****

**SRI RACHAPUDY NAGABHUSHANAM DEGREE & PG COLLEGE, BADVEL**

**COMPUTER STUDY MATERIAL**

**SEMESTER: V**

**SUBJECT: Web Interface Designing Technologies**

**GROUP : MPCS/MSCS**

**YEAR : THIRD YEAR**

****

**Semester – V**

**Course 6A: Web Interface Designing Technologies**

**Unit - I**

**HTML:** Introduction to web designing, difference between web applications and desktop applications, introduction to HTML, HTML structure, elements, attributes, headings, paragraphs, styles, colours, HTML formatting, Quotations, Comments, images, tables, lists, blocks and classes, HTML CSS, HTML frames, file paths, layout, symbols, HTML responsive.

**Unit – II**

**HTML forms:** HTML form elements, input types, input attributes, HTML5, HTML graphics, HTML media – video, audio, plug INS, you tube. HTML API’S: Geo location, Drag/drop, local storage, HTML SSE.

**CSS:** CSS home, introduction, syntax, colours, back ground, borders, margins, padding, height/width, text, fonts, icons, tables, lists CSS forms, CSS counters, CSS responsive.

**Unit – III**

**Client side Validation:** Introduction to JavaScript - What is DHTML, JavaScript, basics, variables, string manipulations, mathematical functions, statements, operators, arrays, functions. Objects in JavaScript - Data and objects in JavaScript, regular expressions, exception handling.

**Unit – IV**

**Word press:** Introduction to word press, servers like wamp, bitnami e.tc, installing and configuring word press, understanding admin panel, working with posts and pages, using editor, text formatting with shortcuts, working with media-Adding, editing, deleting media elements, working with widgets, menus.

**Unit – V**

**Working with themes**-parent and child themes, using featured images, configuring settings, user and user roles and profiles, adding external links, protecting word press website from hackers.

**Unit - I**

**HTML:** Introduction to web designing, difference between web applications and desktop applications, introduction to HTML, HTML structure, elements, attributes, headings, paragraphs, styles, colours, HTML formatting, Quotations, Comments, images, tables, lists, blocks and classes, HTML CSS, HTML frames, file paths, layout, symbols, HTML responsive.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**Introduction to Web Designing**

Web design refers to the design of websites that are displayed on the internet. It usually refers to the user experience aspects of website development rather than software development. Web design used to be focused on designing websites for desktop browsers; however, since the mid-2010s, design for mobile and tablet browsers has become ever-increasingly important.

A web designer works on the appearance, layout, and, in some cases, content of a website. Appearance, for instance, relates to the colors, font, and images used. Layout refers to how. information is structured and categorized. A good web design is easy to use, aesthetically pleasing, and suits the user group and brand of the website. Many webpages are designed with a focus on simplicity, so that no extraneous information and functionality that might distract or confuse users appears. As the keystone of a web designer's output is a site that wins and fosters the trust of the target audience, removing as many potential points of user frustration as possible is a critical consideration.

In responsive design, content moves dynamically depending on screen size; in adaptive design, the website content is fixed in layout sizes that match common screen sizes.

**Elements of web design:**

The web design process allows designers to adjust to any preferences and provide effective solutions. There are many standard components of every web design, including:

* Layout
* Images
* Visual hierarchy
* Color scheme
* Typography
* Readability
* Navigation
* Content
* Mobile

**Difference between Web Applications and Desktop Applications**

In general, computer applications can be distinguished into two types;

1. desktop applications (desktop apps)
2. web applications (web apps).

**Desktop applications:** Desktop applications are programs or software that run locally on a computer after installation. These applications take space on the hard drive of the computer and generally run without an internet connection. However, some desktop applications may require an internet connection for updates and some like web browsers are functional only in presence of an internet connection. Microsoft Office applications, Adobe Illustrator, Adobe Photoshop, media players, etc. are a few examples of desktop applications.

**Web applications:**

Web applications are not required to be installed in the computer and are accessible through web browsers in the presence of an internet connection. Contrarily to desktop apps, web applications take space on the server. As most of the web applications are designed to be responsive, interactive, and can run on computers, mobile devices, and all other devices with a web browser, they are considered to be the modern form of computer applications and their use is massively growing day by day. Google docs, YouTube, Facebook, Netflix, etc. are a few examples of web applications.

|  |  |
| --- | --- |
| **Desktop apps** | **Web apps** |
| 1. They require installation on the computer to run. | 1. They are accessible through web browsers and do not require installation |
| 2. Generally, desktop apps do not require an internet connection to run. | . 2. Web apps cannot run without an internet connection. |
| 3.They are accessible only in the machine they are installed in. | 3. They accessible from anywhere and through any device with an internet connection and a web browser. |
| 4. They take space on the hard drive of the local computer | 4. They take space on the remote server. |
| 5. Deployment and updating are to be done individually each on computer. | 5. Deployment and updating done only on the server. |
| 6. They have strict hardware requirements for proper functionality. | 6. Web apps are independent and just require a web browser and inter connection to function |
| 7. Generally, they are faster than web applications | 7. Generally, they are slower than desktop applications. |

**Introduction to HTML**

* HTML stands for Hypertext Markup Language.
* It is used to display the document in the web browsers.
* HTML pages can be developed to be simple text or to be complex multimedia program containing sound, moving images and java applets.
* HTML is considered to be the global publishing format for Internet. It is not a programming language.
* HTML was developed by Tim Berners-Lee.
* HTML standards are created by a group of interested organizations called W3C (World Wide Web consortium).
* HTML is mainly useful to create web pages. HTML contains limited number of tags.
* HTML is initially defined from SGML (standard generalized markup language)

**Points to be remembered for HTML tags:**

* HTML is a set of elements or tags.
* HTML document is a text file made of up elements tags are used to markup elements.
* Tags are always represented by <>. Representations of tags are <element name>.

**HTML Structure**

* Html document begin with <html> tag & ends with </html> tag.
* <html> tag contains two sections.They are

1. head section

2. body section.

* To represent the head section <head></head> tags are used.
* Title element is a part of a head tag. <title>-----------</title>
* Head section includes some stuff (data) that does not show in client area of browser window. It is useful for writing title & technical information (author name, additionally tools) etc of web page.
* Head is not compulsory or must. Body section is compulsory.
* To represent the body section <body> -------- </body> tags are used.
* The data that is to be displayed is written in body tag.

<html>

<head><title> Basic HTML document </title>

</head>

<body>

Welcome to the world of Web Technologies.

A sample HTML program.

</body>

</html>

* Comments in HTML documents start with **<!**and end with **>**. Each comment cancontain as many lines of text as you like. If comment is having more lines, then each linemust start and end with -- and must not contain -- within its body.

<! -- this is a comment line - -

-- which can have more lines - ->

**Elements**

An **HTML element** is defined by a starting tag. If the element contains other content, it ends with a closing tag, where the element name is preceded by a forward slash as shown below with few tags −

|  |  |  |
| --- | --- | --- |
| **Start Tag** | **Content** | **End Tag** |
| <p> | This is paragraph content. | </p> |
| <h1> | This is heading content. | </h1> |
| <div> | This is division content. | </div> |
| <br /> |  |  |

So here **<p>....</p>** is an HTML element, **<h1>...</h1>** is another HTML element. There are some HTML elements which don't need to be closed, such as **<img.../>**, **<hr />** and **<br />** elements. These are known as **void elements**.

HTML documents consists of a tree of these elements and they specify how HTML documents should be built, and what kind of content should be placed in what part of an HTML document.

HTML Tag vs. Element

An HTML element is defined by a *starting tag*. If the element contains other content, it ends with a *closing tag*.

For example, **<p>** is starting tag of a paragraph and **</p>** is closing tag of the same paragraph but **<p>This is paragraph</p>** is a paragraph element.

**Attributes**

An attribute is used to define the characteristics of an HTML element and is placed inside the element's opening tag. All attributes are made up of two parts − a **name** and a **value**

* The **name** is the property you want to set. For example, the paragraph **<p>** element in the example carries an attribute whose name is **align**, which you can use to indicate the alignment of paragraph on the page.
* The **value** is what you want the value of the property to be set and always put within quotations. The below example shows three possible values of align attribute: **left, center** and **right**.

Attribute names and attribute values are case-insensitive. However, the World Wide Web Consortium (W3C) recommends lowercase attributes/attribute values in their HTML 4 recommendation.

Example:

<!DOCTYPE html>

<html>

<head>

<title>Align Attribute Example</title>

</head>

<body>

<p align = "left">This is left aligned</p>

<p align = "center">This is center aligned</p>

<p align = "right">This is right aligned</p>

</body>

</html>

**Headings**

## Heading Tags

Any document starts with a heading. You can use different sizes for your headings. HTML also has six levels of headings, which use the elements **<h1>, <h2>, <h3>, <h4>, <h5>,** and **<h6>**. While displaying any heading, browser adds one line before and one line after that heading.

### Example

### <!DOCTYPE html>

### <html>

### <head>

### <title>Heading Example</title>

### </head>

### <body>

### <h1>This is heading 1</h1>

### <h2>This is heading 2</h2>

### <h3>This is heading 3</h3>

### <h4>This is heading 4</h4>

### <h5>This is heading 5</h5>

### <h6>This is heading 6</h6>

### </body>

### </html>

**Paragraphs**

The **<p>** tag offers a way to structure your text into different paragraphs. Each paragraph of text should go in between an opening <p> and a closing </p> tag as shown below in the example −

Example

<!DOCTYPE html>

<html>

<head>

<title>Paragraph Example</title>

</head>

<body>

<p>Here is a first paragraph of text.</p>

<p>Here is a second paragraph of text.</p>

<p>Here is a third paragraph of text.</p>

</body>

</html>

**Styles**

HTML is quite limited when it comes to the presentation of a web page. It was originally designed as a simple way of presenting information. CSS (Cascading Style Sheets) was introduced in December 1996 by the World Wide Web Consortium (W3C) to provide a better way to style HTML elements.

With CSS, it becomes very easy to specify the things like, size and typeface for the fonts, colors for the text and backgrounds, alignment of the text and images, amount of space between the elements, border and outlines for the elements, and lots of other styling properties.

**Adding Styles to HTML Elements**

Style information can either be attached as a separate document or embedded in the HTML document itself. These are the three methods of implementing styling information to an HTML document.

1. **Inline styles**-Using the style attribute in the HTML start tag.

2**. Embedded style-**Using the <style> element in the head section of the document.

3. **External style sheet** - Using the <link> element, pointing toan external CSS files

**Colours**

Colors are very important to give a good look and feel to your website. You can specify colors on page level using <body> tag or you can set colors for individual tags using **bgcolor** attribute.

The <body> tag has following attributes which can be used to set different colors −

* **bgcolor** − sets a color for the background of the page.
* **text** − sets a color for the body text.
* **alink** − sets a color for active links or selected links.
* **link** − sets a color for linked text.
* **vlink** − sets a color for *visited links* − that is, for linked text that you have already clicked on.

HTML Color Coding Methods

There are following three different methods to set colors in your web page −

* **Color names** − You can specify color names directly like green, blue or red.
* **Hex codes** − A six-digit code representing the amount of red, green, and blue that makes up the color.
* **Color decimal or percentage values** − This value is specified using the rgb( ) property.

**HTML Formatting Tags**

HTML Formatting is a process of formatting text for better look and feel. HTML provides us ability to format text without using CSS. There are many formatting tags in HTML. These tags are used to make text bold, italicized, or underlined. There are almost 14 options available that how text appears in HTML and XHTML

Here, we are going to discuss some HTML formatting tags. Following is the list of HTML formatting text.

**1. bold tag:** This tag is used for implement bold effect on the text

**<b> ……. </b>**

**2. Italic tag:** This implements italic effects on the text.

**<i>…….</i>**

**3. strong tag:** This tag is used to always emphasized the text

**<strong>……….</strong>**

**4. sub and sup tag:** These tags are used for subscript and superscript effects on the text.

**<sub> ……….</sub>**

**<sup>………..</sup>**

**5.** **big tag :** This tag is used to increase the font size by conventional unit.

Ex: <p>SRNB <big>DEGREE COLLEGE</big></p>

**6. Underline tag:** This tag is used to underline text written between it.

Ex: <h5><u>SRNB</u> DEGREE COLLEGE</h5>

**Quotations**

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

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

**EXAMPLE**:

<!DOCTYPE html>

<html>

<head>

<title>Quotations</title>

</head>

<body>

<h3>GeeksforGeeks</h3>

<p>The quick brown fox jumps over the lazy dog<br></p>

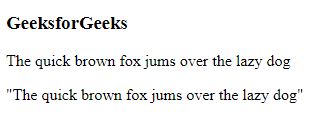
<!--Inside quotes-->

<p><q>The quick brown fox jumps over the lazy dog</q></p>

</body>

</HTML>

**OUTPUT:**



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

<!DOCTYPE html>

<html>

<head>

<title>Blockquote</title>

</head>

<body>

<h3>GeeksforGeeks</h3>

<p>The quick brown fox jumps over the lazy dog<br></p>

<!--Inside blockquotes-->

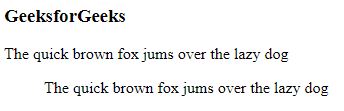
<p><blockquote>The quick brown fox jumps

over the lazy dog</blockquote></p>

</body>

</html>

**OUTPUT:-**



1. **<address> element:** Using the <address> element, we can define an address in a webpage and the text put inside the address tag will be emphasized. Usually line break is added before and after the address tag and content inside this tag is generally renders in italic format. It helps screen reader also It has both opening and closing tags.

Example:-

<!DOCTYPE html>

<html>

<head>

<title>Address</title>

</head>

<body>

<h3>GeeksforGeeks</h3>

<address>

<p>Address:<br>

710-B, Advant Navis Business Park,<br>

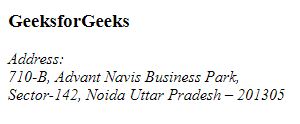
Sector-142, Noida Uttar Pradesh – 201305</p>

</address>

</body>

</html>

**Output**:-



**Comments**

The comment tag ( <! -- comment -->  ) is used to insert comments in the HTML code. It is a good practice of coding, so that coder and the reader can get help to understand the code. It is useful to understand steps of the complex code. The comment tag is helpful while the debugging of codes.

* It is a simple piece of code that is wiped off (ignore) by web browsers i.e. , not displayed by the browser.
* It helps the coder and reader to understand the piece of code used for especially in complex source code.

**Syntax:**

<!-- Comments here -->

**Types of HTML Comments:** There are three types of comments in HTML which are:

* Single-line comment
* Multi-lines comment
* Using <comment> tag

**Single-line comment:**  Single line comment is given inside the ( <!–  comment –> ) tag.

**Multi-line comment:** Multiple lines can be given by the syntax (<!– –>), Basically it’s the same as we used in single line comment, difference is half part of the comment (” –> “), is appended where the intended comment line ends.

**Using <comment> tag:**There used to be an HTML **<comment>** tag, but currently it is not supported by any modern browser.

**Images**

Images are very important to beautify as well as to depict many complex concepts in simple way on your web page. This tutorial will take you through simple steps to use images in your web pages.

**Insert Image**

You can insert any image in your web page by using **<img>** tag. Following is the simple syntax to use this tag.

<img src = "Image URL" ... attributes-list/>

The <img> tag is an empty tag, which means that, it can contain only list of attributes and it has no closing tag.

**Ex**:-

<!DOCTYPE html>

<html>

<head>

<title>Using Image in Webpage</title>

</head>

<body>

<p>Simple Image Insert</p>

<img src = "/html/images/test.png" alt = "Test Image" />

</body>

**Tables**

The HTML table is used to arrange data into rows and columns. In HTML , a table is created by using <table> tag and end with</table>.

**<table align=”center” | “left” | “right” border=”n” bgcolor=”#FFFF”cellpadding=”n”**

**cellspacing=”n” rowspan=”n” colspan=”n”>………………</table>**

Everything that we write between these two tags will be within a table. The attributes ofthe table will control in formatting of the table. The attributes are shown below:

**Cell padding:-** It determines how much spacethere is between the contents of a cell and its border;

**Cell spacing:-** It sets the amount of white space between cells.

**Align:-** aligned to the left, right or center of the page by using an attribute called as align.

**Bgcolor attribute:-** This attribute is used to set background color.

**Colspan and rowspan attributes:-** we will use colspan attribute, if we want to combain two or more columns into a Single cell and rowspan attribute is used to combain two or more rows into a single cell.

**<tr> ….. </tr>**

This is the sub tag of <table>tag, each row of the table has to be delimited by these tags.

**<th>……</th>**

This is again a sub tag of the <tr> tag. This tag is used to show the table heading.

**<td>…..</td>**

This tag is used to give the content of the table.

**Example:** HTML code showing the use of table tag

<html>

<head><title> table page</title>

</head>

<body>

<table align="center" cellpadding="2" cellspacing="2" border="2">

<caption>Timetable for BSC</caption>

<tr>

<th> I period </th>

<th> II peiord</th>

</tr>

<tr>

<td>WT</td>

<td>CLUSTER</td>

</tr>

<tr>

<td>MATHEMATICS</td>

<td>MATHS CLUSTER</td>

</tr>

</table>

</body>

</html>

**OUTPUT**:-

|  |  |
| --- | --- |
| **I period** | **II period** |
| WT | CLUSTER |
| MATHEMATICS | MATHS CLUSTER |

**Lists**

HTML Lists are used to specify a list of information. A list may contain one or more List items. There are 4 types of Lists.

1. UnOrdered List(Bulleted List)
2. Ordered List (Numbered List)
3. Defination List
4. Nested List

**UnOrdered List(Bulleted List):-**

In UnOrdered list, alist of information is specified bulleted formate.The Unordered(bulleted) lists are each made up of sets of listitems. This tag is used to write list items

**<ul type=”disc” | “square” | ”circle” >**

**<li>…….</li>**

**………………**

**<li>…….</li>**

**</ul>**

This tag is used for basic unordered list which uses a bullet in front of each tag; everything between the tag is encapsulated within <li> tags.

**Example:** HTML code showing unordered list.

<html>

<head><title>Fruits Page</title>

</head>

<body>

<h3>Fruits Details </h3><br>

<ul type="disc">

<li>Grapes</li>

<li>Mango</li>

<li>Apple</li>

<li>Guava</li>

<li>Banana</li>

</ul><br>

</body>

</html>

**Ordered List(Bulleted List):-**

In Ordered List, a list of information specified in numbered format. In HTML ordered list is created by using <ol>. The ordered(numbered) lists are each made up of sets of list items. This tag is used to write list items

**<ol type=”1” | ”a” | “I” >**

**<li>…….</li>**

**………………**

**<li>…….</li>**

**</ol>**

**Example:** HTML code showing ordered list.

<html>

<head><title>Fruits Page</title>

</head>

<body>

<h3>Fruits Details </h3>

<ol type="a">

<li>Grapes</li>

<li>Mango</li>

<li>Apple</li>

<li>Guava</li>

<li>Banana</li>

</ol>

</body>

</html>

**Definition lists:**

The definition list specifies a list of information in Descriptive formate. In HTML definition list is created by using <dl>.The definition lists are each made up of sets of list items. This tag is used to write list items

**<dl>….. </dl>**

This tag is used for the third category i.e., definition list, where numbers or bullet is not used infront of the list item, instead it uses definition for the items.

**<dt>…..</dt>**

This is a sub tag of the <dl> tag called as definition term or type, which is used for marking the items whose definition is provided in the next data definition.

**<dd> ….</dd>**

This is a sub tag of the <dt> tag, definition of the terms are enclosed within these tags.The definition may include any text or block.

**Example:** HTML code showing Definitionlist.

<html>

<head><title>Definitions Page</title>

</head>

<body>

<h3>Definition Details </h3><br>

<dl>

<dt>OOP

<dd>Object Oriented Programming</dd>

</dt>

<dt>POP

<dd>Procedure Oriented Programming</dd>

</dt>

</dl>

</body>

</html>

**Nested List:**

In nested list, a list is specified inside of another list.There are two types of Nested lists. They are

1. Nested Ordered List
2. Nested UnOrdered List

Example:-

<html>

<head>

<title>Nested List</title>

</head>

<body>

<ol>

<li>UG

<ul>

<li>BA</li>

<li>B.COM</li>

<li>BSc</li>

</ul>

</li>

</ol>

</body>

</html>

Output:

1.UG

* BA
* B.COM
* BSc

**Blocks and Classes**

All the HTML elements can be categorized into two categories **(a)** Block Level Elements **(b)**Inline Elements.

Block Elements

Block elements appear on the screen as if they have a line break before and after them. For example, the <p>, <h1>, <h2>, <h3>, <h4>, <h5>, <h6>, <ul>, <ol>, <dl>, <pre>, <hr />, <blockquote>, and <address> elements are all block level elements. They all start on their own new line, and anything that follows them appears on its own new line.

Inline Elements

Inline elements, on the other hand, can appear within sentences and do not have to appear on a new line of their own. The <b>, <i>, <u>, <em>, <strong>, <sup>, <sub>, <big>, <small>, <li>, <ins>, <del>, <code>, <cite>, <dfn>, <kbd>, and <var> elements are all inline elements.

Grouping HTML Elements

There are two important tags which we use very frequently to group various other HTML tags (i) <div> tag and (ii) <span> tag

## The <div> tag

This is the very important block level tag which plays a big role in grouping various other HTML tags and applying CSS on group of elements. Even now <div> tag can be used to create webpage layout where we define different parts (Left, Right, Top etc.) of the page using <div> tag. This tag does not provide any visual change on the block but this has more meaning when it is used with CSS.

### Example

Following is a simple example of <div> tag. We will learn Cascading Style Sheet (CSS) in a separate chapter but we used it here to show the usage of <div> tag −

<!DOCTYPE html>

<html>

<head>

<title>HTML div Tag</title>

</head>

<body>

<!-- First group of tags -->

<div style = "color:red">

<h4>This is first group</h4>

<p>Following is a list of vegetables</p>

<ul>

<li>Beetroot</li>

<li>Ginger</li>

<li>Potato</li>

<li>Radish</li>

</ul>

</div>

<!-- Second group of tags -->

<div style = "color:green">

<h4>This is second group</h4>

<p>Following is a list of fruits</p>

<ul>

<li>Apple</li>

<li>Banana</li>

<li>Mango</li>

<li>Strawberry</li>

</ul>

</div>

</body>

</html>

This will produce the following result −

## The <span> tag

The HTML <span> is an inline element and it can be used to group inline-elements in an HTML document. This tag also does not provide any visual change on the block but has more meaning when it is used with CSS.

The difference between the <span> tag and the <div> tag is that the <span> tag is used with inline elements whereas the <div> tag is used with block-level elements.

### Example

Following is a simple example of <span> tag. We will learn Cascading Style Sheet (CSS) in a separate chapter but we used it here to show the usage of <span> tag −

<!DOCTYPE html>

<html>

<head>

<title>HTML span Tag</title>

</head>

<body>

<p>This is <span style = "color:red">red</span> and this is

<span style = "color:green">green</span></p>

</body>

</html>

**HTML CSS**

Cascading Style Sheets (CSS) describe how documents are presented on screens. Cascading Style Sheets (CSS) provide easy and effective alternatives to specify various attributes for the HTML tags. Using CSS, you can specify a number of style properties for a given HTML element. Each property has a name and a value, separated by a colon (:). Each property declaration is separated by a semi-colon (;).

Syntax:-

<style type=”text”/”css”>

P selector

{

Color:green;font-size:15px; declaration

}

</style>

* In the above , the selctor is noemally the HTML element, you want to style.
* Each declaration consists of a property and value.
* Property is a style. Each property has a value.

**HTML Frames**

HTML frames are used to divide your browser window into no. of parts. Where, each part is loaded with a separate HTML document. A collection of frames in the browser window is known as Frameset. There are two tags related to frames i.e., frameset and frame

**<frameset cols=” % , %” | rows=” % , %”>……..</frameset>**

**<frame name=”name” src=”filename” scrolling =” yes” | “no”>**

To create frames in HTML we are using <frameset> tag instead of <body>.

