scope

Like CSS custom properties, mixin and variable definitions do not have to be placed before a line where they are referenced. So the following Less code is identical to the previous example:

```
@var: red;  
  
#page {  
#header {  
color: @var; // white  
}  
@var: white;  
}
```

[See also: Lazy Loading](#)

# Comments

Both block-style and inline comments may be used:

```
/* One heck of a block
```

```
 * style comment! */
```

```
@var: red;
```

```
// Get in line!
```

```
@var: white;
```

# Importing

Importing works pretty much as expected. You can import a .less file, and all the variables in it will be available. The extension is optionally specified for .less files.

```
@import"library"; // library.less  
@import"typo.css";
```

# Variables

Control commonly used values in a single location.

**GitHub link :**

[https://github.com/TopsCode/WEB\\_DESIGNING/blob/main/Module-4%20CSS%20Preprocessors\(%20SASS-LESS%20\)/Less/Less/1.SassVariables.less](https://github.com/TopsCode/WEB_DESIGNING/blob/main/Module-4%20CSS%20Preprocessors(%20SASS-LESS%20)/Less/Less/1.SassVariables.less)

# Overview

It's not uncommon to see the same value repeated dozens *if not hundreds of times* across your stylesheets:

```
a,  
.link {color: #428bca;}  
.widget  
{color: #fff;background: #428bca;  
}
```

# Overview

Variables make your code easier to maintain by giving you a way to control those values from a single location:

```
// Variables  
@link-color:#428bca; // sea blue  
@link-color-hover:darken  
(@link-color, 10%); // Usage a  
.link {color: @link-color;}  
a:hover {color: @link-color-hover;}  
.widget  
{color: #fff;  
background: @link-color;  
}
```

# Variable Interpolation

The examples above focused on using variables to control *values in CSS rules*, but they can also be used in other places as well, such as selector names, property names, URLs and @import statements.

## *Selectors*

v1.4.0

```
// Variables@my-selector: banner; // Usage.@{my-selector} {font-weight: bold;line-height: 40px;margin: 0 auto;}
```

Compiles to:

```
.banner {font-weight: bold;line-height: 40px;margin: 0 auto;}
```

[WEB DESIGNING/Module-4 CSS Preprocessors\( SASS-LESS \)/Less/Less/Variable\\_interpolation.zip at main · TopsCode/WEB\\_DESIGNING · GitHub](#)

# Variable Interpolation

## *URLs*

```
// Variables@images:"..../img"; // Usagebody {color: #444;background: url("@{images}/white-sand.png");}
```

## *Import Statements*

v1.4.0

Syntax: @import "@{themes}/tidal-wave.less";

Note that before v2.0.0, only variables which have been declared in the root or current scope were considered and that only the current file and calling files were considered when looking for a variable.

Example:

```
// Variables@themes:"..../src/themes"; // Usage@import"@{themes}/tidal-wave.less";
```

## *Properties*

v1.6.0

```
@property:color; .widget {@{property}: #0ee;background-@{property}: #999;}
```

Compiles to:

```
.widget {color: #0ee;background-color: #999;}
```

[\*\*WEB DESIGNING/Module-4 CSS Preprocessors\( SASS-LESS \)/Less/Less/Variable interpolation.zip at main · TopsCode/WEB DESIGNING · GitHub\*\*](#)

# Variable Variables

In Less, you can define a variable's name using another variable.

```
@primary: green;  
@secondary: blue;  
.section {  
  @color: primary;  
  .element {color:  
    @@color;  
  }  
}
```

Which compiles to:

```
.section  
.element {color: green;  
}
```

# Lazy Evaluation

Variables do not have to be declared before being used.

Valid Less snippet:

```
.lazy-eval {width: @var;} @var:@a;@a:9%;
```

this is valid Less too:

```
.lazy-eval {width: @var;@a:9%;} @var:@a;@a:100%;
```

both compile into:

```
.lazy-eval {width: 9%;}
```

When defining a variable twice, the last definition of the variable is used, searching from the current scope upwards. This is similar to css itself where the last property inside a definition is used to determine the value.

For instance:

```
@var:0;.class {@var:1;.brass {@var:2;three: @var;@var:3; }one: @var;}
```

Compiles to:

```
.class {one: 1;}.class.brass {three: 3;}
```

[WEB DESIGNING/Module-4 CSS Preprocessors\( SASS-LESS \)/Less/Less/Lazy Evaluation.zip at main - TopsCode/WEB DESIGNING - GitHub](#)

# Lazy Evaluation

Essentially, each scope has a "final" value, similar to properties in the browser, like this example using custom properties:

```
.header {--color: white;color: var(--color); // the color is black--color: black;}
```

This means that, unlike other CSS pre-processing languages, Less variables behave very much like CSS's.

[WEB DESIGNING/Module-4 CSS Preprocessors\( SASS-LESS \)/Less/Less/Lazy Evaluation.zip at main . TopsCode/WEB DESIGNING - GitHub](#)

# Properties as Variables (NEW!)

v3.0.0

You can easily treat properties like variables using the \$prop syntax. Sometimes this can make your code a little lighter.

```
.widget {color: #efefef;background-color: $color;}
```

Compiles to:

```
.widget {color: #efefef;background-color: #efefef;}
```

Note that, like variables, Less will choose the last property within the current/parent scope as being the "final" value.

```
.block {color: red; .inner {background-color: $color; }color: blue;  
}
```

Compiles to:

```
.block {color: red; color: blue; } .block.inner {background-color: blue;  
}
```

# Default Variables

We sometimes get requests for default variables - an ability to set a variable only if it is not already set. This feature is not required because you can easily override a variable by putting the definition afterwards.

For instance:

```
// library@base-color: green; @dark-color: darken(@base-color, 10%); // use of  
library@import "library.less"; @base-color: red;
```

This works fine because of [Lazy Loading](#) - @base-color is overridden and @dark-color is a dark red.

# Parent Selectors

Referencing parent selectors with &

The & operator represents the parent selectors of a [nested rule](#) and is most commonly used when applying a modifying class or pseudo-class to an existing selector:

```
a {color: blue; &:hover {color: green;}}
```

results in:

```
a {color: blue;} a:hover {color: green;}
```

} Notice that without the &, the above example would result in a :hover rule (a descendant selector that matches hovered elements inside of  tags) and this is not what we typically would want with the nested :hover.

The "parent selectors" operator has a variety of uses. Basically any time you need the selectors of the nested rules to be combined in other ways than the default. For example another typical use of the & is to produce repetitive class names:

```
.button {&-ok {background-image: url("ok.png"); } &-cancel {background-image: url("cancel.png"); } &-custom {background-image: url("custom.png"); }}
```

[\*\*WEB DESIGNING/Module-4 CSS Preprocessors\( SASS-LESS \)/Less/Less/Parent Selectors.zip at main · TopsCode/WEB DESIGNING - GitHub\*\*](#)

# Parent Selectors

output:

```
.button-ok {background-image: url("ok.png");}.button-cancel {background-image:  
url("cancel.png");}.button-custom {background-image: url("custom.png");}
```

[WEB DESIGNING/Module-4 CSS Preprocessors\( SASS-LESS \)/Less/Less/Parent\\_Selectors.zip at main · TopsCode/WEB DESIGNING · GitHub](#)

# Multiple &

& may appear more than once within a selector. This makes it possible to repeatedly refer to a parent selector without repeating its name.

.link {& + & {color: red; } && {color: green; } && {color: blue; } &, &ish {color: cyan; }}

will output:

.link + .link {color: red;}.link.link {color: green;}.link.link {color: blue;}.link, .linkish {color: cyan; }

} Note that & represents all parent selectors (not just the nearest ancestor) so the following example:

.grand {.parent {&>& {color: red; } && {color: green; } && {color: blue; } &, &ish {color: cyan; } }}

results in:

.grand.parent>.grand.parent {color: red;}.grand.parent.grand.parent {color: green;}.grand.parent.grand.parent {color: blue;}.grand.parent,.grand.parentish {color: cyan;}

# Changing Selector Order

It can be useful to prepend a selector to the inherited (parent) selectors. This can be done by putting the & after current selector. For example, when using Modernizr, you might want to specify different rules based on supported features:

```
.header {.menu {border-radius: 5px;.no-borderradius& {background-image: url('images/button-background.png'); } }}
```

The selector .no-borderradius& will prepend .no-borderradius to its parent .header .menu to form the.no-borderradius .header .menu on output:

```
.header.menu {border-radius: 5px;}.no-borderradius.header.menu {background-image: url('images/button-background.png');}
```

# Combinatorial Explosion

& can also be used to generate every possible permutation of selectors in a comma separated list:

```
p, a, ul, li {border-top: 2px dotted #366; & + & {border-top: 0; }}
```

This expands to all possible (16) combinations of the specified elements:

```
p,a,ul,li {border-top: 2px dotted #366;}p + p
```

```
,p + a
```

```
,p + ul
```

```
,p + li
```

```
,a + p
```

```
,a + a
```

```
,a + ul
```

```
,a + li
```

```
,ul + p
```

```
,ul + a
```

```
,ul + ul
```

```
,ul + li,li + p,li + a,li + ul,li + li {border-top: 0;}
```

# @import At-Rules

Import styles from other style sheets

In standard CSS, @import at-rules must precede all other types of rules. But Less doesn't care where you put @import statements.

Example:

```
.foo {  
background: #900;  
}  
@import "this-is-valid.less";
```

# File Extensions

@import statements may be treated differently by Less depending on the file extension:

If the file has a .css extension it will be treated as CSS and the @import statement left as-is (see the [inline option](#) below).

If it has *any other extension* it will be treated as Less and imported.

If it does not have an extension, .less will be appended and it will be included as a imported Less file.

Examples:

```
@import"foo"; //foo.less is imported@import"foo.less"; //foo.less is imported@import"foo.php";  
//foo.php imported as a Less file@import"foo.css"; //statement left in place, as-is
```

The following options can be used to override this behavior.

# Import Options

Less offers several extensions to the CSS @import CSS at-rule to provide more flexibility over what you can do with external files.

Syntax: `@import (keyword) "filename";`

The following import options have been implemented:

reference: use a Less file but do not output it

inline: include the source file in the output but do not process it

less: treat the file as a Less file, no matter what the file extension

css: treat the file as a CSS file, no matter what the file extension

once: only include the file once (this is default behavior)

multiple: include the file multiple times

optional: continue compiling when file is not found

More than one keyword per @import is allowed, you will have to use commas to separate the keywords:

Example: `@import (optional, reference) "foo.less";`

# reference

Released [v1.5.0](#)

Example: @import (reference) "foo.less";

Imagine that reference marks every at-rule and selector with a *reference flag* in the imported file, imports as normal, but when the CSS is generated, "reference" selectors (as well as any media queries containing only reference selectors) are not output. Reference styles will not show up in your generated CSS unless the reference styles are used as [mixins](#) or [extended](#).

Additionally, **reference** produces different results depending on which method was used (mixin or extend):

**[extend](#)**: When a selector is extended, only the new selector is marked as *not referenced*, and it is pulled in at the position of the reference @import statement.

**[mixins](#)**: When a reference style is used as an [implicit mixin](#), its rules are mixed-in, marked "not reference", and appear in the referenced place as normal.

# reference

## *reference example*

This allows you to pull in only specific, targeted styles from a library such as [Bootstrap](#) by doing something like this:

```
.navbar:extend(.navbarall) {}
```

And you will pull in only .navbar related styles from Bootstrap.

# inline

Use @import (inline) to include external files, but not process them.

Released [v1.5.0](#)

Example: @import (inline) "not-less-compatible.css";

You will use this when a CSS file may not be Less compatible; this is because although Less supports most known standards CSS, it does not support comments in some places and does not support all known CSS hacks without modifying the CSS.

So you can use this to include the file in the output so that all CSS will be in one file.

# less

Use @import (less) to treat imported files as Less, regardless of file extension.

Released [v1.4.0](#)

Example:

```
@import (less) "foo.css";
```

# CSS

Use @import (css) to treat imported files as regular CSS, regardless of file extension. This means the import statement will be left as it is.

Released [v1.4.0](#)

Example:

```
@import (css) "foo.less";  
outputs  
@import"foo.less";
```

# CSS

## once

The default behavior of @import statements. It means the file is imported only once and subsequent import statements for that file will be ignored.

Released [v1.4.0](#)

This is the default behavior of @import statements.

Example:

```
@import (once) "foo.less";@import (once) "foo.less"; // this statement will be ignored
```

## multiple

Use @import (multiple) to allow importing of multiple files with the same name. This is the opposite behavior to once.

Released [v1.4.0](#)

Example:

```
//file: foo.less.a {color: green;}
```

# CSS

```
//file: main.less@import (multiple) "foo.less";@import (multiple) "foo.less";
```

Outputs

```
.a {color: green;}.a {color: green;}
```

# optional

Use @import (optional) to allow importing of a file only when it exists. Without the optional keyword Less throws a FileError and stops compiling when importing a file that can not be found.

Released [v2.3.0](#)

# Extend

Extend is a Less pseudo-class which merges the selector it is put on with ones that match what it references.

Released [v1.4.0](#)

```
navul {&:extend(.inline);background: blue;}
```

In the rule set above, the :extend selector will apply the "extending selector" (nav ul) onto the .inline class *wherever the .inline class appears*. The declaration block will be kept as-is, but without any reference to the extend (because extend isn't css).

So the following:

```
navul {&:extend(.inline);background: blue;}.inline {color: red;
```

} Outputs

```
navul {background: blue;}.inline,navul {color: red;}
```

Notice how the nav ul:extend(.inline) selector gets output as nav ul - the extend gets removed before output and the selector block left as-is. If no properties are put in that block then it gets removed from the output (but the extend still may affect other selectors).

# Extend Syntax

The extend is either attached to a selector or placed into a ruleset. It looks like a pseudo-class with selector parameter optionally followed by the keyword all:

Example:

```
.a:extend(.b) {} // the above block does the same thing as the below block.a  
{&:extend(.b);}.c:extend(.dall) {// extends all instances of ".d" e.g. ".x.d" or ".d.x"}.c:extend(.d) //  
extends only instances where the selector will be output as just ".d"
```

It can contain one or more classes to extend, separated by commas.

Example:

```
.e:extend(.f) {}.e:extend(.g) {} // the above and the below do the same thing.e:extend(.f, .g) {}
```

# Extend Attached to Selector

Extend attached to a selector looks like an ordinary pseudo-class with selector as a parameter. A selector can contain multiple extend clauses, but all extends must be at the end of the selector.

Extend after the selector: pre:hover:extend(div pre).

Space between selector and extend is allowed: pre:hover :extend(div pre).

Multiple extends are allowed: pre:hover:extend(div pre):extend(.bucket tr) - Note this is the same as pre:hover:extend(div pre, .bucket tr)

This is NOT allowed: pre:hover:extend(div pre).nth-child(odd). Extend must be last.

If a ruleset contains multiple selectors, any of them can have the extend keyword. Multiple selectors with extend in one ruleset:

```
.big-division,.big-bag:extend(.bag),.big-bucket:extend(.bucket) { // body}
```

[WEB DESIGNING/Module-4 CSS Preprocessors\( SASS-LESS \)/Less/Less/Extend Atteched to Selector.zip at main . TopsCode/WEB DESIGNING · GitHub](#)

# Extend Inside Ruleset

Extend can be placed into a ruleset's body using &:extend(selector) syntax. Placing extend into a body is a shortcut for placing it into every single selector of that ruleset.

Extend inside a body:

```
pre:hover,.some-class {&:extend(divpre);}
```

is exactly the same as adding an extend after each selector:

```
pre:hover:extend(divpre),.some-class:extend(divpre) {}
```

[WEB DESIGNING/Module-4 CSS Preprocessors\( SASS-LESS \)/Less/Less/Extend\\_Inside\\_Ruleset.zip at main · TopsCode/WEB DESIGNING · GitHub](#)

# Extending Nested Selectors

Extend is able to match nested selectors. Following less:

Example:

```
.bucket {tr{ // nested ruleset with target selector color: blue; }}.some-class:extend(.bucket tr) {} //  
nested ruleset is recognized
```

Outputs

```
.bucket tr,.some-class {color: blue;}
```

Essentially the extend looks at the compiled css, not the original less.

Example:

```
.bucket {tr&{ // nested ruleset with target selector color: blue; }}.some-class:extend(tr.bucket) {} //  
nested ruleset is recognized
```

Outputs

```
tr.bucket,.some-class {color: blue;}
```

[WEB DESIGNING/Module-4 CSS Preprocessors\( SASS-LESS \)/Less/Less/Extending nested selector.zip at main - TopsCode/WEB DESIGNING - GitHub](#)

# Exact Matching with Extend

Extend by default looks for exact match between selectors. It does matter whether selector uses leading star or not. It does not matter that two nth-expressions have the same meaning, they need to have the same form in order to be matched. The only exception are quotes in attribute selector, less knows they have the same meaning and matches them.

Example:

```
.a.class,.class.a,.class>.a {color: blue;}.test:extend(.class) {} // this will NOT match the any selectors above
```

Leading star does matter. Selectors `*.class` and `.class` are equivalent, but extend will not match them:

```
*.class {color: blue;}.noStar:extend(.class) {} // this will NOT match the *.class selector
```

Outputs

```
*.class {color: blue;}
```

Order of pseudo-classes does matter. Selectors `link:hover:visited` and `link:visited:hover` match the same set of elements, but extend treats them as different:

```
link:hover:visited {color: blue;}.selector:extend(link:visited:hover) {}
```

Outputs

```
link:hover:visited {color: blue;}
```

# nth Expression

Nth expression form does matter. Nth-expressions  $1n+3$  and  $n+3$  are equivalent, but extend will not match them:

```
:nth-child(1n+3) {color: blue;}  
.child:extend(:nth-child(n+3))  
{}
```

Outputs

```
:nth-child(1n+3)  
{color: blue;}
```

Quote type in attribute selector does not matter. All of the following are equivalent.

```
[title=identifier]  
{color: blue;}  
[title='identifier']  
{color: blue;}[title="identifier"] {color: blue;} .noQuote:extend([title=identifier])  
.singleQuote:extend([title='identifier']) {}.doubleQuote:extend([title="identifier"]) {}
```

# nth Expression

## Outputs

```
[title=identifier],  
.noQuote,.singleQuote.,  
doubleQuote {color: blue;}  
[title='identifier'],  
.noQuote,.singleQuote,  
.doubleQuote {color: blue;}  
[title="identifier"],  
.noQuote,  
.singleQuote,  
.doubleQuote {color: blue;}
```

# Extend "all"

When you specify the all keyword last in an extend argument it tells Less to match that selector as part of another selector. The selector will be copied and the matched part of the selector only will then be replaced with the extend, making a new selector.

Example:

```
.a.b.test,.test.c {color: orange;}.test &:hover {color: green; } .replacement:extend(.testall) {}
```

Outputs

```
.a.b.test,.test.c,.a.b.replacement,.replacement.c {color: orange;}.test:hover,.replacement:hover  
{color: green;}
```

*You can think of this mode of operation as essentially doing a non-destructive search and replace.*

[WEB DESIGNING/Module-4 CSS Preprocessors\( SASS-LESS \)/Less/Less/ExtendAll.zip at main · TopsCode/WEB DESIGNING · GitHub](#)

# Selector Interpolation with Extend

Extend is **not** able to match selectors with variables. If selector contains variable, extend will ignore it.

However, extend can be attached to interpolated selector.

Selector with variable will not be matched:

```
@variable: .bucket;  
{@variable}{  
    // interpolated selectorcolor: blue;  
    .some-class:extend(.bucket) {} // does nothing, no match is found  
and extend with variable in target selector matches nothing:  
.bucket {color: blue;}  
.some-class:extend(@{variable})  
{} // interpolated selector matches nothing@variable: .bucket;
```

[WEB DESIGNING/Module-4 CSS Preprocessors\( SASS-LESS \)/Less/Less/Selector interpolation with Extend.zip at main - TopsCode/WEB DESIGNING - GitHub](#)

# Selector Interpolation with Extend

Both of the above examples compile into:

```
.bucket {color: blue;}
```

However,

:extend attached to an interpolated selector works:

```
.bucket {color: blue;}  
{@variable}:extend(.bucket)  
{@variable: .selector;
```

compiles to:

```
.bucket,  
.selector {color: blue;}
```

# Scoping / Extend Inside @media

Currently, an :extend inside a @media declaration will only match selectors inside the same media declaration:

```
@media print {.screenClass:extend(.selector)
{} // extend inside media.selector { // this will be matched - it is in the same media
color: black; }
.selector { // ruleset on top of style sheet - extend ignores it
color: red; }
@media screen {
.selector { // ruleset inside another media - extend ignores it
color: blue; }
}
```

} compiles into:

```
@media print {
.selector,.screenClass /* ruleset inside the same media was extended */color: black; }
.selector { /* ruleset on top of style sheet was ignored */color: red; }
@media screen {.selector /* ruleset inside another media was ignored */color: blue; }
```

# Scoping / Extend Inside @media

Note: extending does not match selectors inside a nested @media declaration:

@media screen

```
{.screenClass:extend(.selector) {} // extend inside media@media (min-width: 1023px) {.selector { //  
ruleset inside nested media - extend ignores itcolor: blue;    } }}
```

This compiles into:

@media screen and (min-width: 1023px)

{.selector

```
{/* ruleset inside another nested media was ignored */color: blue; }}
```

# Scoping / Extend Inside @media

Top level extend matches everything including selectors inside nested media:

```
@media screen {.selector { /* ruleset inside nested media - top level extend works */color: blue; }  
@media (min-width: 1023px)
```

```
{.selector { /* ruleset inside nested media - top level extend works */color:  
blue; } } .topLevel:extend(.selector) {  
/* top level extend matches everything */  
compiles into:
```

```
@media screen {.selector,.topLevel  
 {/* ruleset inside media was extended */color: blue; }}  
@media screen and (min-width: 1023px)  
{.selector,.topLevel  
 {/* ruleset inside nested media was extended */color: blue;  
}}
```

# Duplication Detection

Currently there is no duplication detection.

Example:

```
.alert-info,.widget /* declarations */ .alert:extend(.alert-info, .widget) {}
```

Outputs

```
.alert-info,.widget,.alert,.alert /* declarations */
```

[\*\*WEB DESIGNING/Module-4 CSS Preprocessors\( SASS-LESS \)/Less/Less/Duplication\\_Detection.zip at main · TopsCode/WEB DESIGNING - GitHub\*\*](#)

# Use Cases for Extend

## ***Classic Use Case***

The classic use case is to avoid adding a base class. For example, if you have

```
.animal {background-color: black;color: white;}
```

and you want to have a subtype of animal which overrides the background color then you have two options, firstly change your HTML

```
<a class="animal bear">Bear</a>.animal {background-color: black;color: white;}.bear {background-color: brown;}
```

or have simplified html and use extend in your less. e.g.

```
<a class="bear">Bear</a>.animal {background-color: black;color: white;}.bear  
{&:extend(.animal);background-color: brown;}
```

[WEB DESIGNING/Module-4 CSS Preprocessors\( SASS-LESS \)/Less/Less/Use Case for Extend.zip at main · TopsCode/WEB DESIGNING · GitHub](#)

# Use Cases for Extend

## *Reducing CSS Size*

Mixins copy all of the properties into a selector, which can lead to unnecessary duplication. Therefore you can use extends instead of mixins to move the selector up to the properties you wish to use, which leads to less CSS being generated.

Example - with mixin:

```
.my-inline-block() {display: inline-block;font-size: 0;}.thing1 {.my-inline-block;}.thing2 {.my-inline-block;}
```

Outputs

```
.thing1 {display: inline-block;font-size: 0;}.thing2 {display: inline-block;font-size: 0;}
```

[WEB DESIGNING/Module-4 CSS Preprocessors\( SASS-LESS \)/Less/Less/Use Case for Extend.zip at main . TopsCode/WEB DESIGNING - GitHub](#)

# Use Cases for Extend

Example (with extends):

```
.my-inline-block  
{display: inline-block;font-size: 0;}  
.thing1 {&:extend(.my-inline-block);}  
thing2 {&:extend(.my-inline-block);}
```

Outputs

```
.my-inline-block,.thing1,.thing2  
{display: inline-block;  
font-size: 0;}
```

[WEB DESIGNING/Module-4 CSS Preprocessors\( SASS-LESS \)/Less/Less/Use Case for Extend.zip at main · TopsCode/WEB DESIGNING · GitHub](#)

# Use Cases for Extend

## ***Combining Styles / A More Advanced Mixin***

Another use-case is as an alternative for a mixin - because mixins can only be used with simple selectors, if you have two different blocks of html, but need to apply the same styles to both you can use extends to relate two areas.

Example:

```
li.list>a  
{// list styles}  
button.list-style {  
&:extend(li.list>a);  
// use the same list styles}
```

[\*\*WEB DESIGNING/Module-4 CSS Preprocessors\( SASS-LESS \)/Less/Less/Use Case for Extend.zip at main - TopsCode/WEB DESIGNING - GitHub\*\*](#)

# Merge properties

Combine properties

The merge feature allows for aggregating values from multiple properties into a comma or space separated list under a single property. merge is useful for properties such as background and transform.

# Comma

Append property value with comma

Released [v1.5.0](#)

Example:

```
.mixin() {box-shadow+: inset 0010px#555;}  
.myclass {  
.mixin();  
box-shadow+: 0020px black;}
```

Outputs

```
.myclass {box-shadow: inset 0010px#555, 0020px black;}
```

# Space

Append property value with space

Released [v1.7.0](#)

Example:

```
.mixin() {transform+_: scale(2);}.myclass {.mixin();transform+_: rotate(15deg);}
```

Outputs

```
.myclass {transform: scale(2) rotate(15deg);}
```

To avoid any unintentional joins, merge requires an explicit + or +\_ flag on each join pending declaration.

# Mixins

"mix-in" properties from existing styles

You can mix-in class selectors and id selectors, e.g.

```
.a, #b {color: red;}  
. mixin-class {.a();}  
. mixin-id {#b();}
```

which results in:

```
.a, #b {color: red;}  
. mixin-class {color: red;}  
. mixin-id {color: red;}
```

Historically, the parentheses in a mixin call are optional, but optional parentheses are deprecated and will be required in a future release.

```
.a(); .a; // currently works, but deprecated; don't use  
.a ();
```

**Git Link :** [https://github.com/TopsCode/WEB\\_DESIGNING/blob/main/Module-4%20CSS%20Preprocessors\(%20SASS-LESS%20\)/03-mixins.scss](https://github.com/TopsCode/WEB_DESIGNING/blob/main/Module-4%20CSS%20Preprocessors(%20SASS-LESS%20)/03-mixins.scss)

# Mixins With Parentheses

If you want to create a mixin but you do not want that mixin to be in your CSS output, put parentheses after the mixin definition.

```
.my-mixin {color: black;}  
.my-other-mixin()  
{background: white;}  
.class {.my-mixin();}  
.my-other-mixin();  
outputs  
.my-mixin {color: black;}  
.class {color: black;  
background: white;}
```

[WEB DESIGNING/Module-4 CSS Preprocessors\( SASS-LESS \)/Less/Less/Mixin\\_with\\_parentheses.zip at main · TopsCode/WEB DESIGNING · GitHub](#)

# Selectors in Mixins

Mixins can contain more than just properties, they can contain selectors too.

