D.N.R. COLLEGE (AUTONOMOUS): BHIMAVARAM

DEPARTMENT OF COMPUTER SCIENCE

WEB INTERFACE DESIGNING TECHNOLOGIES

V SEMESTER PAPER-6

WEB INTERFACE DESIGNING TECHNOLOGIES

UNIT-I

WEB DESIGN:

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.

Two of the most common methods for designing websites that work well both on desktop and mobile.

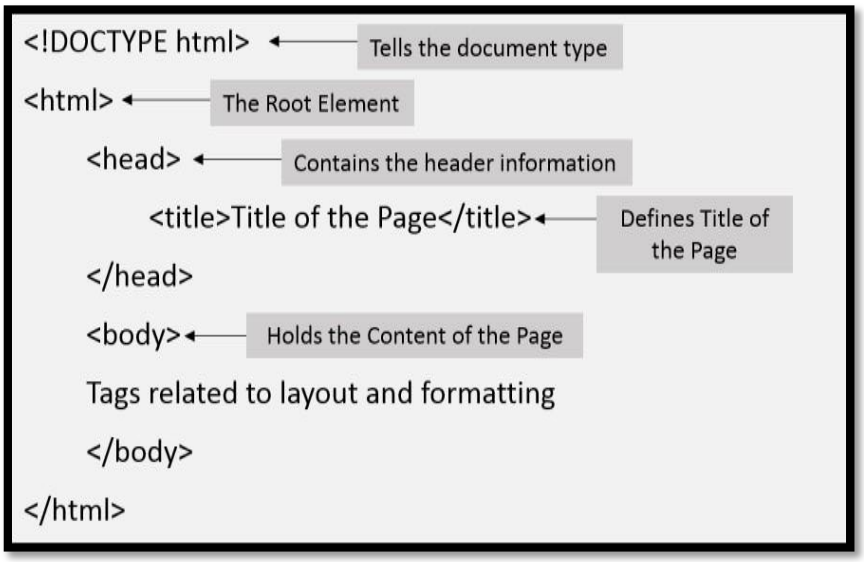
HTML: (HTML is Hyper Text Markup Language.)

- Hyper Text is the method by which you move around on the web –by clicking on special text calledHyper Links which bring you to the next page.
- The fact that it is Hyper just means it is not linear

  -i.e. You can go to any place on the internet whenever you want by clicking on links.
- Markup is what HTML tags do to the text inside them.
- HTML is a language for describing web pages.
- HTML documents contain HTML tags and plain text. HTML documents are also called web pages.
- HTML markup tags are usually called HTML tags
- HTML tags are keywords (tag names) surrounded by angle brackets like <html> HTML tags normally come in pairs like and
- The first tag in a pair is the start tag, the second tag is the end tag
- The end tag is written like the start tag, with a forward slash before the tag name Start and end tags are also called opening tags and closing tags.

STRUCTURE OF AN HTML DOCUMENT:

All HTML documents follow the same basic Structure. The general structure of HTML document (page) is asthe following:



The entire document is surrounded by <html> . . . </html> .The body contains the content that displays on the screen. The tags control the formatting of content by the browser.

The <! DocType> tag: This tag is optional. This tag tells the browser the HTML version with which the documentcomplies.

The <HTML> tag: This tag comes after the <!DocType> tag and identifies the document as an HTML document. It should be the first tag in every HTML document. All other tags include with in <HTML> AND </HTML> tags. The<Head> tag: This tag contains information about the document, including its title, scripts used, style definitionsand document descriptions. The portion included between the <head> and</head> tags is called the header.

The <title > tag: The title does not appear within the browser window, since it is visible in the browsers title bar. The title is given between the tags <title> and </title>.Only one title is allowed per page.

The <body> tag: This tag encloses all tags, attributes and information that one wants the browser to display. Touse the <body> tag, enter below the closing </head> tag and above the closing the </Html> tag.

When an HTML document is complete with all structure tags. it can be saved notepad by selecting the save optionfrom file menu and saving the document with Html extension.

Ex:

```
<html>
<head>
<title>welcome to html</title>
</head>
```

<body>

This is my first html program

</body>

</html>

Output:

HYPER LINKS:

The HTML <a> tag defines a hyperlink. A hyperlink (or link) is a word, group of words, or image that you can click on to jump to another document. When we move the cursor over a link in a Web page, the arrow will turn into a little hand. The most important attribute of the <a> element is the href attribute,qazs indicates the link's destination. By default, links will appear as follows in all browsers:

- An unvisited link is underlined and blue
- A visited link is underlined and purple
- An active link is underlined and red.

HTML Link Syntax

The HTML code for a link is simple. It looks like this:

 Link text

The href attribute specifies the destination of a link.