**Example:-**

<html>

<frameset>

----------------

----------------

</frameset>

</html>

**Attributes**:-

There are two attributes of frameset tag.

1. Rows
2. Cols

**1.rows:-**

we use this attribute, if we want to divide your browser window into no. of parts.

<html>

<frameset rows=”50%,50%”>

----------------

----------------

</frameset>

</html>

**1.cols:-**

we use this attribute, if we want to divide your browser window into no. of columns.

<html>

<frameset cols=”50%,50%”>

----------------

----------------

</frameset>

</html>

To add frames into frameset we use <frame> tag inside of frameset tag.

Let us consider an example an html program to implement frames:

<html>

<frameset rows=”50%,50%”>

<frame src=”a.html”>

<frame src=”b.html”>

</frameset>

</html>

**Output:-**

|  |
| --- |
| **a.html** |
| **b.html** |

**Example** : HTML code that implements frames

<html>

<head><title> My page </title>

</head>

<frameset rows="25%,50%">

<frame name="a" src="f:\feroz\wtlab\asgm1.html">

<frameset cols="25%,50%">

<frame name="b" src="f:\feroz\wtlab\index.html">

<frame name="abc" src="f:\feroz\wtlab\welcome.html">

</frameset>

</frameset>

</html>

**File Paths**

A file path specifies the location of a file inside a web folder structure. Its like an address of a file which helps the web browser to access the files. File paths are used to link external resources such as images, videos, style sheets, JavaScript, displaying other web pages etc.  
To insert a file in a web page its source must be known. For example, the syntax (<img src=” ” alt=” “>) is used to insert an image file, where the path of the file is mentioned in the source (src). 

File paths are of two types:   
 1.Absolute File Paths

2.Relative File Paths

**Absolute File Paths:** It describes the full address(URL) to access an internet file.   
 *<img src=”https://media.geeksforgeeks.org/wp-content/uploads/geek.png” alt=”My Image”>*

**Relative File Path:** It describes the path of the file relative to the location of the current web page file.

**Example 1:** It shows the path of the file present in the same folder of the current web page file.

<!DOCTYPE html>

<html>

<head>

<title>Relative file path</title>

</head>

<body>

<h2>File present in the same folder</h2>

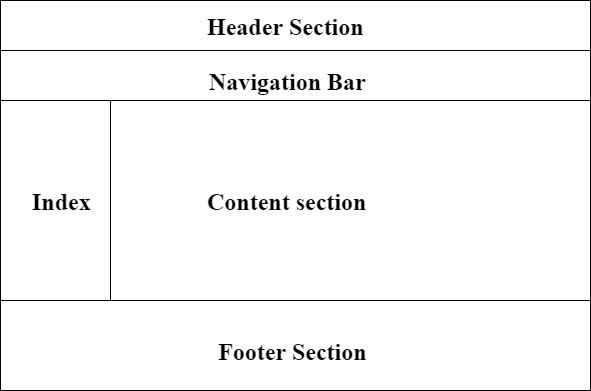
<img src="images/geeks.jpg" alt="My Image" style="width:400px">

</body>

</html>

**Layout**

Page layout is the part of graphic design that deals with the arrangement of visual elements on a page. Page layout is used to make the web pages look better. It establishes the overall appearance, relative importance, and relationships between the graphic elements to achieve a smooth flow of information and eye movement for maximum effectiveness or impact.



* divs have a special class/id associated with them.

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

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

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

**Page Layout Information:**

* **Header:** The part of the front end which is used at the top of the page. <header> tag is used to add a header section on web pages.s

**Syntax:** <header>

<h1> ----- </h1>

<h2> ----- </h2>

----------------

----------------

</header>

* **Navigation bar:** The navigation bar is the same as the menu list. It is used to display the content information using hyperlinks. <nav> tag is used to add the nav section(nav elements) in web pages.

**Syntax:**

<nav>

<ul>

<li> ..... </li>

<li> ..... </li>

</ul>

</nav>

* **Index / Sidebar:** It holds additional information or advertisements and is not always necessary to be added to the page.
* **Content Section:** The content section is the central part where content is displayed.<main> tag is used to add the main content of the webpages.
* **Footer:** The footer section contains the contact information and other query related to web pages. The footer section is always put on the bottom of the web pages. The <footer> tag sets the footer on web pages.

**Syntax:**

<footer>

.....

</footer>

**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.

Mathematical Symbols Supported by HTML

|  |  |  |  |
| --- | --- | --- | --- |
| **Char** | **Number** | **Entity** | **Description** |
| ∀ | &#8704; | &forall; | FOR ALL |
| ∂ | &#8706; | &part; | PARTIAL DIFFERENTIAL |
| ∃ | &#8707; | &exist; | THERE EXISTS |
| ∅ | &#8709; | &empty; | EMPTY SETS |
| ∇ | &#8711; | &nabla; | NABLA |
| ∈ | &#8712; | &isin; | ELEMENT OF |
| ∉ | &#8713; | &notin; | NOT AN ELEMENT OF |
| ∋ | &#8715; | &ni; | CONTAINS AS MEMBER |
| ∏ | &#8719; | &prod; | N-ARY PRODUCT |
| ∑ | &#8721; | &sum; | N-ARY SUMMATION |

**Example:-**

<!DOCTYPE html>

1. <html>
2. <body>
3. <h3>The Currency Symbols</h3>
4. <p>This is Indian Rupee symbol <b>₹<b></p>
5. <p>This is Euro symbol <b>€</b></p>
6. <p> This is Dollar symbol <b>#36;</b></p>
7. </body>
8. </html>

**HTML Responsive**

**Responsive Web design**

Responsive web design is used to make your web page look appropriate, good, and well placedon all devices (desktop, tablet, smartphone etc.)

Responsive web design uses HTML and CSS to resize, hide, shrink, enlarge, or move the content. It makes the content look good on any screen.

**Set the viewport**Play Video

Let's see how to set the viewport.

## Responsive Images

Images which can be scaled nicely to fit any browser size are known as responsive images.

## How to make Image Responsive?

**By using the width property**

Set the CSS width property to 100% to make the image responsive and scale up and down.

**Example**

1. <!DOCTYPE html>
2. <html>
3. <meta name="viewport" content="width=device-width, initial-scale=1.0">
4. <body>
5. <h2>Responsive Image</h2>
6. <p>When we set the CSS width property to 100%, it makes the image responsive.
7. Resize the browser window to see the effect.</p>
8. <img src="img\_girl.jpg" style="width:100%;">( change image)
9. </body>

</html>

## Responsive Text-size

We can make the text size responsive by using the "uv" unit. It means viewport-width. By using this, we can make the text size to follow the browserwindow screen.

**Example**

1. <!DOCTYPE html>
2. <html>
3. <meta name="viewport" content="width=device-width, initial-scale=1.0">
4. <body>
5. <h1 style="font-size:10vw;">Here size is 10vw.</h1>
6. <p style="font-size:6vw;">Here size is 6vw.</p>
7. <p style="font-size:4vw;">Here size is 4vw.</p>
8. <p>Resize the browser window to see how the text size changes.</p>
9. </body>
10. </html>

**Unit – II**

**HTML forms:** HTML form elements, input types, input attributes, HTML5, HTML graphics, **HTML media** – video, audio, plug INS, you tube.

**HTML API’S:** Geo location, Drag/drop, local storage, HTML SSE.

**CSS:** CSS home, introduction, syntax, colours, back ground, borders, margins, padding, height/width, text, fonts, icons, tables, lists CSS forms, CSS counters, CSS responsive.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**HTML forms**

HTML forms are used when we want to collect data from the user. For example we collect user name, father’s name, address and mobile number etc. a form will take input from the user and send to the database.

Forms are the best way of adding interactivity of element in a web page. They are usually used to let the user to send information back to the server but can also be used to simplify navigation on complex web sites. The tags that use to implement forms are as follows.

**<form action=”URL” method = “post” | “get”>…….</form>**

When get is used, the data is included as part of the URL. The post method encodes the data within the body of the message. Post can be used to send large amount of data, and it is more secure than get.

**Example** : HTML code that implements forms

<html>

<head><title>form</title>

</head>

<body>

<h2> HTML Forms</h2>

<form>

Name:<input type="text" maxlength=30 size=15><br>

Password:<input type="password" maxlenght=10 size=15><br>

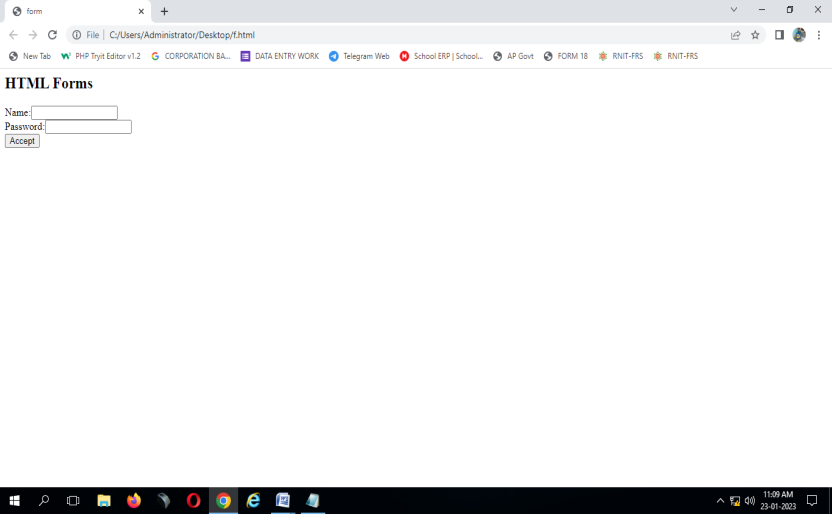
<input type="submit" name="s" value="Accept>

</form>

</body>

</html>

When we run the above code , you will get the following output.



**HTML form elements**

In html a <form> element is used to specify a form. The form element is like a container i.e., it holds various input elements such as textbox, textarea, checkboxes, buttons, dropdown list etc. The HTML <form> element is basically used to create an HTML forms for user input. Following is the syntax to use the HTML <form> element.

**Syntax:- 1. <form action=”URL” method = “post” | “get”>…….</form>**

When get is used, the data is included as part of the URL. The post method encodes the data within the body of the message. Post can be used to send large amount of data, and it is more secure than get.

2.

<form>

Form elements

</form>

**HTML Form Input Types and Attributes**

This is the most commonly used element within HTML forms, It allows you to specify various types of user input fields, depending on the type attribute. An input element can be of type text field, password field, checkbox, radio button, submit button, reset button, file select box, as well as several new input types introduced in HTML5.

**HTML <input> Tag:**

The most used form tag is the <input> tag. This is the most commonly used tag within HTML forms. It give you to specify various types of user information fields such as text, checkboxes radio buttons, Select Submit etc.

Following is a list of all types of <input> element of HTML.

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

**Following is the description about types of <input> element with examples.**

**1. <input type="text">:**

<input> element of type "text" are used to define a single-line input text field.

### Example:

1. **<form>**
2. **<label>**Enter first name**</label><br>**
3. **<input** type="text" name="firstname"**><br>**
4. **<label>**Enter last name**</label><br>**
5. **<input** type="text" name="lastname"**><br>**
6. **<p><strong>**Note:**</strong>**The default maximum cahracter lenght is 20.**</p>**
7. **</form>**

**Output:**

### Input "text" type:

The **"text"**field defines a sinlge line input text field.

Top of Form

Enterfirstname  
  
Enterlastname  


**Note:**The default maximum cahracter lenght is 20.

Bottom of Form

### 2. <input type="password">:

The <input> element of type "password" allow a user to enter the password securely in a webpage. The entered text in password filed converted into "\*" or ".", so that it cannot be read by another user.

### Example:

1. **<form>**
2. **<label>**Enter User name**</label><br>**
3. **<input** type="text" name="firstname"**><br>**
4. **<label>**Enter Password**</label><br>**
5. **<input** type="Password" name="password"**><br>**
6. **<br><input** type="submit" value="submit"**>**
7. **</form>**

**Output:**

### Input "password" type:

The **"password"**field defines a sinlge line input password field to enter the password securely.

Top of Form

EnterUsername  
  
EnterPassword  
  
  


Bottom of Form

### 3. <input type="submit">:

The <input> element of type "submit" defines a submit button to submit the form to the server when the "click" event occurs.

### Example:

1. **<form** action="https://www.javatpoint.com/html-tutorial"**>**
2. **<label>**Enter User name**</label><br>**
3. **<input** type="text" name="firstname"**><br>**
4. **<label>**Enter Password**</label><br>**
5. **<input** type="Password" name="password"**><br>**
6. **<br><input** type="submit" value="submit"**>**
7. **</form>**

**Output:**

### Input "submit" type:

Top of Form

EnterUsername  
  
EnterPassword  
  
  


Bottom of Form

After clicking on submit button, this will submit the form to server and will redirect the page to **action**value.We will learn about "action" attribute in later chapters

### 4. <input type="reset">:

The <input> type "reset" is also defined as a button but when the user performs a click event, it by default reset the all inputted values.

### Example:

1. **<form>**
2. **<label>**User id: **</label>**
3. **<input** type="text" name="user-id" value="user"**>**
4. **<label>**Password: **</label>**
5. **<input** type="password" name="pass" value="pass"**><br><br>**
6. **<input** type="submit" value="login"**>**
7. **<input** type="reset" value="Reset"**>**
8. **</form>**

**Output:**

### Input "reset" type:

Top of Form

Userid:  Password:   
  
 

Bottom of Form

Try to change the input values of user id and password, then when you click on reset, it will reset input fields with default values.

### 5. <input type="radio">:

The <input> type "radio" defines the radio buttons, which allow choosing an option between a set of related options. At a time only one radio button option can be selected at a time.

### Example:

1. **<form>**
2. **<p>**Kindly Select your favorite color**</p>**
3. **<input** type="radio" name="color" value="red"**>** Red **<br>**
4. **<input** type="radio" name="color" value="blue"**>** blue **<br>**
5. **<input** type="radio" name="color" value="green"**>**green **<br>**
6. **<input** type="radio" name="color" value="pink"**>**pink **<br>**
7. **<input** type="submit" value="submit"**>**
8. **</form>**

**Output:**

### Input "radio" type

Top of Form

Kindly Select your favorite color

 Red  
 blue  
green  
pink  


Bottom of Form

### 6. <input type="checkbox">:

The <input> type "checkbox" are displayed as square boxes which can be checked or unchecked to select the choices from the given options.

#### Note: The "radio" buttons are similar to checkboxes, but there is an important difference between both types: radio buttons allow the user to select only one option at a time, whereas checkbox allows a user to select zero to multiple options at a time.

### Example:

1. **<form>**
2. **<label>**Enter your Name:**</label>**
3. **<input** type="text" name="name"**>**
4. **<p>**Kindly Select your favourite sports**</p>**
5. **<input** type="checkbox" name="sport1" value="cricket"**>**Cricket**<br>**
6. **<input** type="checkbox" name="sport2" value="tennis"**>**Tennis**<br>**
7. **<input** type="checkbox" name="sport3" value="football"**>**Football**<br>**
8. **<input** type="checkbox" name="sport4" value="baseball"**>**Baseball**<br>**
9. **<input** type="checkbox" name="sport5" value="badminton"**>**Badminton**<br><br>**
10. **<input** type="submit" value="submit"**>**
11. **</form>**

**Output:**

## Input "checkbox" type

### Registration Form

Top of Form

Enter your Name: 

Kindly Select your favorite sports

Cricket  
Tennis  
Football  
Baseball  
Badminton  
  


Bottom of Form

### 7. <input type="button">:

The <input> type "button" defines a simple push button, which can be programmed to control a functionally on any event such as, click event.

#### Note: It mainly works with JavaScript.

### Example:

1. **<form>**
2. **<input** type="button" value="Clcik me " onclick="alert('you are learning HTML')"**>**
3. **</form>**

**Output:**

## Input "button" type.

Click the button to see the result:

Top of Form

Bottom of Form

#### Note: In the above example we have used the "alert" of JS, which you will learn in our JS tutorial. It is used to show a pop window.

### 8. <input type="file">:

The <input> element with type "file" is used to select one or more files from user device storage. Once you select the file, and after submission, this file can be uploaded to the server with the help of JS code and file API.

### Example:

1. **<form>**
2. **<label>**Select file to upload:**</label>**
3. **<input** type="file" name="newfile"**>**
4. **<input** type="submit" value="submit"**>**
5. **</form>**

**Output:**

## Input "file" type.

We can choose any type of file until we do not specify it! The selected file will appear at next to "choose file" option

Top of Form

Select file to upload:  

Bottom of Form

**HTML input attributes:**

The html value attribute is used to define an initial value for an input element. In the example

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Value** | **Description** |
| Type | “text”  “buttons”  “passwords”  “file”  “checkbox”  “radio”  “reset”  “submit” | Use input text value  Use to input button  Use to text is convert password type  Use to browse file.  Use to select option checkbox.  Use to select in radio type.  Use to input reset data  Use to input submit data. |
| Value | “value” | Specify the default value of element. |
| Name | “value” | Specify unique name for the input element. |
| Size | “number” | Define width of numbered size input box |
| Maxlength | “number” | Define maximum number allow to enter input box. |
| Checked |  | Specify input element to be checked when it is first time loaded. |
| Alt | “text” | Define input name to identify. |
| Align | “left”  “right”  “center” | Define input type align in HTML document. |
| src | “URL” | Define image open to mouse down submit |
| height | “number” | The height attribute specifies the height of an input element. |
| width | “number” | The width attribute specifies the width of an input element. |

**HTML5**

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

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

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

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

New Features

HTML5 introduces a number of new elements and attributes that can help you in building modern websites. Here is a set of some of the most prominent features introduced in HTML5.

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

**HTML graphics**

Graphics are representations used to make web-page or applications visually appealing and also for improving user experience and user interaction. Some examples of graphics are photographs, flowcharts, bar graphs, maps, engineering drawings, constructional blueprints, etc. Usually, the following technologies are used in web graphics with HTML5 Canvas API, SVG,etc.

# HTML SVG

SVG stands for Scalable Vector Graphics. It basically defines vector-based graphics in XML format. SVG graphics do NOT lose any quality if they are zoomed or resized. Every element and every attribute in SVG files can be animated.

**Advantages of SVG:**Advantages of using SVG over other image formats (like JPEG and GIF) are:

* SVG images can be created and edited with any text editor.
* SVG images can be searched, indexed, scripted, and compressed.
* SVG images are scalable.
* SVG images can be printed with high quality at any resolution.

**Example 1:**In this example, We create an SVG line in HTML.

<!DOCTYPE html>

<html>

<body>

<h2>Welcome To GeeksforGeeks</h2>

<svg height="250" width="600">

<line x1="10" y1="10" x2="400" y2="400"

style="stroke:rgb(0,0,255);stroke-width:3" />

</svg>

</body>

</html>

**OUTPUT:-**



**Example 2:** Drawing A SVG Rectangle in HTML

<DOCTYPE html>

<html>

<body>

<!-- html svg tag is used here -->

<svg width="400" height="100">

<rect width="400" height="100"

style="fill: rgb(0, 0, 255);

stroke-width: 10; stroke: rgb(0, 0, 0)" />

</svg>

</body>

</html>

**OUTPUT:-**



**Canvas API:-**

The HTML “canvas” element is used to draw graphics via JavaScript. The “canvas” element is only a container for graphics. One must use JavaScript to actually draw the graphics. Canvas has several methods for drawing paths, boxes, circles, text, and adding images. The canvas would be a rectangular area on an HTML page. By default, a canvas has no border and no content.

**Syntax:**

<canvas>

Content...

</canvas>

It is recommended to have an id attribute (to be referred to in a script), and a width and height attribute to define the size of the canvas. To add a border, use the style attribute. **Supported Properties:**The properties like Colors, Styles, Shadows, Line Styles, Rectangles, Paths, Transformations, Text, Pixel Manipulation, Compositing & Image Drawing, are the global attributes that are supported by all the canvas tags. Please refer to the [HTML Canvas Complete Reference](https://www.geeksforgeeks.org/html-canvas-complete-reference/) article for the details.

**Example 1:**

<!DOCTYPE html>

<html>

<body>

<canvas id="myCanvas"

width="400"

height="200"

style="border:2px solid #000000;">

</canvas>

</body>

</html>

**OUTPUT:-**

:-

**differences between HTML SVG and HTML Canvas:**

* SVG is a language for describing 2D graphics in XML whereas Canvas draws 2D graphics, on the fly with JavaScript.
* If attributes of an SVG object are changed, the browser can automatically re-render the shape whereas Canvas is rendered pixel by pixel. In canvas, once the graphic is drawn, it is forgotten by the browser.
* SVG is resolution independent whereas CANVAS is resolution-dependent.
* SVG supports event handlers whereas CANVAS doesn’t have support for event handlers.

**HTML media – video, audio, plug INS, you tube.**

**HTML MEDIA(OR) MULTIMEDIA:-**

Text, images, graphics, audio, video, and animation are all examples of media used to communicate and share information. All of these forms of communication are referred to as multimedia.

In simple words, the term "multimedia" refers to almost anything that can be heard or seen for e.g., sound, music, images, records, videos, films, animations, etc. Web pages can include multimedia elements of various formats and types.

The simplest approach to add video or audio to our website is to use the HTML <embed> tag. HTML provides different multimedia tags that allow you to add multimedia files to your website, audio, video, embed, and object is some of the tags that are used.

The audio tag is used to display audio files on a Web page, while the video tag is used to display video files. The embed and object tags display multimedia content on a Web page and also allow you to embed files from other websites.

**HTML Multimedia Formats:**

Media files contain multimedia elements such as audio and video. The file extension is the most common way to determine the type of file. Multimedia files are available in a variety of formats like wav, mp3, mp4, mpg,.wmv, and.avi.

**Common Video Formats**

|  |  |
| --- | --- |
| **Formate** | **Description** |
| **MP4** | MP4 is a digital multimedia container format that is most commonly used to store video and audio files. |
| **WebM** | WebM is a file format for audiovisual media. Its primary purpose is to provide a royalty-free alternative to be used in HTML5 video and HTML5 audio elements. |
| **Ogg** | Ogg is an open-source multimedia file format. It is optimized for streaming content and it can contain music, video, text, and metadata Because it is Copyright free, Ogg is freely available for use in software or media projects by anyone. |

**Common Audio Formats**

|  |  |
| --- | --- |
| **Formate** | **Description** |
| **MP3** | MP3 is a digital audio encoding format that employs a type of lossy data compression. |
| **WAV** | WAV or Waveform Audio Filo Format, was created by Microsoft and IBM as an audio file standard for storing digital audio on computers multimedia |
| **Ogg** | Ogg is an open-source file format. It is optimized for streaming content and it can contain music video, text, and metadata. Because it is Copyright-free, is freely available for use in software or media projects by anyone |

**VIDEO**

A video file is a collection of images displayed in a sequence to represent moving scenes. Different video codecs, such as DivX and QuickTime, are often used to encode and decode video files. A video is displayed on a web page using the HTML <video> element

There are three different formats that are commonly supported by web browsers – mp4, Ogg, and WebM. The table below lists the formats supported by different browsers:

| Browser | MP4 | WebM | OGG |
| --- | --- | --- | --- |
| Google Chrome | Yes | Yes | Yes |
| Internet Explorer | Yes | No | No |
| Firefox | Yes | Yes | Yes |
| Opera | Yes | Yes | Yes |
| Safari | Yes | Yes | No |

**Syntax:**

<video src="" controls> </video>

**Attributes that can be used with the “video” tag are listed below :**

1. [**Autoplay**](https://www.geeksforgeeks.org/html-video-autoplay-attribute/)**:**It tells the browser to immediately start downloading the video and play it as soon as it can.
2. [**Preload**](https://www.geeksforgeeks.org/html-video-preload-attribute/)**:**It intends to provide a hint to the browser about what the author thinks will lead to the best user experience.
3. [**Loop**](https://www.geeksforgeeks.org/html-video-loop-attribute/)**:**It tells the browser to automatically loop the video.
4. [**height**](https://www.geeksforgeeks.org/html-video-height-attribute-2/)**:**It sets the height of the video in CSS pixels.
5. [**width**](https://www.geeksforgeeks.org/html-video-width-attribute/)**:**It sets the width of the video in CSS pixels.
6. [**Controls:**](https://www.geeksforgeeks.org/html-video-controls-attribute/)It shows the default video controls like play, pause, volume, etc.
7. [**Muted**](https://www.geeksforgeeks.org/html-video-muted-attribute/)**:**It mutes the audio from the video.
8. [**Poster**](https://www.geeksforgeeks.org/html-video-poster-attribute/)**:**It loads an image to preview before the loading of the video.
9. [**src**](https://www.geeksforgeeks.org/html-video-src-attribute/)**:**It is used to specify the URL of the video file.