For example:

```
.my-hover-mixin()  
&:hover {border: 1px solid red; }  
}  
button {.my-hover-mixin();}  
}
```

Outputs

```
button:hover {border: 1px solid red;}
```

# Namespaces

If you want to mixin properties inside a more complicated selector, you can stack up multiple ids or classes.

```
#outer()  
.inner {color: red; }  
.c [#outer.inner();]
```

Note: legacy Less syntax allows > and whitespace between namespaces and mixins. This syntax is deprecated and may be removed. Currently, these do the same thing.

```
#outer>.inner(); // deprecated#outer.inner(); // deprecated#outer.inner(); // preferred
```

Namespacing your mixins like this reduces conflicts with other library mixins or user mixins, but can also be a way to "organize" groups of mixins.

Example:

```
#my-library {.my-mixin()  
{color: black; }} // which can be used like this.class {#my-library.my-mixin();}
```

[WEB DESIGNING/Module-4 CSS Preprocessors\( SASS-LESS \)/Less/Less/NameSpaces.zip at main - TopsCode/WEB DESIGNING · GitHub](#)

# Guarded Namespaces

If a namespace has a guard, mixins defined by it are used only if the guard condition returns true. A namespace guard is evaluated exactly the same as a guard on a mixin, so the following two mixins work the same way:

```
#namespace when (@mode = huge)
{.mixin() /* */} #namespace
{.mixin() when (@mode = huge)
 /* */}
```

The default function is assumed to have the same value for all nested namespaces and mixin. The following mixin is never evaluated; one of its guards is guaranteed to be false:

```
#sp_1 when (default())
{#sp_2 when (default())
{.mixin() when not(default())
 /* */}}
```

[WEB DESIGNING/Module-4 CSS Preprocessors\( SASS-LESS \)/Less/Less/Guarded Namespaces.zip at main · TopsCode/WEB DESIGNING · GitHub](#)

# The !important keyword

Use the !important keyword after mixin call to mark all properties inherited by it as !important:

Example:

```
.foo (@bg: #f5f5f5, @color: #900)  
{background: @bg; color: @color;}
```

```
.unimportant {  
  .foo();  
}  
.important {  
  .foo() !important;  
}
```

} Results in:

```
.unimportant {background: #f5f5f5; color: #900;}  
.important {background: #f5f5f5 !important; color: #900 !important;}
```

# Parametric Mixins

How to pass arguments to mixins

Mixins can also take arguments, which are variables passed to the block of selectors when it is mixed in.

For example:

```
.border-radius(@radius) {-webkit-border-radius: @radius;-moz-border-radius: @radius;border-radius: @radius;}
```

And here's how we can mix it into various rulesets:

```
#header {.border-radius(4px);}.button {.border-radius(6px);}
```

} Parametric mixins can also have default values for their parameters:

```
.border-radius(@radius: 5px) {-webkit-border-radius: @radius;-moz-border-radius: @radius;border-radius: @radius;}
```

We can invoke it like this now:

```
#header {.border-radius();}
```

[WEB DESIGNING/Module-4 CSS Preprocessors\(SASS-LESS \)/Less/Less/Parametrics\\_Mixin.zip at main - TopsCode/WEB DESIGNING · GitHub](#)

# Parametric Mixins

And it will include a 5px border-radius.

You can also use parametric mixins which don't take parameters. This is useful if you want to hide the ruleset from the CSS output, but want to include its properties in other rulesets:

```
.wrap() {text-wrap: wrap;white-space: -moz-pre-wrap;white-space: pre-wrap;word-wrap: break-word;} pre{ .wrap() }
```

Which would output:

```
pre {text-wrap: wrap;white-space: -moz-pre-wrap;white-space: pre-wrap;word-wrap: break-word;}
```

[\*\*WEB DESIGNING/Module-4 CSS Preprocessors  
SASS-LESS \)/Less/Less/Parametrics\\_Mixin.zip at  
main · TopsCode/WEB DESIGNING · GitHub\*\*](#)

# Parameter separators

Parameters are currently either *semicolon* or *comma* separated.

Originally, parameters were only separated by commas, but the semi-colon was later added to support passing comma-separated list values to single arguments.

Note: As of Less 4.0, you can wrap a list value using a paren escape [~()], e.g. .name(@param1: ~(red, blue)). This is similar to the quote escape syntax: ~"quote". This may make semi-colon separators un-necessary in your code-base.

Examples:

two arguments and each contains comma separated list: .name(1, 2, 3; something, else)

three arguments and each contains one number: .name(1, 2, 3)

use a dummy semicolon to create a mixin call with one argument containing a comma-separated css list: .name(1, 2, 3;). *Note: if the trailing semi-colon seems strange, you may prefer: .name(~(1, 2, 3))*

Ways to write a comma separated default value:

@param-values: red, blue; .name(@param1: @param-values).

.name(@param1: red, blue;)

.name(@param1: ~(red, blue))

[\*\*WEB DESIGNING/Module-4 CSS Preprocessors\( SASS-LESS \)/Less/Less/Parametrics Seprator.zip at main - TopsCode/WEB DESIGNING - GitHub\*\*](#)

# Overloading mixins

It is legal to define multiple mixins with the same name and number of parameters. Less will use properties of all that can apply. If you used the mixin with one parameter e.g. .mixin(green);, then properties of all mixins with exactly one mandatory parameter will be used:

```
.mixin(@color)
{color-1: @color;}
.mixin(@color, @padding: 2)
{color-2: @color;padding-2:
@padding;}
.mixin(@color, @padding, @margin: 2)
{color-3: @color;padding-3: @padding;margin:
@margin@margin@margin@margin;};
some.selectordiv {.mixin(#008000);}
```

[WEB DESIGNING/Module-4 CSS Preprocessors\( SASS-LESS \)/Less/Less/Overloading\\_Mixin.zip at main · TopsCode/WEB DESIGNING · GitHub](#)

# Overloading mixins

compiles into:

```
.some.selectordiv {color-1: #008000;  
color-2: #008000;  
padding-2: 2;}
```

[WEB DESIGNING/Module-4 CSS Preprocessors\( SASS-LESS \)/Less/Less/Overloading\\_Mixin.zip at main · TopsCode/WEB DESIGNING · GitHub](#)

# Named Parameters

A mixin reference can supply parameters values by their names instead of just positions. Any parameter can be referenced by its name and they do not have to be in any special order:

```
.mixin(@color: black; @margin: 10px; @padding: 20px)
{color: @
color; margin:
@margin; padding: @padding;}
.class1 {.mixin(@margin: 20px; @color: #33acfe);}
.class2 {.mixin(#efca44; @padding: 40px);}
```

[WEB DESIGNING/Module-4 CSS Preprocessors\( SASS-LESS \)/Less/Less/Named Parameters.zip at main . TopsCode/WEB DESIGNING - GitHub](#)

# Named Parameters

compiles into:

```
.class1 {color: #33acf; margin: 20px; padding: 20px;}  
.class2 {color: #efca44; margin: 10px; padding: 40px;}
```

[WEB DESIGNING/Module-4 CSS Preprocessors\( SASS-LESS \)/Less/Less/Named Parameters.zip at main · TopsCode/WEB DESIGNING · GitHub](#)

# The @arguments Variable

@arguments has a special meaning inside mixins, it contains all the arguments passed, when the mixin was called. This is useful if you don't want to deal with individual parameters:

```
.box-shadow(@x: 0, @y: 0, @blur: 1px, @color: #000)  
{-webkit-box-shadow: @arguments;  
-moz-box-shadow: @arguments;box-shadow: @arguments;}  
.big-block {.box-shadow(2px, 5px);}
```

Which results in:

```
.big-block {-webkit-box-shadow: 2px5px1px#000;  
-moz-box-shadow: 2px5px1px#000;  
box-shadow: 2px5px1px#000;}
```

# Using Mixins as Functions

Selecting properties and variables from mixin calls

***Property / value accessors***

*Released [v3.5.0](#)*

Starting in Less 3.5, you can use property/variable accessors to select a value from an evaluated mixin's rules. This can allow you to use mixins similar to functions.

Example:

```
.average(@x, @y) {@result: (@@x + @@y) / 2;} div {// call a mixin and look up its "@result"
  padding: .average(16px, 50px)[@result];}
```

Results in:

```
div {padding: 33px;}
```

[WEB DESIGNING/Module-4 CSS Preprocessors\( SASS-LESS \)/Less/Less/Using Mixins As Function.zip at main - TopsCode/WEB DESIGNING - GitHub](#)

# Overriding mixin values

If you have multiple matching mixins, all rules are evaluated and merged, and the last matching value with that identifier is returned. This is similar to the cascade in CSS, and it allows you to "override" mixin values.

```
// library.less#library()
{.mixin() {prop: foo; }}
// customize.less@import"library";#library()
{.mixin() {prop: bar; }}
.box {my-value: #library.mixin[prop];}
```

Outputs:

```
.box {my-value: bar;
}
```

# Variable calls

Entire mixin calls can be aliased and called as variable calls. As in:

```
#library() {.colors() {background: green; }}.box {@alias: #library.colors();@alias();}
```

Outputs:

```
.box {background: green;}
```

Note, unlike mixins used in root, mixin calls assigned to variables and *called with no arguments* always require parentheses. The following is not valid.

```
#library() {.colors() {background: green; }}.box {@alias: #library.colors;@alias(); // ERROR: Could  
not evaluate variable call @alias}
```

# Variable calls

This is because it's ambiguous if variable is assigned a list of selectors or a mixin call. For example, in Less 3.5+, this variable could be used this way.

```
.box {  
  @alias: #library.colors;  
  @alias {  
    a: b;  
  }  
}
```

The above would output:

```
.box  
#library.colors {  
  a: b;  
}
```

# Maps

Released [v3.5.0](#)

Use rulesets and mixins as maps of values

By combining namespacing with the lookup [] syntax, you can turn your rulesets / mixins into maps.

```
@sizes: {mobile: 320px;  
tablet: 768px;  
desktop: 1024px;}  
.navbar {display: block;  
@media (min-width: @sizes[tablet])  
{display: inline-block; }  
}
```

# Maps

Outputs:

```
.navbar {display: block;}@media (min-width: 768px)  
{.navbar {display: inline-block; }}
```

Mixins are a little more versatile as maps because of namespacing and the ability to overload mixins.

```
#library(){  
.colors() {primary: green;secondary: blue;  
}  
#library() {.colors()  
primary: grey; }  
.button  
{color: #library.colors[primary];  
border-color: #library.colors[secondary];  
}
```

# Maps

Outputs:

```
.button {color: grey; border-color: blue;}
```

You can also make this easier by [aliasing mixins](#). That is:

```
.button {@colors: #library.colors();  
color: @colors[primary]; border-color:  
@colors[secondary];}
```

Note, if a lookup value produces another ruleset, you can append a second [] lookup, as in:

```
@config: {@options: {  
library-on: true }} & when (@config[@options][library-on] = true)  
.produce-ruleset {prop: val; }
```

In this way, rulesets and variable calls can emulate a type of "namespacing", similar to mixins. As far as whether to use mixins or rulesets assigned to variables as maps, it's up to you. You may want to replace entire maps by re-declaring a variable assigned to a ruleset. Or you may want to "merge" individual key/value pairs, in which case mixins as maps might be more appropriate

[WEB DESIGNING/Module-4 CSS Preprocessors\( SASS-LESS \)/Less/Less/Map.zip at main · TopsCode/WEB DESIGNING · GitHub](#)

# Scope

Mixin scope features

Intuitively, mixins have access to definition scope.

```
#ns {@a: one;.mixin-1()  
{prop: @a;  
}  
.rule {#ns.mixin-1();  
} /* OUTPUTS:.rule { prop: one; } */
```

# Scope

## Deprecated mixin scope features

This is a list of mixin scope features that may be removed in future releases.

### #1. (*DEPRECATED*) Mixins have access to caller scope.

```
#ns {.mixin-1()
{prop: @a;
}
.rule {@a: one;#ns.mixin-1();
}/* OUTPUTS:.rule { prop: one;}*/
```

# Scope

This is counter-intuitive because:

It is not typical in most other languages.

It's not immediately obvious when looking at the definition what output will be produced by the mixin.

**Preferred approach:** Pass in the variable you want to be visible to the mixin.

```
#ns {  
  .mixin-1(@a)  
  {prop: @a;  
  }  
  .rule {  
    #ns.mixin-1(@a: one);  
  }  
}
```

# Scope

## #2. (*DEPRECATED*) *The caller scope has access to variables from the mixin*

Mixins will push their variables into the caller scope, but *only* if the variable is not locally defined.

```
#ns {.mixin-1()
  [@a: one;@b: two;
  ]}
.rule {@b: three;#ns.mixin-1()
;prop-1: @a;
prop-2: @b;}
/* OUTPUTS:.rule
{ prop-1: one;
prop-2: three;
}*/
```

[WEB DESIGNING/Module-4 CSS Preprocessors\( SASS-LESS \)/Less/Less/Scope.zip at main · TopsCode/WEB DESIGNING · GitHub](#)

# Scope

This is counter-intuitive because:

A variable higher in the caller scope can be overridden.

It's also not a typical language behavior.

It differs from the behavior of detached rulesets.

Also, with the introduction of Maps, you can retrieve variable values (and mixins) directly.

**Preferred approach:**

```
#ns {.mixin-1()
  {@a: one; @b: two;
  }
.rule {@returns: #ns.mixin-1();
prop-1: @returns[@a]; prop-2:
@returns[@b];
/* OUTPUTS:.rule
{ prop-1: one; prop-2: two;}*/}
```

# Scope

## #3. (*DEPRECATED*) *The caller scope has access to mixins from the mixin*

Similarly to deprecated variable behavior, mixins are also pushed into the caller scope. However, unlike variables, mixins with the same name as the merged scope mixin are merged.

```
#ns {  
  .mixin-1() {prop-1: one;  
  prop-2: two; }  
  .rule {#ns();  
  .mixin-1();.mixin-1()  
  {prop-3: three; }  
  /* OUTPUT:.rule {  
    prop-1: one;  
    prop-2: two;  
    prop-3: three;  
  } */
```

[WEB DESIGNING/Module-4 CSS Preprocessors\( SASS-LESS \)/Less/Less/Scope.zip at main · TopsCode/WEB DESIGNING · GitHub](#)

# Scope

**Preferred approach:** Call mixins directly.

```
#ns {  
  .mixin-1()  
    {prop-1: one;  
     prop-2: two; }  
  .rule {.mixin-1()  
    {prop-3: three; }  
  #ns.mixin-1();  
  .mixin-1();}  
•OUTPUT:.rule {  
  • prop-1: one;  
  •prop-2: two;  
  •prop-3: three;  
  •}*/
```

[WEB DESIGNING/Module-4 CSS Preprocessors\( SASS-LESS \)/Less/Less/Scope.zip at main · TopsCode/WEB DESIGNING · GitHub](#)

# Tips & Tricks

Credit: [less/less.js/issues/1472](https://lesscss.org/less.js/issues/1472)

Here is a trick for defining variables and keeping them in some private scope, preventing them from leaking to the global space.

```
& // Vars@height:100px;  
@width:20px; // Don't define any prop:value on this scope (as doing so will generate (wrong)  
output).
```

```
.test {height: @height; width: @width;  
}  
.rest {height: @height; // Name error: variable @height is undefined  
}
```

# Tips & Tricks

Here, @height and @width are only defined for the scope created by &{ ... } You can also nest an scope inside a rule:

```
.some-module {  
  @height:200px;@width:200px;text-align: left;  
  line-height: @height; // 200px & {  
    // Override original values  
    @height:100px;@width: auto;  
    .some-module__element {height: @height; // 100px width: @width; // 200px }  
    .some-module__element.text {line-height: (@height / 2);  
      // 50px } } & {  
    // Override original values  
    @height:50px;  
    .some-module__another-element {  
      height: @height; // 50px width: @width; // 200px  
    }  
    .some-module__another-element.text {  
      line-height: (@height / 2); // 25px }  
  } }
```

# **Module -8**

## **[jQuery]**

# Prerequisite

**HTML**



**CSS**



**JS**



# What is jQuery

- jQuery is a fast, small, cross-platform and feature-rich JavaScript library.
- jQuery is cross-platform.
- jQuery means "write less do more".
- jQuery simplifies AJAX call and DOM manipulation.
- jQuery was first released in January 2006 by John Resig at BarCamp NYC.  
It is currently headed by Timmy Wilson and maintained by a team of developers.



# Applications of jQuery

- HTML manipulation
- DOM manipulation
- DOM element selection
- CSS manipulation
- Effects and Animations
- AJAX
- HTML event methods

# Why jQuery is required

- It is very fast and extensible.
- It facilitates the users to write UI related function codes in minimum possible lines.
- It improves the performance of an application.
- Browser's compatible web applications can be developed.
- It uses mostly new features of new browsers.

# jQuery Example

- jQuery is developed by Google.
- To create the first jQuery example, you need to use JavaScript file for jQuery.
- You can download the jQuery file from [jquery.com](http://jquery.com) or use the absolute URL of jQuery file as shown below:

```
<script type="text/javascript"
src="http://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js">
```

# Example

Change the background color using  
jQuery:

```
<title>First jQuery Example</title>
```

```
<script type="text/javascript"
src="http://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js">
</script>
```

```
<script type="text/javascript" language="javascript">
$(document).ready(function() {
    $("p").css("background-color", "cyan");
});
</script>
```

# **\$(document).ready() and \$()**

```
$(document).ready(function() {  
    $("p").css("color", "red");  
});  
$(function() {  
    $("p").css("color", "red");  
});
```

# jQuery Selectors

- jQuery Selectors are used to select and manipulate HTML elements.
- With jQuery selectors, you can find or select HTML elements based on their id, classes, attributes, types and much more from a DOM.
- All jQuery selectors start with a dollar sign and parenthesis e.g. `\$()`.  
It is known as the factory function.

# The `$()` factory function

- Every jQuery selector starts with this sign `$()`. This sign is known as the factory function.

Selector	Description
Name	It selects all elements that match with the given element name.
#ID	It selects a single element that matches with the given id.
.Class	It selects all elements that match with the given class.
Universal(*)	It selects all elements available in a DOM.
Multiple Elements A,B,C	It selects the combined results of all the specified selectors A,B and C.

# Functions

Function	Functionalities
hide()	hide the selected elements.
show()	show the selected elements.
fadeIn() , fadeOut(), fadeToggle(), fadeTo()	used to fade effect the element.
slideDown() , slideUp(), slideToggle()	used to Slide effect the element.
animate()	
delay()	

# Functions

Function	Functionalities
html()	used to change the entire content of the selected elements
text()	used to set or return the text content of the selected elements.
val()	Get-set value of selected element
append()	add additional content at the end of the selected elements.
appendTo(), prepend(), after(), before(), insertAfter()	AppendTo use for specific element

## jQuery Functions

insertAfter()	detach()	attr()	toggleClass()	
before()	wrap()	prop()	removeClass()	
after()	wrapInner()	css()	toggleClass()	
append()	wrapAll()	offset()		
prepend()	unwrap()	position()		
remove()	clone()	addClass() )		
empty()	scrollTop()	hasClass()		

[WEB DESIGNING/Module-6 jQuery/Jquery Function.html at main · TopsCode/WEB DESIGNING · GitHub](#)

# jQuery Form Methods

- There are mainly two methods that works with Form.
  1. serialize()
  2. serializeArray()

[WEB DESIGNING/Module-6 jQuery/Jquery Form Method.html at  
main - TopsCode/WEB DESIGNING - GitHub](#)

# serialize()

- jQuery serialize() method is used to create a text string in standard URL- encoded notation.
- It is used in form controls like <input>, <textarea>, <select> etc.
- It serializes the form values so that its serialized values can be used in the URL query string while making an AJAX request.

**Syntax:**

`$(selector).serialize()`

[WEB DESIGNING/Module-6 jQuery/Serialise.html at main - TopsCode/WEB DESIGNING - GitHub](#)

# serializeArray()

- The jQuery serializedArray() Method is used to create a JavaScript array of objects by serializing form values.
- It operates on a jQuery collection of forms and form controls.
- You can select one or more form elements such as <input>, <textarea> or the form element itself.

**Syntax:**

`$(selector).serializeArray()`

# jQuery Events

- jQuery events are the actions that can be detected by your web application.
  - They are used to create dynamic web pages.
  - An event shows the exact moment when something happens.
1. Mouse Events
  2. Keyboard Events
  3. Form Events
  4. Document/Window Event

Github link :

[https://github.com/TopsCode/WEB\\_DESIGNING/blob/main/Module-6%20jQuery/43-events.html](https://github.com/TopsCode/WEB_DESIGNING/blob/main/Module-6%20jQuery/43-events.html)

## jQuery Mouse, keyboard, form event, document/window events

Click()	mouseout()	change()	delegate()
Mouseenter()	keydown()	select()	
mouseleave()	keyup()	load()	
hover()	keypress()	unload()	
mousedown()	blur()	resize()	
mouseup()	focus()	bind()	
mouseover()	submit()	unbind()	

# jQuery CSS Methods

- There are total 3 methods used to handle CSS on DOM through jQuery.
  1. `css()`
  2. `offset()`
  3. `position()`

Github link :

[https://github.com/TopsCode/WEB\\_DESIGNING/blob/main/Module-6%20jQuery/15-css.html](https://github.com/TopsCode/WEB_DESIGNING/blob/main/Module-6%20jQuery/15-css.html)

# JQuery manipulating CSS

jQuery simplifies the process of manipulating CSS properties.  
Here's a quick summary of the methods:

- =>css(): Used to get or set individual or multiple CSS properties.
- => .addClass(), .removeClass(), .toggleClass(): Used for managing CSS classes.
- ⇒ .animate(): Used for animating changes to CSS properties.
- ⇒ .hover(): Used to change styles when the mouse enters or leaves an element.
- ⇒ jQuery UI: Offers additional methods for advanced animations and transitions.

# jQuery Dimension Methods

- For manipulating/updating the existing width/height of the element below methods of jQuery are used.
  1. width()
  2. height()
  3. innerWidth()
  4. innerHeight()
  5. outerWidth()
  6. outerHeight()

# width()

- jQuery width() method is used to return or set the width of matched element.
- **to get width:** When this method is used to return the width, it returns the width of first matched element.

`$(selector).width()`

- **to set width:** When this method is used to set the width, it sets the width for every matched element.

`$(selector).width(value)`

Github link :

<https://github.com/TopsCode/WEB DESIGNING/blob/main/Module-6%20jQuery/33-width.html>

# Example - to get width()

```
<script>
$(document).ready(function(){
    $("button").click(function(){
        alert("Width of div: " + $("div").width());
    });
});
</script>
```

# Example - to set width()

```
<script>
var widthValue = 70;
$( "div" ).one( "click", function() {
    $( this ).width( widthValue ).addClass( "mod"
); widthValue-= 10;
});
</script>
```

# height()

- The jQuery height() method is used to return the current computed height for the first element or set the height of every matched element.
- **to get height:** When this method is used to return the height, it returns the height of first matched element.

`$(selector).height()`

- **to set height:** When this method is used to set the height, it sets the height for every matched element.

`$(selector).height(value)`

**Github link :**

[https://github.com/TopsCode/WEB\\_DESIGNING/blob/main/Module-6%20jQuery/34-height.html](https://github.com/TopsCode/WEB_DESIGNING/blob/main/Module-6%20jQuery/34-height.html)

# Example - to get height()

```
<script>
$(document).ready(function(){
    $("button").click(function(){
        alert("Height of div: " + $("div").height());
    });
});
</script>
```

# Example - to set height()

```
<script>
$( "div" ).one( "click", function() {
  $( this ).height( 50 ).css({
    cursor: "auto",
    backgroundColor:
    "green"
  });
});
</script>
```

# innerWidth()

- jQuery innerWidth() method is used to return the inner width of the first matched element without including border and margin.
- This method includes padding but excludes border and margin.
- **to get Inner Width:**  
`$(selector).innerWidth()`

**GitHub link :**

[https://github.com/TopsCode/WEB\\_DESIGNING/blob/main/  
Module-6%20jQuery/35-innerWidth.html](https://github.com/TopsCode/WEB_DESIGNING/blob/main/Module-6%20jQuery/35-innerWidth.html)

# Example - to get innerWidth()

```
<script>
$(document).ready(function(){
    $("button").click(function(){
        alert("Inner width of div is: " + $("div").innerWidth());
    });
});
</script>
```

# innerHeight()

- The jQuery innerHeight () method is used to return the inner height of first matched element.
- It includes padding but not border and margin.
- **to get inner Height:**  
`$(selector).innerHeight()`

Github link :

[https://github.com/TopsCode/WEB\\_DESIGNING/blob/main/Module-6%20jQuery/36-innerHeight.html](https://github.com/TopsCode/WEB_DESIGNING/blob/main/Module-6%20jQuery/36-innerHeight.html)

# Example - to get innerHeight()

```
<script>
$(document).ready(function(){
    $("button").click(function(){
        alert("Inner height of the div is: " + $("div").innerHeight());
    });
});
</script>
```

# outerWidth()

- jQuery outerWidth() method is used to return the outer width of the first matched.
- Element with padding and border.
- **to get outer Width:**

`$(selector).outerWidth(includeMargin)`

include Margin can have two values which true or false

Github link :

[https://github.com/TopsCode/WEB\\_DESIGNING/blob/main/Module-6%20jQuery/37-outerWidth.html](https://github.com/TopsCode/WEB_DESIGNING/blob/main/Module-6%20jQuery/37-outerWidth.html)

# Example - to get outerWidth()

```
<script>
$(document).ready(function(){
    $("button").click(function(){
        alert("Outer width of div is: " + $("div").outerWidth());
    });
});
</script>
```

# outerHeight()

- The jQuery outerHeight () method is used to return the outer height of first matched element.
  - This method includes padding and border both.
- 
- **to get outer Height:**  
\$(selector).outerHeight(includeMargin)

include Margin can have two values which true or false

Github link :

[https://github.com/TopsCode/WEB\\_DESIGNING/blob/main/Module-6%20jQuery/38-outerHeight.html](https://github.com/TopsCode/WEB_DESIGNING/blob/main/Module-6%20jQuery/38-outerHeight.html)

# Example - to get outerHeight()

```
<script>
$(document).ready(function(){
    $("button").click(function(){
        alert("Outer height of the div is: " + $("div").outerHeight());
    });
});
</script>
```

Github link :

[https://github.com/TopsCode/WEB\\_DESIGNING/blob/main/Module-6%20jQuery/37-outerWidth.html](https://github.com/TopsCode/WEB_DESIGNING/blob/main/Module-6%20jQuery/37-outerWidth.html)

# jQuery Traversing

- In jQuery, traversing helps us to find out:
  1. Ancestor
  2. Descendents
  3. Siblings

jQuery Traversing technique provides methods to **filter** the elements.

# jQuery Traversing - Ancestors

- There are total 3 methods to traverse and find out parent elements through jQuery:
  1. parent()
  2. parents()
  3. parentsUntil()

# parent()

- The parent() method in jQuery finds the direct parent of the given selector.
- It is an inbuilt function in jQuery.
- This method only traverses a single level up in the DOM tree and returns the direct parent of the selected element.

Github link :

[https://github.com/TopsCode/WEB\\_DESIGNING/blob/main/Module-6%20jQuery/59-parent.html](https://github.com/TopsCode/WEB_DESIGNING/blob/main/Module-6%20jQuery/59-parent.html)

# parent()

- **Syntax:**

```
$(selector).parent(filter)
```

**selector:** represents the selected element whose parent is to be searched.

**filter:** optional - specifies the selector expression, which is used to narrow down the search.

# Example - parent()

- On click highlight the parent element

```
<script>  
function fun()  
{  
$(document).ready(function(){  
    $("p").parent("h2").css({"color": "blue", "border": "5px dashed  
    blue"});  
});  
}  
</script>
```

# parents()

- The parents() method in jQuery is used to get all ancestor elements of the given selector.
- It is an inbuilt function in jQuery.
- This method traverses upwards from the parent element, all the level up in the DOM tree and returns all ancestors of the selected element.