Ex: Visit 3Schools Which will display like this: Visit google.co.inClicking on this hyperlink will send the user to Google's homepage.

Note: The " Link text " doesn't have to be text. It can be an image or any other HTML element. HTML Links - the target Attribute :

The target attribute specifies where to open the linked document. The following example will open the linkeddocument in a new browser window or a new tab:

Visit Google**Ex:** Linking to Other Pages

Step:1 Open the Notepad and type the following code and save it as "first.html ".

<html>

<head>

<title>First Page</title>

```
</head>
<body>
<h3 >Linking to another page</h3>
<p ><b> DNR College Bhimavaram</b>
</p>
<a href="second.html">GoTo Second</a>
</body>
</html>
```

Step:2 Open the Notepad and type the following code and save it as “second.html ”.

```
<html>
<head>
<title>Second Page</title>
</head>
<body>
<h3 >Linking to another page</h3>
<p >BSc/ BSc Computers
<br>B.com / B.com computer applications
<br>B.A
</p>
<br><a href="first.html" >Go To first</a>
</body>
</html>
```

Output:

LISTS IN HTML:

HTML supports three types of lists. They are

1. Ordered Lists
2. Unordered Lists
3. Definition Lists

HTML List Tags:

Tag	Description	Syntax
	Defines an Order List	<ol [type="1"|"a"|"A"|"I"|"i"] [start="n"] [compact]. . .
	Defines an Unordered List	<ul [type="disc"|"square"|"circle"]>. . .
	Defines a list item	. . .
<dl>	Defines a description List	<dl [compact]. . .</dl>
<dt>	Defines a term/name in a description list	<dt>. . . </dt>
<dd>	Defines a description of a term/name in a description list	<dd>. . .[</dd>]

Ordered Lists:

An ordered list can be numerical or alphabetical. The <ol> tag defines an ordered list. The <li> tag is used todefine list items.

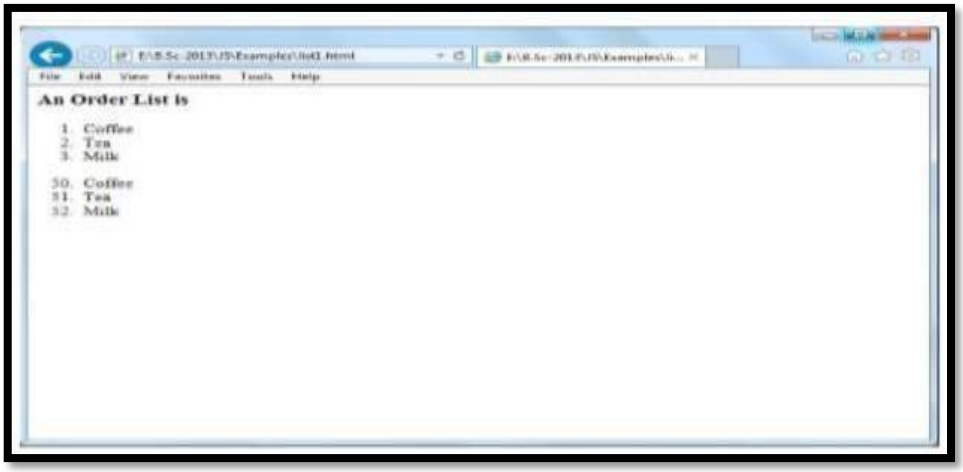
Ex:

```
<html>
<body>
<h3> An Order List is</h3>
<ol>
<li>Coffee</li>
<li>Tea</li>
<li>Milk</li>
</ol>
<ol start="50">
<li>Coffee</li>
<li>Tea</li>
<li>Milk</li>
```


</body>

</html>

Output:



Unordered List :

An unordered list is a list of items. The list items are marked with bullets (typically small black circles). Anunordered list starts with the <ul> tag. Each list item starts with the

 tag.Ex:

<html>

<body>

<h4>An Unordered List:</h4>

Coffee

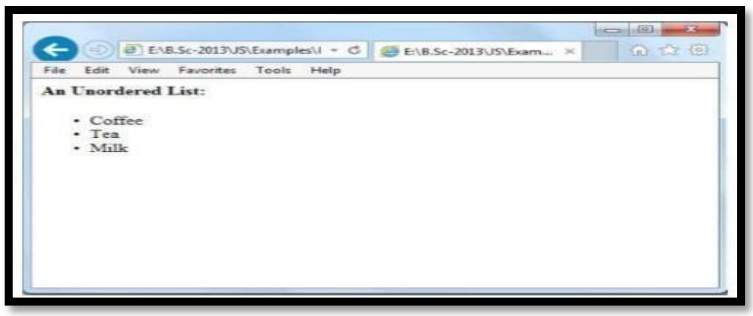
Tea

Milk

</body>

</html>

Output:



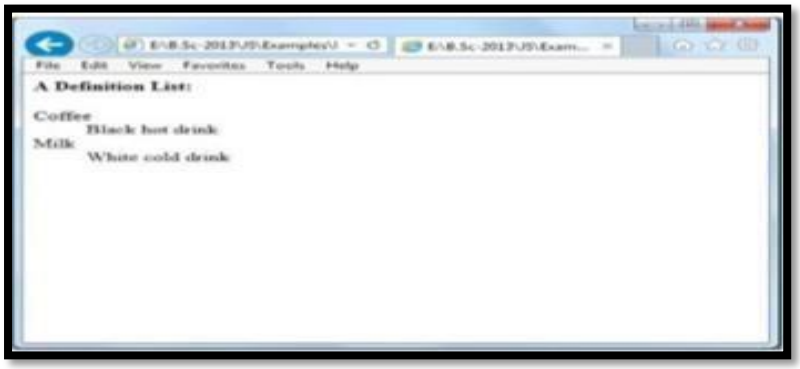
Definition Lists:

A definition list is not a list of single items. It is a list of items (terms), together with a description of each item (term). A definition list starts with a <dl> tag (definition list).Each term starts with a <dt> tag (definition term). Eachdescription starts with a <dd> tag (definition description).

Ex:

```
<html>
<body>
<h4>A Definition List:</h4>
  <dl>
    <dt>Coffee</dt>
      <dd>Black hot drink</dd>
  <dt>Milk</dt>
     <dd>White cold drink</dd>
  </dl>
</body>
</html>
```

Output:



FRAME TAG IN HTML WITH EXAMPLE:

Frames allow for multiple . html documents to be displayed inside of one browser. Each HTML document is calleda frame , and each frame is independent of the others. The <frame> tag defines one particular window (frame) within a <frameset>.

<frame> in a <frameset> can have different attributes, such as border, scrolling, the ability to resize, etc.

The disadvantages of using frames are :

- The web developer must keep track of more HTML documents.
- It is difficult to print the entire page.
- Users often dislike them.
- It presents linking challenges.

People often use frames to wrap their own ads and branding around other people's content.

Syntax:

<frameset [cols=”%,%] [rows=”%,%]>. . .</frameset>

<frame [name=”name”] src=”filename” [scrolling=”yes”|”auto”|”no”] [frame border=”0” |”1”]>

cols: Specifies the number and size of columns in a frameset .

rows: Specifies the number and size of rows in a**frameset. name:** Specifies the name of a frame .

src: Specifies the URL of the document to show in a frame.

scrolling: Specifies whether or not to display scrollbars in a frame . frameborder : Specifies whether or not todisplay a border around a frame.

Ex:

Step 1: Open the Notepad and type the following code:

<html>

<head>

<title>Frame A</title>

</head>

<body bgcolor=”blue”>

<h1 align=”center”>This is FRAME A</h1>

</body>

</html>

***** Save the above program as “frameA.html”*****

Step 2: Open the Notepad and type the following code:

<html>

<head>

<title>Frame B</title>

</head>

<body bgcolor=”pink”>

<h1 align=”center”>This is FRAME B</h1>

</body>

</html>

*****Save the above program as “frameB.html”******

Step 3: Open the Notepad and type the following code:

<html>

<head>

<title>Frame C</title>

</head>

<body bgcolor=”red”>

<h1 align=”center”>This is FRAME C</h1>

</body></html>

***** Save the above program as “frameC.html”*****

Step 4: Open the Notepad and type the following code:

<html>

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

<frame src="frameA.html">

<frame src="frameB.html">

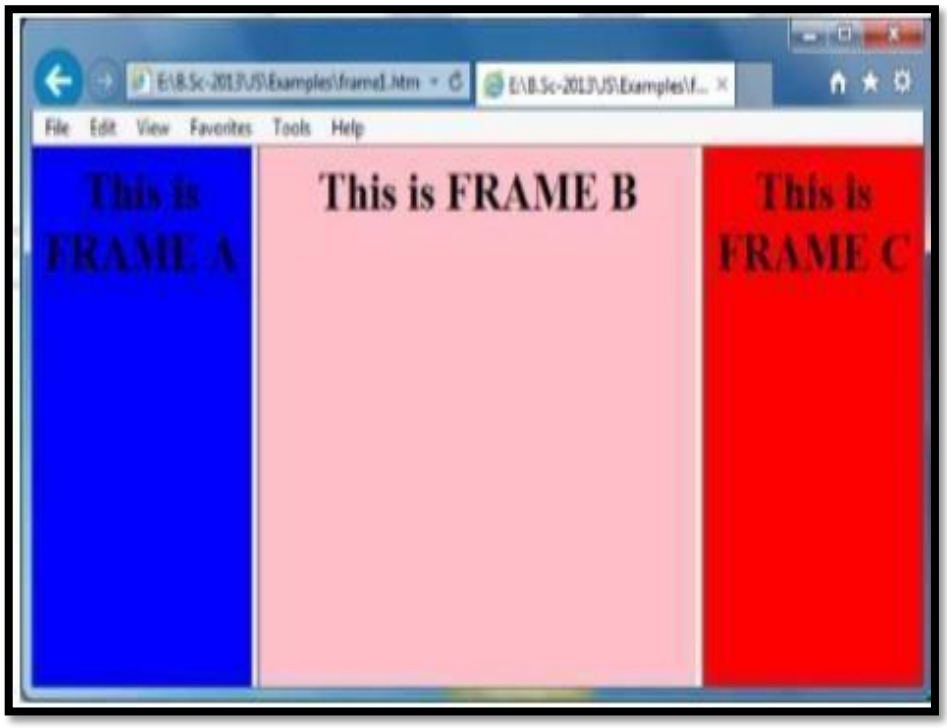
<frame src="frameC.html">

</frameset>

</html>

***** Save the above program as “frame1.html”*****

Output:



DESIGNING OF TABLES IN HTML:

Tables are an excellent way to organize and display information on a page. Tables are defined using the <table> tag. A table is divided into rows with the <tr> tag, and each row is divided into data cells using the <td> tag. The letters td stand for "table data,"which is the content of a data cell. A data cell can contain text,images, lists, paragraphs, forms, horizontal rules, tables, and so on.

Tag	Description	Syntax
<table>	Defines a table	<table [align="center"|"left"|"right"] [border="n"]] [cellpadding="n"][width="nn%"][cellspacing="n"]> . . . </table>
<th>	Defines a header cell in a table	<th [align="center"|"left"|"right"] [border="n"][valign="top"|"center"|"bottom"][nowrap][colspan="n"]
		[rowspan="n"]. . .</th>
<tr>	Defines a row in a table	<tr [align="center"|"left"|"right"] [valign="top"|"center"|"bottom"]> . . .</tr>
<td>	Defines a cell in a table	<td [align="center"|"left"|"right"] [valign="top"|"center"|"bottom"]
		[nowrap][colspan=n][rowspan=n]>. . .</td>
<caption>	Defines a table caption	<caption>string</caption>
<colgroup>	Specifies a group of one or more columns in a table for formatting	<colgroup [span="n"][width="n"]>. . .</colgroup>
<col>	Specifies column properties for each column within a <colgroup> element	<col span="n" [width="n"]>. . .</col>
<thead>	Groups the header content in a table	<thead>. . .</thead>
<tbody>	Groups the body content in a table	<tbody>. . . </tbody>
<tfoot>	Groups the footer content in a table	<tfoot>. . .</tfoot>

Ex:1 The following code creates a table with one row and one column .

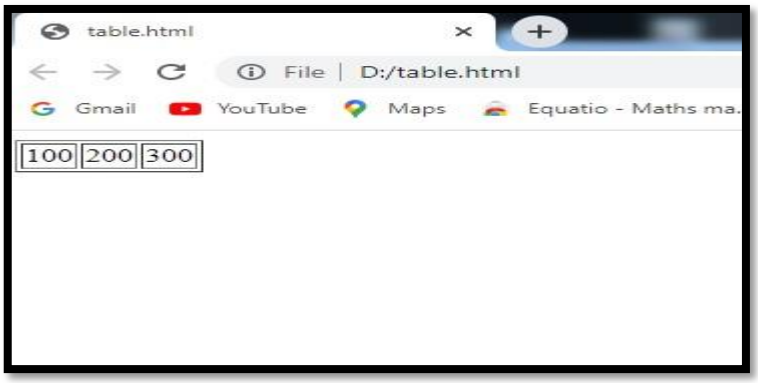
```
<html>
<h4>One column:</h4>
<table border="1">
<tr>td>100</td>
</tr>
</table>
</body>
</html>
```

Output:

Ex:2 The following code creates a table with one row and three columns.

```
<html>
<body>
<table border="1">
<tr>
<td>100</td>
<td>200</td>
<td>300</td>
</tr>
</table>
</body>
</html>
```

Output:



Headings in a Table:

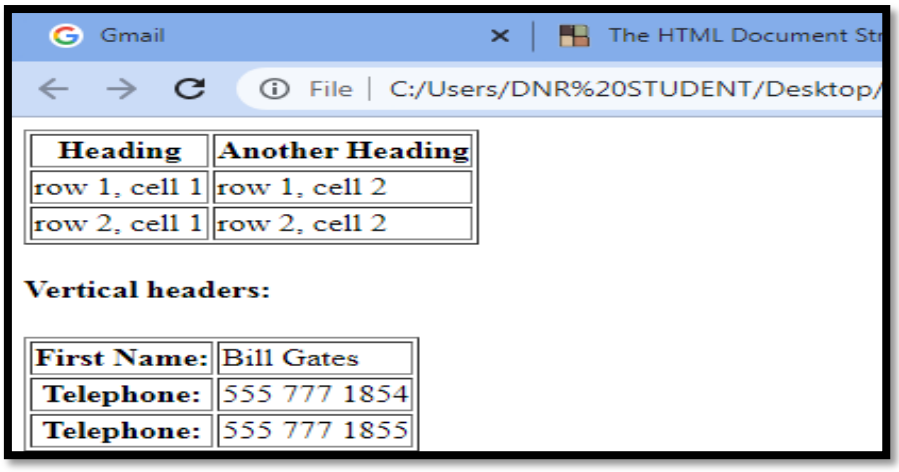
Table headings are defined with the <th> tag.

Ex: The following explains how to use the headings in a table.

```
<html>
<body>
<table border="1">
<tr>
<th>Heading</th>
<th>Another Heading</th>
</tr>
<tr>
<td>row 1, cell 1</td>
<td>row 1, cell 2</td>
</tr>
```

```
<tr>
<td>row 2, cell 1</td>
<td>row 2, cell 2</td>
</tr>
</table>
<h4>Vertical headers:</h4>
<table border="1">
<tr>
<th>First Name:</th>
<td>Bill Gates</td>
</tr>
<tr>
<th>Telephone:</th>
<td>555 777 1854</td>
</tr>
<tr>
<th>Telephone:</th>
<td>555 777 1855</td>
</tr>
</table>
</body>
</html>
```

Output:



TEXT 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.

In HTML the formatting tags are divided into two categories:

Physical tag: These tags are used to provide the visual appearance to the text.

Logical tag: These tags are used to add some logical or semantic value to the text.

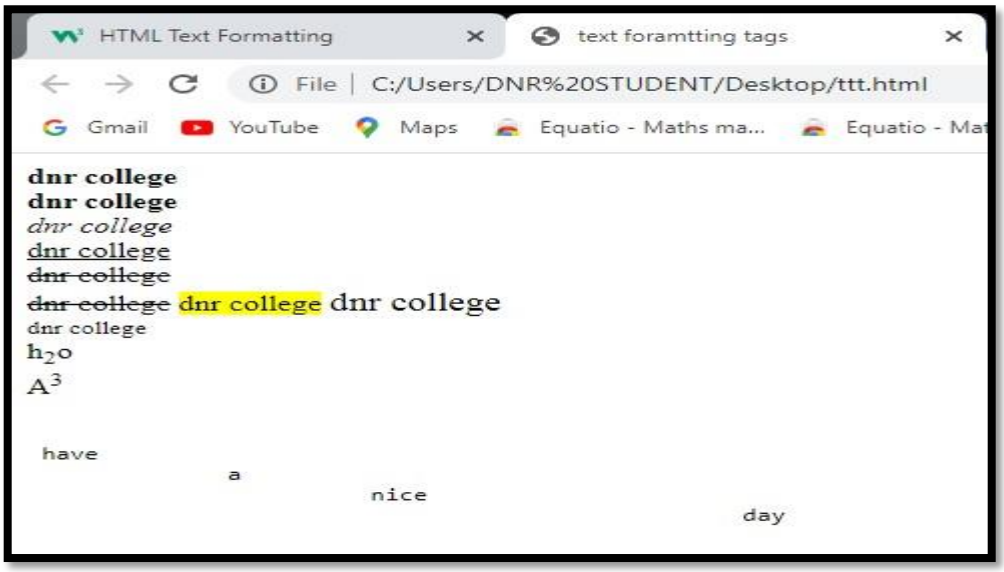
Element name	Description
<b>	This is a physical tag, which is used to bold the text written between it.
	This is a logical tag, which tells the browser that the text is important.
<i>	This is a physical tag which is used to make text italic.
	This is a logical tag which is used to display content in italic.
<mark>	This tag is used to highlight text.
<u>	This tag is used to underline text written between it.
<tt>	This tag is used to appear a text in teletype. (not supported in HTML5)
<strike>	This tag is used to draw a strikethrough on a section of text. (Not supported in HTML5)
<sup>	It displays the content slightly above the normal line.
<sub>	It displays the content slightly below the normal line.
<del>	This tag is used to display the deleted content.
<ins>	This tag displays the content which is added
<big>	This tag is used to increase the font size by one conventional unit.
<small>	This tag is used to decrease the font size by one unit from base font size.