**EXAMPLE**:-

<!DOCTYPE html>

<html>

<body>

<center>

<h1 style="color:green;">SRNB DEGREE COLLEGE</h1>

<h3>HTML video tag</h3>

<p>Adding video on the webpage</p>

<video width="450"

height="250"

controls

preload="auto">

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

</video>

</center>

</body>

</html>

**AUDIO**

Audio files are used to store audio data on a variety of devices, including computers, MP3 players, and cell phones. You must convert audio data into a digital format before storing it.

Encoding the raw audio data is the process of transforming audio data into a digital file. It includes extracting audio data samples and storing them in a compressed way to reduce file size. The HTML5 <audio> element provides a standard way to embed audio in web pages. The audio element, on the other hand, is very new, but still, it works in most modern web browsers. Syntax: The basic syntax of the <audio> element <audio>...</audio>.

**Syntax:**

<audio>

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

</audio>

**Attributes:** The various attributes that can be used with the “audio” tag are listed below:

* [**Controls**](https://www.geeksforgeeks.org/html-controls-attribute/)**:** Designates what controls to display with the audio player.
* [**Autoplay**](https://www.geeksforgeeks.org/html-autoplay-attribute/)**:** Designates that the audio file will play immediately after it loads controls.
* [**Loop**](https://www.geeksforgeeks.org/html-loop-attribute/)**:** Designates that the audio file should continuously repeat.
* [**src**](https://www.geeksforgeeks.org/html-src-attribute/)**:** Designates the URL of the audio file.
* [**muted**](https://www.geeksforgeeks.org/html-muted-attribute/)**:** Designates that the audio file should be muted.

**Supported Formats:** Three formats mp3, ogg, wav are supported by HTML5. The support for each format by different browsers is given below :

|  |  |  |  |
| --- | --- | --- | --- |
| **Browser** | **MP3** | **WAV** | **OGG** |
| **Google Chrome** | **Yes** | **Yes** | **Yes** |
| **Internet Explorer** | **Yes** | **No** | **No** |
| **Firefox** | **Yes** | **Yes** | **Yes** |
| **Opera** | **Yes** | **Yes** | **Yes** |
| **Safari** | **Yes** | **Yes** | **No** |

The below Examples explain the audio Tag:

**Example 1 (Adding audio to Webpage):**The controls attribute is used to add audio controls such as play, pause, and volume. The “source” element is used to specify the audio files which the browser may use. The first recognized format is used by the browser.

<!DOCTYPE html>

<html>

<body>

<p>Audio Sample</p>

<!-- audio tag starts here -->

<audio controls>

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

<source src="test.ogg" type="audio/ogg">

</audio>

<!-- audio tag ends here -->

</body>

</html>

**PlugINS**

Html plugins are computer programs that help enhance a web browser's standard functionality. Java applets are an example of well-known HTML plug-ins. The <object> or <embed> tags can be used to embed plug-ins into web pages. Plug-ins can be for a variety of purposes, like displaying maps, scanning for viruses, and verifying someone's bank account information, etc.

**<object> Element:**

The HTML <object> tag is used to insert multimedia files into a webpage. Multimedia assets, such as video, music, image, PDF, Java Applets, etc. can be included in the <object> tag.

The HTML <param> element can also be used in addition with the <object> tag to specify parameters to a plugin that was included using the <object> tag.

If we insert anything between the <object> and </object> tags, it will only be displayed if the browser doesn't support the object element.

**Syntax**:

<object data=” ”> </object>

**Example**:-

<!DOCTYPE html>

<html>

<style>

body{

display:flex;

justify-content:center;

align-items:center;

height:100vh;

}

</style>

<body>

<object width="500" height="300" data="https://www.youtube.com/embed/A\_YbrEKA4wI"></object>

</body>

</html>

**<embed> Element:**

The HTML <embed> tag is used to include a third-party application, multimedia, plugin, or other external documents in an HTML file. However in HTML5, the newly added <audio> and <video> elements are mostly used to embed multimedia in HTML documents.

**Syntax**:-

<embed src=” ”> </embed>

**Example:-**

<!DOCTYPE html>

<html>

<style>

body{

display:flex;

justify-content:center;

align-items:center;

height:100vh;

}

embed{

width:500px;}

</style>

<body>

<embed src="https://cdn.pixabay.com/photo/2021/05/29/07/06/shiba-6292660\_960\_720.jpg">

</body>

</html>

**Youtube**

Adding a video to a webpage was a real challenge since one had to convert the videos to different formats to make them play in all browsers. Converting videos to different formats can be difficult and time-consuming. Now, adding a video to a webpage has become as easy as copying and pasting and a very apt solution to add videos to a website is using Youtube. Youtube helps to host a video for a user so that they can be further embedded on webpages.

**Steps to add a Youtube video on a Webpage :**

1. Upload the video that you want to embed on your webpage on YouTube.
2. Copy the video id of the video.
3. Use iframe, object or ’embed’ element in your web page for video definition.
4. Use the src attribute to point to the URL of the video.
5. Dimensions of the player can be adjusted using the width and height attributes.

## Adding Youtube video : Using iFrame tag:

<!DOCTYPE html>

<html>

<body>

<iframe height="480" width="500"

src="https://www.youtube.com/embed/il\_t1WVLNxk">

</iframe>

</body>

</html>

**EnablingYouTubeautoplayfeature:**  
Youtube’s autoplay feature can be used to automatically play a video when a user visits that page.

There are two types of parameters that can be used :

1. Value 1 : The video starts playing automatically when the player loads.
2. Value 0 (default case) : The video does not play automatically when the player loads.

**Creating a YouTube playlist :(loop)**

A playlist of youtube videos can be created using comma character which separates the list of videos to play.

The loop parameter is used to loop the number of playbacks on the videos :

1. Value 1 : The video will keep on looping again and again.
2. Value 0 (default case) : The video plays only once.

**Enabling / Disabling Youtube controls :**

The Youtube Player offers controls like play, pause, volume etc that can be disabled or enabled using the controls parameter.  
There are two parameters available that can be used :

1. Value 1 (default case) : Player controls are displayed.
2. Value 0 : Player controls are not displayed.

**HTML API’S: Geo location**

**Geo-location** in HTML5 is used to share the location with some websites and be aware of the exact location. It is mainly used for local businesses, restaurants, or showing locations on the map. It uses JavaScript to give latitude and longitude to the backend server. Most of the browsers support Geolocation API. Geo-location API uses a global navigator object which can be created as follows:

**Syntax:**

var loc = navigator.geolocation

**Location Properties:** The following table determines properties used in getCurrentPosition() and their returning values.

* [**coords.latitude:**](https://www.geeksforgeeks.org/html-dom-geolocation-coords-latitude-property/)Always returns latitude as a decimal number.
* [**coords.accuracy:**](https://www.geeksforgeeks.org/html-dom-geolocation-coords-accuracy-property/)Always returns the accuracy of position.
* [**coords.longitude**](https://www.geeksforgeeks.org/html-dom-geolocation-coords-longitude-property/)**:**Always returns longitude as a decimal number.
* [**coords.altitude:**](https://www.geeksforgeeks.org/html-dom-geolocation-coords-altitude-property/)Returns the altitude in meters above sea level if available.
* [**coords.altitudeAccuracy:**](https://www.geeksforgeeks.org/html-dom-geolocation-coords-altitudeaccuracy-property/)Returns altitude accuracy of position if available.
* [**coords.heading:**](https://www.geeksforgeeks.org/html-dom-geolocation-coords-heading-property/)Returns heading in degree clockwise from North if available
* [**coords.speed:**](https://www.geeksforgeeks.org/html-dom-geolocation-coords-speed-property/)Returns speed in mps, if available.
* **timestamp:**Returns date or time of response if available

**Geolocation Methods:** The Geolocation has following methods which make it interesting and easier to work.

* **getCurrentPosition():**It fetches the current location of the user.
* [**watchPosition():**](https://www.geeksforgeeks.org/html-geolocation-watchposition-method/)It fetches periodic updates of the user’s current location.
* **clearWatch():**It cancels a watchPosition call currently in execution.

**Example:** This example explains returning the user’s current location using the getCurrentPosition() method.

var loc = navigator.geolocation;

function getLoc() {

loc.getCurrentPosition(showLoc, errHand);

}

The above function can also be written without creating a navigator object as shown below:

function getlocation(){

navigator.geolocation.getCurrentPosition(showLoc, errHand);

}

**Example:**In this example, we simply display the current location with the help of latitude and longitude via HTML Geolocation.

<!DOCTYPE html>

<html>

<body>

<p>Displaying location using Latitude and Longitude</p>

<button class="geeks" onclick="getlocation()">

Click Me

</button>

<p id="demo1"></p>

<script>

var variable1 = document.getElementById("demo1");

function getlocation() {

navigator.geolocation.getCurrentPosition(showLoc);

}

function showLoc(pos) {

variable1.innerHTML =

"Latitude: " +

pos.coords.latitude +

"<br>Longitude: " +

pos.coords.longitude;

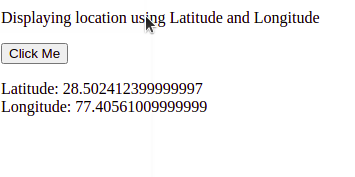
}

</script>

</body>

</html>

**Output:-**



**Drag/Drop**

Drag and Drop is a very interactive and user-friendly concept that makes it easier to move an object to a different location by grabbing it. This allows the user to click and hold the mouse button over an element, drag it to another location, and release the mouse button to drop the element there. In HTML 5 Drag and Drop are much easier to code and any element in it is draggable.

**Drag and Drop Events:** There are various Drag and Drop events, some of them are listed below:

* [**ondrag:**](https://www.geeksforgeeks.org/html-ondrag-event-attribute/)It is used to use when the element or text selection is being dragged in HTML.
* [**ondragstart**](https://www.geeksforgeeks.org/html-ondragstart-event-attribute/): It is used to call a function, drag(event), that specifies what data to be dragged.
* [**ondragenter**](https://www.geeksforgeeks.org/html-ondragenter-event-attribute/): It is used to determine whether or not the drop target is to accept the drop. If the drop is to be accepted, then this event has to be canceled.
* [**ondragleave**](https://www.geeksforgeeks.org/html-ondragleave-event-attribute/): It occurs when the mouse leaves an element before a valid drop target while the drag is occurring.
* [**ondragover**](https://www.geeksforgeeks.org/html-ondragover-event-attribute/): It specifies where the dragged data can be dropped.
* [**ondrop**](https://www.geeksforgeeks.org/html-ondrop-event-attribute/): It specifies where the drop has occurred at the end of the drag operation.
* [**ondragend**](https://www.geeksforgeeks.org/html-ondragend-event-attribute/): It occurs when the user has finished dragging an element.

**Procedure for Drag and Drop:**

* **Step 1:** Make an object draggable
  + First set the draggable attribute to true for that element to be draggable <img draggable = “true”>
  + Then, specify what should happen when the element is dragged. The *ondragstart* attribute calls a function, drag(event), that specifies what data to be dragged. The dataTransfer.setData() method sets the data type and the value of the dragged data
  + The event listener *ondragstart*will set the allowed effects (copy, move, link, or some combination).
* **Step 2:** Dropping the Object
  + The *ondragover* event specifies where the dragged data can be dropped. By default, data/elements cannot be dropped in other elements. To allow a drop, it must prevent the default handling of the element. This is done by calling the event.preventDefault() method
  + Finally, the drop event, which allows the actual drop to be performed

**Example 1:**This example shows the drag & drop of an image in HTML.

<!DOCTYPE HTML>

<html>

<head>

<style>

#getData {

width: 250px;

height: 200px;

padding: 10px;

border: 1px solid #4f4d4d;

}

</style>

<script>

function allowDrop(even) {

even.preventDefault();

}

function drag(even) {

even.dataTransfer.setData("text", even.target.id);

}

function drop(even) {

even.preventDefault();

var fetchData = even.dataTransfer.getData("text");

even.target.appendChild(document.getElementById(fetchData));

}

</script>

</head>

<body>

<h3>Drag the GeekforGeeks image into the rectangle:</h3>

<div id="getData"

ondrop="drop(event)"

ondragover="allowDrop(event)">

</div>

<br>

<img id="dragData"

src="gfg.png"

draggable="true"

ondragstart="drag(event)"

width="250"

height="200">

</body>

</html>

**Local Storage**

HTML5 introduces two mechanisms, similar to HTTP session cookies, for storing structured data on the client side and to overcome following drawbacks.

* Cookies are included with every HTTP request, thereby slowing down your web application by transmitting the same data.
* Cookies are included with every HTTP request, thereby sending data unencrypted over the internet.
* Cookies are limited to about 4 KB of data. Not enough to store required data.

The two storages are **session storage** and **local storage** and they would be used to handle different situations.

The latest versions of pretty much every browser supports HTML5 Storage including Internet Explorer.

## Session Storage

The *Session Storage* is designed for scenarios where the user is carrying out a single transaction, but could be carrying out multiple transactions in different windows at the same time.

### Example

*For example, if a user buying plane tickets in two different windows, using the same site. If the site used cookies to keep track of which ticket the user was buying, then as the user clicked from page to page in both windows, the ticket currently being purchased would "leak" from one window to the other, potentially causing the user to buy two tickets for the same flight without really noticing.*

HTML5 introduces the *sessionStorage* attribute which would be used by the sites to add data to the session storage, and it will be accessible to any page from the same site opened in that window, i.e., **session** and as soon as you close the window, the session would be lost.

Following is the code which would set a session variable and access that variable −

<!DOCTYPE HTML>

<html>

<body>

<script type = "text/javascript">

if( sessionStorage.hits ) {

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

} else {

sessionStorage.hits = 1;

}

document.write("Total Hits :" + sessionStorage.hits );

</script>

<p>Refresh the page to increase number of hits.</p>

<p>Close the window and open it again and check the result.</p>

</body>

</html>

## Local Storage

The *Local Storage* is designed for storage that spans multiple windows, and lasts beyond the current session. In particular, Web applications may wish to store megabytes of user data, such as entire user-authored documents or a user's mailbox, on the client side for performance reasons.

Again, cookies do not handle this case well, because they are transmitted with every request.

### Example

HTML5 introduces the *localStorage* attribute which would be used to access a page's local storage area without no time limit and this local storage will be available whenever you would use that page.

Following is the code which would set a local storage variable and access that variable every time this page is accessed, even next time, when you open the window −

<!DOCTYPE HTML>

<html>

<body>

<script type = "text/javascript">

if( localStorage.hits ) {

localStorage.hits = Number(localStorage.hits) +1;

} else {

localStorage.hits = 1;

}

document.write("Total Hits :" + localStorage.hits );

</script>

<p>Refresh the page to increase number of hits.</p>

<p>Close the window and open it again and check the result.</p>

</body>

</html>

## Delete Web Storage

Storing sensitive data on local machine could be dangerous and could leave a security hole.

The *Session Storage Data* would be deleted by the browsers immediately after the session gets terminated.

To clear a local storage setting you would need to call **localStorage.remove('key')**; where 'key' is the key of the value you want to remove. If you want to clear all settings, you need to call **localStorage.clear()** method.

**HTML SSE**

A server-sent event in HTML5 is a new technique for web pages to communicate with the webserver. Social media status updates, stock chart patterns, news feeds, and other automatic data push systems are a few examples of SSE.

Such things are possible due to HTML5 server-sent events. It enables a web page to maintain an open connection with a web server so that the webserver may automatically provide a fresh response at any moment, eliminating the need to reconnect and run the same server script from start.

The HTML5 server-sent event allows a browser to receive automatic updates and information from a server using HTTP connections.

**Server-Sent Events in HTML:**

Whenever we perform some event and send it to the server, such an event that flows from web browser to web-server is called a client-side event. for example, submitting the form data to the server.

On the other hand, Server-sent events occur when the server sends updates or information to the browser. The server-sent events are unidirectional. It's also known as one-way messaging. For example, a simple click on a button requests a new page from the server. **SSE Connection States:**

The following values are assigned to each SSE connection's state:

* Connecting: The connection is yet to be established the process of being established.
* Open: The connection is up and running, and it's ready to send out events.
* Closed: The connection is not open and is not being restored at this time.

**Web Application for SSE:**

In order to use Server-Sent Events in a web application, the page must have an <eventsource> element. The <eventsource> element's sre property should link to a URL that provides a persistent HTTP connection that transmits a data stream containing the events. In the example below we have created an SSE that displays the current time on the screen.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**CSS:** CSS home, introduction, syntax, colours, back ground, borders, margins, padding, height/width, text, fonts, icons, tables, lists CSS forms, CSS counters, CSS responsive.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**CSS Home & INTRODUCTION**

**CSS** is used to control the style of a web document in a simple and easy way.

**CSS** is the acronym for **"Cascading Style Sheet"**. This tutorial covers both the versions CSS1,CSS2 and CSS3, and gives a complete understanding of CSS, starting from its basics to advanced concepts.

## Why to Learn CSS?

**Cascading Style Sheets**, fondly referred to as **CSS**, is a simple design language intended to simplify the process of making web pages presentable.

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

* **Create Stunning Web site** - CSS handles the look and feel part of a web page. Using CSS, you can control the color of the text, the style of fonts, the spacing between paragraphs, how columns are sized and laid out, what background images or colors are used, layout designs,variations in display for different devices and screen sizes as well as a variety of other effects.
* **Become a web designer** - If you want to start a carrer as a professional web designer, HTML and CSS designing is a must skill.
* **Control web** - CSS is easy to learn and understand but it provides powerful control over the presentation of an HTML document. Most commonly, CSS is combined with the markup languages HTML or XHTML.
* **Learn other languages** - Once you understands the basic of HTML and CSS then other related technologies like javascript, php, or angular are become easier to understand.

## Hello World using CSS.

Just to give you a little excitement about CSS, I'm going to give you a small conventional CSS Hello World program, You can try it using Demo link.

<!DOCTYPE html>

<html>

<head>

<title>This is document title</title>

<style>

h1 {

color: #36CFFF;

}

</style>

</head>

<body>

<h1>Hello World!</h1>

</body>

</html>

## Applications of CSS

As mentioned before, CSS is one of the most widely used style language over the web. I'm going to list few of them here:

* **CSS saves time** - You can write CSS once and then reuse same sheet in multiple HTML pages. You can define a style for each HTML element and apply it to as many Web pages as you want.
* **Pages load faster** - If you are using CSS, you do not need to write HTML tag attributes every time. Just write one CSS rule of a tag and apply it to all the occurrences of that tag. So less code means faster download times.
* **Easy maintenance** - To make a global change, simply change the style, and all elements in all the web pages will be updated automatically.
* **Superior styles to HTML** - CSS has a much wider array of attributes than HTML, so you can give a far better look to your HTML page in comparison to HTML attributes.
* **Multiple Device Compatibility** - Style sheets allow content to be optimized for more than one type of device. By using the same HTML document, different versions of a website can be presented for handheld devices such as PDAs and cell phones or for printing.
* **Global web standards**- Now HTML attributes are being deprecated and it is being recommended to use CSS. So its a good idea to start using CSS in all the HTML pages to make them compatible to future browsers.

# What is CSS?

**C**ascading **S**tyle **S**heets, fondly referred to as CSS, is a simple design language intended to simplify the process of making web pages presentable.

CSS handles the look and feel part of a web page. Using CSS, you can control the color of the text, the style of fonts, the spacing between paragraphs, how columns are sized and laid out, what background images or colors are used, layout designs,variations in display for different devices and screen sizes as well as a variety of other effects.

CSS is easy to learn and understand but it provides powerful control over the presentation of an HTML document. Most commonly, CSS is combined with the markup languages HTML or XHTML.

Advantages of CSS

* **CSS saves time** − You can write CSS once and then reuse same sheet in multiple HTML pages. You can define a style for each HTML element and apply it to as many Web pages as you want.
* **Pages load faster** − If you are using CSS, you do not need to write HTML tag attributes every time. Just write one CSS rule of a tag and apply it to all the occurrences of that tag. So less code means faster download times.
* **Easy maintenance** − To make a global change, simply change the style, and all elements in all the web pages will be updated automatically.
* **Superior styles to HTML** − CSS has a much wider array of attributes than HTML, so you can give a far better look to your HTML page in comparison to HTML attributes.
* **Multiple Device Compatibility** − Style sheets allow content to be optimized for more than one type of device. By using the same HTML document, different versions of a website can be presented for handheld devices such as PDAs and cell phones or for printing.
* **Global web standards** − Now HTML attributes are being deprecated and it is being recommended to use CSS. So its a good idea to start using CSS in all the HTML pages to make them compatible to future browsers.

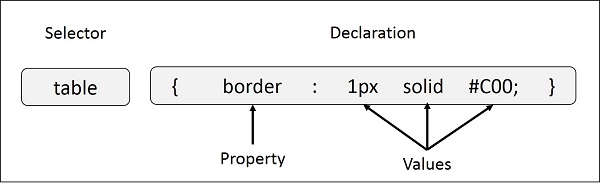
# CSS - Syntax

A CSS comprises of style rules that are interpreted by the browser and then applied to the corresponding elements in your document. A style rule is made of three parts −

* **Selector** − A selector is an HTML tag at which a style will be applied. This could be any tag like <h1> or <table> etc.
* **Property** − A property is a type of attribute of HTML tag. Put simply, all the HTML attributes are converted into CSS properties. They could be *color*, *border* etc.
* **Value** − Values are assigned to properties. For example, *color* property can have value either *red* or *#F1F1F1* etc.

You can put CSS Style Rule Syntax as follows −

selector { property: value }



**Example** − You can define a table border as follows −

table{ border :1px solid #C00; }

Here table is a selector and border is a property and given value *1px solid #C00* is the value of that property.

# CSS - Colors

CSS uses color values to specify a color. Typically, these are used to set a color either for the foreground of an element (i.e., its text) or else for the background of the element. They can also be used to affect the color of borders and other decorative effects.

You can specify your color values in various formats. Following table lists all the possible formats −

|  |  |  |
| --- | --- | --- |
| **Format** | **Syntax** | **Example** |
| Hex Code | #RRGGBB | p{color:#FF0000;} |
| Short Hex Code | #RGB | p{color:#6A7;} |
| RGB % | rgb(rrr%,ggg%,bbb%) | p{color:rgb(50%,50%,50%);} |
| RGB Absolute | rgb(rrr,ggg,bbb) | p{color:rgb(0,0,255);} |
| keyword | aqua, black, etc. | p{color:teal;} |

# CSS - Backgrounds

To set backgrounds of various HTML elements.You can set the following background properties of an element −

* The **background-color** property is used to set the background color of an element.
* The **background-image** property is used to set the background image of an element.
* The **background-repeat** property is used to control the repetition of an image in the background.
* The **background-position** property is used to control the position of an image in the background.
* The **background-attachment** property is used to control the scrolling of an image in the background.
* The **background** property is used as a shorthand to specify a number of other background properties.