# **difference between parent parents()**

- The parents() method is similar to the parent() method, as both travel up to the DOM tree and return the parent element.
- But the difference is that the parents() method traverses multiple levels up in the DOM tree and returns all ancestors, including a grandparent, great-grandparent, etc. of the given selector, while the parent() method traverses a single level up and only returns the direct parent of the given selector.

# parents()

- **Syntax:**

```
$(selector).parents(filter)
```

**selector:** represents the selected element whose parent is to be searched.

**filter:** optional - specifies the selector expression, which is used to narrow down the search.

# Example - parents()

- On click highlight the parent elements

```
<script>
function fun(){
$(document).ready(function(){
    $("h4").parents().css({ "color": "blue", "border": "3px dashed
    blue"});
});
}
</script>
```

# parentsUntil()

- The parentsUntil() method is an inbuilt jQuery method.
- It is used to get all ancestor elements between the selector and stop.
- This method traverses upwards from the parent element along with the ancestors.
- If the selector is not supplied or not matched, the parentsUntil() method will return all selected ancestor elements as similar to the parents() method.

# parentsUntil()

- **Syntax:**

```
$(selector).parentsUntil(stop,filter)
```

**srop:** optional - It can be a selector expression, a jQuery object or an element that indicate where to stop the search in between the parameters selector and stop.

**filter:** optional - specifies the selector expression, which is used to narrow down the search.

[WEB DESIGNING/Module-6 jQuery/Jquery\\_until.html](#)  
[at main · TopsCode/WEB DESIGNING · GitHub](#)

# Example - parentsUntil()

- On click highlight the parents until a specifir element is reached.

```
<script>
function
fun(){
$(document).ready(function(){
    $("p").parentsUntil().css({ "color": "blue", "border": "3px dashed
    blue"});
});
}
</script>
```

[WEB DESIGNING/Module-6 jQuery/Jquery\\_until.html](#)  
[at main · TopsCode/WEB DESIGNING · GitHub](#)

# jQuery Traversing - Descendents

- Below are the methods that helps to find out the descendents - child elements:
  1. children()
  2. find()

Github link :

[https://github.com/TopsCode/WEB\\_DESIGNING/blob/main/Module-6%20jQuery/53-children.html](https://github.com/TopsCode/WEB_DESIGNING/blob/main/Module-6%20jQuery/53-children.html)

# children()

- The children() method in jQuery returns the direct children of the given selector.
- It is an inbuilt method in JQuery.
- This method traverses only a single level down the DOM tree.
- We can use the find() method to traverse multiple levels down or return the descendants(such as grandchildren, great-grandchildren, etc.).

Github link :

[https://github.com/TopsCode/WEB\\_DESIGNING/blob/main/Module-6%20jQuery/53-children.html](https://github.com/TopsCode/WEB_DESIGNING/blob/main/Module-6%20jQuery/53-children.html)

# children()

- **Syntax:**

```
$(selector).children(filter)
```

**filter:** It is an optional value that is used to narrow down the search.

# Example - children()

- Example to find out the first child element:

```
<script>
function fun(){
$(document).ready(function(){
    $("div").children().css({ "font-size": "30px", "color": "blue", "border": "6px
dashed blue"});
});
}
</script>
```

# find()

- The find() method in jQuery finds the descendant elements of the given selector.
- A descendant can be a child, grandchild, and so on. It is an inbuilt method in jQuery.
- For searching the descendant, the find() method traverse downwards from the selected element in the DOM tree.
- We can use the "\*" selector for returning all descendant elements. To return all descendant elements of the given selector, we have to write it as follows.

Github link :

[https://github.com/TopsCode/WEB\\_DESIGNING/blob/main/Module-6%20jQuery/56-find.html](https://github.com/TopsCode/WEB_DESIGNING/blob/main/Module-6%20jQuery/56-find.html)

# find()

- **Syntax:**

`$(selector).find(filter)`

**\*:** The \* selector used to return all the descendants (child) elements of the selected element.

**filter:** It is an optional value that is used to narrow down the search.

**Github link :**

[https://github.com/TopsCode/WEB\\_DESIGNING/blob/main/Module-6%20jQuery/56-find.html](https://github.com/TopsCode/WEB_DESIGNING/blob/main/Module-6%20jQuery/56-find.html)

# Example - children()

- Example to find out the all child and selected child element:

```
<script>
function fun(){
$(document).ready(function(){
    $("#div1").find("ul").css({ "font-size": "30px", "color": "blue", "border": "6px dashed blue"});
});
}
</script>
```

GitHub link :

[https://github.com/TopsCode/WEB\\_DESIGNING/blob/main/Module-6%20jQuery/53-children.html](https://github.com/TopsCode/WEB_DESIGNING/blob/main/Module-6%20jQuery/53-children.html)

# jQuery Traversing - Siblings

- jQuery Traversing provides below methods to search for the siblings
  - next/previous elements.
- 1. siblings()
- 2. next()
- 3. nextAll()
- 4. nextUntil()
- 5. prev()
- 6. prevAll()
- 7. prevUntil()

Github link :

<https://github.com/TopsCode/WEB DESIGNING/blob/main/Module-6%20jQuery/62.-siblings.html>

# siblings()

- siblings() method returns all siblings of the selected element.
- It is an inbuilt method in jQuery.
- Siblings are those elements that share a common parent.

Github link :

[https://github.com/TopsCode/WEB\\_DESIGNING/blob/main/Module-6%20jQuery/62.-siblings.html](https://github.com/TopsCode/WEB_DESIGNING/blob/main/Module-6%20jQuery/62.-siblings.html)

# siblings()

- **Syntax:**

```
$(selector).siblings(filter)
```

**filter:** It is a selector expression that is used to narrow down the search for sibling elements. It can have more than one value. The values should be separated by a comma.

# Example - siblings()

- **to display all the siblings of the selected element**

```
<script>
$(document).ready(function() {
    $("button").click(function() {
        $("span").siblings().css({ "color": "red", "border": "2px dashed blue" });
    });
});
</script>
```

# next()

- The next() method is an inbuilt method in jQuery, which returns the next sibling of the selected element.
- This method traverses forwards along with the next sibling of DOM elements.

# next()

- **Syntax:**

```
$(selector).next( [selector] )
```

**filter:** It is a selector expression that is used to narrow down the search for sibling elements. It can have more than one value. The values should be separated by a comma.

# Example - next()

- **to display the next element of the selected element**

```
<script>
function
fun(){
$(document).ready(function(){
$("h2").next().css({ "font-size": "30px", "color": "blue", "border": "6px
dashed blue"});
});
}
</script>
```

# nextAll()

- The nextAll() method is used to return all next siblings of the specified selector.
- It is an inbuilt method in jQuery.
- This method traverses forwards along with the next siblings of DOM elements.

# nextAll()

- **Syntax:**

```
selector.nextAll( filter )
```

**filter:** It is a selector expression that is used to narrow down the search for sibling elements. It can have more than one value. The values should be separated by a comma.

# Example - nextAll()

- **to display all the next elements of the selected element**

```
<script>  
function fun(){  
$(document).ready(function(){  
$("p").nextAll().css({"color": "blue", "border": "3px dashed  
blue"});  
});  
}  
</script>
```

# nextUntil()

- The nextUntil() method is an inbuilt jQuery method.
- It is used to find the next sibling elements between selector and stop.
- This method traverses forwards along with the siblings of DOM elements.
- If the selector is not supplied or not matched, the nextUntil() method will return all next sibling elements as similar to the nextAll() method.

# nextUntil()

- **Syntax:**

```
$(selector).nextUntil(stop,filter)
```

**stop:** optional - It can be a selector expression, a jQuery object or an element that indicate where to stop the search.

**filter:** optional - It is a selector expression to narrow down the search.

# Example - nextUntil()

- to display the next element until the matched element is found

```
<script>
function
fun(){
$(document).ready(function(){
    $("h1").nextUntil("h3").css({ "color": "blue", "border": "3px solid
    blue"});
}
</script>
```

[WEB DESIGNING/Module-6 jQuery/Jquery\\_NextUntill.html at main · TopsCode/WEB DESIGNING · GitHub](#)

# prev()

- The prev() method is an inbuilt method in jQuery, which returns an immediately previous sibling of the selected element.
- This method traverses backward along with the previous sibling of DOM elements.

# prev()

- **Syntax:**

```
selector.prev( [selector] )
```

**selector:** to specify a selector expression that narrows down the searching of the previous sibling.

# Example - prev()

- to display the previous element of the selected element

```
<script>
function fun(){
$(document).ready(function(){
$("p").prev().css({ "font-size": "30px", "color": "blue", "border": "6px
dashed blue"});
});
}
</script>
```

[WEB DESIGNING/Module-6 jQuery/Jquery\\_Prev.html](#)  
[at main · TopsCode/WEB DESIGNING · GitHub](#)

# prevAll()

- The prevAll() method is jQuery's inbuilt function that returns all previous siblings of the selected element.
- This method traverses backward along with the previous siblings of DOM elements.

# prevAll()

- **Syntax:**

selector.prevAll( filter )

**filter:** It is a selector expression that is used to narrow down the search for sibling elements. It can have more than one value. The values should be separated by a comma.

[WEB DESIGNING/Module-6 jQuery/Jquery\\_PrevAll.html at main · TopsCode/WEB DESIGNING · GitHub](#)

# Example - prevAll()

- to display all the previous element of the selected element

```
<script>
function
fun(){
$(document).ready(function(){
$("p").prevAll().css({"color": "blue", "border": "3px dashed
blue"});
});
}
</script>
```

[WEB DESIGNING/Module-6 jQuery/Jquery\\_PrevAll.html at main · TopsCode/WEB DESIGNING · GitHub](#)

# prevUntil()

- The prevUntil() method returns all previous sibling elements between the selector and stop.
- Sibling elements are elements that share the same parent.

# prevUntil()

- **Syntax:**

```
$(selector).prevUntil(stop,filter)
```

**stop:** Optional - A selector expression, element or jQuery object indicating where to stop the search for previous matching siblings elements

**filter:** optional - Specifies a selector expression to narrow down the search for sibling elements between the selector and stop

# Example - prevUntil()

- to display all the previous element untill the matched element found

```
<script>
function
fun(){
$(document).ready(function(){
    $("li.start").prevUntil("li.stop").css({"color": "red", "border": "2px
solid red"});
});
}
</script>
```

[WEB DESIGNING/Module-6 jQuery/Jquery PrevUntil.html at main · TopsCode/WEB DESIGNING · GitHub](#)

# jQuery Filtering

- To Filter specific element, object through jQuery, various filtering methods are available which are as below:
  1. `first()`
  2. `last()`
  3. `eq()`
  4. `filter()`
  5. `not()`

Github link :

[https://github.com/TopsCode/WEB\\_DESIGNING/blob/main/Module-6%20jQuery/55-filter.html](https://github.com/TopsCode/WEB_DESIGNING/blob/main/Module-6%20jQuery/55-filter.html)

# first()

- The first() method returns the first element of the selected elements.
- **Syntax:**  
`$(selector).first()`

# Example - first()

- To highlight the very first element/selected very first element.

```
<script>
$(document).ready(function() {
    $("button").click(function(){
        $("p").first().css("background-color", "lightblue");
    });
});
</script>
```

# last()

- The first() method returns the last element of the selected elements.
- **Syntax:**  
`$(selector).last()`

[WEB DESIGNING/Module-6 jQuery/Jquery\\_Last.html](#)  
[at main · TopsCode/WEB DESIGNING · GitHub](#)

# Example - last()

- To highlight the last element/selected very first element.

```
<script>
$(document).ready(function() {
    $("button").click(function(){
        $("div span").last().css("background-color", "lightblue");
    });
});
</script>
```

# eq()

- The eq() method in jQuery returns an element with the given index.
- This method reduces the set of elements to the one with a specific index.
- The index can either be positive or negative. If we use a negative index, the index count starts from the end of the list of selected elements.

- **Syntax:**

`$(selector).eq(index)`

Github link :

[https://github.com/TopsCode/WEB\\_DESIGNING/blob/main/Module-6%20jQuery/54-eq.html](https://github.com/TopsCode/WEB_DESIGNING/blob/main/Module-6%20jQuery/54-eq.html)

# Example - eq()

- To highlight the first and third element.

```
<script>
$(document).ready(function() {
  $(".para").eq(1).css({ "color": "red", "fontSize": "20px", "fontWeight": "bold" });
  $(".para").eq(3).css({ "color": "blue", "fontSize": "20px", "fontWeight": "bold" });
});
</script>
```

[WEB DESIGNING/Module-6 jQuery/Jquery Eq.html at  
main · TopsCode/WEB DESIGNING · GitHub](#)

# filter()

- The filter() method returns the elements matching the specified criteria.
- If elements do not match the criteria, they are removed from the selection

## Syntax:

`$(selector).filter(selector)`

**selector:** It is an optional attribute. It could be a JQuery object or a selector expression. We can also use the comma-separated list of expressions to apply multiple filters at once.

GitHub link :

[https://github.com/TopsCode/WEB\\_DESIGNING/blob/main/Module-6%20jQuery/55-filter.html](https://github.com/TopsCode/WEB_DESIGNING/blob/main/Module-6%20jQuery/55-filter.html)

# Example - filter()

- To filter out all the paragraph elements.

```
<script> function fun(){
$(document).ready(function(){
    $("p").filter(".para").css({"background": "yellow"});
});
}
```

# not()

- The not() method returns the elements that are not matching the specified criteria.
- If elements do not match the criteria, they are returned from the selection, while the matching elements will be removed.
- It is a jQuery's inbuilt method and works opposite to the filter() method.

# not()

- **Syntax:**

```
$(selector).not(criteria)
```

**criteria:** optional - It could be a jQuery object or the comma-separated list of elements to be removed from the set of elements. It is a selector expression and can be the id and class of the particular element.

# Example - not()

- To filter out all the paragraph that doesn't have specific id.

```
<script>  
function fun(){  
$(document).ready(function(){  
    $("p").not(".para").css({"background": "yellow"});  
});  
}  
</script>
```

# Selectors

- jQuery Selectors are used to select and manipulate HTML elements. They are very important part of jQuery library.
- With jQuery selectors, you can find or select HTML elements based on their id, classes, attributes, types and much more from a DOM.
- Selectors are used to select one or more HTML elements using jQuery and once the element is selected then you can perform various operation on that.

# :enabled

- The :enabled selector is used to select all enabled form elements.
- It is a pseudo-class selector that can also be used to style the enabled UI elements.
- By default, the form elements are in an enabled state.
- If some of the form elements are disabled, then using the :enabled selector, we can highlight the enabled elements.
- This selector can only be used for the HTML elements that support the disabled attribute that are `<input>`, `<textarea>`, `<button>`, `<option>`, `<fieldset>`, `<optgroup>`, `<select>`, and `<menuitem>`.

[WEB DESIGNING/Module-6 jQuery/JQuery Enable Disable.html at main .](#)

[TopsCode/WEB DESIGNING · GitHub](#)

# :enabled

- **Syntax:**

```
$(":enabled")
```

Ex.

```
$("button:enabled")
```

[WEB DESIGNING/Module-6 jQuery/JQuery Enable Disable.html at  
main · TopsCode/WEB DESIGNING · GitHub](#)

# Example - :enabled

```
<script>
$(document).ready(function() {
  $('button').click(function(){
    $(":enabled").css({"background-color": "yellow", "border": "2px dashed blue"});
  });
});
</script>
```

# :disabled

- The :disabled selector is used to select all disabled form elements.
- It is a pseudo-class selector that can also be used to style the disabled UI elements.
- This selector can only be used for the HTML elements that support the disabled attribute that are `<input>`, `<textarea>`, `<button>`, `<option>`, `<fieldset>`, `<optgroup>`, `<select>`, and `<menuitem>`.

# :disabled

- **Syntax:**

```
$(":disabled")
```

Ex.

```
$("button:disabled")
```

# Example - :disabled

```
<script>
$(document).ready(function() {
  $('#inp').click(function(){
    $(".:disabled").css({"background-color": "yellow", "border": "2px dashed blue"});
  });
});
</script>
```

# :nth-child

- The :nth-child() selector is used to match the elements based on their position regardless of the type of its parent.
- This selector is used to match the elements based on their position within a group of siblings.
- It matches each element, which is the nth-child.

**Github link :**

[https://github.com/TopsCode/WEB\\_DESIGNING/blob/main/Module-6%20jQuery/58-nth-child.html](https://github.com/TopsCode/WEB_DESIGNING/blob/main/Module-6%20jQuery/58-nth-child.html)

# :nth-child

- **Syntax:**

:nth-child( n | odd | even | formula)

**n:** The n in the above syntax must be an integer.

**odd:** It selects every odd child element.

**even:** It selects every even child element.

**formula:** It selects the child element present at the **value** of the formula. The formula for finding the different children is  $an+b$ .

**Github link :**

[https://github.com/TopsCode/WEB\\_DESIGNING/blob/main/Module-6%20jQuery/58-nth-child.html](https://github.com/TopsCode/WEB_DESIGNING/blob/main/Module-6%20jQuery/58-nth-child.html)

# Example - :nth-child

```
<script>
$(document).ready(function() {
  $("#odd").click(function () {
    $("li").css("background-color", "white");
    $("li:nth-child(odd)").css("background-color", "orange");
  });
});
</script>
```

Github link :

[https://github.com/TopsCode/WEB\\_DESIGNING/blob/main/Module-6%20jQuery/58-nth-child.html](https://github.com/TopsCode/WEB_DESIGNING/blob/main/Module-6%20jQuery/58-nth-child.html)

# :reset

- the :reset selector is used to select the elements of type reset.
- It selects the input elements and buttons with type = reset.
- If we write input:reset, the selector will only select the input elements with type reset.
- Whereas, if we write :reset, the selector will select both input as well as button elements.

# :reset

- **Syntax:**

```
$(":reset")
```

# Example - :reset

```
<script>
  $(document).ready(function() {
    $("button:reset").css({"background-color": "coral", "font-size": "20px",
"border": "4px dashed blue"});
  });
</script>
```

# :lang

- The :lang() selector is used to select the elements with the specified language code.
- It selects the elements whose language value is equal to the supplied language code.
- This selector can also select the elements whose language value starts with the supplied language immediately followed by a hyphen ("-").
- We can use the lang attribute in HTML to determine the language value of the elements.

# :lang

- **Syntax:**

```
$(":lang(language)")
```

# Example - :lang

```
<script>
$(document).ready(function() {
    $("button").click(function() {
        $("p:lang(en)").css("background-color", "lightgreen");
    });
});
</script>
```

# :lt

- The :lt() selector selects the elements that have lesser index value than the value specified in the index parameter of the :lt() selector.
- The index starts at 0.
- We can use the :gt() selector to select the elements having a larger index value than the specified index.

# :lt

- **Syntax:**

```
$(":lt(index)")
```

**index:** required - specifies which element is to be selected. The elements that have a lower index value than the specified value will get selected.

# Example - :lt

```
<script>
$(document).ready(function() {
    $("button").click(function(){
        $("td:lt(4)").css("backgroundColor", "yellow");
    });
});
</script>
```

[WEB DESIGNING/Module-6 jQuery/Jquery\\_it.html at  
main · TopsCode/WEB DESIGNING · GitHub](#)

# :gt

- The :gt() selector selects the elements that have a greater index value than the value specified in the index parameter of the :gt() selector.
- The index starts at 0.
- We can use the :lt() selector to select the elements having lesser index value than the specified index.

# :gt

- **Syntax:**

```
$(":gt(index)")
```

**index:** required - specifies which element is to be selected. The elements that have a lower index value than the specified value will get selected.

# Example - :gt

```
<script>
$(document).ready(function() {
    $("button").click(function(){
        $("td:gt(4)").css("backgroundColor", "yellow");
    });
});
</script>
```

[WEB DESIGNING/Module-6 jQuery/Jquery GT.html at  
main · TopsCode/WEB DESIGNING · GitHub](#)

# **Module - 9**

## **[ Framework ]**

# Bootstrap

- Bootstrap is the most popular HTML, CSS, and JavaScript framework for developing responsive, mobile-first websites.
- Bootstrap is completely free to download and use!
- Quickly design and customize responsive mobile-first sites with Bootstrap, the world's most popular front-end open source toolkit, featuring Sass variables and mixins, responsive grid system, extensive prebuilt components, and powerful JavaScript plugins.

Github link :

[https://github.com/TopsCode/WEB\\_DESIGNING/tree/main/Module6%20Bootstrap](https://github.com/TopsCode/WEB_DESIGNING/tree/main/Module6%20Bootstrap)

- 1) Bootstrap Introduction
- 2) Bootstrap Getting Started
- 3) Bootstrap Grid System
- 4) Bootstrap Fixed Layout
- 5) Bootstrap Fluid Layout
- 6) Bootstrap Responsive Layout
  - Bootstrap Utilities
- 7) Bootstrap Typography
- 8) Bootstrap Tables
- 9) Bootstrap Lists
- 10) Bootstrap List Groups
- 11) Bootstrap Forms
- 12) Bootstrap Custom Forms
- 13) Bootstrap Input Groups
- 14) Bootstrap Buttons
- 15) Bootstrap Button Groups

- Bootstrap with CSS: Grid System
- 1) Bootstrap Images
- 2) Bootstrap Cards
- 3) Bootstrap Media Objects
- 4) Bootstrap Icons
- 5) Bootstrap Navs
- 6) Bootstrap Navbar
- 7) Bootstrap Breadcrumbs
- 8) Bootstrap Pagination
- 9) Bootstrap Badges
- 10) Bootstrap Progress Bars
- 11) Bootstrap Spinners
- 12) Bootstrap Jumbotron
- 13) Bootstrap Helper Classes
- Bootstrap with CSS - typography
- Bootstrap Advanced

- 1) Bootstrap Modals
- 2) Bootstrap Dropdowns
- 3) Bootstrap Tabs
- 4) Bootstrap Tooltips
- 5) Bootstrap Popovers
- 6) Bootstrap Alerts
- 7) Bootstrap Stateful Buttons
- 8) Bootstrap Accordion
- 9) Bootstrap Carousel
- 10) Bootstrap Typeahead
- 11) Bootstrap ScrollSpy
- 12) Bootstrap Toasts

- Bootstrap Contextual Classes
- Bootstrap with CSS - Responsive Utilities
- Bootstrap Layout - Glyphicon
- bootstrap Layout - Dropdowns
- Bootstrap Dropup
- Bootstrap - Button Group
- Bootstrap Layout - Navigation Elements
- Navbar
- Breadcrumb
- pagination
- Input Group
- Labels
- Badges
- Jumbotron
- Page Headers
- Alerts
- Progress Bar
- List Group
- Panels
- Wells

# Prerequisites for Bootstrap

- HTML (Mandatory)
- CSS (Mandatory)
- JavaScript/jQuery (Moderate)

# Bootstrap Alternatives

## 1. Skeleton

Cons – Limited Templates

## 2. Foundation

Cons – Difficult to modify codes

## 3. UIKit

Cons – Limited Utility classes as compared  
to Bootstrap

## 4. Bulma

Cons – Still in the development phase

## 5. Pure

Cons – Limited CSS selectors

## 6. Powertocss

Cons – No longer  
Maintained

## 7. Kickstrap

Cons – No longer  
Maintained

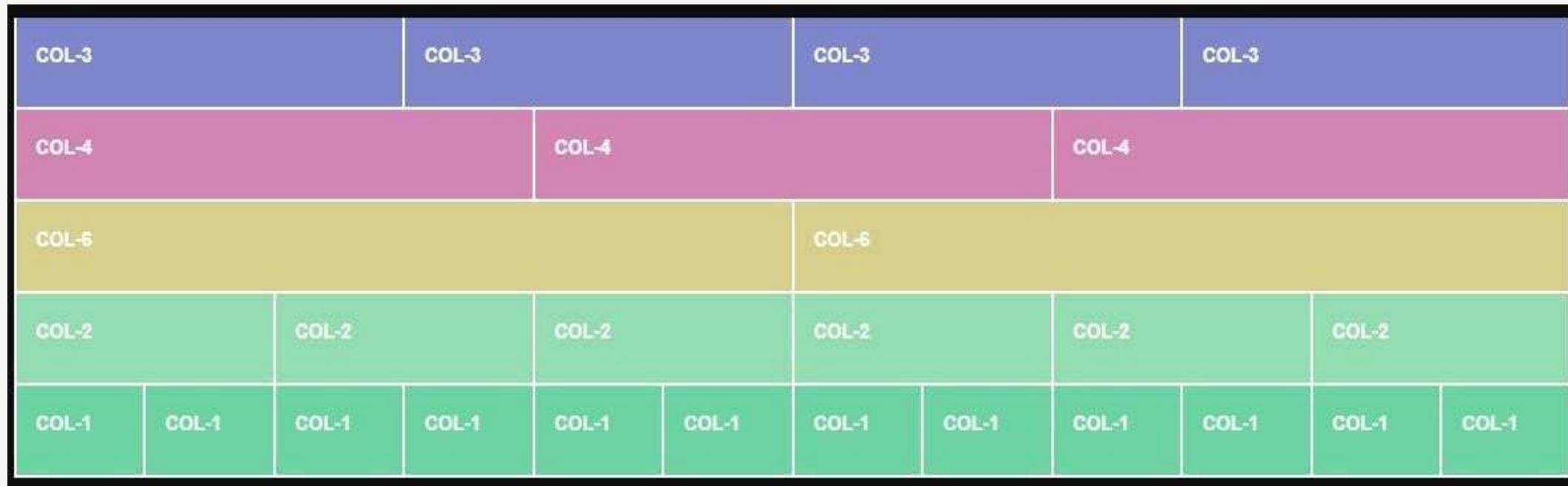
# Bootstrap Grid System

- The Bootstrap Grid System allows up to 12 columns across the page. You can use all 12 columns individually or you can groups the columns together to create wider columns.
- Bootstrap Grid System is responsive and the columns are re-arranged automatically according to the screen size.

**Github link :**

[https://github.com/TopsCode/WEB\\_DESIGNING/blob/main/Module6%20Bootstrap/css/bootstrap-grid.css](https://github.com/TopsCode/WEB_DESIGNING/blob/main/Module6%20Bootstrap/css/bootstrap-grid.css)

# Bootstrap Grid System



# Bootstrap Grid System

There are four classes in Bootstrap Grid System:

- xs (for phones)
- sm (for tablets)
- md (for desktops)
- lg (for larger desktops)
- xl (for larger desktops)
- xxl (for larger desktops)

# Bootstrap Utilities

- Bootstrap provides some handful helper classes, for faster mobile-friendly development. These can be used for showing and hiding content by device via media query, combined with large, small, and medium devices.

Github link :

[https://github.com/TopsCode/WEB\\_DESIGNING/blob/main/Module6%20Bootstrap/css/bootstrap-utilities.css](https://github.com/TopsCode/WEB_DESIGNING/blob/main/Module6%20Bootstrap/css/bootstrap-utilities.css)

# Bootstrap Fixed Layout

In Bootstrap, a fixed layout refers to the use of fixed-width containers or elements that don't scale with the viewport size. These elements maintain their width even when the window is resized, rather than fluidly expanding or contracting based on the screen size. Fixed layouts are often used when you want your website to have a consistent and controlled layout, regardless of the user's screen size. Bootstrap provides a variety of tools for creating fixed layouts, such as the container, container-fluid, and various fixed positioning classes. Here's how to implement a fixed layout in Bootstrap

[WEB DESIGNING/Module6 Bootstrap/Bootstrap Fix Layout.html at main · TopsCode/WEB DESIGNING · GitHub](#)

# Bootstrap Fluid Layout

A fluid layout in Bootstrap refers to a layout that expands and contracts to fill the available space in the viewport. Unlike fixed layouts where elements have a fixed width, fluid layouts adapt to different screen sizes, providing a more flexible and responsive design. In Bootstrap, you can achieve fluid layouts using the container-fluid class, which spans the full width of the viewport, regardless of the screen size. This allows your page elements to stretch across the screen and adjust based on the browser size.