Ex:

```
<html>
<head>
<title>text foramtting tags</title>
</head>
<body>
<b>dnr college</b> <br>
<strong>dnr college</strong> <br>
```

```
<em>dnr college</em> <br>

<u>dnr college</u> <br>

<strike>dnr college</strike> <br>

<del>dnr college</del>

<mark>dnr college</mark>

<big>dnr college</big> <br>

<small>dnr college</small> <br> h<sub>2</sub>o <br> A<sup>3</sup> <br> <br>
  <pre> have
                a
                        nice
                                day
                                        </pre>

</body>

</html>
```

Output:



SHORTS

COMMENTS IN HTML:

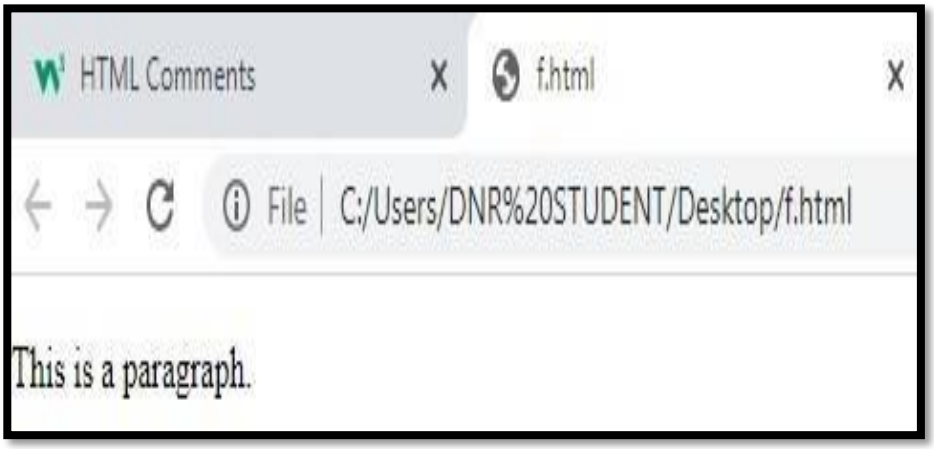
HTML comments are not displayed in the browser, but they can help document your HTML source code. You can add comments to your HTML source by using the following syntax:

<! -- Write your comments here -->

Ex:

```
<! DOCTYPE html>
<Html>
<body>
<! -- This is a comment -->
<p>This is a paragraph.</p>
<!-- Comments are not displayed in the browser -->
</body>
</html>
```

Output:



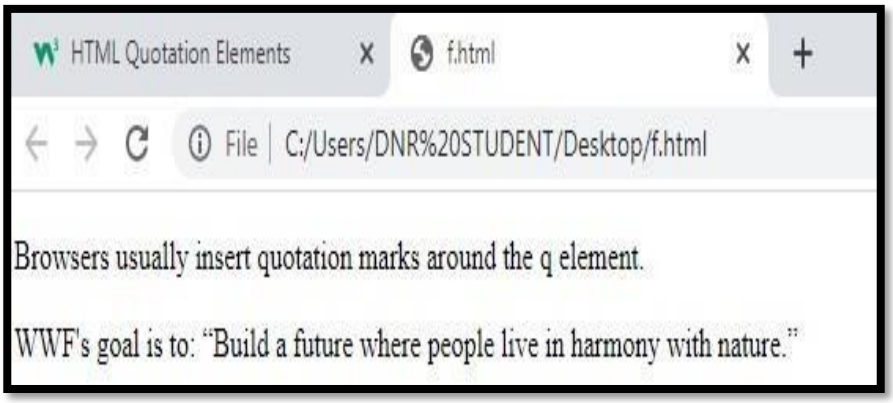
QUOTATIONS IN HTML:

The HTML <q> tag defines a short quotation.

Browsers normally insert quotation marks around the quotation.Ex:

Ex:

```
<!DOCTYPE html>
<html>
<body>
<p>Browsers usually insert quotation marks around the q element.</p>
<p>WWF's goal is to: <q>Build a future where people live in harmony with nature.</q></p>
</body>
</html>
```



Web Applications	Desktop Applications
Deployment and up-gradation for a web-based application require deployment on a single set of server machines.	Deployment and any up-gradation/patch are done on individual client machines separately.
Web applications can be accessed from anywhere,so there is no location constraint.	As desktop are confined to a standalone machine, so they can be only accessed from the machines they are deployed in.
Web applications are platform-independent, they can work in different types of platforms with the only requirement of a web browser.	Desktop applications need to be developed separatelyfor different platform machines.
Web applications are at higher security risks as they are inherently designed to increase accessibility.	Desktop applications, on the other hand, have better authorization and administrators have better control, hence more secure.
Web applications rely heavily on internet connectivity, for there operation.	Desktop applications don't require the internet for their operations. Some applications just require internet connectivity at the time of updations.