## Set the Background Color

Following is the example which demonstrates how to set the background color for an element.

<html>

<head>

</head>

<body>

<p **style = "background-color:yellow;"**>

This text has a yellow background color.

</p>

</body>

</html>

# CSS - Borders

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

* The **border-color** specifies the color of a border.
* The **border-style** specifies whether a border should be solid, dashed line, double line, or one of the other possible values.
* The **border-width** specifies the width of a border.

Now, we will see how to use these properties with examples.

The border-color Property

The border-color property allows you to change the color of the border surrounding an element. You can individually change the color of the bottom, left, top and right sides of an element's border using the properties −

* **border-bottom-color** changes the color of bottom border.
* **border-top-color** changes the color of top border.
* **border-left-color** changes the color of left border.
* **border-right-color** changes the color of right border.

# CSS - Margins

The *margin* property defines the space around an HTML element. It is possible to use negative values to overlap content.

The values of the margin property are not inherited by the child elements. Remember that the adjacent vertical margins (top and bottom margins) will collapse into each other so that the distance between the blocks is not the sum of the margins, but only the greater of the two margins or the same size as one margin if both are equal.

We have the following properties to set an element margin.

* The **margin** specifies a shorthand property for setting the margin properties in one declaration.
* The **margin-bottom** specifies the bottom margin of an element.
* The **margin-top** specifies the top margin of an element.
* The **margin-left** specifies the left margin of an element.
* The **margin-right** specifies the right margin of an element.

# CSS - Paddings

The *padding* property allows you to specify how much space should appear between the content of an element and its border −

The value of this attribute should be either a length, a percentage, or the word *inherit*. If the value is *inherit*, it will have the same padding as its parent element. If a percentage is used, the percentage is of the containing box.

The following CSS properties can be used to control lists. You can also set different values for the padding on each side of the box using the following properties −

* The **padding-bottom** specifies the bottom padding of an element.
* The **padding-top** specifies the top padding of an element.
* The **padding-left** specifies the left padding of an element.
* The **padding-right** specifies the right padding of an element.
* The **padding** serves as shorthand for the preceding properties.

**CSS - height/width**

The CSS height and width properties are used to set the height and width of an element. The height and width properties do not include padding, borders, or margins. It sets the height and width of the area inside the padding, border, and margin of the element. The CSS Height and Width properties may have the following values:

|  |  |
| --- | --- |
| **Property** | **Description** |
| Auto | This is the default value. The browser calculates the height and width |
| Length | Defines the height/width in px, cm etc. |
| % | Defines the height/width in percent of the containing block. |
| Initial | Sets the height/width to its default value. |
| inherit | The height/width will be inherited from its parent value. |

# CSS - Text

You can set following text properties of an element −

* The **color** property is used to set the color of a text.
* The **direction** property is used to set the text direction.
* The **letter-spacing** property is used to add or subtract space between the letters that make up a word.
* The **word-spacing** property is used to add or subtract space between the words of a sentence.
* The **text-indent** property is used to indent the text of a paragraph.
* The **text-align** property is used to align the text of a document.
* The **text-decoration** property is used to underline, overline, and strikethrough text.
* The **text-transform** property is used to capitalize text or convert text to uppercase or lowercase letters.
* The **white-space** property is used to control the flow and formatting of text.
* The **text-shadow** property is used to set the text shadow around a text.

**Set the Text Color**

The following example demonstrates how to set the text color. Possible value could be any color name in any valid format.

<html>

<head>

</head>

<body>

<p **style = "color:red;"**>

This text will be written in red.

</p>

</body>

</html>

**Set the Text Alignment**

The following example demonstrates how to align a text. Possible values are *left, right, center, justify*.

<html>

<head>

</head>

<body>

<p **style = "text-align:right;"**>

This will be right aligned.

</p>

<p **style = "text-align:center;"**>

This will be center aligned.

</p>

<p **style = "text-align:left;"**>

This will be left aligned.

</p>

</body>

</html>

# CSS - Fonts

You can set following font properties of an element −

* The **font-family** property is used to change the face of a font.
* The **font-style** property is used to make a font italic or oblique.
* The **font-variant** property is used to create a small-caps effect.
* The **font-weight** property is used to increase or decrease how bold or light a font appears.
* The **font-size** property is used to increase or decrease the size of a font.
* The **font** property is used as shorthand to specify a number of other font properties.

Set the Font Family

Following is the example, which demonstrates how to set the font family of an element. Possible value could be any font family name.

<html>

<head>

</head>

<body>

<p **style = "font-family:georgia,garamond,serif;"**>

This text is rendered in either georgia, garamond, or the

default serif font depending on which font you have at your system.

</p>

</body>

</html>

Set the Font Style

Following is the example, which demonstrates how to set the font style of an element. Possible values are *normal, italic and oblique*.

<html>

<head>

</head>

<body>

<p **style = "font-style:italic;"**>

This text will be rendered in italic style

</p>

</body>

</html>

**CSS-Icons**

Icons can easily be added to your HTML page, by using an icon library.

The simplest way to add an icon to your HTML page, is with an icon library, such as Font Awesome.

All the icons in the icon libraries below, are scalable vectors that can be customized with CSS (size, color, shadow, etc.)

1.Font Awesome Icons

2.Bootstrap Icons

3.Google Icons

**1.Font Awesome Icons:** To use the Font Awesome icons, go to fontawesome.com, sign in, and get a code to add in the <head> section of your HTML page:

<script src="https://kit.fontawesome.com/yourcode.js" crossorigin="anonymous"></script>

**2. Bootstrap Icons:** To use the Bootstrap glyphicons, add the following line inside the <head> section of your HTML page:

<link rel="stylesheet"

href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.>

**3.Google Icons:** To use the Google icons, add the following line inside the <head> section of your HTML page:

<link rel="stylesheet" href="https://fonts.googleapis.com/icon?family=Material+Icons">

# CSS - Tables

A table in CSS is used to apply the various styling properties to the HTML Table elements to arrange the data in rows and columns, or possibly in a more complex structure in a properly organized manner. Tables are widely used in communication, research, and data analysis. We can set following properties of a table-The **border-collapse** specifies whether the browser should control the appearance of the adjacent borders that touch each other or whether each cell should maintain its style.

* The **border-spacing** specifies the width that should appear between table cells.
* The **caption-side** captions are presented in the <caption> element. By default, these are rendered above the table in the document. You use the *caption-side* property to control the placement of the table caption.
* The **empty-cells** specifies whether the border should be shown if a cell is empty.
* The **table-layout** allows browsers to speed up layout of a table by using the first width properties it comes across for the rest of a column rather than having to load the whole table before rendering it.

Now, we will see how to use these properties with examples.

The border-collapse Property

This property can have two values *collapse* and *separate*. The following example uses both the values −

<html>

<head>

<style type = "text/css">

table.one {border-collapse:collapse;}

table.two {border-collapse:separate;}

td.a {

border-style:dotted;

border-width:3px;

border-color:#000000;

padding: 10px;

}

td.b {

border-style:solid;

border-width:3px;

border-color:#333333;

padding:10px;

}

</style>

</head>

<body>

<table class = "one">

<caption>Collapse Border Example</caption>

<tr><td class = "a"> Cell A Collapse Example</td></tr>

<tr><td class = "b"> Cell B Collapse Example</td></tr>

</table>

<br />

<table class = "two">

<caption>Separate Border Example</caption>

<tr><td class = "a"> Cell A Separate Example</td></tr>

<tr><td class = "b"> Cell B Separate Example</td></tr>

</table>

</body>

</html>

It will produce the following result −

|  |
| --- |
| Collapse Border Example |
| Cell A Collapse Example |
| Cell B Collapse Example |

|  |
| --- |
| Separate Border Example |
| Cell A Separate Example |
| Cell B Separate Example |

**CSS – Lists**

Lists are very helpful in conveying a set of either numbered or bullet points. This chapter teaches you how to control list type, position, style, etc., using CSS.

We have the following five CSS properties, which can be used to control lists −

* The **list-style-type** allows you to control the shape or appearance of the marker.
* The **list-style-position** specifies whether a long point that wraps to a second line should align with the first line or start underneath the start of the marker.
* The **list-style-image** specifies an image for the marker rather than a bullet point or number.
* The **list-style** serves as shorthand for the preceding properties.
* The **marker-offset** specifies the distance between a marker and the text in the list.

<html>

<head>

</head>

<body>

<ul style = "list-style-type:circle;">

<li>Maths</li>

<li>Social Science</li>

<li>Physics</li>

</ul>

<ul style = "list-style-type:square;">

<li>Maths</li>

<li>Social Science</li>

<li>Physics</li>

</ul>

<ol style = "list-style-type:decimal;">

<li>Maths</li>

<li>Social Science</li>

<li>Physics</li>

</ol>

<ol style = "list-style-type:lower-alpha;">

<li>Maths</li>

<li>Social Science</li>

<li>Physics</li>

</ol>

<ol style = "list-style-type:lower-roman;">

<li>Maths</li>

<li>Social Science</li>

<li>Physics</li>

</ol>

</body>

</html>

**It will produce the following result** −

* Maths
* Social Science
* Physics
* Maths
* Social Science
* Physics

1. Maths
2. Social Science
3. Physics
4. Maths
5. Social Science
6. Physics
7. Maths
8. Social Science
9. Physic

**CSS Forms**

* Using HTML, we can create a simple form but that will not look attractive and beautiful.
* Using CSS, we can apply styles on the HTML forms to make it colorful and attractive.
* The Inline Forms are those forms in which the fields are in the same line and are at the same level.
* The Stacked Form is a form where each field starts on a new line

**Example**:

<!DOCTYPE html>

<html>

<head>

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

<head>

<style>

\*{

margin-top:0;

box-sizing: border-box;

}

input[type=text], input[type=email], input[type=date], select

{

width:100%;

padding: 12px 15px;

margin: 10px 3px;

border: none;

border-bottom: 1px solid tomato;

}

.headingstyle{

text-align:center

}

Input[type=submit]{

padding 12px 15px;

margin: 10px 3px;

font-size:1.5em;

}

Input[type=submit]:hover{

box-shadow: 0 14px 28px rgba(0,0,0,0.25), 0 10px 10px rgba(0,0,0,0.22);

}

input [type=text].focus, input[type=email]:focus, input[type=date]:focus select :focus {

background: lightblue;

}

div

{

background-color: #90acd3;

font-family: Helvetica;

padding: 10px;

border: 1px solid black;

box-shadow: 0 14px 28px rgba(0,0,0,0.25), 0 10px 10px rgba(0,0,0,0.22);

}

input[type=submit]{

color:white;

cursor:pointer;

max-width: 140px;

background:black;

margin:0 0 5px;

padding: 10px;

font-size:15px;

}

<style>

<body>

<div>

<form action="action\_submit.php">

<fieldset>

<legend>KSR Form</legend>

<h2 class="headingstyle">Contact Form</h2>

<label for="testnameval">Name</label>

<input type="text" id="testnameval" name="name"><br><br>

<label for="mailid">Email</label>

<input type="email" id="mailid" name="mailid">

<br><br>

<label for="mailid">Joining Date</label>

<input type="date" name="jday"><br><br>

<label for="sub">Tutorial</label>

<select id="sub" name="subject">

<option value="css">CSS</option>

<option value="java">JAVA</option>

<option value="html">HTML</option>

</select><br>

<input type="submit" value="Submit">

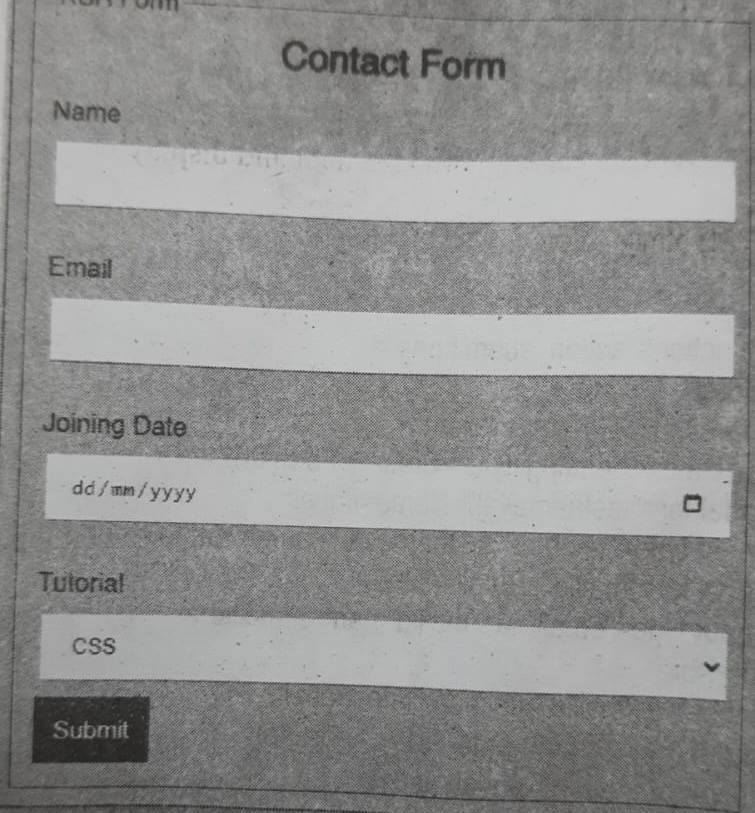
</fieldset>

</form>

</div>

</body> </html>

**Output**:



**CSS Counters,**

Counters in CSS are basically variables which can be used for numbering and values of CSS counters may be incremented by CSS rules. For example, CSS counters can be used to increment the numbering of the headings automatically. In HTML, <ol> tag is used to give the ordered numbers to list items but CSS contains counter to give order elements in some other fashion.

**CSS counters properties:** CSS counters contains the following properties:

* **counter-reset:** It is used to reset a counter.
* **counter-increment:** It basically increments a counter value.
* **content:** It is used to generate content.
* **counter() or counters() function:** The value of a counter can be displayed using either the counter() or counters() function in a content property. These two functions basically used to add the value of a counter to the element.

**Initialization the CSS Counter:** To use CSS counter property firstly it must be created with the counter-reset property and the first step is resetting the counter. The myCounter by default initialized to a value 0(zero) with the counter-reset property.

**Syntax:**

**counter-reset: myCounter;**

**Incrementation and Use of CSS Counter:**To incrementing the counter use CSS counter-increment property.

**Syntax:**

**counter-increment: myCounter;**

The counter() or counters() function in a content is used to display the content in a particular order.

**Syntax:**

**content: counter(myCounter);**

**Example:**

<!DOCTYPE html>

<html>

<head>

<title>CSS counter property</title>

<style>

body {

counter-reset: myCounter;

}

h4::before {

counter-increment: myCounter;

content: "Heading " counter(myCounter) ": ";

}

.geeks {

color:#090;

font-size:40px;

font-weight:bold;

text-align:center;

}

.gfg {

text-align:center;

font-size:18px;

}

</style>

</head>

<body>

<div class = "geeks">GeeksforGeeks</div>

<div class = "gfg">CSS counter property</div>

<h3>Example of automatic numbering with CSS counters</h3>

<h4>GeekforGeeks</h4>

<h4>Computer Science portal</h4>

<h4>Geeks</h4>

</body>

</html>

**OUTPUT:-**



**CSS Responsive**

Responsive web design provides an optimal experience, easy reading and easy navigation with a minimum of resizing on different devices such as desktops, mobiles and tabs.

The website should adjust properly in all the devices as per the shape and size of the devices. A responsive website will look good and attractive in all the devices and none of the links, images, fonts etc should break while the website is seen from any devices.

It is difficult to make a responsive design by only HTML. It is better to combine with CSS Floats, CSS media queries, CSS flexbox, CSS grids or a CSS framework like Bootstrap.

## Bootstrap responsive web design

Bootstrap is most popular web design framework based on HTML,CSS and Java script and it helps you to design web pages in responsive way for all devices.

<html>

<head>

<meta charset = "utf-8">

<meta name = "viewport" content = "width=device-width, initial-scale = 1">

<link rel = "stylesheet"

href = "http://maxcdn.bootstrapcdn.com/bootstrap/3.2.0/css/bootstrap.min.css">

<style>

body {

color:green;

}

</style>

</head>

<body>

<div class = "container">

<div class = "jumbotron">

<h1>Tutorials point</h1>

<p>

Tutorials Point originated from the idea that there exists a class

of readers who respond better to online content and prefer to learn

new skills at their own pace from the comforts of their drawing rooms.

</p>

</div>

<div class = "row">

<div class = "col-md-4">

<h2>Android</h2>

<p>

Android is an open source and Linux-based operating system for mobile

devices such as smartphones and tablet computers. Android was developed

by the Open Handset Alliance, led by Google, and other companies.

</p>

</div>

<div class = "col-md-4">

<h2>CSS</h2>

<p>

Cascading Style Sheets, fondly referred to as CSS, is a simple design

language intended to simplify the process of making web pages presentable.

</p>

</div>

<div class = "col-md-4">

<h2>Java</h2>

<p>

Java is a high-level programming language originally developed by Sun

Microsystems and released in 1995. Java runs on a variety of platforms,

such as Windows, Mac OS, and the various versions of UNIX. This tutorial

gives a complete understanding of Java.

</p>

</div>

</div>

</body>

</html>

# OUTPUT:-

# Tutorials point

Tutorials Point originated from the idea that there exists a class of readers who respond better to online content and prefer to learn new skills at their own pace from the comforts of their drawing rooms.

## Android

Android is an open source and Linux-based operating system for mobile devices such as smartphones and tablet computers. Android was developed by the Open Handset Alliance, led by Google, and other companies.

## CSS

Cascading Style Sheets, fondly referred to as CSS, is a simple design language intended to simplify the process of making web pages presentable.

## Java

Java is a high-level programming language originally developed by Sun Microsystems and released in 1995. Java runs on a variety of platforms, such as Windows, Mac OS, and the various versions of UNIX. This tutorial gives a complete understanding of Java.

**Unit – III**

**Client side Validation:** Introduction to JavaScript - What is DHTML, JavaScript, basics, variables, string manipulations, mathematical functions, statements, operators, arrays, functions. Objects in JavaScript - Data and objects in JavaScript, regular expressions, exception handling.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**What is DHTML**

**DHTML** stands for **Dynamic Hypertext Markup language** i.e., **Dynamic HTML**.

Dynamic HTML is not a markup or programming language but it is a term that combines the features of various web development technologies for creating the web pages dynamic and interactive.

The DHTML application was introduced by Microsoft with the release of the 4th version of IE (Internet Explorer) in 1997.

**Components of Dynamic HTML**

DHTML consists of the following four components or languages:

* HTML 4.0
* CSS
* JavaScript
* DOM.

**HTML 4.0**

HTML is a client-side markup language, which is a core component of the DHTML. It defines the structure of a web page with various defined basic elements or tags.

**CSS**

CSS stands for Cascading Style Sheet, which allows the web users or developers for controlling the style and layout of the HTML elements on the web pages.

**JavaScript**

JavaScript is a scripting language which is done on a client-side. The various browser supports JavaScript technology. DHTML uses the JavaScript technology for accessing, controlling, and manipulating the HTML elements. The statements in JavaScript are the commands which tell the browser for performing an action.

**DOM**

DOM is the document object model. It is a w3c standard, which is a standard interface of programming for HTML. It is mainly used for defining the objects and properties of all elements in HTML.

**Features of DHTML**

Following are the various characteristics or features of DHTML (Dynamic HTML):

* Its simplest and main feature is that we can create the web page dynamically.
* **Dynamic Style** is a feature, that allows the users to alter the font, size, color, and content of a web page.
* It provides the facility for using the events, methods, and properties. And, also provides the feature of code reusability.
* It also provides the feature in browsers for data binding.
* Using DHTML, users can easily create dynamic fonts for their web sites or web pages.
* With the help of DHTML, users can easily change the tags and their properties.
* The web page functionality is enhanced because the DHTML uses low-bandwidth effect.

**Difference between HTML and DHTML**

Following table describes the differences between HTML and DHTML:

|  |  |
| --- | --- |
| **HTML (Hypertext Markup language)** | **DHTML (Dynamic Hypertext Markup language)** |
| 1. HTML is simply a markup language. | 1. DHTML is not a language, but it is a set of technologies of web development. |
| 2. It is used for developing and creating web pages. | 2. It is used for creating and designing the animated and interactive web sites or pages. |
| 3. This markup language creates static web pages. | 3. This concept creates dynamic web pages. |
| 4. It does not contain any server-side scripting code. | 4. It may contain the code of server-side scripting. |
| 5. The files of HTML are stored with the .html or .htm extension in a system. | 5. The files of DHTML are stored with the .dhtm extension in a system. |
| 6. A simple page which is created by a user without using the scripts or styles called as an HTML page. | 6. A page which is created by a user using the HTML, CSS, DOM, and JavaScript technologies called a DHTML page. |
| 7. This markup language does not need database connectivity. | 7. This concept needs database connectivity because it interacts with users. |

**What is JavaScript ?**

* JavaScript is the most popular scripting language on the internet, and works in all major browsers, such as Internet Explorer, Firefox, Chrome, Opera, and Safari.
* JavaScript was designed to add interactivity to HTML pages
* A scripting language is a lightweight programming language
* JavaScript is usually embedded directly into HTML pages
* JavaScript is an interpreted language (means that scripts execute without preliminary compilation)
* JavaScript originates from a language called LiveScript.
* JavaScript was developed in 1995
* The HTML <script> tag is used to insert a JavaScript into an HTML page.
* JavaScripts can be put in the body and in the head sections of an HTML page.

**Example:**

<html>

<head><title>java script program</title>

<script type="text/javascript">

document.write("Hai...");

</script>

</head>

<body >

</body>

</html>

* To insert a JavaScript into an HTML page, we use the <script> tag. Inside the <script> tag we use the type attribute to define the scripting language.
* The document.write command is a standard JavaScript command for writing output to a page.
* JavaScript in a page will be executed immediately while the page loads into the browser.
* Scripts to be executed when they are called, or when an event is triggered, are placed in functions.

Advantages of JavaScript

The merits of using JavaScript are −

* **Less server interaction** − You can validate user input before sending the page off to the server. This saves server traffic, which means less load on your server.
* **Immediate feedback to the visitors** − They don't have to wait for a page reload to see if they have forgotten to enter something.
* **Increased interactivity** − You can create interfaces that react when the user hovers over them with a mouse or activates them via the keyboard.
* **Richer interfaces** − You can use JavaScript to include such items as drag-and-drop components and sliders to give a Rich Interface to your site visitors.

Limitations of JavaScript

We cannot treat JavaScript as a full-fledged programming language. It lacks the following important features −

* Client-side JavaScript does not allow the reading or writing of files. This has been kept for security reason.
* JavaScript cannot be used for networking applications because there is no such support available.
* JavaScript doesn't have any multi-threading or multiprocessor capabilities.

Once again, JavaScript is a lightweight, interpreted programming language that allows you to build interactivity into otherwise static HTML pages.

**JAVASCRIPT BASICS**

1. Comments
2. Variables
3. Data Types
4. Operators
5. Statements
6. Arrays
7. Functions

**JAVASCRIPT COMMENTS**

Javascript comments are the meaningful way to deliver message. It is to add information about the code, warnings or suggestions so that end use can easily interpret in the code.

There are two types of comments in javascript:

1. Single line comments
2. Multi line comments

**VARIABLES**

* Like any other programming language, JavaScript has variables. A variable is a named value that we can use in our programs.
* Variable names must begin with a letter, digit;
* We cannot use spaces in variable names
* Variable names are case sensitive, so that fred, FRED and frED all refer to different variables.
* Reserved words are not used as variable names.

**e.g.:** **var a = 10;**

**var name = “Krishna”;** where var is optional

<html>

<head>

<title>Variables in java script program</title>

</head>

<body bgcolor=”pink”>

<script type="text/javascript">

Var a=25;

document.write(a);

</script>

</body>

</html>

Variable are two types. They are 1. Local Variable and 2. Global Variables

**Local Variables:-**

Local variables are declared inside a function.