Key Concepts for Fluid Layout in Bootstrap:

1. **container-fluid:=>**This class is used to create a container that takes up the full width of the viewport. It is used for creating fluid layouts that expand and contract with the screen size.

[WEB DESIGNING/Module6 Bootstrap/Bootstrap fluid Layout.html at main · TopsCode/WEB DESIGNING · GitHub](#)

# Bootstrap Fluid Layout

2. Grid System: => Bootstrap's grid system allows you to create fluid layouts using rows and columns, which are responsive. The grid system is based on 12 columns, and it adapts to different screen sizes using breakpoints.
3. Fluid Elements=> You can use fluid design principles for various elements, like images, videos, and other containers that automatically resize based on the screen.

[WEB DESIGNING/Module6 Bootstrap/Bootstrap fluid Layout.html at  
main · TopsCode/WEB DESIGNING · GitHub](#)

# Bootstrap Typography

Typography provides some utilities to add additional styles to texts.

These utilities are:

1. Text alignment
2. Text transform
3. Font weight and italics

[WEB DESIGNING/Module6 Bootstrap/Bootstrap\\_Typography.html at main · TopsCode/WEB DESIGNING · GitHub](#)

# Bootstrap Tables

- We can create different types of Bootstrap tables by using different classes to style them.

GitHub link :

[https://github.com/TopsCode/WEB\\_DESIGNING/blob/main/Module6%20Bootstrap/05table.html](https://github.com/TopsCode/WEB_DESIGNING/blob/main/Module6%20Bootstrap/05table.html)

# Bootstrap List

In Bootstrap, lists are a powerful and flexible component used to display items in a vertical or horizontal format. Bootstrap provides several utility classes and components to create styled lists, such as basic lists, unordered lists, ordered lists, and custom lists with icons, links, and more. You can also use the grid system or other Bootstrap components to enhance your lists.

## Types of Lists in Bootstrap:

### 1. Unordered List:

[WEB DESIGNING/Module6 Bootstrap/Bootstrap List.html at main · TopsCode/WEB DESIGNING · GitHub](#)

# Bootstrap List

An unordered list displays a list of items with bullet points. By default, Bootstrap uses list-style-type: disc to display bullet points. You can customize it using various classes.

## 2. Ordered List:

. Ordered List An ordered list is used when you want to display a list of items in a numbered sequence.

## 3. Definition List ;

A definition list is used to display terms and their definitions. It's useful for glossaries or lists with term-definition pairs.

# Bootstrap List

## 4. Horizontal Lists :

A horizontal list displays list items in a single row, typically used for navigation menus or inline elements. You can use the list-inline class to make a list horizontal.

## 5. Custom Lists :

You can customize lists with icons to improve the design and functionality. For example, you can use Font Awesome or Bootstrap Icons to display icons next to list items.

[WEB DESIGNING/Module6 Bootstrap/Bootstrap\\_List.html at main · TopsCode/WEB DESIGNING · GitHub](#)

# Bootstrap List Groups

Bootstrap List Groups are a flexible and customizable component used to group a list of items into a visually distinct group. List groups allow you to create styled lists, including clickable items, badges, and links. This component is often used for navigation menus, contact lists, or any set of related items in a list format

## Key Features of Bootstrap List Groups:

1. Basic List Groups: Styled lists with default or custom items.
2. Active Items: Highlight the current item in the list (useful for navigation).

[WEB DESIGNING/Module6 Bootstrap/Bootstrap List Group.html at main · TopsCode/WEB DESIGNING · GitHub](#)

# Bootstrap List Groups

3. Badges: Display badges next to list items to show additional information.
4. Links in List Groups: Use list-group-item as links for interactive list items.
5. Flush Lists: Remove borders and padding around the list to make it flush with the container.

# Bootstrap Cards



[WEB DESIGNING/Module6 Bootstrap/Bootstrap\\_Card.html at main · TopsCode/WEB DESIGNING · GitHub](#)

# Bootstrap Media Objects

Bootstrap Media Objects are a component used to display content with a media object, typically an image or an icon, next to text content. This is particularly useful for creating layouts like user profiles, comment sections, or any content where an image or icon accompanies textual information.

## Key Features of Media Objects in Bootstrap:

**Media:** The media object refers to an image or an icon.

**Media Body:** The text content that accompanies the media.

# Bootstrap Media Objects

**Alignment:** Media objects can be aligned vertically or horizontally.

**Customizable:** You can apply various styles to the media, such as size adjustments and alignment.

# Bootstrap Icons

Bootstrap Icons are a collection of free, high-quality icons designed specifically to work with Bootstrap. These icons are scalable, customizable, and easy to use with Bootstrap's components and utilities. They can be used in a variety of applications, including buttons, navigation menus, and more.

## Getting Started with Bootstrap Icons :

- 1. Via CDN (Content Delivery Network):** This is the easiest way to get started.
- 2. Download and Host:** You can download the icon library and host it locally

[WEB DESIGNING/Module6 Bootstrap/Bootstrap Condensed Table.html](#)  
[at main · TopsCode/WEB DESIGNING · GitHub](#)

# Bootstrap Navs

Bootstrap Navs are navigation components that allow you to easily create menus and navigation systems for your website. Bootstrap provides several types of navigation styles, including navbars, tabs, and pills, which can be customized and styled using various utility classes and components. .

## Types of Navs in Bootstrap:

- 1. Navbars:** A responsive navigation bar for links, logos, and other elements.

[WEB DESIGNING/Module6 Bootstrap/Bootstrap\\_Nav.html at main · TopsCode/WEB DESIGNING · GitHub](#)

# Bootstrap Navs

- 2. Tabs:** Horizontal navigation for dividing content into sections.
- 3. Pills:** Similar to tabs, but with rounded corners and often used for navigation with a modern look.
- 4. Vertical Navs:** A vertical list of links often used for sidebar menus.

# Bootstrap Helper Classes

Bootstrap Helper Classes are utility classes that help you quickly style and manage elements without needing to write custom CSS. These classes provide a set of predefined styles for layout, spacing, alignment, text, colors, visibility, and more. Bootstrap's utility classes are highly customizable and flexible, making them a core part of Bootstrap's rapid development workflow.

# Bootstrap Modals

A Modal is a dialog box or popup window that is displayed on top of the current page. It can be used to display content, alerts, or forms without navigating away from the current page. Bootstrap provides an easy way to implement modals with minimal code.

[WEB DESIGNING/Module6 Bootstrap/Bootstrap\\_model.html at main · TopsCode/WEB DESIGNING · GitHub](#)

# Bootstrap Popovers

Popovers are small overlay elements that provide additional information when triggered, similar to tooltips but with more content and features.

Bootstrap popovers are built on the Popover plugin, and you can use them to display rich content, like headers, paragraphs, or custom HTML, when the user interacts with an element.

Bootstrap popovers can be triggered by clicking, hovering, or focusing on an element. You can customize the position, content, and behavior of popovers.

[WEB DESIGNING/Module6 Bootstrap/Bootstrap\\_Popvers.html at main · TopsCode/WEB DESIGNING - GitHub](#)

# Bootstrap Stateful Buttons

In Bootstrap, a stateful button is typically one that changes its appearance depending on its current state (e.g., "active", "disabled", "selected"). You can achieve this functionality by combining Bootstrap's classes with custom JavaScript or jQuery to toggle states.

[WEB DESIGNING/Module6 Bootstrap/Bootstrap Stateful Button.html at main · TopsCode/WEB DESIGNING · GitHub](#)

# Bootstrap Toasts

Bootstrap Toasts are a great way to display brief messages or notifications to users in a non-intrusive manner. They typically appear as small pop-up messages on the screen that fade out after a short period or can be dismissed by the user

[WEB DESIGNING/Module6 Bootstrap/Bootstrap\\_Tost.html at main · TopsCode/WEB DESIGNING · GitHub](#)

# Bootstrap Bordered Table

A bordered table in Bootstrap adds borders around all the table cells, which can help clearly separate the content in each row and column. You can easily create a bordered table using the table-bordered class in Bootstrap.

```
<table class="table table-bordered table-hover">
```

[WEB DESIGNING/Module6 Bootstrap/Bootstrap Bordered Table.html](#)  
[at main · TopsCode/WEB DESIGNING · GitHub](#)

# Bootstrap Hover Row Table

In Bootstrap, you can add the hover effect to table rows to make it easier for users to visually identify which row they are interacting with. The table-hover class is used to highlight table rows when a user hovers over them.

```
<table class="table table-bordered table-striped table-hover">
```

# Bootstrap Condensed Table

A condensed table in Bootstrap is a table with reduced padding in the table cells. This allows you to fit more content into a smaller area, making the table more compact. Bootstrap provides the table-sm class to create a condensed table.

```
<table class="table table-bordered table-sm table-striped table-hover">
```

# Bootstrap Responsive Table

A responsive table in Bootstrap allows the table to scroll horizontally when the screen size is too small to display all the columns. This ensures that the table remains readable on devices with smaller screen widths, such as mobile phones.

```
<div class="table-responsive-sm">
```

[WEB DESIGNING/Module6 Bootstrap/Bootstrap Responsive Table.html](#)  
[at main · TopsCode/WEB DESIGNING · GitHub](#)

# Bootstrap 4 Form

In Bootstrap 4, creating forms is very straightforward thanks to the builtin classes that help structure the form elements neatly and responsively. Below is an example of how to build a simple form using Bootstrap 4, including form controls like input fields, labels, buttons, and form validation.

[WEB DESIGNING/Module6 Bootstrap/Bootstrap 4 Form.html at main · TopsCode/WEB DESIGNING · GitHub](#)

# Border Classes

you can easily apply borders to elements using the predefined border classes. These classes allow you to set borders around elements, such as div elements, buttons, form fields, etc

**General Border:** border: Adds a border to all sides of the element.

**Border Colors:** border-primary, border-secondary, border-success, etc.

**Border Width:** border-0: No border.

=> border-1: Thin border.

=> border-2: Thick border.

# Border Classes

## **Border Radius:**

- ⇒ rounded: Default rounded corners.
- ⇒ rounded-sm: Small rounded corners.
- ⇒ rounded-lg: Large rounded corners.
- ⇒ border-top, border-right, border-bottom, border-left. No Border on

## **Specific Sides:**

- => border-top-0, border-right-0, border-bottom-0, border-left-0.

# Bootstrap with CSS - Responsive Utilities

In Bootstrap 5, responsive utilities continue to be a core feature that makes designing responsive web pages easy. These utilities allow you to control the visibility, layout, spacing, and alignment of elements based on the viewport size.

# Bootstrap - Button Group

In Bootstrap, a Button Group is a collection of buttons that are grouped together to form a single component. Button groups allow you to align multiple buttons together and provide a cleaner and more organized design for related actions. You can create a vertical or horizontal group of buttons depending on your layout requirements.

# Bootstrap Forms

- In Bootstrap, there are three types of form layouts:
  1. Vertical form (this is default)
  2. Horizontal form
  3. Inline form

**Github link :**

<https://github.com/TopsCode/WEB DESIGNING/blob/main/Module6%20Bootstrap/11 %20Bootstrap %20Forms%20.html>

# Bootstrap Rules for Form

- There are three standard rules for these 3 form layouts:
  1. Always use `<form role="form">` (helps improve accessibility for people using screen readers)
  2. Wrap labels and form controls in `<div class="form-group">` (needed for optimum spacing)
  3. Add class `.form-control` to all textual `<input>`, `<textarea>`, and `<select>` elements

# Input Groups

- The .input-group class is a container to enhance an input by adding an icon, text or a button in front or behind the input field as a "help text".
- Use .input-group-prepend to add the help text in front of the input, and .input-group-append to add it behind the input.
- At last, add the .input-group-text class to style the specified help text.

**Github link :**

[https://github.com/TopsCode/WEB\\_DESIGNING/blob/main/Module6%20Bootstrap/13\\_Bootstrap%20Input%20Groups.html](https://github.com/TopsCode/WEB_DESIGNING/blob/main/Module6%20Bootstrap/13_Bootstrap%20Input%20Groups.html)

# Sizing

- Add the relative form sizing classes to the .input-group itself and contents within will automatically resize—no need for repeating the form control size classes on each element.

# Checkboxes and Radios

- Place any checkbox or radio option within an input group's addon instead of text.

# Multiple inputs

- While multiple <input>s are supported visually, validation styles are only available for input groups with a single <input>.

# Multiple addons

- Multiple add-ons are supported and can be mixed with checkbox and radio input versions.

# Custom forms

- Input groups include support for custom selects and custom file inputs. Browser default versions of these are not supported.

Github link :

[https://github.com/TopsCode/WEB\\_DESIGNING/blob/main/Module6%20Bootstrap/12\\_Bootstrap%20Custom%20Forms.html](https://github.com/TopsCode/WEB_DESIGNING/blob/main/Module6%20Bootstrap/12_Bootstrap%20Custom%20Forms.html)

# Bootstrap Buttons

- There are seven styles to add a button in Bootstrap. Use the following classes to achieve the different button styles:
  1. .btn-default
  2. .btn-primary
  3. .btn-success
  4. .btn-info
  5. .btn-warning
  6. .btn-danger
  7. .btn-link

**Github link :**

[https://github.com/TopsCode/WEB\\_DESIGNING/blob/main/Module6%20Bootstrap/17-button-group.html](https://github.com/TopsCode/WEB_DESIGNING/blob/main/Module6%20Bootstrap/17-button-group.html)

# Bootstrap Images

- Bootstrap supports for images. There are three classes in Bootstrap that can be used to apply some simple style to the images.

For Bootstrap 3

1. img-rounded
2. img-circle
3. img-thumbnail

For Bootstrap 4

1. rounded
2. rounded-circle
3. img-thumbnail

**Github link :**

[https://github.com/TopsCode/WEB\\_DESIGNING/blob/main/Module6%20Bootstrap/30\\_bootstrap\\_images.html](https://github.com/TopsCode/WEB_DESIGNING/blob/main/Module6%20Bootstrap/30_bootstrap_images.html)

# Navigation Elements

- Bootstrap provides a few different options for styling navigation elements. All of them share the same markup and base class, `.nav`. Bootstrap also provides a helper class, to share markup and states. Swap modifier classes to switch between each style.
- To create a tabbed navigation menu –
  1. Start with a basic unordered list with the base class of `.nav`
  2. Add class `.nav-tabs`.

Github link :

[https://github.com/TopsCode/WEB\\_DESIGNING/blob/main/Module6%20Bootstrap/Tables/19.Bootstrap%20Layout%20-%20Navigation%20Elements.html](https://github.com/TopsCode/WEB_DESIGNING/blob/main/Module6%20Bootstrap/Tables/19.Bootstrap%20Layout%20-%20Navigation%20Elements.html)

# Navbar

- The navbar is one of the prominent features of Bootstrap sites.
- Navbars are responsive 'meta' components that serve as navigation headers for your application or site.
- Navbars collapse in mobile views and become horizontal as the available viewport width increases.
- At its core, the navbar includes styling for site names and basic navigation.

**Github link :**

[https://github.com/TopsCode/WEB\\_DESIGNING/blob/main/Module6%20Bootstrap/13-navbar.html](https://github.com/TopsCode/WEB_DESIGNING/blob/main/Module6%20Bootstrap/13-navbar.html)

# Default Navbar

- Add the classes **.navbar**, **.navbar-default** to the **<nav>** tag.
- Add **role = "navigation"** to the above element, to help with accessibility.
- Add a header class **.navbar-header** to the **<div>** element. Include an **<a>** element with class **navbar-brand**. This will give the text a slightly larger size.
- To add links to the navbar, simply add an unordered list with the classes of **.nav**, **.navbar-nav**.

# Responsive Navbar

- To add responsive features to the navbar, the content that you want to be collapsed needs to be wrapped in a <div> with classes **.collapse**, **.navbar-collapse**.
- The collapsing nature is tripped by a button that has the class of **.navbar-toggle** and then features two data- elements.
- The first, **data-toggle**, is used to tell the JavaScript what to do with the button, and the second, **data-target**, indicates which element to toggle.
- Then with a class **.icon-bar** create what we like to call the hamburger button. This will toggle the elements that are in the **.nav-collapse** <div>

GitHub link :

[https://github.com/TopsCode/WEB\\_DESIGNING/blob/main/Module6%20Bootstrap  
trap/13-navbar.html](https://github.com/TopsCode/WEB_DESIGNING/blob/main/Module6%20Bootstrap/13-navbar.html)

# Navbar Brands

- Adding images to the .navbar-brand will likely always require custom styles or utilities to properly size.

**Github link :**

[https://github.com/TopsCode/WEB\\_DESIGNING/blob/main/Module6%20Bootstrap/13-navbar.html](https://github.com/TopsCode/WEB_DESIGNING/blob/main/Module6%20Bootstrap/13-navbar.html)

# Navbar Forms

- To add form elements inside the navbar, add the .navbar-form class to a form element and add an input(s). Note that we have added a .form-group class to the div container holding the input.

# Navbar Text

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

[WEB DESIGNING/Module6 Bootstrap/Navbar\\_Text.html at main · TopsCode/WEB DESIGNING · GitHub](#)

# Fixed Navigation Bar

- The navigation bar can also be fixed at the top or at the bottom of the page.
- A fixed navigation bar stays visible in a fixed position (top or bottom) independent of the page scroll.

[WEB DESIGNING/Module6 Bootstrap/Fixed Navigation Bar.html at main · TopsCode/WEB DESIGNING · GitHub](#)

# Breadcrumb

- Breadcrumbs are a great way to show hierarchy-based information for a site.
- In the case of blogs, breadcrumbs can show the dates of publishing, categories, or tags.
- They indicate the current page's location within a navigational hierarchy.

**GitHub link :**

[https://github.com/TopsCode/WEB\\_DESIGNING/blob/main/Module6%20Bootstrap/36\\_breadcrumbs.html](https://github.com/TopsCode/WEB_DESIGNING/blob/main/Module6%20Bootstrap/36_breadcrumbs.html)

# Pagination

- Pagination, an unordered list is handled by Bootstrap like a lot of other interface elements.

[Github link :](#)

[https://github.com/TopsCode/WEB\\_DESIGNING/blob/main/Module6%20Bootstrap/14-pagination.html](https://github.com/TopsCode/WEB_DESIGNING/blob/main/Module6%20Bootstrap/14-pagination.html)

# Pagination with Icons

- We can use icons in place of Next and Previous text in the patination. Be sure to provide proper screen reader support with aria attributes and the .sr-only utility

# Disabled and Active State

- Pagination links are customizable for different circumstances. Use `.disabled` for links that appear un-clickable and `.active` to indicate the current page.

[WEB DESIGNING/Module6 Bootstrap/Disabled and Active.html at main · TopsCode/WEB DESIGNING · GitHub](#)

# Sizing

- Add .pagination-lg or .pagination-sm for additional sizes.

# Spinner

- Spinner is also called a **loading indicator**. It is used to display/indicate the loading state of our projects. Bootstrap uses a **.spinner** class to create a Spinner.

Github link :

[https://github.com/TopsCode/WEB\\_DESIGNING/blob/main/Module6%20Bootstrap/40\\_spinners.html](https://github.com/TopsCode/WEB_DESIGNING/blob/main/Module6%20Bootstrap/40_spinners.html)

# Bootstrap Contextual Classes

- Contextual classes are used to color table rows (`<tr>`) or table cells (`<td>`)

Class/Context	Color	Description
.active	gray	Used to apply the hover color to the table row or table cell
.success	green	Indicates a successful or positive action
.info	cyan	Indicates a neutral informative change or action
.warning	yellow	Indicates a warning that might need attention
.danger	red	Indicates a dangerous or potentially negative action

# Glyphicon

- Glyphicons are easily understandable icons and symbols. They look great and hence are widely used in web projects.
- Bootstrap provides total 260 Glyphicons from the Glyphicons Halflings set.
- **Note:** Glyphicons are only supported up to Bootstrap version-3: <https://getbootstrap.com/docs/3.3/components/>

**GitHub link :**

[https://github.com/TopsCode/WEB\\_DESIGNING/blob/main/Module6%20Bootstrap/Tables/15.Bootstrap%20Layout%20-%20glyphicon.html](https://github.com/TopsCode/WEB_DESIGNING/blob/main/Module6%20Bootstrap/Tables/15.Bootstrap%20Layout%20-%20glyphicon.html)

# Bootstrap Dropdowns

- Dropdown menus are toggleable, contextual menus, used for displaying links in a list format.
- It facilitates users to choose one value from a predefined list.
- You have to wrap dropdown menu within the class .dropdown to create Bootstrap Dropdown.

Github link :

[https://github.com/TopsCode/WEB\\_DESIGNING/blob/main/Module6%20Bootstrap/19\\_Bootstrap%20Dropdowns.html](https://github.com/TopsCode/WEB_DESIGNING/blob/main/Module6%20Bootstrap/19_Bootstrap%20Dropdowns.html)

# Bootstrap Dropdown Divider

- The **class .divider** is used to separate links inside the dropdown menu

# Bootstrap Dropdown Header

- The **class .dropdown-header** is used to add headers inside the dropdown menu.

# Bootstrap Dropdown Disable an item

- Use the **class .disabled** to disable an item in the dropdown menu.

# Bootstrap Dropdown

- If you want to open the menu upwards instead of downwards, you need to change the element with class="dropdown" instead of class="dropdown":

GitHub link :

[https://github.com/TopsCode/WEB\\_DESIGNING/blob/main/Module6%20Bootstrap/Tables/17.Bootstrap%20Dropdown.html](https://github.com/TopsCode/WEB_DESIGNING/blob/main/Module6%20Bootstrap/Tables/17.Bootstrap%20Dropdown.html)

# Split Button Dropdowns

- It is used to show dropdowns as split buttons. Here we use all contextual classes.

# Labels

- Labels are used to provide additional information about something:
- Use the `.label` class, followed by one of the six contextual classes `.label-default`, `.label-primary`, `.label-success`, `.label-info`, `.label-warning` or `.label-danger`, within a `<span>` element to create a label:

**Github link :**

[https://github.com/TopsCode/WEB\\_DESIGNING/blob/main/Module6%20Bootstrap/Tables/24.labels.html](https://github.com/TopsCode/WEB_DESIGNING/blob/main/Module6%20Bootstrap/Tables/24.labels.html)

# Badges

- Badges scale to match the size of the immediate parent element by using relative font sizing and em units.
- As of v4, badges no longer have focus or hover styles for links

**Github link :**

[https://github.com/TopsCode/WEB\\_DESIGNING/blob/main/Module6%20Bootstrap/38\\_badges.html](https://github.com/TopsCode/WEB_DESIGNING/blob/main/Module6%20Bootstrap/38_badges.html)

# Button

- Badges can be used as part of links or buttons to provide a counter.

Github link :

[https://github.com/TopsCode/WEB\\_DESIGNING/blob/main/Module6%20Bootstrap/17-button-group.html](https://github.com/TopsCode/WEB_DESIGNING/blob/main/Module6%20Bootstrap/17-button-group.html)

# Jumbotron

- A lightweight, flexible component that can optionally extend the entire viewport to showcase key marketing messages on your site.

**GitHub link :**

[https://github.com/TopsCode/WEB\\_DESIGNING/blob/main/Module6%20Bootstrap/41\\_Jumbotrons.html](https://github.com/TopsCode/WEB_DESIGNING/blob/main/Module6%20Bootstrap/41_Jumbotrons.html)

# Fluid Jumbotron

- To make the jumbotron full width, and without rounded corners, add the .jumbotron-fluid modifier class and add a .container or .container-fluid within.

[WEB DESIGNING/Module6 Bootstrap/Fluid Jumbotron.html at main · TopsCode/WEB DESIGNING · GitHub](#)

# Page Headers

- A page header is like a section divider.
- The .page-header class adds a horizontal line under the heading (+ adds some extra space around the element):

Github link :

[https://github.com/TopsCode/WEB\\_DESIGNING/blob/main/Module6%20Bootstrap/Tables/27.Page%20Headers.html](https://github.com/TopsCode/WEB_DESIGNING/blob/main/Module6%20Bootstrap/Tables/27.Page%20Headers.html)

# Alerts

- Alerts are available for any length of text, as well as an optional dismiss button. For proper styling, use one of the eight **required** contextual classes (e.g., .alert-success).

Github link :

[https://github.com/TopsCode/WEB\\_DESIGNING/blob/main/Module6%20Bootstrap/23\\_Bootstrap%20Alerts.html](https://github.com/TopsCode/WEB_DESIGNING/blob/main/Module6%20Bootstrap/23_Bootstrap%20Alerts.html)

# Link Color

- Use the .alert-link utility class to quickly provide matching colored links within any alert.

# Additional content

- Alerts can also contain additional HTML elements like headings, paragraphs and dividers.

# Dismissing

- Using the alert JavaScript plugin, it's possible to dismiss any alert inline.

# Progress Bar

- Progress components are built with two HTML elements, some CSS to set the width, and a few attributes.
- Bootstrap doesn't use [the HTML5 <progress> element](#), ensuring you can stack progress bars, animate them, and place text labels over them.

**Github link :**

[https://github.com/TopsCode/WEB\\_DESIGNING/blob/main/Module6%20Bootstrap/03progressBar.htm](https://github.com/TopsCode/WEB_DESIGNING/blob/main/Module6%20Bootstrap/03progressBar.htm)

# Progress Bar

- Use the `.progress` as a wrapper to indicate the max value of the progress bar.
- Use the inner `.progress-bar` to indicate the progress so far.
- The `.progress-bar` requires an inline style, utility class, or custom CSS to set their width.
- The `.progress-bar` also requires some role and aria attributes to make it accessible.

# Labels

- Add labels to your progress bars by placing text within the .progress-bar.

# Height

- We only set a height value on the .progress, so if you change that value the inner .progress-bar will automatically resize accordingly.

# Backgrounds

- Use background utility classes to change the appearance of individual progress bars.

# Multiple Bars

- Include multiple progress bars in a progress component if you need.

# Striped

- Add .progress-bar-striped to any .progress-bar to apply a stripe via CSS gradient over the progress bar's background color.

Github link :

[https://github.com/TopsCode/WEB\\_DESIGNING/blob/main/Module6%20Bootstrap/Tables/2.BootstrapWithStripedTables.html](https://github.com/TopsCode/WEB_DESIGNING/blob/main/Module6%20Bootstrap/Tables/2.BootstrapWithStripedTables.html)

# Animated Stripes

- The striped gradient can also be animated. Add `.progress-bar-animated` to `.progress-bar` to animate the stripes right to left via CSS3 animations.

# List Group

- List groups are a flexible and powerful component for displaying a series of content.

Github link :

[https://github.com/TopsCode/WEB\\_DESIGNING/blob/main/Module6%20Bootstrap/11-listgroup.html](https://github.com/TopsCode/WEB_DESIGNING/blob/main/Module6%20Bootstrap/11-listgroup.html)

# ActiveItems

- Add `.active` to a `.list-group-item` to indicate the current active selection.

# Disabled Item

- Add `.disabled` to a `.list-group-item` to make it *appear* disabled.

# Links and Buttons

- Use `<a>`s or `<button>`s to create *actionable* list group items with hover, disabled, and active states by adding `.list-group-item-action`.

# Flush

- Add .list-group-flush to remove some borders and rounded corners to render list group items edge- to-edge in a parent container (e.g., cards).