HTML HEADINGS:

HTML headings are defined with the <h1> to <h6> tags.

<h1> defines the most important heading. <h6> defines the least important heading.

Ex:<!DOCTYPE html>

<html>

<body>

<h1>Heading 1</h1>

<h2>Heading 2</h2>

<h3>Heading 3</h3>

<h4>Heading 4</h4>

<h5>Heading 5</h5>

<h6>Heading 6</h6>

</body>

</html>

Output:

HTML PARAGRAPHS:

HTML paragraph or HTML p tag is used to define a paragraph in a webpage. Let's take a simple example to see how it work. It is a notable point that a browser itself add an empty line before and after a paragraph. An HTML tag indicates starting of new paragraph.

The HTML element defines a paragraph.

A paragraph always starts on a new line, and browsers automatically add some white space (a margin) before and aftera paragraph.

Ex:

<!DOCTYPE html>

<html>

<head>

<title>paragraph</title>

</head>

<body>

This is a paragraph1.

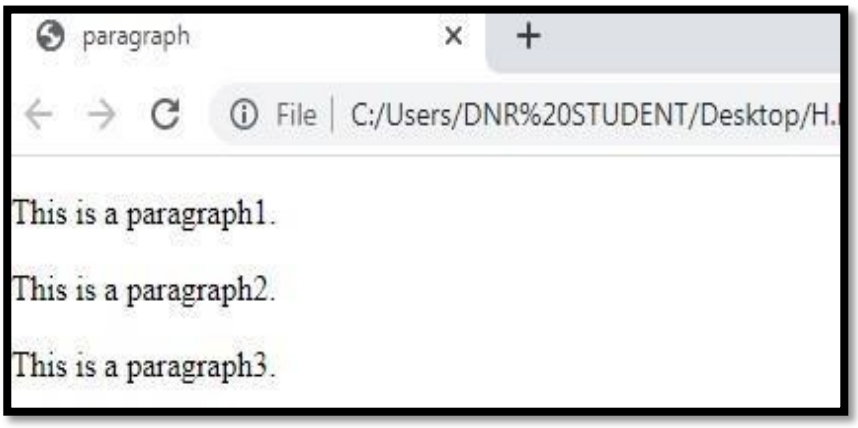
This is a paragraph2.

This is a paragraph3.

</body>

</html>

Output:



STYLES IN HTML:

HTML Style is used to change or add the style on existing HTML elements. There is a default style for every HTML element e.g. background color is white, text color is black etc.

The style attribute can by used with any HTML tag. To apply style on HTML tag, you should have the basic knowledge of css properties e.g. color, background-color, text-align, font-family, font-size etc.

<tag name style="property: value;">

Ex:1

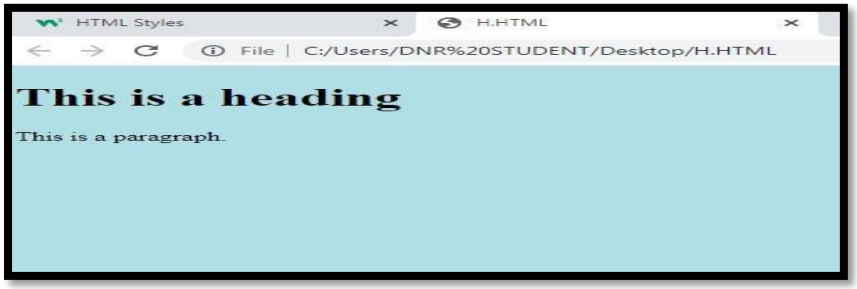
```
<!DOCTYPE html>
<html>
<body>
<p>I am normal</p>
<p style="color:red;">I am red</p>
<p style="color:blue;">I am blue</p>
<p style="font-size:50px;">I am big</p>
</body>
</html>
```

Output:

Ex:2

```
<!DOCTYPE html>
<html>
<body style="background-color:powderblue;">
<h1>This is a heading</h1>
<p>This is a paragraph.</p>
</body>
</html>
```

Output:



IMAGES IN HTML:

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

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

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

The <img> tag has two required attributes:

src - Specifies the path to the image

alt - Specifies an alternate text for the imageSyntax:

```
<img src="url" alt="alternate text">
```

Ex:

```
<!DOCTYPE html>
<html>
<head>
<title>images in html</title>
</head>
<body>
<center>
<img src="flower1.jpg" alt="Flowers" width="400" height="500">
<center>
</body></html>
```

Output:

CLASSES:

The HTML class attribute is used to specify a class for an HTML element.