Let us consider an example program for local variables

<html>

<head>

<title>Variables in java script program</title>

</head>

<body bgcolor=”pink”>

<script type="text/javascript">

Function add()

{

Var a=25; //local variables

document.write(a);

}

Add();

</script>

</body>

</html>

**global Variables:-**

global variables are declared outside a function, and they can be accessed in the any function in the program.

Let us consider an example program for global variables

<html>

<head>

<title>Variables in java script program</title>

</head>

<body bgcolor=”pink”>

<script type="text/javascript">

Var x=10; //Gloal variables

Function add()

{

document.write(x);

}

Add();

document.write(x+5);

</script>

</body>

</html>

**STRING MANIPULATION FUNCTIONS**

**String Manipulation:** A String is defined as a group of characters. We can create a String in two ways:

* var name1 = “kiran”;
* var name2 = new String(“kumar”);

The following methods are String object methods.

|  |  |
| --- | --- |
| **Method** | **Description** |
| [charAt(index)](http://www.tutorialspoint.com/javascript/string_charat.htm) | Returns the character at the specified index. |
| [concat(“string”,”string”,…)](http://www.tutorialspoint.com/javascript/string_concat.htm) | Combines the text of two strings and returns a new string. |
| [indexOf(“string”)](http://www.tutorialspoint.com/javascript/string_indexof.htm) | Returns the index within the calling String object of the first occurrence of the specified value, or -1 if not found. |
| [lastIndexOf(“string”)](http://www.tutorialspoint.com/javascript/string_lastindexof.htm) | Returns the index within the calling String object of the last occurrence of the specified value, or -1 if not found. |
| [slice(begin index, last index)](http://www.tutorialspoint.com/javascript/string_split.htm) | This function is used to cut a string between begin index and last index. |
| [toLowerCase()](http://www.tutorialspoint.com/javascript/string_tolowercase.htm) | Returns the calling string value converted to lower case. |
| [toUpperCase()](http://www.tutorialspoint.com/javascript/string_touppercase.htm) | Returns the calling string value converted to uppercase. |
| [length()](http://www.tutorialspoint.com/javascript/string_valueof.htm) | Returns number of characters. |

**Example:**

<html>

<head><title>String Functions</title>

<script language="javascript">

var name1 = "kiran";

var name2 = "kumar";

document.write("concatenation of name1,name2 is : " + name1.concat(name2)+ "<br>" );

document.write("1st character in name1 is : " + name1.charAt(0)+ "<br>");

document.write("name2 in uppercase is : " + name2.toUpperCase()+ "<br>");

</script></head>

<body></body>

</html>

**MATHEMATICAL FUNCTIONS**

**Mathematical Functions: In** java script we can perform mathematical functions like sin,cos, tan etc by using the math object.the general syntax for using math object is as follows.

**math.function-name(value)**

|  |  |
| --- | --- |
| **Method** | **Description** |
| [abs(value)](http://www.tutorialspoint.com/javascript/math_abs.htm) | Returns the absolute value of a number. |
| [ceil(value)](http://www.tutorialspoint.com/javascript/math_ceil.htm) | Returns the smallest integer greater than or equal to a number. |
| [cos(value)](http://www.tutorialspoint.com/javascript/math_cos.htm) | Returns the cosine of a number. |
| [exp(value)](http://www.tutorialspoint.com/javascript/math_exp.htm) | Returns EN, where N is the argument, and E is Euler”s constant, the base of the natural logarithm. |
| [floor(value)](http://www.tutorialspoint.com/javascript/math_floor.htm) | Returns the largest integer less than or equal to a number. |
| [log(value)](http://www.tutorialspoint.com/javascript/math_log.htm) | Returns the natural logarithm (base E) of a number. |
| [max(value1, value2)](http://www.tutorialspoint.com/javascript/math_max.htm) | Returns the largest of zero or more numbers. |
| [min(value1,value2)](http://www.tutorialspoint.com/javascript/math_min.htm) | Returns the smallest of zero or more numbers. |
| [pow(value, power)](http://www.tutorialspoint.com/javascript/math_pow.htm) | Returns base to the exponent power, that is, base exponent. |
| [random()](http://www.tutorialspoint.com/javascript/math_random.htm) | Returns a pseudo-random number between 0 and 1. |
| [round(value)](http://www.tutorialspoint.com/javascript/math_round.htm) | Returns the value of a number rounded to the nearest integer. |
| [sin(value)](http://www.tutorialspoint.com/javascript/math_sin.htm) | Returns the sine of a number. |
| [sqrt(value)](http://www.tutorialspoint.com/javascript/math_sqrt.htm) | Returns the square root of a number. |
| [tan(value)](http://www.tutorialspoint.com/javascript/math_tan.htm) | Returns the tangent of a number. |

**Example:**

<html>

<head><title>Mathematical Functions</title>

<script language="javascript">

document.write(Math.cos(90));

document.write(Math.sin(90) );

document.write(Math.tan(90));

document.write(Math.max(90,10) );

document.write("min of 90,10 value is : "+ Math.min(90,10));

</script></head>

<body></body>

</html>

**STATEMENTS**

**Statements:** JavaScript supports three types of statements.

* Conditional Statements – if, switch..case
* Looping/ Iteration Statements – while, do..while, for
* Jump statements – break, continue etc.

**Conditional Statements***:-* These statements help to jump from one part of the program to another depending on whether a particular condition is satisfied or not. These decision control statements include:

(a) if statement,

(b) if–else statement,

(c) if–else–if statement,

(d) nested if statement and

(e) switch–case statement.

**if *Statement***

if statement is the simplest decision control statement that is frequently used in decision making. The general

form of a simple if statement is:

**syntax**:-

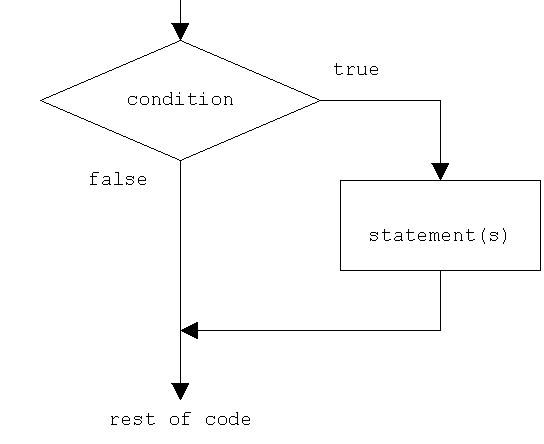
if(condition)

{

Statement 1

}

Statement X



The if block may include 1 statement or *n* statements enclosed within curly brackets. First the test expression is

evaluated. If the test expression is true, the statements of the if block are executed, otherwise these statements will be skipped and the execution will jump to statement x.

**if – else Statement**

in the if-else statement, The condition is evaluated, if the result is true, the statement 1 is executed and then statement X will be executed, else if the expression is false, the statement 2 is executed and then statement X will be executed.. The general form of simple if–else statement is shown

**Syntax**:-

if(condition)

{

Statement 1

}

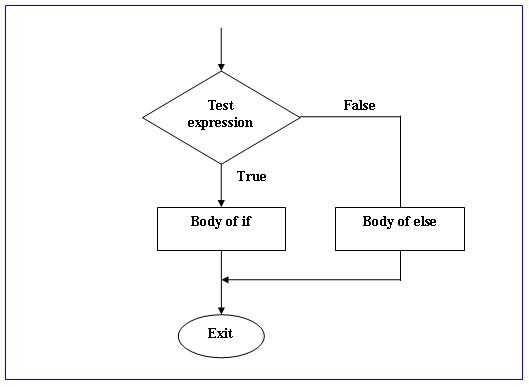
Else

{

Statement 2

}

Statement X



**If-else-if statement**:-there is another way of putting if…else statement together when multipath decisions are involved.if-else-if statements is a chain of if-else statements. The general form of this statement is as follows

If (condition)

Statement 1

Else if (condition)

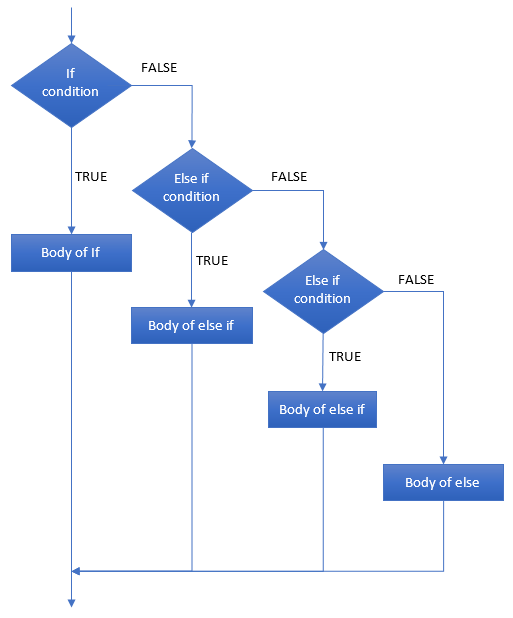
Statement 2

Else if(condition)

Statement 3

Else

Statement



**Nested If:-**

When a series of decisions are involved we may have to use more than one if—else statements in nested form as follows:

If (condition 1)

If( condition 2)

Statement 1

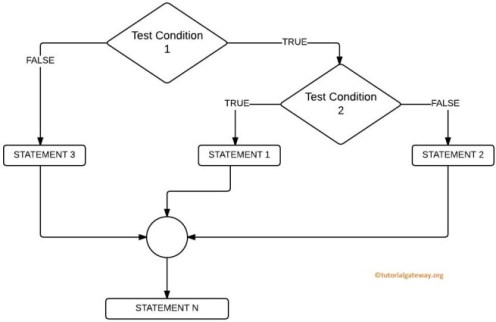
Else

Statements 2

Else

Statement 3

Statement X

****  
switch statement:-**

A switch-case statement is a multi-way decision statement that is a simplified version of an if– else–

if block. The general form of a switch statement is shown

**syntax:-**

switch(expression)

{

Case 1 : block 1;

Break;

Case 2 : block 2;

Break;

Case 3 : block 3;

Break;

.

.

.

.

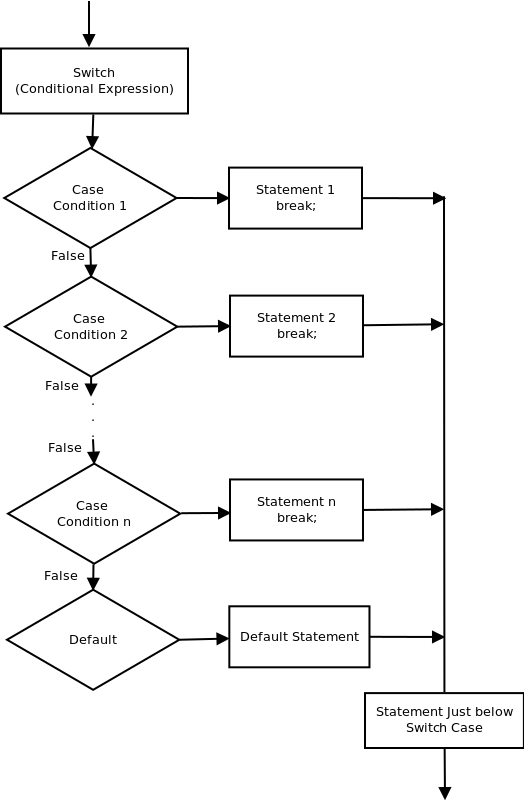
Case n : block n;

Break;

Default : statement

}

Statement X

**

**Looping statement** are the statements execute one or more statement repeatedly several number of times. In programming language there are three types of loops; while, for and do-while.

### Why use loop ?

When you need to execute a block of code several number of times then you need to use looping concept.

### Types of Loops.

There are three type of Loops available

* while loop
* for loop
* do..while

## While loop

In **While Loop ,** First check the condition if condition is true then control goes inside the loop body other wise goes outside the body. **while loop** will be repeats in clock wise direction.

**Syntax**:-

Assignment;

while(condition)

{

Statements;

......

Increment/decrements (++ or --);

}

## For loop

**For Loop**  is a statement which allows code to be repeatedly executed. For loop contains 3 parts Initialization, Condition and Increment or Decrements.

Syntax:-

For(initialization;condition;increment/decrement)

{

Loop body;

}

## do-while

A **do-while Loop**  is similar to a while loop, except that a do-while loop is execute at least one time.

A do while loop is a control flow statement that executes a block of code at least once, and then repeatedly executes the block, or not, depending on a given condition at the end of the block (in while).

Syntax:-

do

{

Statements;

........

Increment/decrement (++ or --)

} while();

**Jump statements** are used to interrupt the normal flow of program.

### Types of Jump Statements

* Break
* Continue

## Break Statement

The break statement is used inside loop or switch statement.

## Continue Statement

The continue statement is also used inside loop. When compiler finds the break statement inside a loop, compiler will skip all the followling statements in the loop and resume the loop.

**OPERATORS**

**Operators:** An operator is a symbol, which is used to perform mathematical and logical operations on operand. JavaScript language supports following type of operators.

* Arithmetic Operators
* Relational/Comparison Operators
* Logical Operators
* Bitwise operators
* Assignment Operators
* Conditional (or ternary) Operators

**Arithmetic Operators: A**rithmetic Operators are used to perform basic mathematical operations like +,-,\*,/,etc.There are following arithmetic operators supported by JavaScript language: Assume variable A holds 10 and variable B holds 20 then:

|  |  |  |
| --- | --- | --- |
| **Operator** | **Description** | **Example** |
| + | Adds two operands | A + B will give 30 |
| - | Subtracts second operand from the first | A - B will give -10 |
| \* | Multiply both operands | A \* B will give 200 |
| / | Divide numerator by denumerator | B / A will give 2 |
| % | Modulus Operator and remainder of after an integer division | B % A will give 0 |
| ++ | Increment operator, increases integer value by one | A++ will give 11 |
| -- | Decrement operator, decreases integer value by one | A-- will give 9 |

**Comparison/ Relational Operators:** this operators are used to comparisons. There are following comparison operators supported by JavaScript language. Assume variable A holds 10 and variable B holds 20 then:

|  |  |  |
| --- | --- | --- |
| **Operator** | **Description** | **Example** |
| == | Checks if the value of two operands are equal or not, if yes then condition becomes true. | (A == B) is not true. |
| != | Checks if the value of two operands are equal or not, if values are not equal then condition becomes true. | (A != B) is true. |
| > | Checks if the value of left operand is greater than the value of right operand, if yes then condition becomes true. | (A > B) is not true. |
| < | Checks if the value of left operand is less than the value of right operand, if yes then condition becomes true. | (A < B) is true. |
| >= | Checks if the value of left operand is greater than or equal to the value of right operand, if yes then condition becomes true. | (A >= B) is not true. |
| <= | Checks if the value of left operand is less than or equal to the value of right operand, if yes then condition becomes true. | (A <= B) is true. |

**Logical Operators:** Logical operatorsare used to test two or morw conditions at a time.There are following logical operators supported by JavaScript language. Assume variable A holds 10 and variable B holds 20 then:

|  |  |  |
| --- | --- | --- |
| **Operator** | **Description** | **Example** |
| && | Called Logical AND operator. If both the operands are non zero then then condition becomes true. | (A && B) is true. |
| || | Called Logical OR Operator. If any of the two operands are non zero then then condition becomes true. | (A || B) is true. |
| ! | Called Logical NOT Operator. Use to reverses the logical state of its operand. If a condition is true then Logical NOT operator will make false. | !(A && B) is false. |

**The** working of logical operator are will under stand by truth table:

|  |  |  |  |
| --- | --- | --- | --- |
| Op 1 | Op 2 | && | 11 |
| 1 | 1 | 1 | 1 |
| 1 | 0 | 0 | 1 |
| 0 | 1 | 0 | 1 |
| 0 | 0 | 0 | 0 |

**Bitwise Operators:** There are following bitwise operators supported by JavaScript language

Assume variable A holds 2 and variable B holds 3 then:

|  |  |  |
| --- | --- | --- |
| **Operator** | **Description** | **Example** |
| & | Called Bitwise AND operator. It performs a Boolean AND operation on each bit of its integer arguments. | (A & B) is 2 . |
| | | Called Bitwise OR Operator. It performs a Boolean OR operation on each bit of its integer arguments. | (A | B) is 3. |
| ^ | Called Bitwise XOR Operator. It performs a Boolean exclusive OR operation on each bit of its integer arguments. Exclusive OR means that either operand one is true or operand two is true, but not both. | (A ^ B) is 1. |
| ~ | Called Bitwise NOT Operator. It is a is a unary operator and operates by reversing all bits in the operand. | (~B) is -4 . |
| << | Called Bitwise Shift Left Operator. It moves all bits in its first operand to the left by the number of places specified in the second operand. New bits are filled with zeros. Shifting a value left by one position is equivalent to multiplying by 2, shifting two positions is equivalent to multiplying by 4, etc. | (A << 1) is 4. |
| >> | Called Bitwise Shift Right with Sign Operator. It moves all bits in its first operand to the right by the number of places specified in the second operand. The bits filled in on the left depend on the sign bit of the original operand, in order to preserve the sign of the result. If the first operand is positive, the result has zeros placed in the high bits; if the first operand is negative, the result has ones placed in the high bits. Shifting a value right one place is equivalent to dividing by 2 (discarding the remainder), shifting right two places is equivalent to integer division by 4, and so on. | (A >> 1) is 1. |
| >>> | Called Bitwise Shift Right with Zero Operator. This operator is just like the >> operator, except that the bits shifted in on the left are always zero, | (A >>> 1) is 1. |

**Assignment Operators:** This operator is used to assign a value of a variable. It is denoted by “=”. There are following assignment operators supported by JavaScript language:

|  |  |  |
| --- | --- | --- |
| **Operator** | **Description** | **Example** |
| = | Simple assignment operator, Assigns values from right side operands to left side operand | C = A + B will assigne value of A + B into C |
| += | Add AND assignment operator, It adds right operand to the left operand and assign the result to left operand | C += A is equivalent to C = C + A |
| -= | Subtract AND assignment operator, It subtracts right operand from the left operand and assign the result to left operand | C -= A is equivalent to C = C - A |
| \*= | Multiply AND assignment operator, It multiplies right operand with the left operand and assign the result to left operand | C \*= A is equivalent to C = C \* A |
| /= | Divide AND assignment operator, It divides left operand with the right operand and assign the result to left operand | C /= A is equivalent to C = C / A |
| %= | Modulus AND assignment operator, It takes modulus using two operands and assign the result to left operand | C %= A is equivalent to C = C % A |

**Note:** Same logic applies to Bitwise operators so they will become like <<=, >>=, >>=, &=, |= and ^=.

## The Conditional Operator (? :)

There is an operator called conditional operator. This first evaluates an expression for a true or false value and then execute one of the two given statements depending upon the result of the evaluation. The conditional operator has this syntax:

|  |  |  |
| --- | --- | --- |
| **Operator** | **Description** | **Example** |
| ? : | Conditional Expression | Exp 1? Exp 2 : Exp 3 |

**ARRAYS**

**Arrays:** An array is a collection of elements of same data type.

* **Creating Arrays:** We can create an array in JavaScript in several ways.
* The easiest way is simply to declare a variable and pass it some elements in array format:

**var days = [“Monday”, “Tuesday”, “Wednesday”, “Thursday”];**

The above example creates an array of four elements each holding a text string. Notice that the array of elements is surrounded by square brackets.

* The second approach is to create an array object using the keyword new and a set of elements to store:

**var days = new Array("Monday", "Tuesday", "Wednesday", "Thursday");**

In the above example the contents of the array are surrounded by parenthesis because they are parameters to the constructor of the Array Object.

* Finally an empty array object which has space for a number of elements can be created:

**var days = new Array(4);**

**Example:**

<html>

<head><title>Accessing Array</title>

</head>

<body>

<script language = "javascript" type="text/javascript">

var a = [10, 20, 30, 40, 50];

for(i=0; i< a.length; i++)

document.writeln("<h1>" + a[i] + "</h1>");

</script>

</body>

</html>

* **Object-based Array Functions**

|  |  |
| --- | --- |
| **Method** | **Description** |
| concat() | Returns a new array comprised of this array joined with other array(s) and/or value(s). |
| [pop()](http://www.tutorialspoint.com/javascript/array_pop.htm) | Removes the last element from an array and returns that element. |
| push () | Adds one or more elements to the end of an array and returns the new length of the array. |
| [reverse()](http://www.tutorialspoint.com/javascript/array_reverse.htm) | Reverses the order of the elements of an array -- the first becomes the last, and the last becomes the first. |
| [shift()](http://www.tutorialspoint.com/javascript/array_shift.htm) | Removes the first element from an array and returns that element. |
| slice() | Extracts a section of an array and returns a new array. |
| [sort()](http://www.tutorialspoint.com/javascript/array_sort.htm) | Sorts the elements of an array. |

**FUNCTONS**

**JavaScript Functions:**

A function is a group of reusable code which can be called anywhere in our program. This eliminates the need of writing same code again and again. We can divide our big program in a number of small and manageable functions.

The most common way to define a function in JavaScript is by using the function keyword, followed by a unique function name, a list of parameters (that might be empty), and a statement block surrounded by curly braces. The basic syntax is shown here:

<script type="text/javascript">

function functionname(parameter-list)

{

statements

}

</script>

## Example:

A simple function that takes no parameters called sayHello is defined here:

<script type="text/javascript">

function sayHello()

{

alert("Hello there");

}</script>

**Calling a Function:**

To invoke a function somewhere later in the script, we need to write the name of that function as follows:

<script type="text/javascript">

sayHello();

</script>

**Function Parameters:**

We can pass different parameters while calling a function. These passed parameters can be captured inside the function and any manipulation can be done over those parameters. A function can take multiple parameters separated by comma.

## Example:

<script type="text/javascript">

function sayHello(name, age)

{

alert( name + " is " + age + " years old.");

}

</script>

Now we can call this function as follows:

<script type="text/javascript">

sayHello(“Zara”, 7 );

</script>

**Objects in JavaScript - Data and objects in JavaScript**

**JavaScript Objects:**

* JavaScript is an Object Oriented Programming (OOP) language.
* Objects are composed of attributes. If an attribute contains a function, it is considered to be a method of the object otherwise; the attribute is considered a property.

**Object Properties:**

Object properties are usually variables that are used internally in the object's methods, but can also be globally visible variables that are used throughout the page.

The syntax for adding a property to an object is:

**objectName.objectProperty = propertyValue;**

**Object Methods:**

The methods are functions that let the object do something or let something be done to it. There is little difference between a function and a method, except that a function is a standalone unit of statements and a method is attached to an object and can be referenced by the this keyword.

**Example:**

Following is a simple example to show how to use write() method of document object to write any content on the document:

**document.write ("This is test");**

**User-Defined Objects:**

All user-defined objects and built-in objects are descendants of an object called Object.

**The new Operator:**

The new operator is used to create an instance of an object.

To create an object, the new operator is followed by the constructor method.

**Example:** In the following example, the constructor methods are Object (), Array (), and Date (). These constructors are built-in JavaScript functions.

var employee = new Object();

var books = new Array("C++", "Perl", "Java");

var day = new Date("August 15, 1947");

The built-in objects in javascript are:

1. Date and time objects
2. Array abjects
3. Math objaects
4. String objects

## Date and time Object

The Date object is used to work with dates and times.

Date objects are created with new Date().