# Numbered

- Add the .list-group-numbered modifier class (and optionally use an <ol> element) to opt into numbered list group items.

# Horizontal

- Add .list-group-horizontal to change the layout of list group items from vertical to horizontal across all breakpoints.

# Contextual Class

- Use contextual classes to style list items with a stateful background and color.

[WEB DESIGNING/Module6 Bootstrap/Bootstrap Contextual Classes.html at main · TopsCode/WEB DESIGNING · GitHub](#)

# Badges

- Add badges to any list group item to show unread counts, activity, and more with the help of some utilities.

Github link :

[https://github.com/TopsCode/WEB\\_DESIGNING/blob/main/Module6%20Bootstrap/38\\_badges.html](https://github.com/TopsCode/WEB_DESIGNING/blob/main/Module6%20Bootstrap/38_badges.html)

# Panels

- A panel in bootstrap is a bordered box with some padding around its content.

**Github link :**

[https://github.com/TopsCode/WEB\\_DESIGNING/blob/main/Module6%20Bootstrap/Tables/31.Panels.html](https://github.com/TopsCode/WEB_DESIGNING/blob/main/Module6%20Bootstrap/Tables/31.Panels.html)

# Panel Heading

- The .panel-heading class adds a heading to the panel.

# Panel Footer

- The .panel-footer class adds a footer to the panel.

# Panel Group

- To group many panels together, wrap a <div> with class .panel-group around them.
- The .panel-group class clears the bottom-margin of each panel

# Panel with contextual classes

- To color the panel, use contextual classes (.panel-default, .panel-primary, .panel-success, .panel-info, .panel-warning, or .panel-danger)

# Wells

- The .well class adds a rounded border around an element with a gray background color and some padding

Github link :

[https://github.com/TopsCode/WEB\\_DESIGNING/blob/main/Module6%20Bootstrap/Tables/32.Wells.html](https://github.com/TopsCode/WEB_DESIGNING/blob/main/Module6%20Bootstrap/Tables/32.Wells.html)

# Well Size

- Change the size of the well by adding the .well-sm class for small wells or .well-lg class for large wells

# Modal

- The Modal plugin is a dialog box/popup window that is displayed on top of the current page

**Github link :**

[https://github.com/TopsCode/WEB\\_DESIGNING/blob/main/Module6%20Bootstrap/12-modal.html](https://github.com/TopsCode/WEB_DESIGNING/blob/main/Module6%20Bootstrap/12-modal.html)

# Static Backdrop

- When backdrop is set to static, the modal will not close when clicking outside it. Click the button below to try it.

# Scrolling Long Content

- When modals become too long for the user's viewport or device, they scroll independent of the page itself. Try the demo below to see what we mean.
- You can also create a scrollable modal that allows scroll the modal body by adding `.modal-dialog-scrollable` to `.modal-dialog`.

[WEB DESIGNING/Module6 Bootstrap/Scrolling Long Content.HTML at main · TopsCode/WEB DESIGNING - GitHub](#)

# Vertically Centered

- Add .modal-dialog-centered to .modal-dialog to vertically center the modal.

# Tooltips

- Tooltips can be placed within modals as needed. When modals are closed, any tooltip within are also automatically dismissed.

**Github link :**

[https://github.com/TopsCode/WEB\\_DESIGNING/blob/main/Module6%20Bootstrap/16-tooltips.html](https://github.com/TopsCode/WEB_DESIGNING/blob/main/Module6%20Bootstrap/16-tooltips.html)

# Using the Grid

- Utilize the Bootstrap grid system within a modal by nesting `.container-fluid` within the `.modal-body`. Then, use the normal grid system classes as you would anywhere else.

# Scrollspy

- Automatically update Bootstrap navigation or list group components based on scroll position to indicate which link is currently active in the viewport.

**Github link :**

[https://github.com/TopsCode/WEB\\_DESIGNING/blob/main/Module6%20Bootstrap/28\\_Bootstrap%20ScollSpy.html](https://github.com/TopsCode/WEB_DESIGNING/blob/main/Module6%20Bootstrap/28_Bootstrap%20ScollSpy.html)

# Popover

- The Popover component is similar to tooltips; it is a pop-up box that appears when the user clicks on an element. The difference is that the popover can contain much more content.

**Github link :**

[https://github.com/TopsCode/WEB\\_DESIGNING/blob/main/Module6%20Bootstrap/22\\_Bootstrap%20Popovers.html](https://github.com/TopsCode/WEB_DESIGNING/blob/main/Module6%20Bootstrap/22_Bootstrap%20Popovers.html)

# Four Directions

- Four options are available: top, right, bottom, and left aligned.

# Dismiss on next click

- Use the focus trigger to dismiss popovers on the user's next click of a different element than the toggle element.

# Collapse

- Collapsibles are useful when you want to hide and show large amount of content.
- The collapse JavaScript plugin is used to show and hide content. Buttons or anchors are used as triggers that are mapped to specific elements you toggle.

[WEB DESIGNING/Module6 Bootstrap/Collapse.HTML](#)  
[at main · TopsCode/WEB\\_DESIGNING · GitHub](#)

# Accordion

- Using the [card](#) component, you can extend the default collapse behavior to create an accordion. To properly achieve the accordion style, be sure to use .accordion as a wrapper.

Github link :

[https://github.com/TopsCode/WEB\\_DESIGNING/blob/main/Module6%20Bootstrap/08-accordion.html](https://github.com/TopsCode/WEB_DESIGNING/blob/main/Module6%20Bootstrap/08-accordion.html)

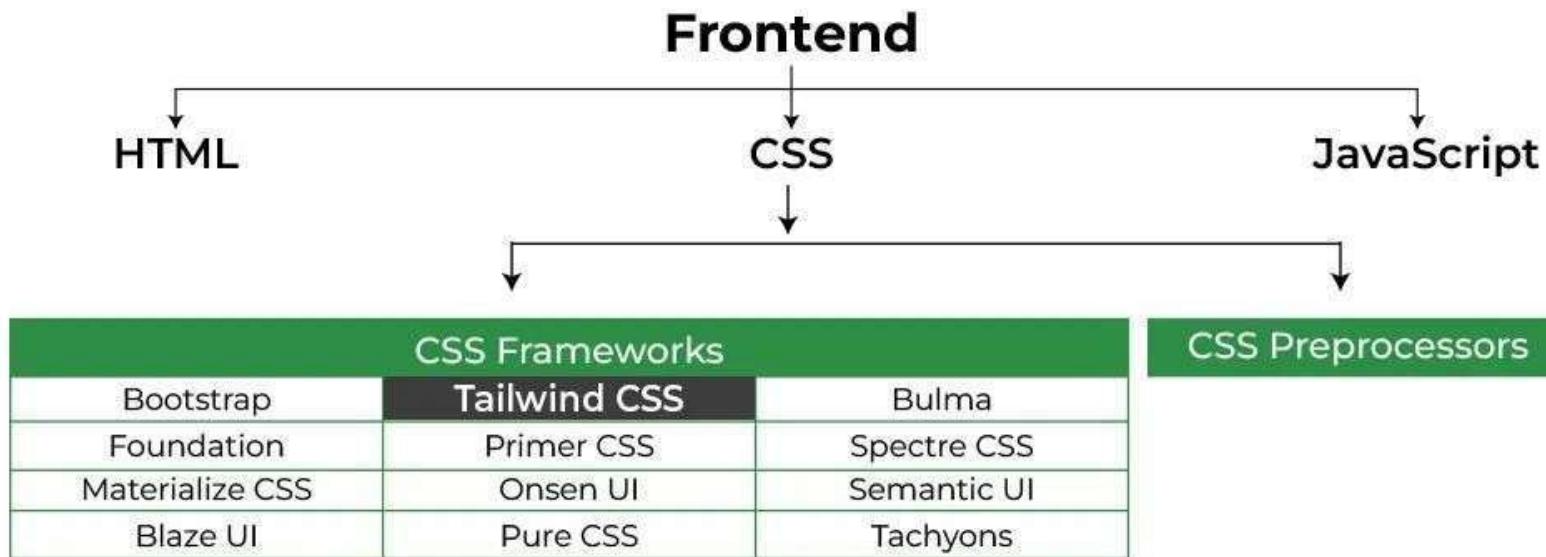
# Carousel

- It works with a series of images, text or custom markup.
- It also includes support for previous/next controls and indicators.
- The carousel is a slideshow for cycling through a series of content, built with CSS 3D transforms and a bit of JavaScript.

**GitHub link :**

[https://github.com/TopsCode/WEB\\_DESIGNING/blob/main/Module6%20Bootstrap/10-carousel.html](https://github.com/TopsCode/WEB_DESIGNING/blob/main/Module6%20Bootstrap/10-carousel.html)

# Tailwind CSS



# Prerequisites for Tailwind CSS

- HTML (Mandatory)
- CSS (Mandatory)
- JavaScript/jQuery (Moderate)

# Why use Tailwind CSS?

- Tailwind CSS has quickly become a popular choice for web developers due to its utility-first approach, offering a range of benefits that improve productivity, flexibility, and the overall development experience. Here are several reasons why developers choose Tailwind CSS:
  1. Utility-First Approach
  2. Faster Development
  3. Highly Customizable
  4. Responsive Design Made Easy

# Why use Tailwind CSS?

5. No Unused CSS (Tree-Shaking)
6. Consistency
7. No Need to Write Custom CSS
8. Large Ecosystem and Community
9. Built-In Dark Mode Support
10. Great for Prototyping
11. Easy to Learn
12. Flexibility in Design
13. Improved Maintainability
14. Integration with JavaScript Frameworks

# Introduction to Tailwind CSS

**Tailwind CSS** is a utility-first CSS framework that streamlines **web development** by providing a set of pre-designed utility classes. These classes enable rapid styling without writing custom CSS, promoting consistency and scalability. Tailwind's approach shifts focus from traditional CSS components to functional classes, empowering developers to efficiently build responsive and visually appealing interfaces with minimal effort.

**Git Link :**

[https://github.com/TopsCode/WEB\\_DESIGNING/blob/main/Module\\_8\\_Tailwind/01-tailwind-intro.html](https://github.com/TopsCode/WEB_DESIGNING/blob/main/Module_8_Tailwind/01-tailwind-intro.html)

# Installation of Tailwind CSS

## Through npm:

- Basically Tailwind is available on npm and you can install it using the following command:

```
npm install tailwindcss
```

- After that create ad Tailwind configuration file using the following command:

```
npm tailwind init {name of file}
```

## Through yarn:

- You can install tailwind by using the yarn command:

```
yarn add tailwindcss
```

- After that create ad Tailwind configuration file using the following command:

```
yarn tailwind init {name of file}
```

# Base Styles preflight

Preflight is a fundamental feature in Tailwind CSS that ensures consistent rendering across different browsers by normalizing and resetting default styles. It reduces the need for custom CSS resets, making your development process faster and more consistent

[WEB DESIGNING/Module 8 Tailwind/Base Styles preflight.html at main · TopsCode/WEB DESIGNING · GitHub](#)

# Tailwind CSS Container

In **Tailwind CSS**, a container is used to fix the *max-width* of an element to match the *min-width* of the breakpoint. It comes very handy when content has to be displayed in a responsive manner to every breakpoint. Breakpoints in tailwind CSS are as shown.

[WEB DESIGNING/Module 8 Tailwind/Tailwind CSS Container.html at main · TopsCode/WEB DESIGNING · GitHub](#)

<b>Breakpoint</b>	<b>min-width</b>
sm	640px
md	768px
lg	1024px
xl	1280px
2xl	1536px

# Tailwind CSS Container

- Tailwind CSS does not center itself automatically and also does not contain any pre-defined padding. The following are some utility classes that make the container class stand out.
- **mx-auto:** To center the container, we use *mx-auto* utility class. It adjust the margin of the container automatically so that the ~~container always has to be in center.~~
- **Syntax:** `<div class="mx-auto">...</div>`

[https://github.com/TopsCode/WEB\\_DESIGNING/blob/main/Module\\_8\\_Tailwind  
d/03-container.html](https://github.com/TopsCode/WEB_DESIGNING/blob/main/Module_8_Tailwind/03-container.html)

# Tailwind CSS Box Sizing

This class accepts more than one value in tailwind CSS all the properties [are covered as in class form. It is the alternative to the CSS box-sizing property.](#) This class is used to defines how the user should calculate the total width and height of an element i.e. padding and borders, are to be included or not.

## **Box Sizing:**

- **box-border**
- **box-content**

**box-border:** In this mode, the width and height properties include content, padding, and borders i.e. if we set an element's width to 200 pixels, that 200 pixels will include any border or padding we added, and the content box will shrink to absorb that extra width. This typically makes it much easier to

```
<element class="box-border">..</element>
```

## **Syntax:**

# Tailwind CSS Box Sizing

```
<!DOCTYPE html>
<head>
    <title>Tailwind box-border Class</title>
    <link href= "https://unpkg.com/tailwindcss@^1.0/dist/tailwind.min.css" rel="stylesheet">
</head>
<body class="text-center">
<center>
    <h1 class="text-green-600 text-5xl font-bold">
        Tops Tech
    </h1>
    <b>Tailwind CSS box-border Class</b>
    <div class="box-border h-28 w-32 p-4
                border-4 bg-green-500 m4">
        A Computer Science Portal
    </div>
</center>
</body>
</html>
```

# Tailwind CSS Box Sizing

**box-content:** This is the default value of the box-sizing class. In this mode, the width and height class include only the content. Border and padding are not included in it i.e if we set an element's width to 200 pixels, then the element's content box will be 200 pixels wide, and the width of any border or padding will be added to the final rendered width.

Syntax: <element class="box-content">..</element>

[https://github.com/TopsCode/WEB\\_DESIGNING/blob/main/Module\\_8\\_Tailwind/05-box-sizing.html](https://github.com/TopsCode/WEB_DESIGNING/blob/main/Module_8_Tailwind/05-box-sizing.html)

# Tailwind CSS Box Sizing

```
<!DOCTYPE html>
<head>
    <title>Tailwind box-content Class</title>
    <link href= https://unpkg.com/tailwindcss@^1.0/dist/tailwind.min.css rel="stylesheet">
</head>
<body class="text-center">
<center>
    <h1 class="text-green-600 text-5xl font-bold">
        Tops Tech
    </h1>
    <b>Tailwind CSS box-content Class</b>
    <div class="box-content h-28 w-32 p-4 border-4 bg-green-500 m4">
        A Computer Science Portal
    </div>
</center>
</body>
</html>
```

# Tailwind CSS Display

This class accepts more than one value in tailwind CSS. All the properties are covered as in class form. It is the alternative to the CSS display property. This class is used to define how the components (div, hyperlink, heading, etc) are going to be placed on the web page. As the name suggests, this property is used to define the display of the different parts of a web page.

## Display Classes:

- **block**: It is used to display an element as a block element.
- **inline-block**: It is used to display an element as an inline-level block container.
- **inline**: It is used to display an element as an inline element.
- **flex**: It is used to display an element as a block-level flex container.
- **inline-flex**: It is used to display an element as an inline-level flex container.

# Tailwind CSS Display

- **table**: It is used to set the behavior as <table> for all elements.
- **table-caption**: It is used to set the behavior as <caption> for all elements.
- **table-cell**: It is used to set the behavior as <td> for all elements.
- **table-column**: It is used to set the behavior as <col> for all elements.
- **table-column-group**: It is used to set the behavior as <column> for all elements.
- **table-footer-group**: It is used to set the behavior as <footer> for all elements.
- **table-header-group**: It is used to set the behavior as <header> for all elements.
- **table-row-group**: It is used to set the behavior as <row> for all elements.
- **table-row**: It is used to set the behavior as <tr> for all elements.
- **flow-root**: It is used to set the default value.
- **grid**: It is used to display an element as a block-level grid container.
- **inline-grid**: It is used to display an element as an inline-level grid container.
- **contents**: It is used to disappear the container.
- **hidden**: It is used to remove the element.

# Tailwind CSS Float

This class accepts more than one value in tailwind CSS. It is the alternative to the CSS float property. The float class defines the flow of content for controlling the wrapping of content around an element.

## Float Classes:

- float-right
- float-left
- float-none

[https://github.com/TopsCode/WEB\\_DESIGNING/blob/main/Module\\_8\\_Tailwind/07-float.html](https://github.com/TopsCode/WEB_DESIGNING/blob/main/Module_8_Tailwind/07-float.html)

# line-clamp

The line-clamp utility in Tailwind CSS allows you to easily limit the number of visible text lines and add an ellipsis when the content overflows. By installing the `@tailwindcss/line-clamp` plugin and using the `line-clamp-{n}` class, you can manage multi-line truncation without writing custom CSS.

Line clamping is a powerful tool for managing text overflow, especially in cases like card components, previews, or summaries, where you want to display a limited number of lines and hide the rest.

[WEB DESIGNING/Module 8 Tailwind/line clamp.html at main · TopsCode/WEB DESIGNING · GitHub](#)

# line height

In Tailwind CSS, line height is controlled using the `leading-{value}` utility, where you can choose from various pre-defined values like `leadingnone`, `leading-tight`, `leading-normal`, etc., or create custom line heights by extending the configuration. Adjusting line height is important for improving the readability, visual appeal, and overall layout of text content on your website or application.

# text align

In Tailwind CSS, the text alignment of an element can be easily controlled using the `text-{alignment}` utility class. This class allows you to align text to the left, right, center, or justify it within its container.

Tailwind provides several classes to control text alignment:

1. `text-left`: Aligns text to the left.
2. `text-center`: Aligns text to the center.
3. `text-right`: Aligns text to the right.
4. `text-justify`: Justifies text, meaning the text stretches to fit the container, with even spacing between words.

# text color

In Tailwind CSS, you can control the text color of elements using the `text-{color}` utility class. Tailwind provides a wide range of built-in colors that you can apply to text, making it easy to style your content.

## Text Color Utilities in Tailwind CSS

To apply a text color in Tailwind, you use the `text-{color}` class, where `{color}` refers to the color you want to apply. Tailwind provides several color names out of the box, as well as support for hex values, RGB, HSL, and opacity.

[WEB DESIGNING/Module 8 Tailwind/text\\_color.html at main · TopsCode/WEB DESIGNING - GitHub](#)

# text decoration

In Tailwind CSS, text decoration refers to controlling the appearance of text decorations like underline, line-through, no underline, and overline. Tailwind provides several utility classes to control text decoration.

**The main text decoration utilities in Tailwind are:**

1. `underline`: Applies an underline to the text.
2. `line-through`: Adds a line-through (strikethrough) effect to the text.
3. `no-underline`: Removes any underline from the text.
4. `overline`: Applies an overline (a line above the text).

# Offset

While Tailwind CSS does not have a specific offset utility, you can achieve the same effect using the positioning utilities like relative, absolute, fixed, and sticky in combination with the top, right, bottom, and left utilities. These classes let you move elements from their normal position or their container's edges, giving you full control over layout and positioning.

# text transform

In Tailwind CSS, the text transform utilities allow you to easily modify the case of text or transform its appearance by applying various text transformation styles. These utilities are based on the CSS text-transform property

**Tailwind provides the following text transform utilities:**

1. uppercase: Transforms text to uppercase (all letters are capitalized).
2. lowercase: Transforms text to lowercase (all letters are lowercase).
3. capitalize: Capitalizes the first letter of each word.
4. normal-case: Resets any text transform to the default, making the text case-sensitive as per the content's natural case.
5. text-transform-none: A more explicit class to ensure that text is not transformed (similar to normal-case).

# Indent

In Tailwind CSS, you can control the text indentation using the `indent-{value}` utility. This utility allows you to add indentation to the beginning of a block-level element, which is typically used for paragraphs or other block-level content.

# vertical alignment

In Tailwind CSS, vertical alignment of elements is typically controlled using the align-items property for flex containers, the vertical-align property for inline elements, and position utilities for absolutely or relatively positioned elements. Let's break down the various ways to achieve vertical alignment in Tailwind.

[WEB DESIGNING/Module 8 Tailwind/Vertical Alignment.html at main · TopsCode/WEB DESIGNING · GitHub](#)

# word break

In Tailwind CSS, the word break utilities allow you to control how long words or text should be handled when they exceed the container's width. This is important for ensuring text doesn't overflow or break awkwardly across lines. Tailwind provides utilities for controlling word breaks using the CSS word-break property.

## Available Word Break Utilities in Tailwind CSS:

1. `break-normal`: This is the default setting. It allows the text to break as normal, meaning it breaks lines at normal word boundaries (spaces).
2. `break-words`: This forces the text to break at any character to prevent overflow, even in the middle of words, if needed.
3. `truncate`: This utility limits the text to a single line, truncating it with an ellipsis (...) if the text overflows.

[WEB DESIGNING/Module 8 Tailwind/Word Break.html at main · TopsCode/WEB DESIGNING - GitHub](#)

# background attachment

In Tailwind CSS, the background-attachment property controls how a background image behaves when the page is scrolled. Specifically, it determines whether the background image scrolls with the content or stays fixed in place.

## Available Background Attachment Utilities:

1. bg-fixed: This utility fixes the background image in place, so it doesn't scroll with the rest of the content.
2. bg-local: The background image scrolls with the content inside the element.
3. bg-scroll: The default behavior, where the background image scrolls with the content of the page (this is the default in CSS).

# background-clip

In Tailwind CSS, the background-clip property controls the area within which the background is drawn. It determines whether the background should extend into the padding area, the content area, or even the border of an element.

## Available background-clip Utilities in Tailwind CSS:

1. bg-clip-border: The background is clipped to the border box (default behavior in CSS).
2. bg-clip-padding: The background is clipped to the padding box.
3. bg-clip-content: The background is clipped to the content box.

# background-Origin

Background Origin=> In CSS, the background-origin property is used to control where the background image is positioned relative to the element's box model. By default, a background image is placed relative to the content box, but you can adjust this by using the background-origin property to change its positioning.

1. `background-origin: border-box;` The background is positioned relative to the border box. The background image will start at the outer edge of the border.
2. `background-origin: padding-box;` The background is positioned relative to the padding box. The background will start at the inner edge of the padding (ignoring the border).
3. `background-origin: content-box;` The background is positioned relative to the content box (the default). The background will start at the edge of the content area (ignoring both padding and border).

# background-position

In CSS, the background-position property specifies the initial position of a background image within an element. It determines where the background image is placed in relation to the element's box.

You can set background-position using specific keywords like top, right, bottom, left, or by using percentage values or length values to provide more control.

## Keywords for background-position:

- ⇒ left: Aligns the background to the left edge.
- ⇒ center: Centers the background.
- ⇒ right: Aligns the background to the right edge.
- ⇒ top: Aligns the background to the top edge.
- ⇒ center: Aligns the background to the center vertically.
- ⇒ bottom: Aligns the background to the bottom edge.

# background-repeat

In CSS, the background-repeat property controls whether and how the background image is repeated (tiled) to cover the element's background. By default, a background image is repeated both horizontally and vertically unless specified otherwise.

## Values for background-repeat:

1. repeat: The background image is repeated both horizontally and vertically.
2. no-repeat: The background image is not repeated; it appears only once.
3. repeat-x: The background image is repeated only horizontally (along the x-axis).
4. repeat-y: The background image is repeated only vertically (along the y-axis).

# background-repeat

5. space: The background image is repeated, but the space between each repetition is distributed evenly (e.g., if there's extra space, it's distributed evenly).
6. round: The background image is repeated and stretched/shrunk to fit the available space.

[WEB DESIGNING/Module 8 Tailwind/background\\_repeat.html at main · TopsCode/WEB DESIGNING · GitHub](#)

# background-repeat

5. space: The background image is repeated, but the space between each repetition is distributed evenly (e.g., if there's extra space, it's distributed evenly).
6. round: The background image is repeated and stretched/shrunk to fit the available space.

[WEB DESIGNING/Module 8 Tailwind/background\\_repeat.html at main · TopsCode/WEB DESIGNING · GitHub](#)

# Background size

In CSS, the background-size property defines the size of the background image. It determines how the background image should be scaled to fit the element's dimensions.

**auto:** The background image's natural size is used (no scaling).

**cover:** The background image is scaled to cover the entire element's background area.

**Contain:** The background image is scaled to fit within the element's background area, but the entire image will be visible. This may result in empty spaces within the element (depending on the aspect ratio of the element and the image).

**Percentage or Length values:** You can define exact width and height values using percentages or fixed lengths (e.g., 100% 50%, 200px 100px, etc.)

[WEB DESIGNING/Module 8 Tailwind/Background size.html at main · TopsCode/WEB DESIGNING · GitHub](#)

# imageBorder

In CSS, an image border is a border applied around an image element. This border can be customized in terms of color, width, and style. You can also use various techniques to create borders for images, such as using the border property, border-radius (to create rounded corners), and box-shadow for added effects.

[WEB DESIGNING/Module 8 Tailwind/imageBorder.html at main · TopsCode/WEB DESIGNING - GitHub](#)

# Offset

The term offset in CSS can refer to various techniques that involve positioning or moving elements. It can be used with properties like top, left, right, bottom, transform, and clip-path, or in a more technical sense with JavaScript's offsetTop and offsetLeft properties..

[WEB DESIGNING/Module 8 Tailwind/Offset Tailwind.html at main · TopsCode/WEB DESIGNING · GitHub](#)

# Outline color

The outline property in CSS is used to create a line around an element, outside the border edge. Unlike the border, the outline does not take up space in the layout and can be used for styling purposes, often for focus indication or accessibility. The outline-color property specifically sets the color of the outline.

# ring width

In CSS and Tailwind CSS, the ring width typically refers to the thickness of a ring or outline that is applied around an element. The ring effect is commonly used in modern web design, especially for focus styles, to provide a clear visual indication of an element being active or focused, such as a button, input, or link.

[WEB DESIGNING/Module 8 Tailwind/ring width.html at main · TopsCode/WEB DESIGNING · GitHub](#)

# ring offset width

The ring-offset property in CSS is used to create space between the element's border (or background) and the ring outline applied around the element. This is useful for giving the ring a more distinct look or separating it from the element it surrounds.

In Tailwind CSS, the ring-offset utility allows you to define the distance between the element and the ring.

Tailwind CSS Ring Offset The ring-offset class in Tailwind CSS controls the amount of space between the element and the ring. It adds an offset between the element and the ring itself. You can use classes like ring-offset-0, ring-offset-1, and so on. Syntax:

```
<element class="ring-offset-{width}">
```

[WEB DESIGNING/Module 8 Tailwind/ring offset width.html at main · TopsCode/WEB DESIGNING · GitHub](#)

# box-decoration-break

The box-decoration-break property in CSS controls how the background and borders are applied to a broken element, like when text is wrapped in multiple lines or a multi-column layout is used. This property determines whether the background or border should be clipped or continue across line breaks.

[WEB DESIGNING/Module 8 Tailwind/box decoration break.html at main · TopsCode/WEB DESIGNING · GitHub](#)

# align-content

In Tailwind CSS, the align-content utility class allows you to control the alignment of the content in multi-line flexbox or grid layouts along the cross axis. This is useful when you have multiple rows or columns and want to control the positioning of those items when there is extra space on the cross axis;

- =>align-content-start;
- =>align-content: flex-start;
- =>align-content-end;
- =>align-content: flex-end;
- =>align-content-center:

[WEB DESIGNING/Module 8 Tailwind/align\\_content.html at main · TopsCode/WEB\\_DESIGNING · GitHub](#)

# align-items

In Tailwind CSS, the align-items property is used to control the alignment of items along the cross axis (perpendicular to the main axis) within a flexbox or grid container. This is useful for adjusting the positioning of items when they don't fill the container along the cross axis.