## Date Object Methods

|  |  |
| --- | --- |
| **Method** | **Description** |
| [getDate()](http://www.w3schools.com/jsref/jsref_getdate.asp) | Returns the day of the month (from 1-31) |
| [getDay()](http://www.w3schools.com/jsref/jsref_getday.asp) | Returns the day of the week (from 0-6) |
| [getFullYear()](http://www.w3schools.com/jsref/jsref_getfullyear.asp) | Returns the year (four digits) |
| [getHours()](http://www.w3schools.com/jsref/jsref_gethours.asp) | Returns the hour (from 0-23) |
| [getMilliseconds()](http://www.w3schools.com/jsref/jsref_getmilliseconds.asp) | Returns the milliseconds (from 0-999) |
| [getMinutes()](http://www.w3schools.com/jsref/jsref_getminutes.asp) | Returns the minutes (from 0-59) |
| [getMonth()](http://www.w3schools.com/jsref/jsref_getmonth.asp) | Returns the month (from 0-11) |
| [getSeconds()](http://www.w3schools.com/jsref/jsref_getseconds.asp) | Returns the seconds (from 0-59) |
| [getTime()](http://www.w3schools.com/jsref/jsref_gettime.asp) | Returns the number of milliseconds since midnight Jan 1, 1970 |
| [setDate()](http://www.w3schools.com/jsref/jsref_setdate.asp) | Sets the day of the month of a date object |
| [setFullYear()](http://www.w3schools.com/jsref/jsref_setfullyear.asp) | Sets the year (four digits) of a date object |
| [setHours()](http://www.w3schools.com/jsref/jsref_sethours.asp) | Sets the hour of a date object |
| [setMilliseconds()](http://www.w3schools.com/jsref/jsref_setmilliseconds.asp) | Sets the milliseconds of a date object |
| [setMinutes()](http://www.w3schools.com/jsref/jsref_setminutes.asp) | Set the minutes of a date object |
| [setMonth()](http://www.w3schools.com/jsref/jsref_setmonth.asp) | Sets the month of a date object |
| [setSeconds()](http://www.w3schools.com/jsref/jsref_setseconds.asp) | Sets the seconds of a date object |
| [setTime()](http://www.w3schools.com/jsref/jsref_settime.asp) | Sets a date and time by adding or subtracting a specified number of milliseconds to/from midnight January 1, 1970 |

**Regular Expressions**

* JavaScript include a set of routines to manipulate strings and search patterns. These are wrapped up as regular expression objects.
* JavaScript regular expressions are more than patterns: they include functions which we can call from scripts when we need a pattern finding.

**var pattern = “target”;**

**var string = “can you find the target?”;**

**string.match(pattern);**

But we can also work the other way. A string can be passed into a regular expression as a parameter:

**var pattern = new RegExp(“target”);**

**var string = “can you find the target?”;**

**pattern.exec(string);**

**Creating Regular Expressions:**

A regular expression is an object that describes a pattern of characters.

**Syntax: 1.** Dynamic patterns are created using the new keyword to create an instance of the RegExp class:

**regex = new RegExp(“fish | fowl”);**

2.A static regular expression is created as follows:

**regex = /pattern/modifier;**

**Modifiers or flags:**

In JavaScript, there are three optional flags that allow you to change how the regular expressions engine will perform the actual matching:

**G :** Global match — find all matches rather than only the first one

**Example:**

var str = "Joe sells cogs from his cog store.";

var regex = /cog/g;

var result = str.replace(regex, "bed");

document.write(result); // Joe sells beds from his bed store.

**I : -** Ignore case — match both lower and upper case letters

**Example:**

var str = "I want to fIGhT.";

var regex = /fight/i;

var result = str.replace(regex, "fart");

document.write(result); // I want to fart.

**M:** Multi-line match — tells the engine to treat the subject string as multiple lines. ^ and $

match next to \n instead of the start or end of the entire string Special characters and

character classes.

**Example:**

var str = "pop pop\npop pop";

var regex = /p$/gm;

var result = str.replace(regex, "X");

document.write(result); // pop poX pop poX

**JavaScript Regular Expression Grammar**

|  |  |  |  |
| --- | --- | --- | --- |
| **Token** | **Description** | **Syntax** | |
| ^ | Match at the start of the input string | /^abc/ | |
| $ | Match at the end of the input string | /abc$/ | |
| \* | Match 0 or more times | /abc\*/ | |
| + | Match 1 or more times | /abc+/ | |
| ? | Match 0 or 1 time | /abc?/ | |
| a| b | Match a or b | /abc|def/ | |
| {n} | Match the string n times | /abc{i}/ | |
| \d | Match a digit | /\d/ | |
| \D | Match anything except digits |  | |
| \w | Match any alphanumeric character or the underscore |  | |
| \W | Match anything except alphanumeric character or the underscore | |  |
| \s | Match a white space character | |  |
| \S | Match anything except a white space character | |  |
| […] | Creates a set of characters, one of which must match if the operation is to be successful. If you need to specify a range of characters then separate the first and last with a hyphen: [0-9] or [D-G]. | | |
| [^…] | Creates a set of characters, which must not match. If any character in the set matches then the operation is failed. This fails if any lowercase letter from d to q is matched: [^d-q]. | | |

**class RegExp Functions:**

The RegExp objects provides two methods for working with regular expressions:

**test(text):** The test(text) method tests for a match in the input string. It searches the string for the specified pattern and returns true if the pattern matches the string or false otherwise.

**Example:**

<html>

<head></head>

<body><script language="javascript">

var str = "abc 123";

var regex = /bc 12/;

var result = regex.test(str);

document.write(result); // true

</script></body>

</html>

**exec(text):** The exec(text) method executes the specified pattern on the input string and returns an array of matched strings if it succeeds or null if it fails. The first element of the array contains the text matched by the entire pattern while the other elements correspond to text that matched captured subpatterns.

**Example:**

<html>

<head></head>

<body><script language="javascript">

var str = "Birds fly in the sky.";

var regex = /fly/;

var result = regex.exec(str);

document.write(result); // fly</script></body>

</html>

**Exception Handling**

* Many object oriented programming languages provide a mechanism for dealing with general classes of error. This mechanism is called exception handling.
* Exception handling allows program to continue execution.
* It helps to identify the problem.
* It helps to terminate the program in a controlled manner.

There are 3 types of errors in java script.

1. Syntax error
2. Runtime error

**Syntax error:-**

Syntax error also called pasing errors, occur at compile time.

Ex:-

<html>

<head><title>syntax error</title></head>

<body>

Document.write(“welcome to the program” ;

</body>

In the above example program missing braces. This is the syntax error.

**runtime error:-**

runtime errors are called exceptions, occur during execution of a program.

Ex:-

<html>

<head><title>runtime error</title></head>

<body>

aler(“welcome to the program” );

</body>

In the above example program the alert method is mistakes. This is the runtime error.these runtime error purpose in javas we are havingexception handling mechanisms.they are

Try

Catch

finally

**JavaScript try and catch:**

* The **try** statement allows you to define a block of code to be tested for errors while it is being executed.
* The **catch** statement allows you to define a block of code to be executed, if an error occurs in the try block.
* The JavaScript statements try and catch come in pairs.

**Syntax:** try

{ //Run some code here

}

catch(err)

{ //Handle errors here

}

**Example1:**

<html>

<head>

<title>exception handling</title>

</head>

<body>

<script>

Try

{

Aler(“welcome to exception”);

}

Catch(e)

{

Alert(e.description);

}

Finally // finally block is always executed

{

Alert(“successfully executed”);

}

alert(“Thank U” );

</script>

</body>

</html

**Unit – IV**

**Word press:** Introduction to word press, servers like wamp, bitnami e.tc, installing and configuring word press, understanding admin panel, working with posts and pages, using editor, text formatting with shortcuts, working with media-Adding, editing, deleting media elements, working with widgets, menus.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**Word press:**

WordPress is an open source **Content Management System (CMS),** which allows the users to build dynamic websites and blogs. Wordpress is the most popular blogging system on the web and allows updating, customizing and managing the website from its back-end CMS and components.

WordPress

## What is Content Management System (CMS)?

The **Content Management System (CMS)** is a software which stores all the data such as text, photos, music, documents, etc. and is made available on your website. It helps in editing, publishing and modifying the content of the website.

WordPress was initially released on 27th May, 2003 by Matt Mullenweg and Mike Little. WordPress was announced as open source in October 2009.

## Features

* **User Management:** It allows managing the user information such as changing the role of the users to (subscriber, contributor, author, editor or administrator), create or delete the user, change the password and user information. The main role of the user manager is **Authentication**.
* **Media Management:** It is the tool for managing the media files and folder, in which you can easily upload, organize and manage the media files on your website.
* **Theme System:** It allows modifying the site view and functionality. It includes images, stylesheet, template files and custom pages.
* **Extend with Plugins:** Several plugins are available which provides custom functions and features according to the users need.
* **earch Engine Optimization:** It provides several search engine optimization (SEO) tools which makes on-site SEO simple.
* **Multilingual:** It allows translating the entire content into the language preferred by the user.
* **Importers:** It allows importing data in the form of posts. It imports custom files, comments, post pages and tags.

## Advantages

* It is an open source platform and available for free.
* CSS files can be modified according to the design as per users need.
* There are many plugins and templates available for free. Users can customize the various plugins as per their need.
* It is very easy to edit the content as it uses WYSIWYG editor (What You See Is What You Get is a user interface that allows the user to directly manipulate the layout of document without having a layout command).
* Media files can be uploaded easily and quickly.
* It offers several SEO tools which makes on-site SEO simple.
* Customization is easy according to the user's needs.
* It allows creating different roles for users for website such as admin, author, editor and contributor.

## Disadvantages

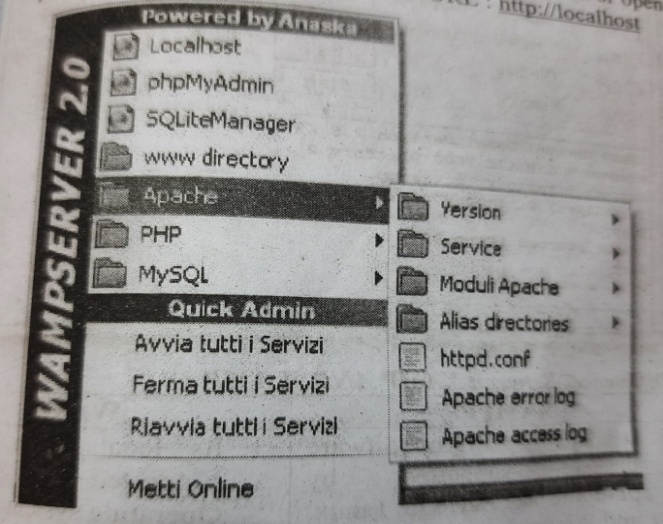
* Using several plugins can make the website heavy to load and run.
* PHP knowledge is required to make modifications or changes in the WordPress website.
* Sometimes software needs to be updated to keep the WordPress up-to-date with the current browsers and mobile devices. Updating WordPress version leads to loss of data, so it a backup copy of the website is required.
* Modifying and formatting the graphic images and tables is difficult.

**Servers like wamp, bitnami e.tc,**

WampServer is a Windows web development environment. It allows you to create web applications with Apache2, PHP and a MySQL database. Alongside, PhpMyAdmin allows you to manage easily your database.

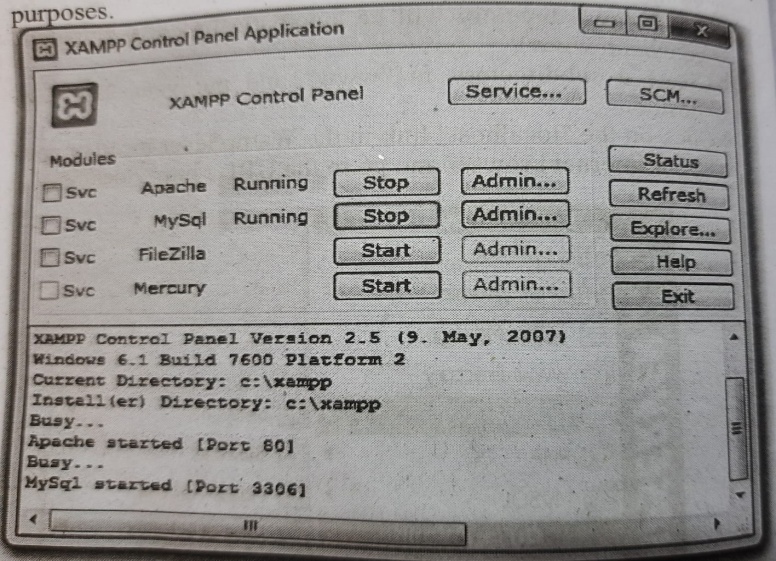
**Using Wampserver:-**

* The "www" directory will be automatically created (usually c:\wamp\www)
* Create a subdirectory in "www" and put your PHP files inside.
* Click on the "localhost" link in the WampSever menu or open your internet browser and go to the URL: <http://localhost>



XAMPP is a free and open source cross-platform web server solution stack package developed by Apache Friends, consisting mainly of the Apache HTTP Server, MariaDB database, and interpreters for scripts written in the PHP and Perl programming languages.

XAMPP stands for Cross-Platform (X Apache (A), MariaDB (M), PHP (P) and Perl (P). It is a simple lightweight Apache distribution that makes it extremely easy for developers to create a local web server for testing and deployment purposes.



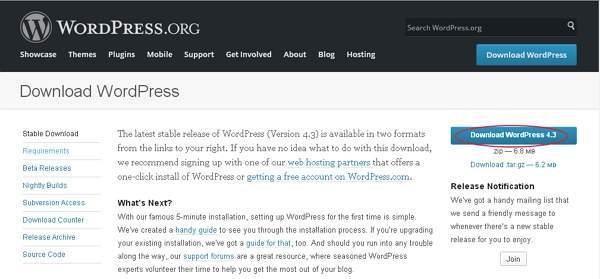
**Installing and Configuring Word Press**

## System Requirements for WordPress

* **Database:** MySQL 5.0+
* **Web Server:**
  + WAMP (Windows)
  + LAMP (Linux)
  + XAMP (Multi-platform)
  + MAMP (Macintosh)
* **Operating System:** Cross-platform
* **Browser Support:** IE (Internet Explorer 8+), Firefox, Google chrome, Safari, Opera
* **PHP Compatibility:** PHP 5.2+

## Download WordPress

When you open the link <https://wordpress.org/download/>, you will get to see a screen as the following snapshot:



Download the WordPress zip file from the official site.

## Create Store Database

* WordPress requires MySQL database. So create a new empty database with user/password (for example, user as "root" and password as "root" or else you can set as per your convenience).
* Then, you can continue with the installation process as discussed further.

## Set Up Wizard

It's very easy to set up WordPress into your system. The following steps describe how to set up WordPress locally on your system.

**Step (1):** Extract the downloaded WordPress folder and upload it into your web server or localhost.

**Step (2):** Open your browser and navigate to your WordPress file path, then you will get the first screen of the WordPress installer as shown in the following screen. In our case, the path is **localhost/< Your\_WordPress\_folder >**.



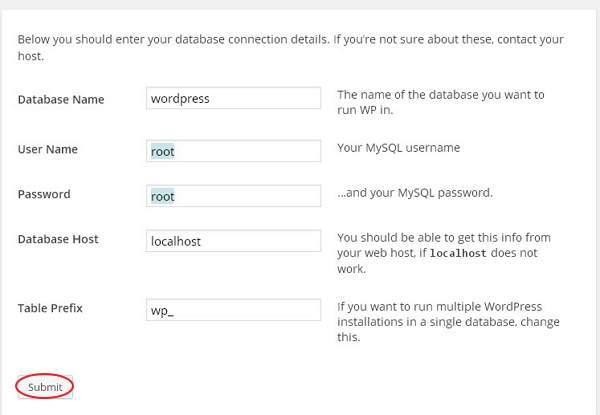
Select your language for the WordPress and click on **Continue**.

**Step (3):** In this step, you can view the information needed for the database before proceeding with WordPress installation.



Click on **Let's go!**

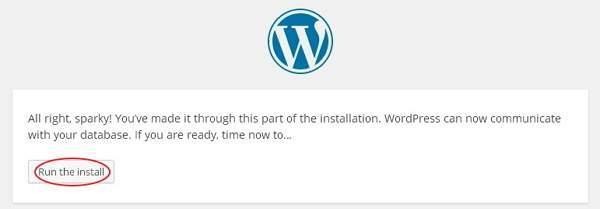
**Step (4):** Here, you have to enter the information about the MySQL database as described in the following screen.



* **Database Name:** Enter the database name which you have created in MySQL database for WordPress.
* **Username:** Enter the user name of your MySQL database.
* **Password:** Enter the password which you had set for MySQL database.
* **Database Host:** Write the host name, by default it will be localhost.
* **Table Prefix:** It is used to add prefix in the database tables which helps to run multiple sites on the same database. It takes the default value.

After filling all information, click on **Submit** button.

**Step (5):** WordPress checks the database setting and gives you the confirmation screen as shown in the following snapshot.



Click on **Run the install**

**Step (6):** Enter administrative information.

It contains the following fields:

* **Site Title:** Enter the name of the site which you are going to create in WordPress.
* **Username:** Enter the username as per your choice while logging in the WordPress.
* **Password twice:** Enter password two times to protect your site.
* **Your E-mail:** Enter your e-mail address which helps to recover the password or any update.
* **Privacy:** It allows the search engine to index this site after checking the checkbox.

After filling all the information, click on the **Install WordPress** button.

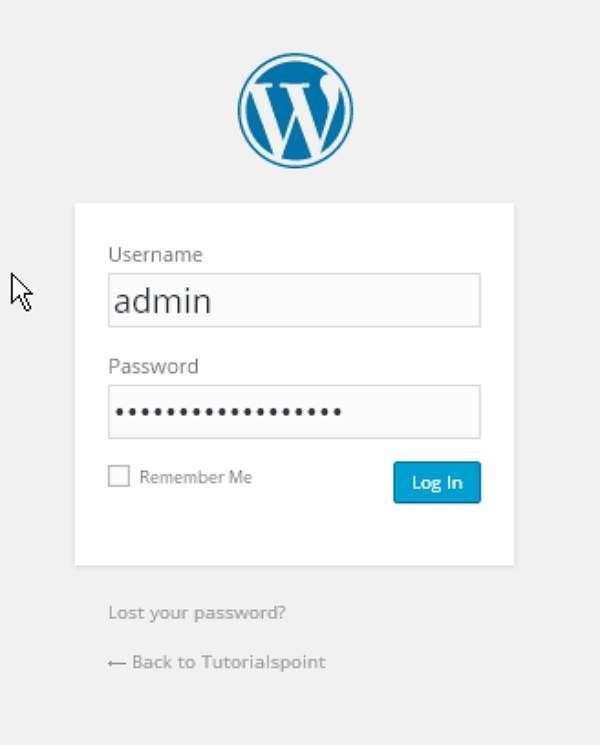
**Step (7):** After installation being successful, you will get a screen of the stating success as seen in the following screen.

You can view your username and password detail added in WordPress.



Click on **Log In** button.

**Step (8):** After clicking on login, you will get a WordPress Admin Panel as depicted in the following screen.



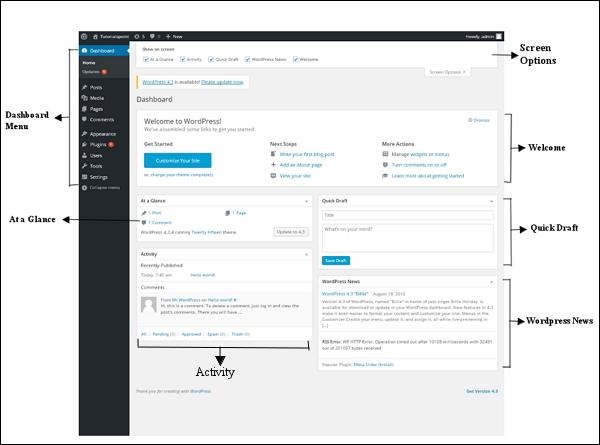
Enter the username and password which you had mentioned during installation as shown in step 6 and click on the **Log In** button.

**Understanding Admin Panel(or) Dash Board**

The WordPress Dashboard is a first screen which will be seen when you log into the administration area of your blog which will display the overview of the website. It is a collection of gadgets that provide information and provide an overview of what's happening with your blog. You can customize your needs by using some quick links such as writing quick draft, replying to latest comment, etc.

WordPress

Dashboard can be categorized as shown in the following snapshot. Each of these categories are discussed in the following sections:



**Dashboard Menu**

The WordPress Dashboard provides navigation menu that contains some menu options such as posts, media library, pages, comments, appearance options, plugins, users, tools and settings on the left side.

**Screen Options**

The dashboard contains different types of widgets which can be shown or hidden on some screens. It contains check boxes to show or hide screen options and also allows us to customize sections on the admin screen.

**Welcome**

It includes the **Customize Your Site** button which allows customizing your WordPress theme. The center column provides some of the useful links such as creating a blog post, creating a page and view the front end of your website. Last column contains links to widgets, menus, settings related to comments and also a link to the **First Steps With WordPress** page in the WordPress codex.

**Quick Draft**

The **Quick Draft** is a mini post editor which allows writing, saving and publishing a post from admin dashboard. It includes the title for the draft, some notes about the draft and save it as a Draft.

**WordPress News**

The **WordPress News** widget displays the latest news such as latest software version, updates, alerts, news regarding the software etc. from the official WordPress blog.

**Activity**

The **Activity** widget includes latest comments on your blog, recent posts and recently published posts. It allows you to approve, disapprove, reply, edit, or delete a comment. It also allows you to move a comment to spam.

**At a Glance**

This section gives an overview of your blog's posts, number of published posts and pages, and number of comments. When you click on these links, you will be taken to the respective screen. It displays the current version of running WordPress along with the currently running theme on the site.

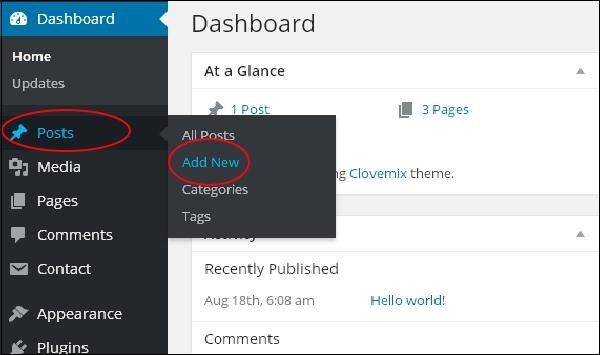
**Working with Posts and Pages**

# WordPress - Add Posts

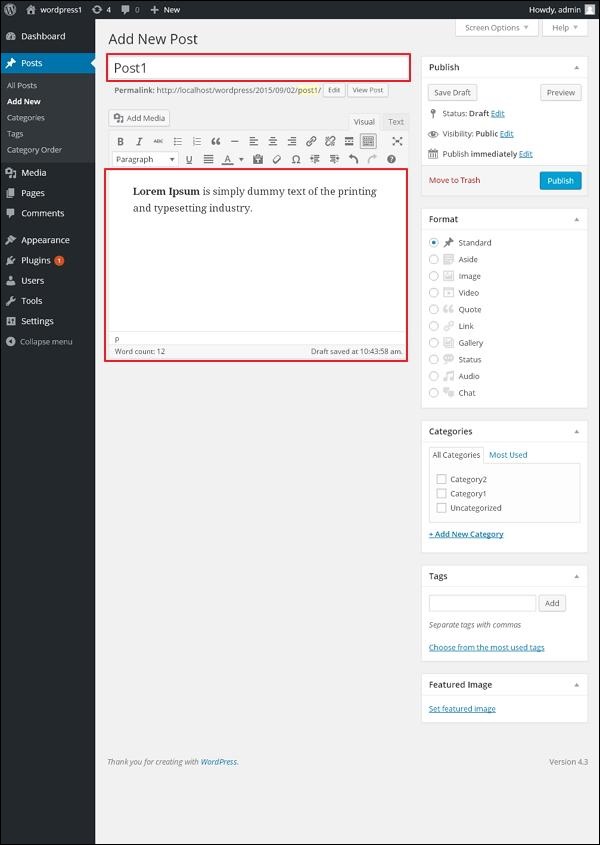
Posts are also known as *articles* and sometimes referred as blogs or blog posts. These are used to popularize your blogs.

Following are the simple steps to Add Posts in WordPress.

**Step (1)** − Click on **Posts → Add New** in WordPress.



**Step (2)** − You will get the editor page of the Post as shown in the following screen. You can use the WordPress WYSIWYG editor to add the actual content of your post. We will study in detail about WYSIWYG editor in the chapter [WordPress - Add Pages](https://www.tutorialspoint.com/wordpress/wordpress_add_pages.htm).



Following are the fields on the editor page of the Add Posts Page.

* **Post Title** − Enter the title of the post, i.e., **Post1**.
* **Post Content** − Enter the content of your post.

**Step (3)** − Click on **Publish** button to publish your respective post.



Following are the few other options present in the Publish section.

* **Save Draft** − It saves the post as a draft.
* **Preview** − You can preview your post before publishing.
* **Move to Trash** − Deletes the post.
* **Status** − Change the status of your post to *Published, Pending,* or *Reviewer Draft*.
* **Visibility** − Change the visibility of the post to *Public, Private* or *Password protected*.
* **Published** − Change the publishded post date and time.

# Delete Pages

Following are the steps to **Delete pages**in wordPress.

**Step (1)** − Click on **Pages → All Pages** in WordPress.

**Step (2)** − You can delete Sample Page (Sample Page is created by default in WordPress). When the cursor hovers on the pages, then a few options gets displayed below the Sample Page. Click on **Trash** option to delete the post.

Or alternatively you can also delete your page directly while editing or adding page by clicking on the **Move to Trash** button as shown in the following screen.

**Step (3)** − To confirm that you have deleted the page, view your page list.

# Edit Posts

Following are the simple steps to Edit Posts in WordPress.

**Step (1)** − Click on **Posts → All Posts** in WordPress.

**Step (2)** − You can view Post1 (Post1 was created in the chapter [WordPress - Add Posts](https://www.tutorialspoint.com/wordpress/wordpress_add_posts.htm)). When the cursor hovers on the Post, few options get displayed below the Post name. There are two ways to edit the Post i.e. **Edit** and **Quick Edit**.

**Edit** − Click on **Edit** option in Post1 as shown in the following screen.

You can edit or change the content or title of the post as per your needs, and then click on **Update** button as shown in the following screen.

**Quick Edit** − Click on **Quick Edit** option in Post1 as shown in the following screenshot.

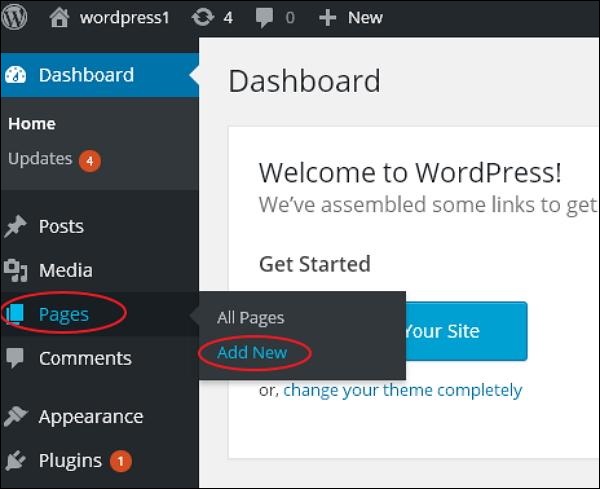
Here you can edit the **Title, Slug** and **date** of the posts and can also select the categories for your post as shown in the following screenshot and then click on **Update** button to confirm post edits.

# WordPress - Add Pages

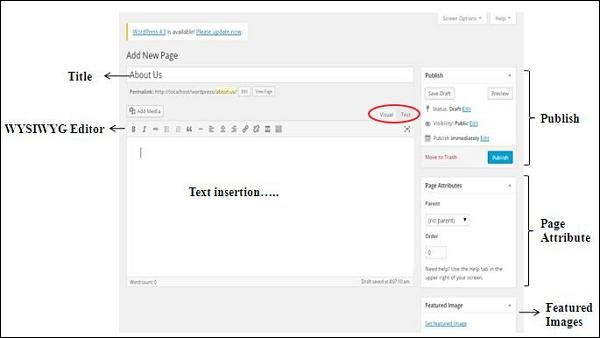
A page is an area on your site where you can display content. A page can be about anything at all. Some of the most common pages on a website include Home, About, and Contact pages. You can add as many pages to your site as you would like, and you can update your pages as many times as you want. After you create a page, you can add it to your site’s navigation menu so your visitors can find it. Adding pages is similar to adding posts in WordPress. Pages are static content and often do not change its displayed information.

Following are the simple steps to add pages in WordPress.

**Step (1)** − Click on **Pages → Add New** as shown in the following screenshot.



**Step (2)** − You will get the editor page as seen in the following screenshot. The editor page has two tabs, Visual and Text. You can insert text in either of these. Here, we'll study about inserting text into Visual format.



Following are the details of the fields on editor page of the **Add New Page**.

**Title**

It is used to write the title of the article, which is later displayed on the page.

**Text Insertion**

For writing the content of an article.

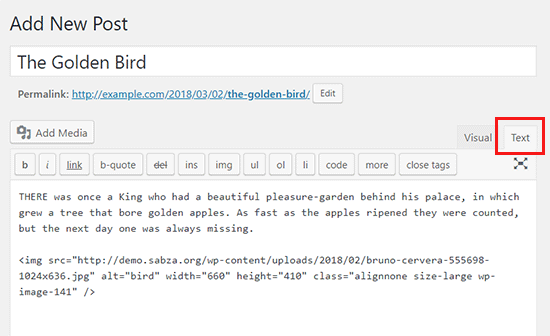
**Publish**

To publish the page to the user on the website.

**Using Editor**

The post edit screen in WordPress comes with two editors to write posts, Visual and Text. Unlike the visual editor, the text editor requires you to add any formatting such as italics, alignment, and spacing manually using HTML.

**Text Editor:-**



The Text editor do have some basic buttons in the bar along the top which quickly insert commonly used HTML elements into the content. Plain text editor is more useful for users who are comfortable with HTML because most formatting is done manually by writing HTML directly into the editor. One bit of formatting the text editor will perform, however, is converting line breaks into paragraphs. This means that every time you start a new line, you are beginning a new paragraph just like the visual editor.

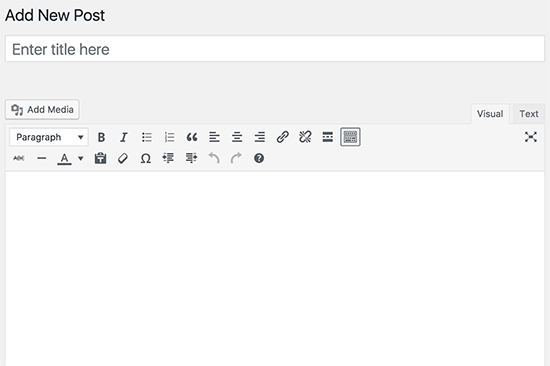
Often advanced users prefer to use the text editor in WordPress instead of the visual editor to prevent WordPress from adding its own styles which it tends to do every now and again.

**visual editor**

Visual Editor is a rich text editor in the WordPress post edit screen. The WordPress post edit screen has two editors, Visual and Text.

The visual editor is a WYSIWG editor. WYSIWIG stands for “what you see is what you get” which means that however the content shows up on your display is exactly the way it will be when it is published. Editors like Microsoft Word and Pages fall into this category as well.

WordPress comes with a modified version of TinyMCE, which is an open source WYSIWYG HTML editor. It mimics the behavior of desktop publishing tools such as Microsoft Word and even has many of their features such as a toolbar along the top allowing you to format your content.



So, instead of writing HTML, users can write posts and format them using the buttons in the toolbar. Since it is a visual text editor when a user adds images TinyMCE can show them right inside the editor. Users can then easily align the images with their text using the editor’s text align buttons.

You can see the toolbar in the image below with all of the formatting options we just mentioned including making font bold and italic, aligning paragraphs, adding quotes, and spell checking. In the top left you can see the “add media” button that allows you to include images and other media in your post while in the top right you can see the button that will let you toggle between the visual and [text editor](https://www.wpbeginner.com/glossary/text-editor/) modes.

**Text Formatting with Shortcuts**

WordPress text patterns, also known as formatting shortcuts, were introduced in version 4.3. They provide an easy way for a WordPress user to format text within the content editor, adding lists, quotes, heading tags, and more.

Text patterns are very similar to [keyboard shortcuts](https://wpsmackdown.com/wordpress-keyboard-shortcuts-writing/), but are utilized in a slightly different way.

In the WordPress visual editor you can use a combination of keys to do things that normally need a mouse, trackpad or other input device. Rather than reaching for your mouse to click on the toolbar, you can use the following keyboard shortcuts:

### [Ctrl + key](https://wordpress.org/documentation/article/keyboard-shortcuts-classic-editor/#ctrl-key)

Letter Action  
c ... Copy  
v ... Paste  
a ... Select all  
x ... Cut  
z ... Undo  
y ... Redo  
b ... Bold  
i ... Italic  
u ... Underline  
k ... Insert/edit link

### [Alt + Shift + key](https://wordpress.org/documentation/article/keyboard-shortcuts-classic-editor/#alt-shift-key)

The following shortcuts use a different key combination: Windows/Linux: “Alt + Shift (⇧) + letter”. Mac: “Ctrl + Option (alt ⌥) + letter”. (Macs running any WordPress version below 4.2 use “Alt + Shift (⇧) + letter”).

Letter Action  
n ... Check Spelling (This requires a plugin.)  
l ... Align Left  
j ... Justify Text  
c ... Align Center  
d ... Strikethrough  
r ... Align Right  
u ... • List  
a ... Insert link  
o ... 1. List  
s ... Remove link  
q ... Quote  
m ... Insert Image  
w ... Distraction Free Writing mode  
t ... Insert More Tag  
p ... Insert Page Break tag  
h ... Help  
x ... Add/remove code tag  
1 ... Heading 1  
2 ... Heading 2  
3 ... Heading 3  
4 ... Heading 4  
5 ... Heading 5  
6 ... Heading 6  
9 ... Address

### [Formatting Shortcuts](https://wordpress.org/documentation/article/keyboard-shortcuts-classic-editor/#formatting-shortcuts)

Formatting Shortcuts while using visual editor (Since [Version 4.3](https://codex.wordpress.org/Version%204.3))

Letter . Action  
\* ...... Start an unordered list  
- ...... Start an unordered list  
1. ..... Start an ordered list  
1) ..... Start an ordered list  
## ..... H2  
### .... H3  
#### ... H4  
##### .. H5  
###### . H6  
> ...... transform text into blockquote  
--- .... horizontal line  
`..` ... transform text into code block

**Working with Media-Adding, editing, deleting media elements**

Media is a tab in your Wordpress admin sidebar. Which is to manage user uploads(images, audio, video and other files). Under the media menu , there are two screens.

The first screen Library, lists all the files in the media library. These files can be edited and deleted from the library.

The second screen is Add New, which allows users to upload files.

Remember users can also upload media (images, videos, etc) while writing a post or page. However, the Add New link under allows users to upload files without attaching them to a specific post or page. **How to Add Media (pdfs, jpgs, etc.) to Posts and Pages:**

1. Log into your WordPress website

2. Click on Media (left navigation) > Add New

You will see a 'Multi-file uploader' you can do the same tasks with the 'Browser uploader' as well, however these instructions are for using the 'Multi-file uploader'. File types you can upload: .jpg, .gif, .png and .pdf.

1. Either:

drag and drop the media files you wish to use on your website to the space inside the dotted lines; OR click on 'Select Files', find the file you wish to upload, and click to upload.

4. (Optional) You can now edit the image by clicking the 'Edit Image' button

5. Fill in the Alternate Text field

This is the text that would appear if a visitor's images are turned off or if a visitor was using a text or screen reader.

6.Take note of the File URL

This is the URL you use to link to the image or file.

7.Click 'Save all changes' to keep your file in the system

You will be directed to the Library page that lists all uploaded media for your website.

**Delete a media file:**

If you no longer wish to have a media file in your system:

1. Click on Media > Library

2. Hover over the name of the file you wish to delete

Three options will appear: 'Edit', 'Delete Permanently and 'View

3. Click on 'Delete Permanently'

A pop-up window will appear.

4. Click on 'Ok

Your file will now be permanently deleted and no longer available or in an archive.

**Edit a media file:**

Following are the steps to Edit Media in WordPress.

**Step (1)** − Click on **Media → Library** and click on the name of the media item or the edit link.

**Step (2)** − You will view a list of media files. Select any one image to edit.

**Step (3)** − You can view the edit media page with few options on the right side.

* **URL** − You can read only a link from media file.
* **Title** − This displays the name of the media. Title is often shown in galleries and attachment pages, if themes or plugins are designed to display it.
* **Permalink** − Permalink is the URL of the media attachment page. This is also a link to view the attachment page.
* **Edit image button** − This allows you to edit the image position, such as rotate counter-clockwise, rotate clockwise, scale, crop, flip vertically, and flip horizontally.
* **Caption** − A brief explanation of the media.
* **Alternate Text** − The alt text for the image, which is used to describe media. Used for availability.
* **Description** − An explanation of your media file.
* **Delete Permanently** − Delete your media file permanently.

**Step (4)** − Once you complete editing the required fields, click on **Update** button to save the changes made to an image.

**Working with Widgets**

Widgets are elements you can use to easily display various types of content within your WordPress sidebar and footer areas. These include text, links, images, and more.

For many users, especially beginners without much technical experience, widgets are incredibly helpful. You can use these pre-built modules to enhance both the design and functionality of your

pages without having to code in new features manually.

Widgets are small blocks of content, and usually aren't the main focus of a page. However, that doesn't mean they don't play essential roles. For instance, you can use them to display the weather, highlight a location on a map, or provide visitors with a convenient navigation menu..

Whereas WordPress themes enable you to change the entire look and feel of your website, widgets let you add and customize specific features to certain areas. WordPress comes with a variety default options:

You can use these widgets for everything from displaying search bars to showcasing a list of recent comments or posts. In addition to these out-of-the-box options, there are also ways to create your own widgets.

Plus, some themes and plugins come with their own unique widgets.

**Adding a Widget:-**

1. On the left-hand admin panel, click on Appearance and select the Widgets option.

Here you can view all of the widgets currently available on your website and the areas you can add them. The areas you can add them are dependent on the theme you are using. Each theme is unique so you may have different options available. However, they all work the same.

1. Simply click on a widget and drag it to the appropriate area.

When you add a widget, you will be presented with options to configure it in a variety of ways. However, each widget is unique. Adding widgets to WordPress is a relatively easy process however, some people may struggle with the drag and drop method.

**Accessibility Mode:-**

First, you must enable accessibility mode. To do this, click on the Enable accessibility mode option in the top right corner of the Widgets area.

You should now notice an "Add" button next to all of the widgets Click on the "Add" button for a widget you want to add.

At the top, you will be able to customize the widget settings. Again, this is unique for each plugin. At the bottom, you will be able to select where to add the widget and the position it takes.

For instance, if you want to add it to your sidebar and a top, you would pick the sidebar and position 1. Click on the "Sa Widget" button when you are done.

While in this mode, you can click on the "Edit" button at any time to make changes to the widget.

**Removing Widgets:-**

It is very common to regularly change the widgets you are using, as such, you are going to need to know how to remove widgets so you can make room for new ones. Luckily, this is equally as easy.

If you are using the default Widget settings, simply expand the widget settings and click on the Delete option.

If you are using accessibility mode,click on the Edit button and then click on the “Delete” option.

**Menus**

A WordPress menu is a list of defined links to pages on our website. Commonly placed in the header section of websites, it lets visitors navigate our website with ease. With menus, we can direct our visitors to a specific post or page. Moreover, we can even provide custom links through our menus.

**Defining a Menu:**

we must define a menu before we can add items to it.

1. Login to the WordPress Dashboard.

2. From the 'Appearance' menu on the left-hand side of the Dashboard, select the 'Menus' option to bring up the Menu Editor.

3. Select Create a new menu at the top of the page

4. Enter a name for our new menu in the Menu Name box

5. Click the Create Menu button.

Our new custom menu' has now been defined.

**Adding Items to a Menu:**

1. Locate the pane entitled Pages.

2. Within this pane, select the View All link to bring up a list of all the currently published Pages on our site.

3. Select the Pages that we want to add by clicking the checkbox next to each Page's title.

4. Click the Add to Menu button located at the bottom of this pane to add our selection(s) to the menu that we created in the previous step.

5. Click the Save Menu button once we've added all the menu items we want.

Our custom menu has now been saved.

**Deleting a Menu Item:**

1. Locate the menu item that we want to remove in the menu editor window

2. Click on the arrow icon in the top right-hand corner of the menu item/box to expand it.

3. Click on the Remove link. The menu item/box will be immediately removed.

4. Click the Save Menu button to save our changes.

**Unit – V**

**Working with themes**-parent and child themes, using featured images, configuring settings, user and user roles and profiles, adding external links, protecting word press website from hackers.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**Working with themes-Parent and Child Themes( Theme Management)**

A WordPress theme is a tool to change the layout and design of your website. Themes customize the appearance of your site, including the layout, typography, color, and other design elements.

A theme is a zipped folder with a group of files, including a set of page templates written in the PHP programming language combined with some CSS stylesheets.

When you first create a WordPress site, it will come with a default theme installed. You can change the theme in WordPress to match your needs and industry.

Choosing the right theme can make your website more attractive, easier to use, and improve visitor engagement.

A WordPress theme is a folder of files that work together to create the design of your site. A theme includes template files, stylesheets, images, and possibly JavaScript files. All those files change how your blog posts and pages are displayed.

Changing your WordPress theme won't necessarily change your blog content, pages, users, or other information stored in your database. Your theme only changes how all that information is displayed to your website visitors.

There are thousands of free and premium WordPress themes vailable for you to use. theme Free themes can be downloaded from the official WordPress directory, or you can purchase premium themes with more advanced features from third-party developers. You can even create your own custom themes for a unique web design.

**Most Popular WordPress Themes:**

The best and most popular WordPress themes available for download are:

**Divi:** A multi-purpose theme by Elegant Themes that's customizable with a drag-and-drop builder

**Astra**: A customizable theme that comes with pre-designed demos so you can get started in minutes

**Genesis by StudioPress**: A clean and fast theme framework that you can style with child themes

**Ocean WP:** A lightweight and customizable theme that's 100% free and great for beginners

All of these themes are easy to use, well-supported, and have dozens of beautiful layouts. They're also all considered multi-purpose themes, which means they'll work well for any type of website. You can use them for personal blogs, business sites, eCommerce sites, and more.

**Discuss about working with themes- parent and child themes?**

It includes image files, templates, CSS stylesheets, etc. that can help to make your website look great.

**Parent Theme:**

* A parent theme has all the WordPress template files and asset required for the theme to work. In short, a parent theme is independent of any other theme and has a collection of all the assets and files required to display a theme.
* Originally WordPress had just the parent theme but later with time, came the feature of creating a child theme to make it easy for a user or developer to make some small changes in the parent theme.

**Child Theme:**

* A parent theme is independent of any other theme but a child theme is not.
* As per the name, a child theme is a copy of the parent theme and doesn't contain all the files and therefore is dependent on the parent theme to work.
* The child theme was not originally present in WordPress, but it was added later to help users or developers customize the parent theme easily. A child theme generally has two files - 'style.css and functions.php'.
* You can customize the child theme, make changes in style.css or functions.php files according to your needs and interest and your parent theme will remain unchanged.
* This helps developers and designers to customize the theme and still retain the original theme template.

**Adding New Theme:**

You can download any theme from WordPress Theme Directory directly to your site by using the Add New option in the Appearance sub-menu.

1. Log in to the WordPress Administration Screens.

2. Select the Appearance screen, then Themes.

3. Select Add New.

4.Either use the Search or Filter options to locate a Theme you would like to use.

5. Click on the Preview link to preview the Theme or the Install link to upload the Theme to your site.

Or use the Upload Theme button at the top of page to upload a zipped copy of a Theme that you have previously downloaded to your machine.

**Using Featured Images**

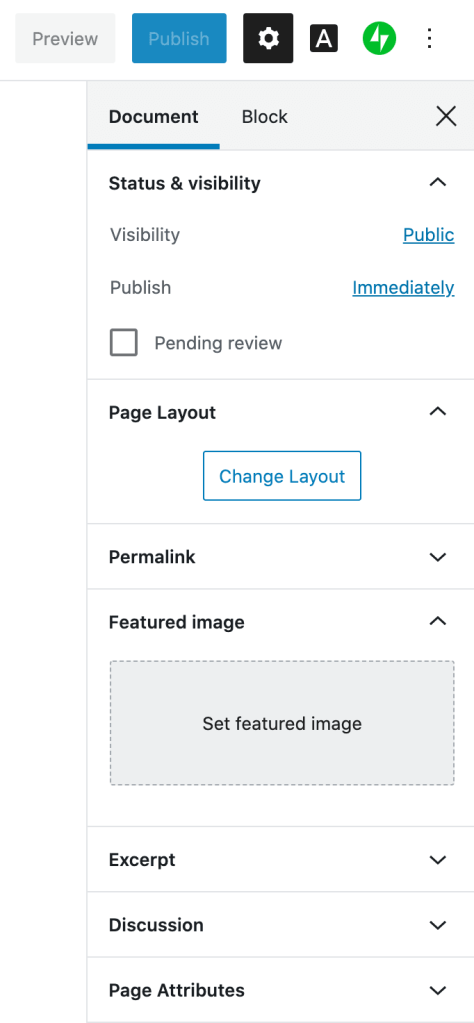
A featured image represents the contents, mood, or theme of a post or page. Posts and pages can have a single featured image, which many themes and tools can use to enhance the presentation of your site.

#### Set a Featured Image

To set a featured image on a post or page, follow these steps:

1. Visit your [dashboard](https://wordpress.com/home).
2. Click on Pages or Posts from the options on the left.
3. Click on the title of the page or post to open the editor.
4. In the settings on the right, locate the **Featured Image**section. If you do not see the settings area, you may need to click on the  icon in the top right corner to make it appear.
5. Click **Set** **Featured Image** as shown in the image on the right.
6. You will then see options to choose an image from your site’s Media Library, Google Photos, [Pexels Free Photos](https://wordpress.com/support/free-photo-library/), or [Openverse](https://wordpress.com/support/openverse/).
   * If you want to upload an image from your computer, choose Media Library and then click the **Upload files** tab.
7. After selecting your image, click the button **Set Featured Image** at the bottom right.
8. Click **Update** on your post/page to save your changes. Your featured image is now set!

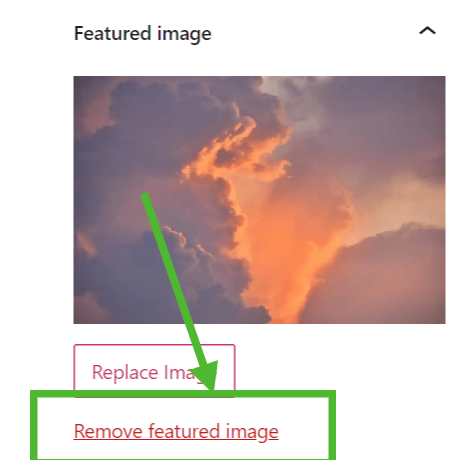
[Optional] If your theme does not display the featured image by default, add a **Cover block** to the top of the page/post. You can then [click one button](https://wordpress.com/support/wordpress-editor/blocks/cover-block/#show-the-featured-image-in-a-cover-block) to display the featured image in the **Cover block.**



Certain themes use featured images in other ways that may require a specifically sized image to work properly. Please see the documentation for your theme for those exact dimensions.