=>items-start: align-items: flex-start; (Aligns items to the start of the cross axis) |

=>items-end: align-items: flex-end; (Aligns items to the end of the cross axis)

=>items-center: align-items: center; (Aligns items to the center of the cross axis)

=>items-baseline: align-items: baseline; (Aligns items along their baseline)  
items-stretch: align-items: stretch; (Stretches items to fill the container; this is the default value)

[WEB DESIGNING/Module 8 Tailwind/align\\_items.html at main · TopsCode/WEB DESIGNING · GitHub](#)

# Alignself

In Tailwind CSS, the align-self property is used to control the alignment of a specific item within a flex or grid container along the cross axis (perpendicular to the main axis). This allows you to override the container's alignment behavior for a single item, enabling more finegrained control.

- =>self-auto: align-self: auto
- =>self-start: align-self: flex-start;
- =>self-end: align-self: flex-end;
- =>self-center: align-self: center;
- =>self-baseline: align-self: baseline;
- =>self-stretch: align-self: stretch;

# Placecontent

In Tailwind CSS, the place-content utility is used to control the alignment of content within a grid container along both the main axis and cross axis simultaneously. This property is especially useful when you're working with grid layouts and want to align all content inside the grid without manually setting alignment for both axes separately.

The place-content property combines the effects of align-content and justify-content:

=>align-content: Aligns the entire grid along the cross axis (vertical axis in a typical grid layout).

=> justify-content: Aligns the entire grid along the main axis (horizontal axis in a typical grid layout).

[WEB DESIGNING/Module 8 Tailwind/Placecontent.html at  
main · TopsCode/WEB DESIGNING · GitHub](#)

# Placecontent

=>place-content-start: place-content: flex-start;  
=>place-content-end: place-content: flex-end;  
=>place-content-center: place-content: center;  
=>place-content-between: place-content: space-between;  
=>place-content-around: place-content: space-around;  
=>place-content-evenly: place-content: space-evenly;  
=>place-content-stretch: place-content: stretch;

# place-items

The place-items utility allows you to set both horizontal and vertical alignment for grid items in one declaration. This is useful when you want to control the position of items inside the grid container simultaneously along both axes.

- =>place-items-start: place-items: start;
- =>place-items-end: place-items: end;
- =>place-items-center: place-items: center;
- =>place-items-stretch: place-items: stretch;

# place-self

The place-self utility is a shorthand for the combination of:

- ⇒ align-self: Aligns the item along the cross axis (vertical axis in a typical grid or flex layout).
- ⇒ justify-self: Aligns the item along the main axis (horizontal axis in a typical grid or flex layout).
  
- ⇒ place-self-auto: place-self: auto;
- ⇒ place-self-start: place-self: start;
- ⇒ place-self-end: place-self: end;
- ⇒ place-self-center: place-self: center;
- ⇒ place-self-stretch: place-self: stretch;
- ⇒ place-self-baseline: place-self: baseline;

# flex-basis

Tailwind provides a set of classes for flex-basis, and they can be used to control the size of flex items relative to their parent container.

- =>basis-auto: flex-basis: auto;
- =>basis-0: flex-basis: 0;
- =>basis-1/4: flex-basis: 25%;
- =>basis-1/3: flex-basis: 33.3333%;
- =>basis-1/2: flex-basis: 50%;
- =>basis-2/3: flex-basis: 66.6667%;
- =>basis-3/4: flex-basis: 75%;
- =>basis-full: flex-basis: 100%;
- =>basis-px: flex-basis: 1px;

[WEB DESIGNING/Module 8 Tailwind/flex basis.html at main · TopsCode/WEB DESIGNING · GitHub](#)

# mix blend mode

In Tailwind CSS, the mix-blend-mode utility allows you to control how an element's content blends with its background or other elements behind it.

- =>mix-blend-normal: mix-blend-mode: normal;
- =>mix-blend-multiply: mix-blend-mode: multiply;
- =>mix-blend-screen: mix-blend-mode: screen;
- =>mix-blend-overlay: mix-blend-mode: overlay;
- =>mix-blend-darken: mix-blend-mode: darken;
- =>mix-blend-lighten: mix-blend-mode: lighten;
- =>mix-blend-color-dodge: mix-blend-mode: color-dodge;
- =>mix-blend-color-burn: mix-blend-mode: color-burn;

[WEB DESIGNING/Module 8 Tailwind/mix blend mode.html at main · TopsCode/WEB DESIGNING · GitHub](#)

# mix blend mode

=>mix-blend-hard-light: mix-blend-mode: hard-light;  
=>mix-blend-soft-light: mix-blend-mode: soft-light;  
=>mix-blend-difference: mix-blend-mode: difference;  
=>mix-blend-exclusion: mix-blend-mode: exclusion;  
=>mix-blend-saturation: mix-blend-mode: saturation;  
=>mix-blend-color: mix-blend-mode: color;  
=>mix-blend-luminosity: mix-blend-mode: luminosity;

# mix blend mode

=>mix-blend-hard-light: mix-blend-mode: hard-light;  
=>mix-blend-soft-light: mix-blend-mode: soft-light;  
=>mix-blend-difference: mix-blend-mode: difference;  
=>mix-blend-exclusion: mix-blend-mode: exclusion;  
=>mix-blend-saturation: mix-blend-mode: saturation;  
=>mix-blend-color: mix-blend-mode: color;  
=>mix-blend-luminosity: mix-blend-mode: luminosity;

# background-blend-mode

In Tailwind CSS, the background-blend-mode utility is used to control how background images blend with the background color.

- =>bg-blend-normal: background-blend-mode: normal;
- =>bg-blend-multiply: background-blend-mode: multiply;
- =>bg-blend-screen: background-blend-mode: screen;
- =>bg-blend-overlay: background-blend-mode: overlay;
- =>bg-blend-darken: background-blend-mode: darken;
- =>bg-blend-lighten: background-blend-mode: lighten;
- =>bg-blend-color-dodge: background-blend-mode: color-dodge;
- =>bg-blend-color-burn: background-blend-mode: color-burn;

[WEB DESIGNING/Module 8 Tailwind/background blend mode.html at main · TopsCode/WEB DESIGNING · GitHub](#)

# background-blend-mode

=>bg-blend-hard-light: background-blend-mode: hard-light;  
=>bg-blend-soft-light: background-blend-mode: soft-light;  
=>bg-blend-difference: background-blend-mode: difference;  
=>bg-blend-exclusion: background-blend-mode: exclusion;  
=>bg-blend-hue: background-blend-mode: hue;  
=>bg-blend-saturation: background-blend-mode: saturation;  
=>bg-blend-color: background-blend-mode: color;  
=>bg-blend-luminosity: background-blend-mode: luminosity;

# backdrop-blur

In Tailwind CSS, the backdrop-blur utility applies a blur effect to the area behind an element. This effect is part of the Backdrop Filter feature, which allows you to apply visual effects such as blur or color manipulation to the area behind an element

=>backdrop-blur-none: No blur effect;

=>backdrop-blur-sm: Small blur

=>backdrop-blur: Medium blur

=>backdrop-blur-md: Medium blur

=>backdrop-blur-lg: Large blur

=>backdrop-blur-xl: Extra large blur

=>backdrop-blur-2xl: Even larger blur

=>backdrop-blur-3xl: Very large blur

[WEB DESIGNING/Module 8 Tailwind/backdro  
p blur.html at main .](#)

[TopsCode/WEB DESIGNING - GitHub](#)

# border-spacing

In Tailwind CSS, the border-spacing utility controls the space between the borders of adjacent table cells. This utility is particularly useful when working with tables where you want to adjust the spacing between the cells' borders.

- => border-spacing-0: No spacing between table cells.
- ⇒ border-spacing-1: 0.25rem (4px) spacing between table cells.
- ⇒ border-spacing-2: 0.5rem (8px) spacing between table cells.
- ⇒ border-spacing-4: 1rem (16px) spacing between table cells.
- ⇒ border-spacing-8: 2rem (32px) spacing between table cells.
- ⇒ border-spacing-16: 4rem (64px) spacing between table cells.
- ⇒ border-spacing-{value}: You can also set custom spacing using any of Tailwind's spacing units

# border-spacing

In Tailwind CSS, the border-spacing utility controls the space between the borders of adjacent table cells. This utility is particularly useful when working with tables where you want to adjust the spacing between the cells' borders.

- => border-spacing-0: No spacing between table cells.
- ⇒ border-spacing-1: 0.25rem (4px) spacing between table cells.
- ⇒ border-spacing-2: 0.5rem (8px) spacing between table cells.
- ⇒ border-spacing-4: 1rem (16px) spacing between table cells.
- ⇒ border-spacing-8: 2rem (32px) spacing between table cells.
- ⇒ border-spacing-16: 4rem (64px) spacing between table cells.
- ⇒ border-spacing-{value}: You can also set custom spacing using any of Tailwind's spacing units

# caption side

In Tailwind CSS, the `caption-side` utility allows you to control the placement of the caption element in a table. The `caption-side` CSS property is used to position the table's caption either at the top or bottom of the table

=>`caption-top`: Places the caption at the top of the table.

=>`caption-bottom`: Places the caption at the bottom of the table.

# Scale

In Tailwind CSS, the scale utility is used to apply scaling transformations to an element. This allows you to resize an element by scaling it along the X, Y, or both axes

- =>scale-0: Scales the element to 0 (completely shrunk).
- =>scale-50: Scales the element to 50% of its original size.
- =>scale-75: Scales the element to 75% of its original size.
- =>scale-90: Scales the element to 90% of its original size.
- =>scale-100: No scaling (default size, 100%).
- =>scale-110: Scales the element to 110% of its original size.
- =>scale-125: Scales the element to 125% of its original size.
- =>scale-150: Scales the element to 150% of its original size.
- =>scale-200: Scales the element to 200% of its original size.

# Translate

In Tailwind CSS, the translate utility is used to apply translation transformations to an element. Translation moves an element along the X, Y, or both axes.

=>translate-x-0: No horizontal translation.

=>translate-x-px: 1 pixel horizontal translation.

=>translate-x-1: 0.25rem (4px) horizontal translation.

=>translate-x-2: 0.5rem (8px) horizontal translation.

⇒translate-x-4: 1rem (16px) horizontal translation.

⇒translate-x-8: 2rem (32px) horizontal translation.

⇒translate-x-16: 4rem (64px) horizontal translation.

⇒translate-x-full: 100% of the element's width in horizontal translation.

# Skew

In Tailwind CSS, the skew utility is used to apply a skewed transformation to an element, which tilts the element along the X or Y axis.

=>skew-x-0: No horizontal skew.

=>skew-x-1: Skew the element by 1 degree along the X axis.

=>skew-x-2: Skew the element by 2 degrees along the X axis.

=>skew-x-3: Skew the element by 3 degrees along the X axis.

⇒ skew-y-0: No vertical skew.

⇒ skew-y-1: Skew the element by 1 degree along the Y axis.

⇒ skew-y-2: Skew the element by 2 degrees along the Y axis.

⇒ skew-y-3: Skew the element by 3 degrees along the Y axis.

⇒ skew-x-[value] and skew-y-[value]: You can also use custom values (e.g., skew-x-[10deg])

# Animations

In Tailwind CSS, animations allow you to add dynamic motion to elements, such as fading, sliding, or bouncing, without writing custom CSS. Tailwind CSS provides utilities for defining animations and custom keyframes to easily implement animations and transitions.

- =>animate-spin: Spins an element indefinitely.
- ⇒ animate-ping: Makes an element "ping," which involves growing and shrinking.
- ⇒ animate-pulse: Creates a pulsing effect (element fades in and out).
- ⇒ animate-bounce: Bounces an element up and down.

# Duration

In Tailwind CSS, the duration utility defines the length of time a transition effect will take to complete. It is specified in milliseconds (ms), and you can apply it to any transition to control how long the transition should last.

- =>duration-75: 75ms
- ⇒duration-100: 100ms
- ⇒duration-150: 150ms
- ⇒duration-200: 200ms
- ⇒duration-300: 300ms
- ⇒ duration-500: 500ms
- ⇒duration-700: 700ms
- ⇒duration-1000: 1000ms (1 second)

# Duration

In Tailwind CSS, the duration utility defines the length of time a transition effect will take to complete. It is specified in milliseconds (ms), and you can apply it to any transition to control how long the transition should last.

- =>duration-75: 75ms
- ⇒duration-100: 100ms
- ⇒duration-150: 150ms
- ⇒duration-200: 200ms
- ⇒duration-300: 300ms
- ⇒ duration-500: 500ms
- ⇒duration-700: 700ms
- ⇒duration-1000: 1000ms (1 second)

# timing function

In Tailwind CSS, the timing function (or easing function) controls the speed curve of an animation or transition

- =>ease-linear: The transition occurs at a constant speed from start to finish.
- ⇒ ease-in: The transition starts slow and then speeds up toward the end.
- ⇒ ease-out: The transition starts fast and then slows down toward the end.
- ⇒ ease-in-out: The transition starts slow, speeds up in the middle, and slows down toward the end.

[WEB DESIGNING/Module 8 Tailwind/timing\\_function.html at main · TopsCode/WEB DESIGNING · GitHub](#)

# Delay

In Tailwind CSS, the delay utility specifies the delay time before a transition or animation starts. This is useful if you want to apply a pause before the transition begins

- =>delay-75: 75ms delay
- =>delay-100: 100ms delay
- =>delay-150: 150ms delay
- =>delay-200: 200ms delay
- =>delay-300: 300ms delay
- =>delay-500: 500ms delay
- =>delay-700: 700ms delay
- =>delay-1000: 1000ms delay (1 second)

# Tailwind CSS Overflow

This class accepts more than one value in Tailwind CSS. It is the alternative to the **CSS Overflow property**. This overflow is for controlling how an element content is handled that is too large for the container. It tells whether to clip content or to add scroll bars

There is separate property in CSS for **CSS Overflow-x** and **CSS Overflow-y**,

## Overflow classes:

- overflow-auto
- overflow-hidden
- overflow-visible

# Tailwind CSS Overflow

## **Overflow classes:**

- overflow-scroll
- overflow-x-auto
- overflow-y-auto
- overflow-x-hidden
- overflow-y-hidden
- overflow-x-visible
- overflow-y-visible
- overflow-x-scroll
- overflow-y-scroll

# Tailwind CSS Positions

This class accepts more than one value in tailwind CSS. It is the alternative to the CSS Position property. This class is used for controlling how an element is positioned in the DOM.

## Position classes:

- static
- fixed
- absolute
- relative
- sticky

[https://github.com/TopsCode/WEB\\_DESIGNING/blob/  
main/Module\\_8\\_Tailwind/12-postions.html](https://github.com/TopsCode/WEB_DESIGNING/blob/main/Module_8_Tailwind/12-postions.html)

# Tailwind CSS Top/Right/Bottom/Left

These classes accept many values in tailwind CSS in which all the properties are covered in class form. These are the alternative to the CSS Top/Right/Bottom/Left properties. These classes are used to control the alignment of a positioned element. Remember we can use these properties only with positioned elements.

## **Top/Right/Bottom/Left classes:**

- .inset-0
- .inset-y-0
- .inset-x-0
- .top-0
- .right-0
- .bottom-0
- .left-0

[https://github.com/TopsCode/WEB\\_DESIGNING/blob/main/Module\\_8\\_Tailwind/13-positions1.html](https://github.com/TopsCode/WEB_DESIGNING/blob/main/Module_8_Tailwind/13-positions1.html)

The default value of top/right/bottom/left/inset utilities in Tailwind is 0 and auto.

**Note:** You can change the number "0" with the valid "rem" values.

# Tailwind CSS Visibility

This class accepts two values in tailwind CSS. It is the alternative to the [CSS visibility property](#). This class is used to specify whether an element is visible or not in a web document but the hidden elements take up space in the web document. Use the *display* property to remove or hide and delete an element from the browser.

## **Visibility:**

- Invisible
- Visible

[https://github.com/TopsCode/WEB\\_DESIGNING/blob/main/Module\\_8\\_Tailwind/14-visibility.html](https://github.com/TopsCode/WEB_DESIGNING/blob/main/Module_8_Tailwind/14-visibility.html)

# Tailwind CSS Z-index

The tailwind CSS is a utility CSS framework that provides classes to manage our HTML content in the use of CSS. The **tailwind CSS** makes our designing part easier and responsive for multiple platforms. The z-Index utility is for controlling the stack order of an element. It is the alternative to the CSS z-index property. This class is used to describe the z-index along the three-dimensional plane, as given in the below examples.

## Tailwind CSS z-index classes:

- z-0
- z-10
- z-20
- z-30
- z-40
- z-50
- z-auto

[https://github.com/TopsCode/WEB\\_DESIGNING/blob/main/Module\\_8\\_Tailwind/15-z-index.html](https://github.com/TopsCode/WEB_DESIGNING/blob/main/Module_8_Tailwind/15-z-index.html)

# Tailwind CSS Flexbox

The CSS flexbox is a vital feature to develop the frontend, there are four directions available in CSS so in tailwind CSS all the properties are covered as in class form. It is the alternative of CSS flex-direction Property for fast development of front-end.

**Note:** To activate the flex-direction you have to include the flex class in your element before the flex-direction class.

- **Flex Direction:**
- **flex-row**
- **flex-row-reverse**
- **flex-col**
- **flex-col-reverse**

[https://github.com/TopsCode/WEB\\_DESIGNING/blob/main/Module\\_8\\_Tailwind/16-flex-direction.html](https://github.com/TopsCode/WEB_DESIGNING/blob/main/Module_8_Tailwind/16-flex-direction.html)

# Tailwind CSS Flexbox

**flex-row:** It arranges the row the same as the text direction. The default value of flex-direction is a row. It is used to specify that the item has a normal text direction. It makes the item follow the normal text direction in lines.

**Syntax:** <element class="flex flex-row"> Contents... </element>

[https://github.com/TopsCode/WEB\\_DESIGNING/blob/  
main/Module\\_8\\_Tailwind/18-flex.html](https://github.com/TopsCode/WEB_DESIGNING/blob/main/Module_8_Tailwind/18-flex.html)

# Tailwind CSS Flexbox

```
<!DOCTYPE html>

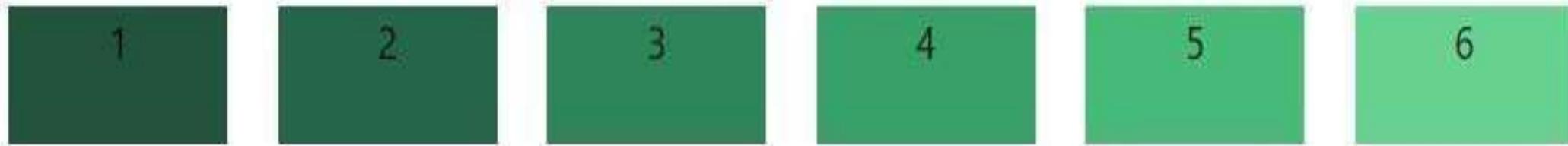
<head>
    <title>Tailwind flex-row Class</title>

    <link href= https://unpkg.com/tailwindcss@^1.0/dist/tailwind.min.css rel="stylesheet">
</head>
<body class="text-center">
    <h1 class="text-green-600 text-5xl font-bold">
        Tops Tech
    </h1>
<b>Tailwind CSS flex-row Class</b>
    <div id="main" class="flex flex-row justify-evenly">
        <div class="bg-green-900 w-24 h-12">1</div>
```

# Tailwind CSS Flexbox

Output of above code :

**Tailwind CSS flex-row Class**



# Tailwind CSS Flex Wrap

The CSS flexbox is a vital feature to develop the frontend, there are three wraps available in CSS so in tailwind CSS all the properties are covered as in class form. It is the alternative of CSS flex-wrap Property for fast development of front-end.

**Note:** To activate the flex-wrap you have to include the flex class in your element before the flex-wrap class.

- **Flex Wrap:**
- **flex-wrap**
- **flexnowrap**
- **flex-wrap-reverse**

## Syntax:

```
<element class="flex flex-wrap"> Contents... </element>
```

[https://github.com/TopsCode/WEB\\_DESIGNING/blob/main/Module\\_8\\_Tailwind/17-flex-wrap.html](https://github.com/TopsCode/WEB_DESIGNING/blob/main/Module_8_Tailwind/17-flex-wrap.html)

# Tailwind CSS Flex

The CSS flexbox is a vital feature to develop the frontend, there is four flex available in tailwind CSS all the properties are covered as in class form. It is the alternative of CSS flex Property for fast development of front-end. It is used to set the length of flexible items. The flex class is much responsive and mobile-friendly. It is easy to positioning child elements and the main container. The margin doesn't collapse with the content margins. The order of any element can be easily changed without editing the HTML section.

## Flex:

- **flex-1**
- **flex-initial**
- **flex-auto**
- **flex-none**

**Syntax :** <element class="flex-1"> Contents... </element>

# Tailwind CSS Flex Grow

The CSS flexbox is a vital feature to develop the frontend, there is two flex-grow available in tailwind CSS all the properties are covered as in class form. It is the alternative of CSS flex grow Property for the fast development of the front-end. This class specifies how much the item will grow compared to other items inside that container. In other words, it is the ability of an item to grow compared to other items present inside the same container.

## Flex Grow:

- **flex-grow-0**
- **flex-grow**
- **Syntax:** <element class="flex-grow-0"> Contents... </element>

# Tailwind CSS Flex Shrink

The CSS flexbox is a vital feature to develop the frontend, there are two flex-shrink available in tailwind CSS all the properties are covered as in class form. It is the alternative of CSS flex-shrink Property for the fast development of the front-end. This class specifies how much the item will shrink compared to other items inside that container. It defines the ability of an element to shrink in comparison to the other elements which are placed inside the same container.

## **Flex shrink :**

- **flex-shrink-0**
- **flex-shrink**

```
<element class="flex-shrink-0"> Contents... </element>
```

## **Syntax :**

[https://github.com/TopsCode/WEB\\_DESIGNING/blob/main/Module\\_8\\_Tailwind/20-flex-shrink.html](https://github.com/TopsCode/WEB_DESIGNING/blob/main/Module_8_Tailwind/20-flex-shrink.html)

# Tailwind CSS Grid

## Tailwind CSS Grid Template Columns

It is used to set the number of columns and size of the columns of the grid, here we will do the same but for fast development of front-end. The number of columns is set by the number of values given to this class.

[https://github.com/TopsCode/WEB\\_DESIGNING/blob/main/Module\\_8\\_Tailwind/22.grid.html](https://github.com/TopsCode/WEB_DESIGNING/blob/main/Module_8_Tailwind/22.grid.html)

# Tailwind CSS Grid

## Grid Template Columns:

- **grid-cols-1:** Each row concedes only one column.
- **grid-cols-2:** Each row concedes only two columns.
- **grid-cols-3:** Each row concedes only three columns.
- **grid-cols-4:** Each row concedes only four columns.
- **grid-cols-5:** Each row concedes only five columns.
- **grid-cols-6:** Each row concedes six columns.
- **grid-cols-7:** Each row concedes seven-columns.
- **grid-cols-8:** Each row concedes eight columns.
- **grid-cols-9:** Each row concedes nine columns.
- **grid-cols-10:** Each row concedes ten columns.
- **grid-cols-11:** Each row concedes eleven columns.
- **grid-cols-12:** Each row concedes twelve columns.
- **grid-cols-none:** Does not follow the grid-column property.

# Tailwind CSS Grid

```
<element class="grid grid-cols-number"> Contents... </element>
```

Example:

```
<!DOCTYPE html>

<html>
  <head> <title>Tailwind grid-cols Class</title>
    <link
      href="https://unpkg.com/tailwindcss@^1.0/dist/tailwind.min.css" rel="stylesheet">
  </head>

  <body class="text-center">
    <h1 class="text-green-600 text-5xl font-bold"> Tops Tech </h1>
    <b>Tailwind CSS grid-cols Class</b>
  </body>
</html>
```

# Tailwind CSS Grid

```
<div id="main" class="grid grid-cols-3 gap-1 justify-evenly">  
    <div class="bg-green-700 w-26 h-12">1</div>  
    <div class="bg-green-500 w-26 h-12">2</div>  
    <div class="bg-green-300 w-26 h-12">3</div>  
    <div class="bg-green-700 w-26 h-12">4</div>  
    <div class="bg-green-500 w-26 h-12">5</div>  
    <div class="bg-green-300 w-26 h-12">6</div> </div>  
</body>  
</html>
```

# Tailwind CSS Grid Column Start / End

It can change the layout of grid items irrespective of their source order, which allows moving grid items around to cater to these varying contexts without having to modify the underlying markup. but for fast development of front-end. The number of columns is set by the number of values given to this class.

# Tailwind CSS Grid Column Start / End

**Spanning columns (col-span):** This class will cover the span area, mentioned number after the class will holds the area of a span, we all know that there are 12 grid column or you can say 12 grid span.

**Syntax:**

```
<element class="col-span-number"> Contents... </element>
```

[https://github.com/TopsCode/WEB\\_DESIGNING/blob/main/Module\\_8\\_Tailwind/23-grid-alignments.html](https://github.com/TopsCode/WEB_DESIGNING/blob/main/Module_8_Tailwind/23-grid-alignments.html)

# Tailwind CSS Grid Column Start / End

**Class Grid Value:** This class accepts a single value as mentioned above and described below:

**number:** It holds the number of spans a grid column will take.

```
<!DOCTYPE html>
<head>
<title>Tailwind col-span Class</title>
<link href=
"https://unpkg.com/tailwindcss@^1.0/dist/tailwind.min.css" rel="stylesheet">
</head>
<body class="text-center">
    <h1 class="text-green-600 text-5xl font-bold"> Tops Tech </h1>
    <b>Tailwind CSS col-span Class</b>
```

# Tailwind CSS Grid Column Start / End

```
<div id="main" class="m-8 grid grid-cols-3 gap-1 justify-evenly">
    <div class="bg-green-300 rounded-lg h-12">1</div>
    <div class="bg-green-300 rounded-lg h-12">2</div>
    <div class="bg-green-300 rounded-lg h-12">3</div>
    <div class="bg-green-500 col-span-2 rounded-lg h-12">4</div>
    <div class="bg-green-300 rounded-lg h-12">5</div>
    <div class="bg-green-500 col-span-3 rounded-lg h-12">6</div>
</div>
</body>
</html>
```

# Tailwind CSS Grid Template

## Rows

It is used to set the number of rows and size of the rows of the grid, here we will do the same but for fast development of front-end. The number of rows is set by the number of values given to this class.

[WEB DESIGNING/Module 8 Tailwind/Tailwind CSS Grid Template.html at main · TopsCode/WEB DESIGNING · GitHub](#)

# Tailwind CSS Grid Template

## Rows

### Grid Template Rows:

- grid-rows-1**: Each row concedes only one row.
- grid-rows-2**: Each row concedes only two rows.
- grid-rows-3**: Each row concedes only three rows.
- grid-rows-4**: Each row concedes only four rows.
- grid-rows-5**: Each row concedes only five rows.
- grid-rows-6**: Each row concedes six rows.
- grid-rows-none**: Does not follow the grid-row property.