#### Remove a Featured Image

To remove a featured image, follow steps 1 to 4 in the previous section to locate the featured image. Then, click the **Remove featured image** link:



You can also change the featured image by clicking the **Replace Image** button

**Configuring Settings**

The **Settings** tab in WordPress Admin sidebar is the central hub to configure settings for different sections of a WordPress website. It contains multiple sub-panels and many WordPress plugins also add their settings page as a menu under the Settings tab. Clicking on it takes users to the **Settings » General** screen. Other default sub panels under the settings tabs are Reading, Writing, Discussion, Media, and Permalinks.

Only a user with the administrator user role has the access to the settings tab.

**General** contains settings such as Site Title, Tagline, WordPress URL, Site URL, E-mail, Registration option, and many other general options for the WordPress site.

**Writing** contains settings related to the post formatting and category options.

**Reading** contains settings related to Site Front Page, How many posts to show per page to readers, RSS items to be shown, whether full text or summary should be shown, and an option to discourage search engines from indexing the website.

**Discussion** contains settings related to linking posts with other blogs that are referenced, comment settings, e-mail notifications for comments, and options for comment moderation.

**Media** contains settings related to upload and management of images and other media within WordPress. Things like image sizes and cropping options can be managed here.

**Permalinks** contains settings related to URL structure of the website. These settings make it easier for users to find and remember URLs to specific posts. Using the correct structure will also help to better organize posts within the various categories. Options in this setting allow you to control whether you want dates reflected in your URLs or just titles and categories, or some combination of both.

**User and User Roles and Profiles**

WordPress is a powerful, flexible [Content Management System (CMS)](https://www.dreamhost.com/blog/history-of-wordpress/) that can be an excellent solution for collaboration. However, to make the most of the CMS, it’s important to understand how to navigate and leverage its [user roles and permissions](https://wordpress.org/support/article/roles-and-capabilities/) features.

WordPress user roles let you assign certain levels of access to people who are registered to your website. This can help you manage and control what tasks are possible and can ultimately help strengthen your site’s security and performance.

Both user roles and permissions are set by the Administrator, which is typically the WordPress site owner. By default, there are six different user roles: Super Admin, Administrator, Editor, Author, Contributor, and Subscriber. Each role has its own set of capabilities, which we’ll discuss in more detail below.

### Super Admin

The Super Admin is the highest level of user on a WordPress site. This user has complete control over the site, including the ability to add and delete users, install and activate plugins, manage themes, and more. Super Admins are typically only found on multisite installations of WordPress.

Super Admins can manage every setting and feature for each site within a multi-site network. They can add and delete other Administrators, create new sites, and [control content](https://www.dreamhost.com/blog/how-to-create-content-marketing-strategy/) across each site.

### Administrator

Administrators have complete control over a single WordPress site. They can add and delete users, [install and activate plugins](https://www.dreamhost.com/blog/wordpress-plugin-primer/), manage themes, etc. Usually, they are the site owners or main authors:

This powerful role has complete access to content, features, and site settings. They can update the CMS as well as plugins [and themes](https://www.dreamhost.com/blog/how-to-pick-wordpress-theme/). The Admin is also responsible for assigning user roles and capabilities to other registered users. Ideally, you should only have one Administrator per website.

### Editor

Editors can manage and publish posts and pages, as well as moderate comments. They can also [schedule content](https://www.dreamhost.com/blog/how-to-schedule-posts-in-wordpress/) and edit categories. However, they cannot install or activate plugins, or manage themes:

In a nutshell, an editor can [modify content](https://www.dreamhost.com/blog/keep-your-blog-fresh-how-to-repurpose-old-posts/) created by themselves and other users with a lower status, such as Authors and Contributors. They can’t change content for users with permissions higher than theirs, such as an Administrator. Typically, this role is reserved for content managers or similar titles.

### Author

As you may have guessed, authors can write and publish their own posts and pages. They can also delete their own posts.  However, they cannot publish, edit, or delete anyone else’s posts. Additionally, authors cannot add or delete users, install or activate plugins, or manage themes:

Unlike Contributors, Authors have access to the WordPress Media Library. While they can edit reader comments, they can only do so on their own posts.

### Contributor

[WordPress Contributors](https://www.dreamhost.com/blog/become-a-wordpress-contributor/) can write and submit their own posts for review by an Administrator or Editor. Once a post is published, they cannot edit it. Furthermore, contributors cannot add or delete users, install or activate plugins, or manage themes.

Contributors are usually roles assigned to freelance writers or guest [bloggers](https://www.dreamhost.com/blog/wordpress-blogger-checklist/). This role is also commonly used for new hires whose content needs editing or reviewing before it can be published on the site.

Once submitted for review, only the Editor or Administrator can publish their posts. Contributors cannot access the Media Library.

**Subscriber**

Subscribers can manage their own profiles and read posts and pages on a WordPress site. They cannot write or publish their own posts or pages, nor can they add or delete users, install or activate plugins, or manage themes:

Subscribers have the fewest permissions and capabilities of all the WordPress roles. It is the default user role set for new registrations.

There are a few additional user role options available on some WordPress sites. For example, if you’re [running a WooCommerce site](https://www.dreamhost.com/blog/woocommerce-online-store-in-an-hour/), Shop Managers have similar capabilities to Administrators, but with some added features specifically for managing WooCommerce stores. For instance, they can add and delete products, manage orders, and more.

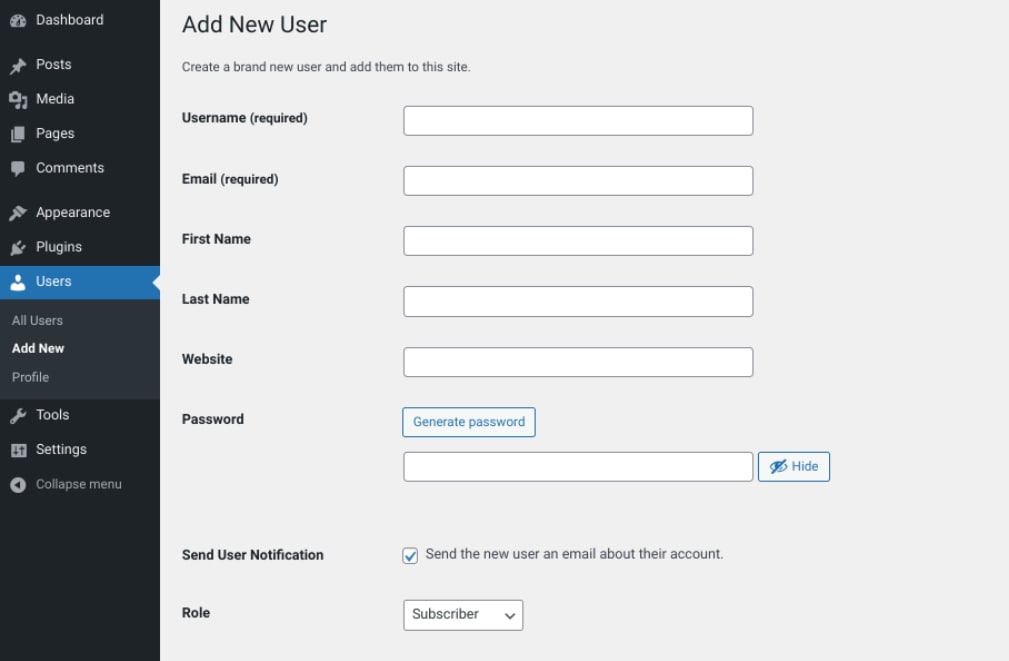
**How to Manage User Roles in WordPress**

Now that you have a better sense of what each user role can do, let’s get into how to manage them. Below, you’ll find instructions for how to add, delete, and update users and user roles in WordPress.

**1. Creating and Deleting Users in WordPress**

Before you assign a user role in WordPress, you first need to have a user to attach it to. To add a new user in WordPress, you can navigate to *Users > Add New*, then fill in the information. This will include details such as username, email, and password:

This will include details such as username, email, and password:



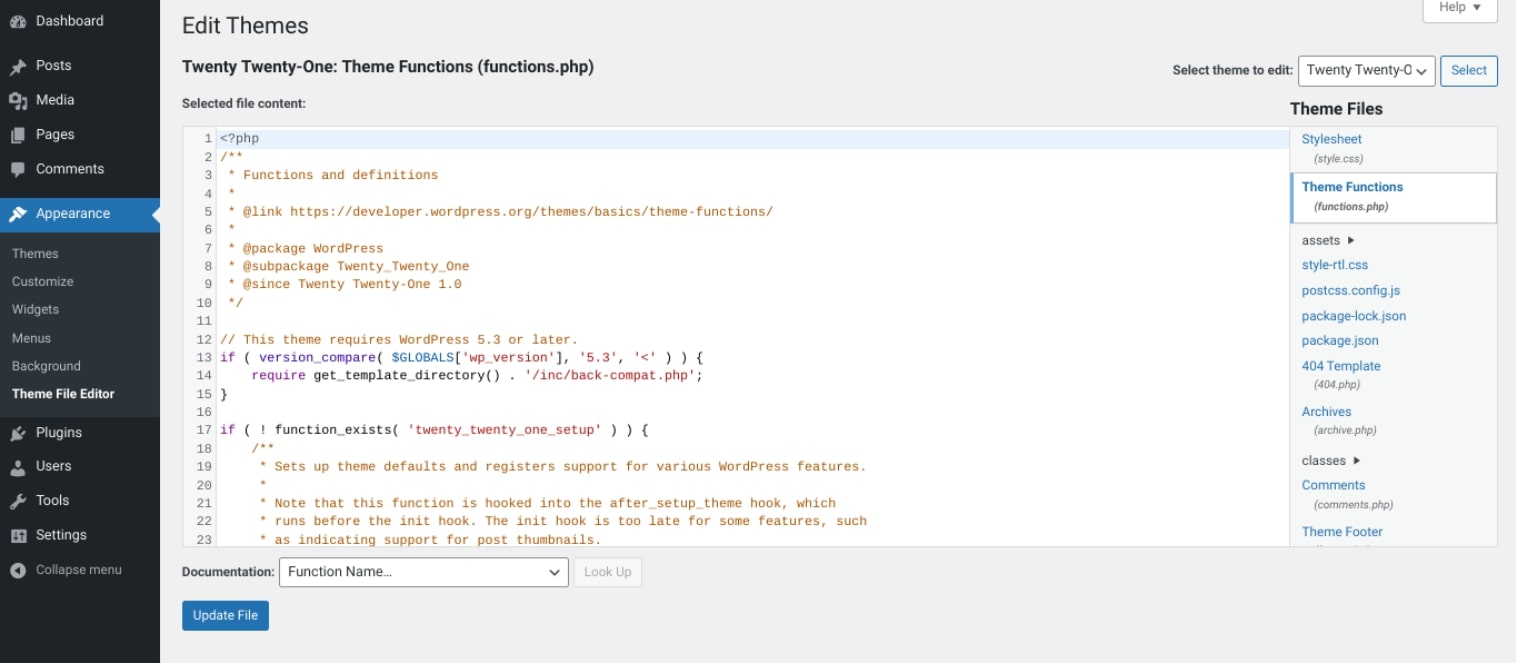
Note that, by default, the *Role* is automatically set to *Subscriber.* When you’re done, you can click on the *Add New User* button at the bottom of the screen.

Alternatively, you can create a new user through your database. To do this, you can navigate to phpMyAdmin from your cPanel dashboard (or whichever system your host uses), then select your WordPress database.

### Deleting a User Role in WordPress

If you want to delete a user role in WordPress so that it is no longer an option, you can do so by [editing your theme’s files](https://developer.wordpress.org/themes/basics/theme-functions/). Keep in mind that modifying theme files can be risky, so it’s best to [create a backup](https://www.dreamhost.com/blog/10-important-reasons-to-perform-website-backup/) of your site before you continue on.

To get started, go to *Appearance > Theme File Editor* in your WordPress dashboard. Next, locate and open the **Theme Functions** file:



In this file, you can add one (or all) of the following code snippets, depending on which user role(s) you want to remove:

remove\_role( 'subscriber' );

remove\_role( 'editor' );

remove\_role( 'contributor' );

remove\_role( 'author' );

When you’re done, select the Update*File*to save your changes.

**Adding External Links**

An external link icon is a small picture that appears next to a link that will take the user to a different website. By using external link icons on your [WordPress blog](https://www.wpbeginner.com/start-a-wordpress-blog/), your visitors will easily be able to tell the difference between outbound links and internal links. Many sites also use external link icons to make it clear that the link will open in a new window or tab.

These icons can keep visitors on your site for longer and [increase your pageviews](https://www.wpbeginner.com/beginners-guide/how-to-increase-pageviews-and-reduce-bounce-rate-in-wordpress/) since there’s less risk of them clicking on an external link and leaving your website by accident.

With that being said, let’s see how you can add an external link icon to your WordPress site.

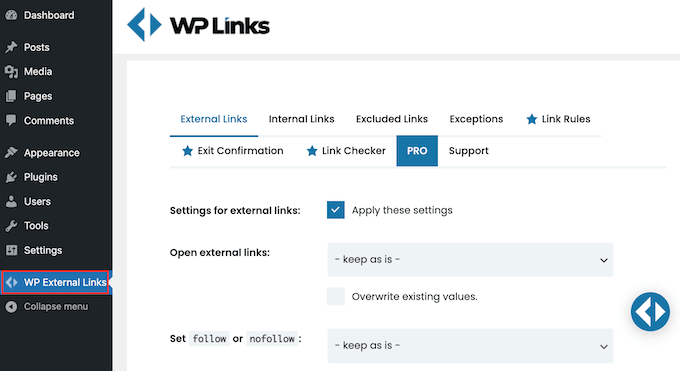
#### ****How to Add an External Link Icon on Your WordPress Site****

The easiest way to add an external link icon to your website is by using [WP External Links](https://wordpress.org/plugins/wp-external-links/).

You can use this plugin to add different images, [Dashicons](https://www.wpbeginner.com/wp-themes/how-to-easily-add-icon-fonts-in-your-wordpress-theme/" \o "How to Easily Add Icon Fonts in Your WordPress Theme), and Font Awesome icons to your external links automatically.

First, you’ll need to install and activate the [WP External Links](https://wordpress.org/plugins/wp-external-links/) plugin. If you need help, then please see our guide on [how to install a WordPress plugin](https://www.wpbeginner.com/beginners-guide/step-by-step-guide-to-install-a-wordpress-plugin-for-beginners/).

After that, just click on the new **WP External Links** setting in the left-hand menu.

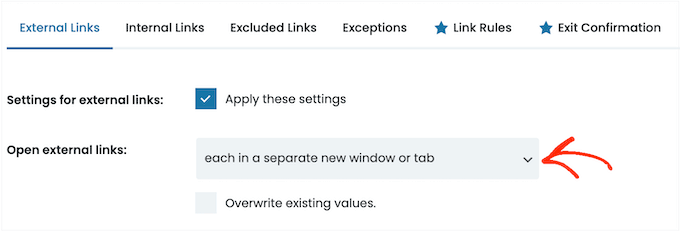


If you get a popup asking you to upgrade, then you can simply click anywhere outside the popup to close it.

As you can see, the plugin can do a lot more than just add an external link icon to your site. For example, it can automatically [add nofollow to all external links in WordPress](https://www.wpbeginner.com/plugins/how-to-nofollow-all-external-links-in-wordpress/).

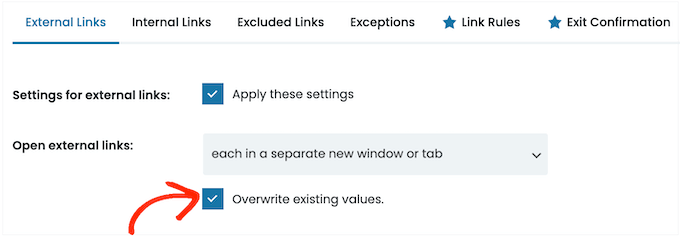
WP External Links can also keep people on your site by [opening all external links in a new window or tab](https://www.wpbeginner.com/plugins/how-to-open-external-links-in-a-new-window-in-wordpress/). We recommend enabling this feature since many users assume links with an external link icon will open in a new window or tab, anyway.

To open all external URLs in a new window or tab, open the ‘Open external links’ dropdown and click on ‘each in a separate new window or tab.’



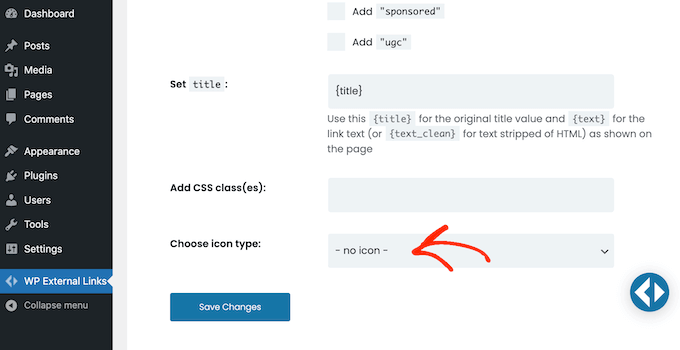
By default, WP External Links will apply this rule to every new external link that you create.

However, it can also scan your site and change any external URLs that you’ve previously added to your website. This will provide a more consistent experience for your users, so it’s a good idea to go ahead and click on the ‘Overwrite existing values’ box.



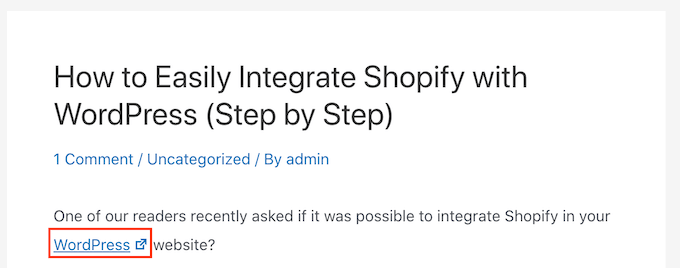
Once you’ve done that, you’re ready to create an icon for your external links. To start, scroll to the ‘Choose icon type’ section.

Now you can click where it says ‘no icon’ to open the dropdown.



Your options are Image, Font Awesome, or Dashicon.

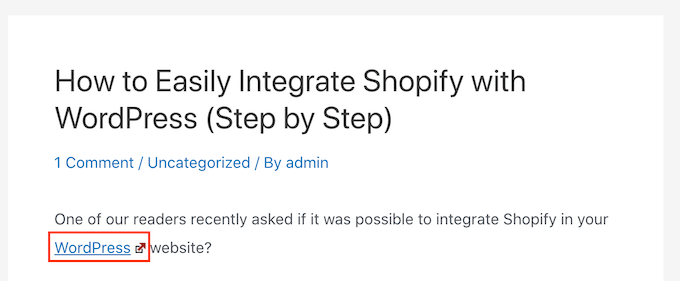
Both Font Awesome and Dashicon come with a single box-with-arrow icon that you can use as your external link icon. This icon will always appear blue on your site, as you can see in the following image.



This is the exact icon that many websites use for their external links, so most visitors will understand what the box-with-arrow icon means.

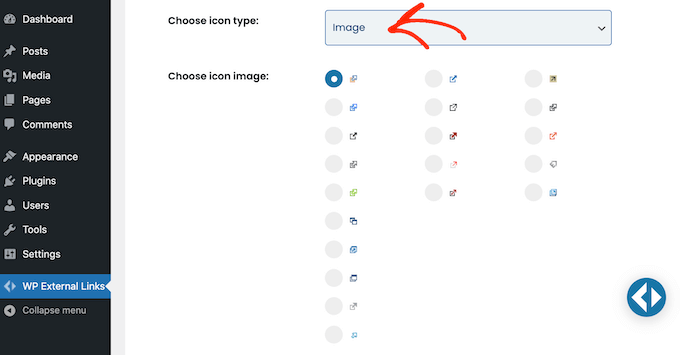
However, ‘Image’ lets you choose between several box-with-arrow icons including some that have a slightly different style and color.

In the following image, you can see an example of how a colorful box-with-arrow icon will look on your [WordPress website](https://www.wpbeginner.com/guides/).



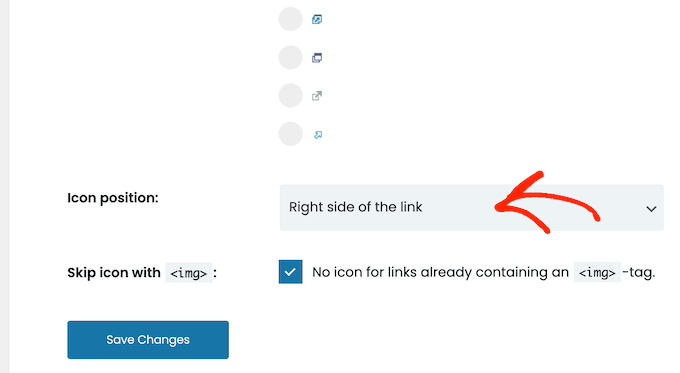
If you want to add a colorful icon to your site, then you’ll need to select ‘Image’ from the ‘Choose icon type’ dropdown.

You can then click on the one that you want to use for your external links.



Once you’ve done that, open the ‘Icon position’ dropdown.

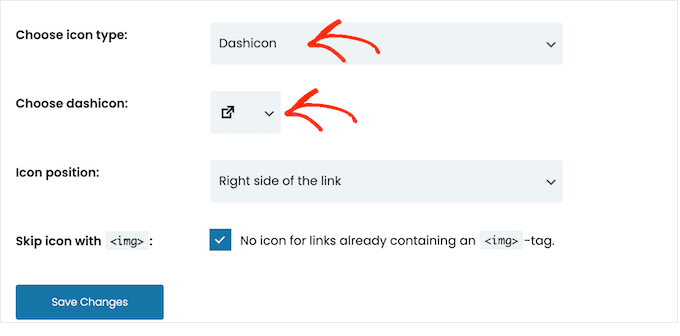
Here, choose whether the icon will appear on the ‘Right side of the link’ or the ‘Left side of the link.’ Most websites show the external icon to the right of the link.



Do you want to use the standard blue box-with-arrow icon instead?

Then you can open the ‘Choose icon type’ dropdown and click on either ‘Font Awesome’ or ‘Dashicon’ instead.

Next, open the dropdown menu beside ‘Choose Dashicon’ or ‘Choose FA,’ and then click on the box-with-arrow icon.



**Note:** Both Font Awesome and Dashicons give you access to dozens of other icons. However, we recommend using the box-with-arrow icon as more people will recognize this as an external link icon.

**Protecting Word Press website from Hackers**

The solution is to follow the WordPress security tips to keep your WP website secure from hackers.

**1. Keeping your WordPress website, theme and plugin updated:** The WordPress web application has to keep update and maintain on a regular basis. They have an update for automatically from WP admin console and also manually.

**2. Change your admin user name:** Change your default admin username to stop the brute force attack.

**3. Disable your Appearance > file editing:** After disable file editing no one can edit your theme files from admin console.

**4 Strong passwords and user permission**: Hacker can easily steal the password. You need to provide some strong password and characteristics must be unique.

**5. Keep secure WordPress hosting:** This is one type of good shared hosting providers. Please read reviews about secure WP hosting companies and select one of them for your website. However, this hosting will provide a more secure platform to the websites.

**6. Install a WordPress backup services or software:** For any WordPress attack, backup is the main thing. The backup will help to quickly restore your data if you have lost it.

**7. By installation of WordPress securities plugins:** you need to be install some WP security plugin that can keep your business or blog WP website safe and secure.

**8. Protect Your WordPress Admin Area:** You can protect your admin section for your IP address only.

**9. Limit login attempts:** You may use the Login LockDown plugin to limit the login attempts for admin user. **10. Change WordPress database prefix:** The default databas WordPress prefix is WP.

**11. Disable directory browsing and indexing**: By the directory browsing hackers are able to view known vulnerabilities file on your hosting server. It is also help in other people about your files and images on your website

**12. Logout automatically logged in user after some time** : You can use the plugin Idea User logout for sign out of inactive user.