### Syntax:

```
<element class="grid grid-rows-number"> Contents...</element>
```

# Tailwind CSS Grid Template

## Rows

Example :

```
<!DOCTYPE html>

<head> <title>Tailwind grid-rows Class</title> <link href=
https://unpkg.com/tailwindcss@^1.0/dist/tailwind.min.css rel="stylesheet"> </head>

<body class="text-center"> <h1 class="text-green-600 text-5xl font-bold"> Tops tech
</h1> <b>Tailwind CSS grid-rows Class</b> <div id="main" class="grid grid-rows-3 grid-
flow-col"> <div class="bg-green-500 rounded-lg m-4 h-12">1</div> <div class="bg-
green-500 rounded-lg m-4 h-12">2</div> <div class="bg-green-500 rounded-lg m-4 h-
12">3</div> <div class="bg-green-500 rounded-lg m-4 h-12">4</div> <div class="bg-
green-500 rounded-lg m-4 h-12">5</div> <div class="bg-green-500 rounded-lg m-4 h-
12">6</div> </div> </body> </html>
```

# Tailwind CSS Grid Row Start / End

It is used to describes the number of properties that allow to design of grid structures and control the placement of grid items using Tailwind CSS. It can change the layout of grid items irrespective of their source order, which allows moving grid items around to cater to these varying contexts without having to modify the underlying markup. But for fast development of front-end. The number of rows is set by the number of values given to this class.

# Tailwind CSS Grid Row Start / End

## Grid Rows Start / End:

- row-auto
- row-span-1
- row-span-2
- row-span-3
- row-span-4
- row-span-5
- row-span-6
- row-span-7
- row-span-8
- row-span-9
- row-span-10
- row-span-11

# Tailwind CSS Grid Row Start / End

row-span-12

row-span-full

row-start-1

row-start-2

row-start-3

row-start-4

row-start-5

row-start-6

row-start-7

row-start-8

row-start-9

row-start-10

row-start-11

# Tailwind CSS Grid Row Start / End

row-start-12

row-start-13

row-start-auto

row-end-1

row-end-2

row-end-3

row-end-4

row-end-5

row-end-6

row-end-7

row-end-8

row-end-9

row-end-10

row-end-11

row-end-12

row-end-13

# Tailwind CSS Grid Row Start / End

**row-end-auto**

**Spanning rows (row-span):** This class will cover the span area, mentioned number after the class will hold the area of a span, we all know that there are 12 grid rows or you can say 12 grid span.

**Syntax:**

```
<element class="row-span-number"> Contents... </element>
```

**Parameter:** This class accepts a single parameter as mentioned above and described below

# Tailwind CSS Grid Row Start / End

```
<!DOCTYPE html> <head>

<title>Tailwind row-span Class</title> <link href=
https://unpkg.com/tailwindcss@^1.0/dist/tailwind.min.css rel="stylesheet"> </head>
<body class="text-center"> <h1 class="text-green-600 text-5xl font-bold"> Tops tech
</h1> <b>Tailwind CSS row-span Class</b> <div id="main" class="m-8 grid grid-row-3
grid-flow-col gap-1"> <div class="bg-green-300 rounded-lg">1</div> <div class="bg-
green-300 rounded-lg">2</div> <div class="bg-green-300 rounded-lg">3</div> <div
class="bg-green-500 row-span-2 rounded-lg">4</div> <div class="bg-green-300
rounded-lg">5</div> <div class="bg-green-500 row-span-3 rounded-lg">6</div> </div>

</body> </html>
```

# Tailwind CSS Grid Auto Flow

it is used to specify how auto-placed items get flowed into the grid items using Tailwind CSS.

## Grid Auto Flow:

- grid-flow-row
- grid-flow-col
- grid-flow-row-dense
- grid-flow-col-dense

**grid-flow-row:** Auto-placement algorithm place the items by filling each row in turn, adding new rows as necessary.

## Syntax:

```
<element class="grid-flow-row"> Contents... </element>
```

# Tailwind CSS Grid Auto Flow

```
<!DOCTYPE html> <html> <head> <title>Tailwind grid-flow-row Class</title> <link href= https://unpkg.com/tailwindcss@^1.0/dist/tailwind.min.css rel="stylesheet"> </head> <body class="text-center"> <h1 class="text-green-600 text-5xl font-bold"> Tops Tech </h1> <b>Tailwind CSS grid-flow-row Class</b> <div class = "m-8 grid grid-flow-row gap-1"> <div class = "bg-green-300 rounded-lg">1</div> <div class = "bg-green-300 rounded-lg">2</div> <div class = "bg-green-300 rounded-lg">3</div> <div class = "bg-green-300 rounded-lg">4</div> </div> </body> </html>
```

# Tailwind CSS Grid Auto Columns

This class is used for utilities to control the size of implicitly-created grid columns.

## Grid Auto columns classes:

- **auto-cols-auto**
- **auto-cols-min**
- **auto-cols-max**
- **auto-cols-fr**

**auto-cols-auto:** It is the default value. The size is determined implicitly according to the size of the container.

## Syntax:

```
<element class="auto-cols-auto">..</element>
```

# Tailwind CSS Grid Auto Columns

```
<!DOCTYPE html> <html> <head> <link href=
https://unpkg.com/tailwindcss@^1.0/dist/tailwind.min.css rel="stylesheet">
</head> <body class="text-center"> <h1 class="text-green-600 text-5xl font-bold">
Tops tech </h1> <b>Tailwind CSS Grid Auto Columns Class</b> <div class ="m-8 p-8
grid bg-green-600 grid-rows-2 grid-flow-col gap-4 auto-cols-auto"> <div class = "p-4 bg-
green-200"> HyperText Markup language </div> <div class = "p-4 bg-green-200">
Cascading Style Sheet </div> <div class = "p-4 bg-green-200"> LiveScript Become
JavaScript </div> </div> </body> </html>
```

# Tailwind CSS Grid Auto Rows

This class is used to specify the size for the rows of implicitly generated grid containers. This class is used to utilities to control the size implicitly-created grid rows.

## Grid Auto Rows classes:

**auto-rows-auto**

**auto-rows-min**

**auto-rows-max**

**auto-rows-fr**

**auto-rows-auto:** It is the default value. The size is determined implicitly according to the size of the container.

## Syntax:

```
<element class="auto-rows-auto">..</element>
```

[WEB DESIGNING/Module 8 Tailwind/Tailwind CSS Grid Auto Rows.html at main · TopsCode/WEB DESIGNING · GitHub](#)

# Tailwind CSS Gap

This class is used to set the spacing also called gutter between the rows and columns. As like column-gap and row-gap using separately both so that one can use simply gap property that can give column as well as row gap.

# Tailwind CSS Gap

- gap-0
- gap-x-0
- gap-y-0
- gap-0.5
- gap-x-0.5
- gap-y-0.5
- gap-1
- gap-x-1
- gap-y-1
- gap-1.5
- gap-x-1.5
- gap-y-1.5
- gap-2
- gap-x-2
- gap-y-2

# Tailwind CSS Gap

gap-2.5

gap-x-2.5

gap-y-2.5

gap-3

gap-x-3

gap-y-3

gap-3.5

gap-x-3.5

gap-y-3.5

gap-4

gap-x-4

gap-y-4

gap-5

gap-x-5

gap-y-5 gap-6 gap-x-6 gap-y-6 gap-7 gap-x-7

# Tailwind CSS Gap

gap-y-7  
gap-8  
gap-x-8  
gap-y-8  
gap-9  
gap-x-9  
gap-y-9  
gap-10  
gap-x-10  
gap-y-10  
gap-11  
gap-x-11  
gap-y-11  
gap-12  
gap-x-12  
gap-y-12

# Tailwind CSS Gap

gap-14

gap-x-14

gap-y-14

gap-16

gap-x-16

gap-y-16

gap-20

gap-x-20

gap-y-20

gap-24

gap-x-24

gap-y-24

gap-28

gap-x-28

gap-y-28

gap-32

# Tailwind CSS Gap

gap-x-32

gap-y-32

gap-36

gap-x-36

gap-y-36

gap-40

gap-x-40

gap-y-40

gap-44

gap-x-44

gap-y-44

gap-48

gap-x-48

gap-y-48

gap-52

gap-x-52

# Tailwind CSS Gap

- gap-y-52
- gap-56
- gap-x-56
- gap-y-56
- gap-60
- gap-x-60
- gap-y-60
- gap-64
- gap-x-64
- gap-y-64
- gap-72
- gap-x-72
- gap-y-72
- gap-80
- gap-x-80
- gap-y-80

# Tailwind CSS Gap

- gap-96
- gap-x-96
- gap-y-96
- gap-px
- gap-x-px
- gap-y-px

# Tailwind CSS Gap

**gap-number:** By using only gap class without mentioning the axis will keep the in both axis according to the mentioned number:

**Syntax:**

```
<element class="gap-number"> Contents... </element>
```

**Parameter:** This class accept a single value as mentioned above and described below:

- number:** Holds the gap size of the element.

[WEB DESIGNING/Module 8 Tailwind/Tailwind CSS Gap.html at main · TopsCode/WEB DESIGNING · GitHub](#)

# Tailwind CSS Justify Content

This class is used to describe the alignment of the flexible box container. It contains the space between and around content items along the main axis of a flex container. It is basically used for controlling how flex and grid items are positioned along a container's main axis.

## Justify Content classes:

- justify-start
- justify-end
- justify-center
- justify-between
- justify-around
- justify-evenly

**justify-start:** It is used to align flex items from the start of the container.

### Syntax:

```
<element class="justify-start">...</element>
```

[WEB DESIGNING/Module 8 Tailwind/Tailwind CSS Justify Content.html at main · TopsCode/WEB DESIGNING · GitHub](#)

# Tailwind CSS Justify Content

```
<!DOCTYPE html><head><link href=
https://unpkg.com/tailwindcss@^1.0/dist/tailwind.min.css rel="stylesheet"> </head>

<body class="text-center"> <h1 class="text-green-600 text-5xl font-bold"> Tops tech
</h1> <b>Tailwind CSS Justify Content Class</b> <div id="main" class="flex justify-
start flex-row"> <div class="bg-green-700 w-24 h-12">1</div>

    <div class="bg-green-600 w-24 h-12">2</div>

        <div class="bg-green-500 w-24 h-12">3</div>

        <div class="bg-green-400 w-24 h-12">4</div>

    </div>
</body>

</html>
```

# Tailwind CSS Padding

This class is used to create space around the element, inside any defined border. We can set different paddings for individual sides (top, right, bottom, left). It is important to add border properties to implement padding properties. There are lots of CSS properties used for padding like CSS *padding-top*, CSS *padding-bottom*, CSS *padding-right*, CSS *padding-left*, etc.

## Padding classes:

- p-0:** This class is used to define the padding on all the sides.
- py-0:** This class is used to define padding on the y-axis i.e *padding-top* and *padding-bottom*.
- px-0:** This class is used to define padding on the x-axis i.e *padding-left* and *padding-right*.
- pt-0:** This class is specially used to add padding on top.
- pr-0:** This class is specially used to add padding on right.
- pb-0:** This class is specially used to add padding on the bottom.
- pl-0:** This class is specially used to add padding on left.

**Note:** You can change the number "0" with the valid "rem" values.

**p-0:** This class is used to define the padding on all the sides.

## Syntax:

```
<element class="p-0">...</element>
```

# Tailwind CSS Padding

```
<!DOCTYPE html> <head> <link href=
https://unpkg.com/tailwindcss@^1.0/dist/tailwind.min.css rel="stylesheet"> </head>
<body class="text-center"> <h1 class="text-green-600 text-5xl font-bold"> Tops tech
</h1> <b>Tailwind CSS Padding Class</b>
<div class="flex justify-center">
<div class="p-4 bg-green-300 w-24 h-24">
<div class="border-2 border-green-800
bg-green-600 w-16 h-16">
</div>
</div> </div> </body> </html>
```

[WEB DESIGNING/Module 8 Tailwind/Tailwind CSS Padding.html at main · TopsCode/WEB DESIGNING · GitHub](#)

# Tailwind CSS Margin

This class is used to create space around the element, outside any defined border. We can set different margins for individual sides (top, right, bottom, left). It is important to add border properties to implement margin classes. There are lots of CSS properties used for margin like *CSS padding-top*, *CSS padding-bottom*, *CSS padding-right*, *CSS padding-left*, etc.

## Margin Classes:

- **m-0:** This class is used to define the margin on all sides.
- **-m-0:** This class is used to define the negative margin on all the sides.
- **my-0:** This class is used to define margin on the y-axis i.e *margin-top* and *margin-bottom*.
- **-my-0:** This class is used to define negative margin on the y-axis i.e *margin-top* and *margin-bottom*.

# Tailwind CSS Margin

- **mx-0:** This class is used to define margin on the x-axis i.e *margin-left* and *margin-right*.
- **-mx-0:** This class is used to add a negative margin on right.
- **mt-0:** This class is specially used to add a margin on top.
- **-mt-0:** This class is specially used to add a negative margin on top.
- **mr-0:** This class is specially used to add a margin on right.
- **-mr-0:** This class is specially used to add a negative margin on right.
- **mb-0:** This class is specially used to add a margin on the bottom.
- **-mb-0:** This class is specially used to add a negative margin on the bottom.
- **ml-0:** This class is specially used to add a margin on left.
- **-ml-0:** This class is specially used to add a negative margin on left.

**Note:** You can change the number “0” with the valid “rem” values.

**m-0:** This class is used to define the margin on all sides.

**Syntax:** `<element class="m-0">...</element>`

[WEB DESIGNING/Module 8 Tailwind/Tailwind CSS Margin.html at main · TopsCode/WEB DESIGNING · GitHub](#)

# Tailwind CSS Space Between

This class is used for controlling the space between child elements.

## Space Between classes:

- space-y-0
- space-y-reverse
- -space-y-0
- space-x-0
- space-x-reverse
- -space-x-0

**Note:** You can change the number "0" with the valid "rem" values.

**space-y-0:** This class is used to perform space on the y-axis.

## Syntax:

```
<element class="space-y-0">...</element>
```

# Tailwind CSS Space Between

```
<!DOCTYPE html> <head> <link href=
https://unpkg.com/tailwindcss@^1.0/dist/tailwind.min.css rel="stylesheet"> </head>

<body class="text-center"> <h1 class="text-green-600 text-5xl font-bold"> Tops tech
</h1> <b>Tailwind CSS Space Between Class</b> <div class="mx-4 space-y-4"> <div
class="bg-green-400 h-16 rounded-lg border-2 border-green-800">1</div> <div
class="bg-green-400 h-16 rounded-lg border-2 border-green-800">2</div> <div
class="bg-green-400 h-16 rounded-lg
border-2 border-green-800">3</div>

<div class="bg-green-400 h-16 rounded-lg
border-2 border-green-800">4</div>

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

[WEB DESIGNING/Module 8 Tailwind/Tailwind CSS Space Between.html](#)  
[at main · TopsCode/WEB DESIGNING - GitHub](#)

# Tailwind CSS Width

This class is used to set the width of the text, images. The width can be assigned to the text and images in the form of pixels(px), percentage(%), centimeter(cm) etc.

## Width classes:

- **w-0:** This class means the width is set to zero.
- **w-auto:** This class means the width is set according to the content
- **w-1/2:** This class means the width is set to half of the window.
- **w-1/3:** This class means the width is set to one-third of the window.
- **w-1/4:** This class means the width is set to one-fourth of the window.
- **w-1/5:** This class means the width is set to one-fifth of the window.
- **w-1/6:** This class means the width is set to one-sixth of the window.
- **w-1/12:** This class means the width is set to one-twelfth of the window.

# Tailwind CSS Width

- **w-full:** This class means the width is set to full.
- **w-screen:** This class means the width is set to the screen size.
- **w-min:** This class is used to define the *min-width*.
- **w-max:** This class is used to define the *max-width*.

**Note:** You can change the number with the valid “rem” values or set the percentage value.

**Syntax:**

```
<element class="w-0">...</element>
```

[https://github.com/TopsCode/WEB\\_DESIGNING/blob/main/Module\\_8\\_Tailwind/25-css-sizing.html](https://github.com/TopsCode/WEB_DESIGNING/blob/main/Module_8_Tailwind/25-css-sizing.html)

# Tailwind CSS Min-Width

This class is used to define the minimum width of an element. The value of the width cannot be less than the value of the *min-width*. If the content specified within the element is smaller, *min-width* maintains the specified minimum width.

## Min-Width Classes:

- min-w-0:** This class is used to set the length of *min-width*.
- min-w-full:** This class is used to set the length of *min-width* at full capacity.
- min-w-min:** This class is used to set the length of *min-width* at minimum capacity.
- min-w-max:** This class is used to set the length of *min-width* at maximum capacity.

## Syntax:

```
<element class="min-w-0">...</element>
```

# Tailwind CSS Min-Width

**Example:** The width will change according to the size of the screen.

```
<!DOCTYPE html> <head> <link href=
```

```
"https://unpkg.com/tailwindcss@^1.0/dist/tailwind.min.css" rel="stylesheet">  
</head> <body class="text-center mx-4 space-y-2"> <h1 class="text-green-600 text-5xl  
font-bold"> Tops Tech </h1>  
  
    <b>Tailwind CSS Min-Width Class</b>  
  
    <div class="w-24 h-16 min-w-full md:min-w-0  
          bg-green-400 rounded-lg text-white">  
        </div>  
  
    </body>  
  
</html>
```

[WEB DESIGNING/Module 8 Tailwind/Tailwind CSS Min Width.](#)  
[html at main · TopsCode/WEB DESIGNING · GitHub](#)

# Tailwind CSS Max-Width

This class is used to define the maximum width of an element. The value of the width cannot be greater than the value of the *max-width*. If the content specified within the element is bigger, *max-width* maintains the specified maximum width. In this class, we will see a new, CSS property presentation, the Prose class. It has been used to have an optimal reading width. If you are familiar with the “typography plugin” then you can guess the working of this class.

## Max-Width classes:

- max-w-0:** This class is used to set the screen width as *max-width: 0rem*.
- max-w-none:** This class is used to set the screen width as *max-width: none*.
- max-w-xs:** This class is used to set the screen width as *max-width: 20rem*.

# Tailwind CSS Max-Width

**max-w-sm:** This class is used to set the screen width as *max-width: 24rem*.

**max-w-md:** This class is used to set the screen width as *max-width: 28rem*.

**max-w-lg:** This class is used to set the screen width as *max-width: 32rem*.

**max-w-xl:** This class is used to set the screen width as *max-width: 36rem*.

**max-w-2xl:** This class is used to set the screen width as *max-width: 42rem*;

**max-w-3xl:** This class is used to set the screen width as *max-width: 48rem*.

**max-w-4xl:** This class is used to set the screen width as *max-width: 56rem*.

**max-w-5xl:** This class is used to set the screen width as *max-width: 64rem*.

# Tailwind CSS Max-Width

**max-w-6xl:** This class is used to set the screen width as *max-width: 72rem*.

**max-w-7xl:** This class is used to set the screen width as *max-width: 80rem*.

**max-w-full:** This class is used to set the screen width as *max-width: 100%*.

**max-w-min:** This class is used to set the screen width as *max-width: min-content*.

**max-w-max:** This class is used to set the screen width as *max-width: max-content*.

**max-w-prose:** This class is used to set the screen width as *max-width: 65ch*.

# Tailwind CSS Max-Width

**max-w-screen-sm:** This class is used to set the screen width as *max-width: 640px*.

**max-w-screen-md:** This class is used to set the screen width as *max-width: 768px*.

**max-w-screen-lg:** This class is used to set the screen width as *max-width: 1024px*.

**max-w-screen-xl:** This class is used to set the screen width as *max-width: 1280px*.

**max-w-screen-2xl:** This class is used to set the screen width as *max-width: 1536px*.

## Syntax:

```
<element class="max-w-0">...</element>
```

[WEB DESIGNING/Module 8 Tailwind/Tailwind CSS Max Width.html at main · TopsCode/WEB DESIGNING · GitHub](#)

# Tailwind CSS Height

This class is used to set the height of an element. The height class does not contain padding, margin, and the border of elements.

## Height classes:

- **h-0:** This class sets the *height* to zero.
- **h-auto:** This class sets the *height* according to the content.
- **h-px:** This class is used to set the *height* in 1px fix.
- **h-1/2:** This class sets the *height* to half of the window.
- **h-1/3:** This class sets the *height* to one-third of the window.
- **h-1/4:** This class sets the *height* to one-fourth of the window.
- **h-1/5:** This class sets the *height* to one-fifth of the window.

# Tailwind CSS Height

- h-1/6:** This class sets the height to one-sixth of the window.
- h-full:** This class sets an element's height to 100% of its parent, as long as the parent has a defined height.
- h-screen:** This class used to make an element span the entire height of the viewport.

**Note:** You can change the number with the valid "rem" values or set the percentage value.

**h-0:** This class is used to set the specific height for any element, you can change the number with a valid number of rem units to fix the height of the element.

## Syntax:

```
<element class="h-0">...</element>
```

[https://github.com/TopsCode/WEB\\_DESIGNING/blob/main/Module\\_8\\_Tailwind/25-css-sizing.html](https://github.com/TopsCode/WEB_DESIGNING/blob/main/Module_8_Tailwind/25-css-sizing.html)

# Tailwind CSS Min-Height

This class is used to set the minimum height of an element. The *min-height* class is used when the content of the element is smaller than the *min-height* and if the content is larger than the *min-height*, it has no effect. This class ensures that the value of the *height* class is not less than the specified *min-height* value of the element in consideration.

## Min-Height classes:

- min-h-0
- min-h-full
- min-h-screen

**min-h-0:** This class is used to set the minimum specific height for any element.

## Syntax:

```
<element class="min-h-0">...</element>
```

[WEB DESIGNING/Module 8 Tailwind/Tailwind CSS Min Height.html at main - TopsCode/WEB DESIGNING - GitHub](#)

# Tailwind CSS Max-Height

This class is used to set the maximum height of an element. If the content of the element is larger than the specified maximum-height then the content will overflow otherwise it has no effect. If the content of the element is smaller than it has no effect. The *height* class value can be overridden by the *max-height* class.

## Max-Height classes:

- max-h-0
- max-h-px
- max-h-full
- max-h-screen

**Note:** You can change the number with the valid “rem” values.

**max-h-0:** This class is used to set the maximum specific height for any element.

## Syntax:

```
<element class="max-h-0">...</element>
```

[WEB DESIGNING/Module 8 Tailwind/Tailwind CSS Max Height.html at main - TopsCode/WEB DESIGNING - GitHub](#)

# Tailwind CSS Font Family

This class is used to specify the font of an element. It can have multiple fonts as a backup system i.e. if the browser does not support one font then the other can be used.

## Font family classes:

- .font-sans
- .font-serif
- .font-mono

**font-sans:** This class is used to apply a websafe sans-serif font family, like *ui-sans-serif*, *system-ui*, *-apple-system*, *BlinkMacSystemFont*, "Segoe UI", *Roboto*, "Helvetica Neue", *Arial*, "Noto Sans", *sans-serif*, "Apple Color Emoji", "Segoe UI Emoji", "Segoe UI Symbol", "Noto Color Emoji", etc.

## Syntax:

```
<element class="font-sans">...</element>
```

# Tailwind CSS Font Size

This class is used to set the font size of the text in an HTML document.

## Font size classes:

- text-xs**: This class defines the text size as extra small.
- text-sm**: This class defines the text size as small.
- text-base**: This class defines the text size as base size.
- text-lg**: This class defines the text size as large.
- text-xl**: This class defines the text size as extra-large.
- text-2xl**: This class defines the text size as 2 times extra-large.
- text-3xl**: This class defines the text size as 3 times extra-large.
- text-4xl**: This class defines the text size as 4 times extra-large.
- text-5xl**: This class defines the text size as 5 times extra-large.

# Tailwind CSS Font Size

**text-6xl:** This class defines the text size as 6 times extra-large.

**text-7xl:** This class defines the text size as 7 times extra-large.

**text-8xl:** This class defines the text size as 8 times extra-large.

[https://github.com/TopsCode/WEB\\_DESIGNING/blob/main/Module\\_8\\_Tailwind\\_CSS\\_Font\\_Size.html](https://github.com/TopsCode/WEB_DESIGNING/blob/main/Module_8_Tailwind_CSS_Font_Size.html)

**text-9xl:** This class defines the text size as 9 times extra-large.

**Note:** Change the size in the component with the required size mentioned above.

**Syntax:**

```
<element class="text-size">...</element>
```

# Tailwind CSS Font Smoothing

This class is used for controlling the font smoothing of an element.

## Font smoothing classes:

- antialiased
- subpixel-antialiased

**Note:** This class does not show any effect.

**antialiased:** This utility is to render text using grayscale antialiasing.

## Syntax:

```
<element class="antialiased">...</element>
```

# Tailwind CSS Font Style

If we want to give design to any type of text then we can make the use of Tailwind CSS font style class. It helps to make a better user experience.

## Font Style classes:

- italic
- non-italic

**italic:** This class is used to set the font style as *italic*.

## Syntax:

```
<element class="italic">...</element>
```

# Tailwind CSS Font Weight

This class is used to set the weight or thickness of the font being used with the HTML Text. The font-weight applied will depend on the *font-family* used in the browser. For example, some *font-family* is available only in specific weights.

## Font weight classes:

- font-thin:** This class sets the *font-weight* to 100.
- font-extralight:** This class sets the *font-weight* to 200.
- font-light:** This class sets the *font-weight* to 300.
- font-normal:** This class sets the *font-weight* to 400.
- font-medium:** This class sets the *font-weight* to 500.
- font-semibold:** This class sets the *font-weight* to 600.
- font-bold:** This class sets the *font-weight* to 700.
- font-extrabold:** This class sets the *font-weight* to 800.
- font-black:** This class sets the *font-weight* to 900.

**Note:** Change the weight in the component with the required weight mentioned above.

## Syntax:

```
<element class="font-{weight}">...</element>
```

[WEB DESIGNING/Module 8 Tailwind/Tailwind CSS Font Weight.html at main · TopsCode/WEB DESIGNING · GitHub](#)

# Tailwind CSS Font Variant

## Numeric

This class is used to control the usage of alternate glyphs. This is done in terms of units or markers such as numbers or fractions.

[WEB DESIGNING/Module 8 Tailwind/Tailwind CSS Font Variant.html at main - TopsCode/WEB DESIGNING - GitHub](#)

### Font variant numeric classes:

- normal-nums:** Using *normal-nums* will remove each effect of the *font-variant-numeric* class.
- ordinal:** This value directly indicates the open type values i.e. *ordn*. The term makes use of special glyphs for ordinal markers.
- slashed-zero:** This class *slashed-zero* used a zero with a slash, which proves to be very useful while distinguishing between 0 and O.
- lining-nums:** This class corresponds to the open type values i.e *Inum*. This keyword activates the numbers lying on the baseline.
- oldstyle-nums:** This class corresponds to the open type values i.e *onum*. This keyword activates the set of figures where some numbers have descendant.

# Tailwind CSS Font Variant Numeric

**proportional-nums:** This class activates those norms where not every number is of the same size. Its open type value is *pnum*.

**tabular-nums:** This class open type value is *tnum*. It activates those sets of figures where a set of numbers is of the same size.

**diagonal-fractions:** It's open type value is *frac*. This activates those sets of figures where numerator and denominator are made smaller and are separated by a slash.

**stacked-fractions:** It's open type value is *arac*. This activates those set of figures where numerator and denominator are made smaller, stacked, and are separated by a horizontal line.

## Syntax:

```
<element class="Font-Variant-Numeric">...</element>
```

# Tailwind CSS Letter Spacing

This class is used to set the spacing behavior between text characters i.e, increasing or decreasing the space between characters in a text.

## Letter Spacing classes:

- tracking-tighter**: The *tracking-tighter* letter spacing for the zero space between characters, the value will be -0.05em.
- tracking-tight**: The *tracking-tight* letter spacing for the little space between characters, the value will be -0.025em.
- tracking-normal**: The *tracking-normal* letter spacing for the current font i.e no extra space between characters. This is the default value.
- tracking-wide**: The *tracking-wide* letter spacing for a little more space between characters compared to normal, the value will be 0.025em.
- tracking-wider**: The *tracking-wider* letter spacing for a little more space between characters compared to wide, the value will be 0.05em.
- tracking-widest**: The *tracking-widest* letter spacing for a little more space between characters compared to wider, the value will be 0.1em.

## Syntax:

```
<element class="tracking-{size}">...</element>
```

[WEB DESIGNING/Module 8 Tailwind/Tailwind CSS Letter Spacing.html at main - TopsCode/WEB DESIGNING - GitHub](#)

# Tailwind CSS Background

## Image

This class is used to set one or more background images to an element. By default, it places the image on the top left corner. To specify two or more images, separate the URLs with a comma.

### Background Image classes:

- bg-none:** This class is used not to set any *linear-gradient*.
- bg-gradient-to-t:** This class is used to set the *linear-gradient* to the top.
- bg-gradient-to-tr:** This class is used to set the *linear-gradient* to the top right.
- bg-gradient-to-r:** This class is used to set the *linear-gradient* to right.

# Tailwind CSS Background

## Image

- **bg-gradient-to-br:** This class is used to set the *linear-gradient* to the bottom right.
- **bg-gradient-to-b:** This class is used to set the *linear-gradient* to the bottom.
- **bg-gradient-to-bl:** This class is used to set the *linear-gradient* to the bottom left.
- **bg-gradient-to-l:** This class is used to set the *linear-gradient* to left.
- **bg-gradient-to-tl:** This class is used to set the *linear-gradient* to the top left.

### Syntax:

```
<element class="bg-gradient-to-{direction}">...</element>
```

# Tailwind CSS Background Color

This class is used to specify the background color of an element. The background covers the total size of the element with padding and border but excluding margin. It makes the text so easy to read for the user.

## Background Color classes:

- **background-transparent:** The background color will be transparent.
- **background-current:** The background will color depend on the parent element color.
- **background-black:** The background color will be black.
- **background-white:** The background color will be white.
- **background-gray-50:** The background color will be gray.
- **background-red-50:** The background color will be red.

# Tailwind CSS Background Color

- background-blue-50:** The background color will be blue.
- background-indigo-50:** The background color will be indigo.
- background-purple-50:** The background color will be purple.
- background-green-50:** The background color will be green.
- background-yellow-50:** The background color will be yellow.
- background-pink-50:** The background color will be pink.

**Note:** The color's values can be changeable according to your need from 50-900, the span should be 100, after the 100.

## Syntax:

```
<element class="background-{color}">...</element>
```

# Tailwind CSS Gradient Color Stops

## Gradient Color Stops classes:

- from-transparent:** This class is used to set the start color transparency.
- from-current:** This class is used to adopt the parent color for the element that will use as the start color.
- from-color-number:** This class is used to set the starting color of a gradient.
- via-transparent:** This class is used to set the via color transparency.
- via-current:** This class is used to adopt the parent color for the element that will use as the via color.

# Tailwind CSS Gradient Color Stops

- via-color-number:** This class is used to set via the color of a gradient.
- to-transparent:** This class is used to set the end color transparency.
- to-current:** This class is used to adopt the parent color for the element that will use as the end color.
- to-color-number:** This class is used to set the ending color of a gradient.

[https://github.com/TopsCode/WEB\\_DESIGNING/blob/main/Module\\_8\\_Tailwind/27-bacground-image.html](https://github.com/TopsCode/WEB_DESIGNING/blob/main/Module_8_Tailwind/27-bacground-image.html)

# Tailwind CSS Border Radius

This class is used to set the border-radius.

## Border Radius Classes:

- rounded-none
- rounded-sm
- rounded
- rounded-md
- rounded-lg
- rounded-xl
- rounded-2xl
- rounded-3xl
- rounded-full
- rounded-t-none
- rounded-r-none
- rounded-b-none
- rounded-l-none

# Tailwind CSS Border Radius

- rounded-t-sm
- rounded-r-sm
- rounded-b-sm
- rounded-l-sm
- rounded-t
- rounded-r
- rounded-b
- rounded-l
- rounded-t-md
- rounded-r-md
- rounded-b-md
- rounded-l-md
- rounded-t-lg
- rounded-r-lg
- rounded-b-lg

[WEB DESIGNING/Module 8 Tailwind/Tailwind CSS Border Radius.](#)  
[html at main · TopsCode/WEB DESIGNING · GitHub](#)

# Tailwind CSS Border Radius

- rounded-l-lg
- rounded-t-xl
- rounded-r-xl
- rounded-b-xl
- rounded-l-xl
- rounded-t-2xl
- rounded-r-2xl
- rounded-b-2xl
- rounded-l-2xl
- rounded-t-3xl
- rounded-r-3xl
- rounded-b-3xl
- rounded-l-3xl
- rounded-t-full

[WEB DESIGNING/Module 8 Tailwind/Tailwind CSS Border Radius.html at main · TopsCode/WEB DESIGNING · GitHub](#)

# Tailwind CSS Border Radius

- rounded-r-full
- rounded-b-full
- rounded-l-full
- rounded-tl-none
- rounded-tr-none
- rounded-br-none
- rounded-bl-none
- rounded-tl-sm
- rounded-tr-sm
- rounded-br-sm
- rounded-bl-sm
- rounded-tl
- rounded-tr
- rounded-br

[WEB DESIGNING/Module 8 Tailwind/Tailwind CSS Border Radius.html at main · TopsCode/WEB DESIGNING · GitHub](#)

# Tailwind CSS Border Radius

- rounded-bl
- rounded-tl-md
- rounded-tr-md
- rounded-br-md
- rounded-bl-md
- rounded-tl-lg
- rounded-tr-lg
- rounded-br-lg
- rounded-bl-lg
- rounded-tl-xl
- rounded-tr-xl
- rounded-br-xl
- rounded-bl-xl
- rounded-tl-2xl

[WEB DESIGNING/Module 8 Tailwind/Tailwind CSS Border Radii.html at main · TopsCode/WEB DESIGNING · GitHub](#)

# Tailwind CSS Border Radius

- rounded-tr-2xl
- rounded-br-2xl
- rounded-bl-2xl
- rounded-tl-3xl
- rounded-tr-3xl
- rounded-br-3xl
- rounded-bl-3xl
- rounded-tl-full
- rounded-tr-full
- rounded-br-full
- rounded-bl-full

**Rounded corners:** In this section, classes are covered that have been used to create rounded corners like rounded-sm, rounded-md, rounded-lg, etc, but not the fully circular or the pill shapes.

**Syntax:** `<element class="rounded-{Border-Radius}">...</element>`

[WEB DESIGNING/Module 8 Tailwind/Tailwind CSS Border Radius.html](#)  
[at main · TopsCode/WEB DESIGNING · GitHub](#)

# Tailwind CSS Border Width

This class is used to set the border width of all four sides of an element. The border-width property is the combination of four property

## Border Width Classes:

- .border
- .border-0
- .border-2
- .border-4
- .border-8
- .border-t
- .border-t-0
- .border-t-2
- .border-t-4
- .border-t-8
- .border-r
- .border-r-0
- .border-r-2
- .border-r-4

[WEB DESIGNING/Module 8 Tailwind/Tailwind CSS Border Width.html at main · TopsCode/WEB DESIGNING · GitHub](#)

# Tailwind CSS Border Width

- border-r-8
- border-b
- border-b-0
- border-b-2
- border-b-4
- border-b-8
- border-l
- border-l-0
- border-l-2
- border-l-4
- border-l-8

**All sides:** In this section class has been used to set the border width for all sides of an element. The classes are border, border-0, border-2, border-4, and border-8.

**Syntax:** `<element class="border-{number}">...</element>`

# Tailwind CSS Border Color

This class is used to specify the border color of an element.

## Border Color classes:

- **border-transparent**: The border color will be transparent.
- **border-current**: The border will color depend on the parent element color.
- **border-black**: The border color will be black.
- **border-white**: The border color will be white.
- **border-gray-50**: The border color will be gray.
- **border-red-50**: The border color will be red.
- **border-blue-50**: The border color will be blue.
- **border-indigo-50**: The border color will be indigo.
- **border-purple-50**: The border color will be purple.
- **border-green-50**: The border color will be green.
- **border-yellow-50**: The border color will be yellow.
- **border-pink-50**: The border color will be pink.

**Note:** The color's values can be changeable according to your need from 50-900, the span should be 100, after the 100.

**Syntax:** `<element class="border-{color}">...</element>`

[\*\*WEB DESIGNING/Module 8 Tailwind/Tailwind CSS Border Color.html\*\*](#)  
[\*\*at main · TopsCode/WEB DESIGNING · GitHub\*\*](#)

# Tailwind CSS Border Opacity

## Border Opacity class:

- border-opacity-0
- border-opacity-5
- border-opacity-10
- border-opacity-20
- border-opacity-25
- border-opacity-30
- border-opacity-40
- border-opacity-50
- border-opacity-60
- border-opacity-70
- border-opacity-75
- border-opacity-80
- border-opacity-90
- border-opacity-95
- border-opacity-100

## Syntax:

```
<element class="border-opacity-{amount}">...</element>
```

# Tailwind CSS Border Style

This class is used for controlling the style of an element's borders.

## Border Style Classes:

- **border-solid**
- **border-dashed**
- **border-dotted**
- **border-double**
- **border-none**
- **border-hidden**

## Syntax:

```
<element class="border-{style}">...</element>
```

[https://github.com/TopsCode/WEB\\_DESIGNING/blob/main/Module\\_8\\_Tailwind/29-borders.html](https://github.com/TopsCode/WEB_DESIGNING/blob/main/Module_8_Tailwind/29-borders.html)

# Tailwind CSS Border Style

```
<!DOCTYPE html>

<html><head><link
href="https://unpkg.com/tailwindcss@^1.0/dist/tailwind.min.css"
rel="stylesheet"></head><body class="text-center"><h1 class="text-green-
600 text-5xl font-bold">Tops tech</h1><b>Tailwind CSS Border Style
Class</b><div class="mx-4 grid grid-cols-5 gap-2 bg-green-200 p-2"><!-- First
sub div is not for roundig--><div class="border-4 border-green-900 border-
solidbg-green-400 w-full h-12">border-solid</div><div class="border-4
border-green-900 border-dashedbg-green-400 w-full h-12">border-
dashed</div>

<div class="border-4 border-green-900 border-dottedbg-green-400 w-full h-
12">border-dotted</div>
```

[https://github.com/TopsCode/WEB\\_DESIGNING/blob/main/Module\\_8\\_Tailwind/29-borders.html](https://github.com/TopsCode/WEB_DESIGNING/blob/main/Module_8_Tailwind/29-borders.html)

# Tailwind CSS Border Style

```
<div class="border-4 border-green-900 border-doublebg-green-400 w-full h-12">border-double</div><div class="border-4 border-green-900 border-nonebg-green-400 w-full h-12">border-none</div><div class="border-4 border-green-900 border-hiddenbg-green-400 w-full h-12">border-hidden</div></div>

</body>

</html>
```

# Box Shadow classes:

**shadow-sm:** This class is used to create a faded or small shadow effects on the box.

**shadow:** This class is used to create normal shadow effects on the box.

**shadow-md:** This class is used to create *md* effects on the box.

**shadow-lg:** This class is used to create *lg* shadow effects on the box.

**shadow-xl:** This class is used to create *xl* shadow effects on the box.

**shadow-2xl:** This class is used to create *2xl* shadow effects on the box.

**shadow-inner:** This class is used to create shadow effects inside the box.

**shadow-none:** This class is used to dilute the shadow effects.

**Syntax:** <element class="shadow-{ shadow-depth }">...</element>

# Tailwind CSS Opacity

## Opacity class:

**opacity-0**: Control the opacity of an element.

**Note:** The number of the opacity can be changeable from 0 to 100 with the span of 5.

## Syntax:

```
<element class="opacity-{number}">...</element>
```

[WEB DESIGNING/Module 8 Tailwind/Tailwind CSS Opacity.html at main · TopsCode/WEB DESIGNING · GitHub](#)

# Tailwind CSS Filter

Tailwind CSS newly added feature filter in 2.1 version.

## Filter Classes:

- .filter: This class is used to enable filters.
- .filter-none: This class is used to remove filters.

## Syntax:

```
<element class="filter">..</element>
```

[https://github.com/TopsCode/WEB\\_DESIGNING/blob/main/Module\\_8\\_Tailwind/31-filter.html](https://github.com/TopsCode/WEB_DESIGNING/blob/main/Module_8_Tailwind/31-filter.html)

# Tailwind CSS Blur

## Blur:

- blur-0:** This class is equivalent to no blur effect as blur(0) in CSS.
- blur-sm:** This class is equivalent to small blur effect as blur(4px) in CSS.
- blur:** This class is equivalent to normal blur effect as blur(8px) in CSS.
- blur-md:** This class is equivalent to medium blur effect as blur(12px) in CSS.
- blur-lg:** This class is equivalent to large blur effect as blur(16px) in CSS.
- blur-xl:** This class is equivalent to extra large blur effect as blur(24px) in CSS.
- blur-2xl:** This class is equivalent to double extra large blur effect as blur(40px) in CSS.
- blur-3xl:** This class is equivalent to triple extra large blur effect as blur(64px) in CSS.

## Syntax:

```
<element class="filter blur-{amount}">..</element>
```

# Tailwind CSS Brightness

## Brightness:

- **brightness-0:** This class used to set the brightness equivalent to CSS brightness(0).
- **brightness-50:** This class used to set the brightness equivalent to CSS brightness(0.5).
- **brightness-75:** This class used to set the brightness equivalent to CSS brightness(0.75).
- **brightness-90:** This class used to set the brightness equivalent to CSS brightness(0.9).
- **brightness-95:** This class used to set the brightness equivalent to CSS brightness(0.95).
- **brightness-100:** This class used to set the brightness equivalent to CSS brightness(1.0).

[WEB DESIGNING/Module 8 Tailwind/Tailwind CSS Brightness.html at main · TopsCode/WEB DESIGNING · GitHub](#)

# Tailwind CSS Brightness

- brightness-105:** This class used to set the brightness equivalent to CSS brightness(1.05).
- brightness-110:** This class used to set the brightness equivalent to CSS brightness1.10).
- brightness-125:** This class used to set the brightness equivalent to CSS brightness(1.25).
- brightness-150:** This class used to set the brightness equivalent to CSS brightness(1.50).
- brightness-200:** This class used to set the brightness equivalent to CSS brightness(2.0).

## Syntax:

```
<element class="filter brightness-{amount}">..</element>
```

# Tailwind CSS Contrast

## Contrast:

- Contrast-0:** This class used to set the contrast equivalent to CSS `contrast(0)`.
- Contrast-50:** This class used to set the contrast equivalent to CSS `contrast(0.5)`.
- Contrast-75:** This class used to set the contrast equivalent to CSS `contrast(0.75)`.
- Contrast-100:** This class used to set the contrast equivalent to CSS `contrast(1.0)`.
- Contrast-125:** This class used to set the contrast equivalent to CSS `contrast(1.25)`.
- Contrast-150:** This class used to set the contrast equivalent to CSS `contrast(1.50)`.
- Contrast-200:** This class used to set the contrast equivalent to CSS `contrast(2.0)`.

**Syntax:** `<element class="filter Contrast-{amount}">..</element>`

[WEB DESIGNING/Module 8 Tailwind/Tailwind CSS Contrast.html at main · TopsCode/WEB DESIGNING · GitHub](#)

# Tailwind CSS Drop Shadow

## Drop Shadow:

- **drop-shadow-sm:** This class is used to set the small shadow effect.
- **drop-shadow:** This class is used to set the normal shadow effect.
- **drop-shadow-md:** This class is used to set the medium shadow effect.
- **drop-shadow-lg:** This class is used to set the large shadow effect.
- **drop-shadow-xl:** This class is used to set the extra-large shadow effect.
- **drop-shadow-2xl:** This class is used to set the double extra large shadow effect.
- **drop-shadow-none:** This class is used to remove the shadow effect.

## Syntax:

```
<element class="filter drop-shadow-{amount}">..</element>
```

[WEB DESIGNING/Module 8 Tailwind/Tailwind CSS Drop Shadow.html at main · TopsCode/WEB DESIGNING · GitHub](#)

# Tailwind CSS Grayscale

## Grayscale:

**grayscale-0:** This class is used to represents the original image.

**grayscale:** This class is used to represents the linear multipliers on the effect.

## Syntax:

```
<element class="filter grayscale | grayscale-
0">..</element>
```

[WEB DESIGNING/Module 8 Tailwind/TailwindCSS Grayscale.html at main · TopsCode/WEB DESIGNING · GitHub](#)

# Tailwind CSS Hue Rotate

## Hue Rotate:

- hue-rotate-180:** This class used to set the hue-rotate quivalent to CSS hue-rotate(-180).
- hue-rotate-90:** This class used to set the hue-rotate quivalent to CSS hue-rotate(-90).
- hue-rotate-60:** This class used to set the hue-rotate quivalent to CSS hue-rotate(-60).
- hue-rotate-30:** This class used to set the hue-rotate quivalent to CSS hue-rotate(-30).
- hue-rotate-15:** This class used to set the hue-rotate quivalent to CSS hue-rotate(-15).
- hue-rotate-0:** This class used to set the hue-rotate quivalent to CSS hue-rotate(0).

# Tailwind CSS Hue Rotate

**hue-rotate-15:** This class used to set the hue-rotate equivalent to CSS hue-rotate(15).

**hue-rotate-30:** This class used to set the hue-rotate equivalent to CSS hue-rotate(30).

**hue-rotate-60:** This class used to set the hue-rotate equivalent to CSS hue-rotate(60).

**hue-rotate-90:** This class used to set the hue-rotate equivalent to CSS hue-rotate(90).

**hue-rotate-180:** This class used to set the hue-rotate equivalent to CSS hue-rotate(180).

## Syntax:

```
<element class="filter hue-rotate-{amount}">..</element>
```

# Tailwind CSS Invert

## Invert Classes:

- .**invert-0**: This class is used to represent the original color.
- .**invert**: This class is used to represent the inverted color of that original color.

## Syntax:

```
<element class="filter invert | invert-0">..</element>
```

[WEB DESIGNING/Module 8 Tailwind/Tailwind CSS Invert.html at main · TopsCode/WEB DESIGNING · GitHub](#)

# Tailwind CSS Saturate

## Saturate Classes:

- saturate-0:** This class used to set the saturate value, equivalent to CSS saturate(0).
- saturate-50:** This class used to set the saturate value, equivalent to CSS saturate(50).
- saturate-100:** This class used to set the saturate value, equivalent to CSS saturate(100).
- saturate-150:** This class used to set the saturate value, equivalent to CSS saturate(150).
- saturate-200:** This class used to set the saturate value, equivalent to CSS saturate(200).

## Syntax:

```
<element class="filter saturate-{amount}">..</element>
```

[WEB DESIGNING/Module 8 Tailwind/Tailwind CSS Saturate.html at main · TopsCode/WEB DESIGNING · GitHub](#)

# Tailwind CSS Sepia

## Sepia:

- . **sepia-0**: This class is used to represent the original image
- . **sepia**: This class is used to represent the sepia image.

## Syntax:

```
<element class="filter sepia| sepia-0">..</element>
```

[WEB DESIGNING/Module 8 Tailwind/Tailwind CSS Sepia.html at  
main · TopsCode/WEB DESIGNING · GitHub](#)

# Tailwind CSS Border Collapse

```
<!DOCTYPE html> <html> <head> <link href="https://unpkg.com/tailwindcss@^1.0/dist/tailwind.min.css" rel="stylesheet"> </head> <body class="text-center mx-4 space-y-2"> <h1 class="text-green-600 text-5xl font-bold"> Tops tech </h1> <b>Tailwind CSS Border Collapse Class</b> <div class="grid grid-flow-col text-center p-2"> <table class="border-collapse border border-green-900">
```

Frameworks	Release
Year	<a href="#">WEB DESIGNING/Module 8 Tailwind/Tailwind CSS Border Collapse.html at main · TopsCode/WEB DESIGNING · GitHub</a>

```
</tr> </thead>
```

# Tailwind CSS Table Layout

## Table Layout classes:

**table-auto**

**table-fixed**

**table-auto:** This class is used to allow the table to automatically size columns to fit the contents of the cell.

## Syntax:

```
<element class="table-auto">...</element>
```

[https://github.com/TopsCode/WEB\\_DESIGNING/blob/main/Module\\_8\\_Tailwind/32-tables.html](https://github.com/TopsCode/WEB_DESIGNING/blob/main/Module_8_Tailwind/32-tables.html)

# Tailwind CSS Table Layout

```
<!DOCTYPE html> <html> <head> <link href=
"https://unpkg.com/tailwindcss@^1.0/dist/tailwind.min.css" rel="stylesheet"
></head> <body class="text-center mx-4 space-y-2"> <h1 class="text-green-
600 text-5xl font-bold"> Tops tech </h1> <b>Tailwind CSS Table Layout
Class</b> <div class="bg-green-200"> <table class="table-auto border-
separate border border-green-900"> <thead> <tr> <th class="border border-
green-600">Frameworks</th>
<th class="border border-green-600">About</th>
</tr>
</thead>
```

---

[https://github.com/TopsCode/WEB\\_DESIGNING/blob/main/Module\\_8\\_Tailwind/32-tables.html](https://github.com/TopsCode/WEB_DESIGNING/blob/main/Module_8_Tailwind/32-tables.html)

# Tailwind CSS Table Layout

```
<tbody> <tr> <td class="border border-green-600">Tailwind CSS</td><td  
class="border border-green-600"> Tailwind CSS is a highly customizable, low-level CSS  
framework that gives you all of the building blocks </td> </tr> <tr> <td class="border  
border-green-600">Bulma</td> <td class="border border-green-600"> Bulma CSS by  
@jgthms is just perfect. Simple, easily customizable and doesn't impose Javascript  
implementations. </td> </tr> <tr> <td class="border border-green-  
600">Bootstrap</td> <td class="border border-green-600"> Bootstrap is a free and  
open-source CSS framework directed at responsive, mobile-first front-end web  
development. </td> </tr> </tbody> </table>  
  
 </div>  
</body>  
  
</html>
```

[https://github.com/TopsCode/WEB\\_DESIGNING/blob/  
main/Module\\_8\\_Tailwind/32-tables.html](https://github.com/TopsCode/WEB_DESIGNING/blob/main/Module_8_Tailwind/32-tables.html)

# Tailwind CSS Transition Property

## Transition Property classes:

- transition-none:** This value is used to specify that no class will get a transition effect.
- transition-all:** All the class will get a transition effect. This is also the default value for this class.
- transition:** We can specify the names of CSS properties for which transition effect will be applied. We can also specify multiple properties by separating them with a comma.
- transition-colors:** This value is used to specify that the color class will get a transition effect.

[https://github.com/TopsCode/WEB\\_DESIGNING/blob/  
main/Module\\_8\\_Tailwind/33-transition.html](https://github.com/TopsCode/WEB_DESIGNING/blob/main/Module_8_Tailwind/33-transition.html)

# Tailwind CSS Transition Property

- transition-opacity:** This value is used to specify that the opacity class will get a transition effect.
- transition-shadow:** This value is used to specify that the shadow class will get a transition effect.
- transition-transform:** This value is used to specify that the transformation into a defined shape.

## Syntax:

```
<element class="transition-{properties}">...</element>
```

# Tailwind CSS Transform

## Transform Classes:

- transform**: To perform any transformation you have to use this class.
- transform-gpu**: By using this class the transformation will be executed on GPU or CPU that will perform in a more soothing way.
- transform-none**: This class is used to deactivate the transformation effect.

## Syntax:

```
<element class="transform-{trans-on}">...</element>
```

[WEB DESIGNING/Module 8 Tailwind/Tailwind CSS Transform.html at main · TopsCode/WEB DESIGNING · GitHub](#)

# Tailwind CSS Rotate

- rotate-0:** This class is for no rotation.
- rotate-1:** This class is used to rotate 1 degree clockwise.
- rotate-2:** This class is used to rotate 2 degrees clockwise.
- rotate-3:** This class is used to rotate 3 degrees clockwise.
- rotate-6:** This class is used to rotate 6 degrees clockwise.
- rotate-12:** This class is used to rotate 12 degrees clockwise.
- rotate-45:** This class is used to rotate 45 degrees clockwise.
- rotate-90:** This class is used to rotate 90 degrees clockwise.
- rotate-180:** This class is used to rotate 180 degrees clockwise.
- rotate-1:** This class is used to rotate 1 degree anti-clockwise.
- rotate-2:** This class is used to rotate 2 degrees anti-clockwise.

# Tailwind CSS Fill

## **Fill class:**

- fill-current:** This class is used to set the fill color of an SVG to the current text color. This makes it easy to set an element's fill color by combining this class with an existing text color utility.

## **Syntax:**

```
<svg class="fill-current">...</svg>
```

[https://github.com/TopsCode/WEB\\_DESIGNING/blob/main/Module\\_8\\_Tailwind/38-svg-fill.html](https://github.com/TopsCode/WEB_DESIGNING/blob/main/Module_8_Tailwind/38-svg-fill.html)

# Tailwind CSS Fill

```
<!DOCTYPE html> <html> <head> <link href=
https://unpkg.com/tailwindcss@^1.0/dist/tailwind.min.css rel="stylesheet"> </head>
<body class="text-center mx-4 space-y-2"> <h1 class="text-green-600 text-5xl font-
bold"> Tops tech </h1> <b>Tailwind CSS Fill Class</b> <div class="bg-green-200 m-4
grid grid-flow-col gap-4 p-5"> <svg height="150px"
width="600px" xmlns="http://www.w3.org/2000/svg" version="1.1"> <circle
class="fill-current text-red-600" cx="100" cy="100" r="50" /> <circle class="fill-current
text-green-600" cx="250" cy="100" r="50" /> <circle class="fill-current text-blue-600"
cx="400" cy="100" r="50" /> <circle class="fill-current text-yellow-600"
cx="550" cy="100" r="50" /> </svg>
</div>
</body> </html>
```

[https://github.com/TopsCode/WEB\\_DESIGNING/blob/main/Module\\_8\\_Tailwind/38-svg-fill.html](https://github.com/TopsCode/WEB_DESIGNING/blob/main/Module_8_Tailwind/38-svg-fill.html)

# Tailwind CSS Stroke

## Stroke Classes:

- stroke-current**: This class is used to set the stroke color of an SVG to the current text color. This makes it easy to set an element's stroke color by combining this class with an existing text color utility.

## Syntax:

```
<svg class="stroke-current">...</svg>
```

[https://github.com/TopsCode/WEB\\_DESIGNING/blob/main/Module\\_8\\_Tailwind/39-svgstroke.html](https://github.com/TopsCode/WEB_DESIGNING/blob/main/Module_8_Tailwind/39-svgstroke.html)

# Tailwind CSS Stroke Width

## Stroke Width classes:

- stroke-0:** This class set the stroke width to zero.
- stroke-1:** This class set the stroke width to one.
- stroke-2:** This class set the stroke width to two.

## Syntax:

```
<svg class="stroke-{width}">...</svg>
```

# Tailwind CSS Stroke

```
<!DOCTYPE html> <html> <head> <link href=
https://unpkg.com/tailwindcss@^1.0/dist/tailwind.min.css rel="stylesheet"> </head>
<body class="text-center mx-4 space-y-2"> <h1 class="text-green-600 text-5xl font-
bold"> Tops tech </h1> <b>Tailwind CSS Stroke Class</b> <div class="bg-green-900 m-
4 grid grid-flow-col gap-4 p-5">
    <svg height="150px" width="600px"
        xmlns="http://www.w3.org/2000/svg"
        version="1.1">
        <circle class="stroke-current text-red-600" cx="100"
            cy="70" r="50" />
        <circle class="stroke-current text-green-600" cx="250"
            cy="70" r="50" />
    </div>
</body>
```