The class attribute is often used to point to a class name in a style sheet. It can also be used by a JavaScript to accessand manipulate elements with the specific class name.

In the following example we have three elements with a class attribute with the value of "city". All of the three elements will be styled equally according to the .city style definition in the head section.

Ex:

```
<html>
<head>
<style>
.city {
background-color: tomato;color: white;
border: 2px solid black;
}
</style>
</head>
<body>
<div class="city">
<h2>London</h2>
<p>London is the capital of England.</p>
</div>
<div class="city">
<h2>Paris</h2>
<p>Paris is the capital of France.</p>
```

</body>

</html>

Output:

FILE PATHS IN HTML:

A file path describes the location of a file in a web site's folder structure. File paths are used when linking to external files, like:

Web pages, Images, Style sheets.

Path	Description
<img src="picture.jpg">	The "picture.jpg" file is located in the same folder as the currentpage
<img src="images/picture.jpg">	The "picture.jpg" file is located in the images folder in the currentfolder
<img src="/images/picture.jpg">	The "picture.jpg" file is located in the images folder at the root ofthe current web
<img src="../picture.jpg">	The "picture.jpg" file is located in the folder one level up from thecurrent folder

Ex:

<!DOCTYPE html>

<html>

<body>

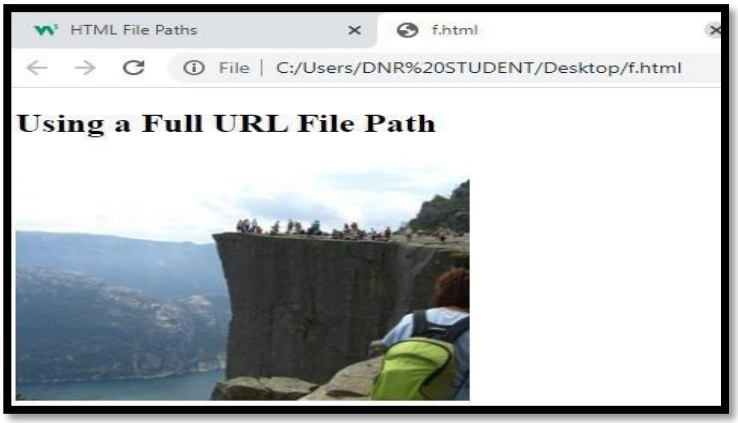
<h2>Using a Full URL File Path</h2>

<img src="https://www.w3schools.com/images/picture.jpg" alt="Mountain" style="width: 300px">

</body>

</html>

Output:


HTML File Paths
f.html
File | C:/Users/DNR%20STUDENT/Desktop/f.html
Using a Full URL File Path

UNIT-II

FORM ELEMENTS:

(i) INPUT

HTML Input Types:

Here are the different input types you can use in HTML:

```
<input type="button">
<input type="checkbox">
<input type="color">
<input type="date">
<input type="date time-local">
<input type="email">
<input type="file">
<input type="hidden">
<input type="image">
<input type="month">
<input type="number">
<input type="password">
<input type="radio">
<input type="range">
<input type="reset">
<input type="search">
<input type="submit">
```

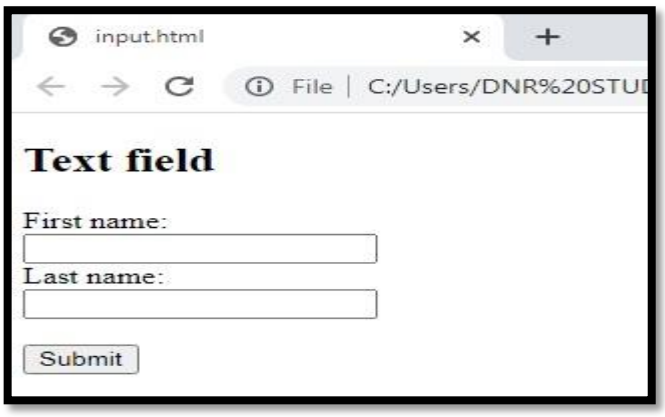
Input Type Text:

<input type="text"> defines a single-line text input field

Ex:

```
!DOCTYPE html>
<html>
<body>
<h2>Text field</h2>
<form action=" ">
  <label for="fname">First name:</label><br>
  <input type="text" id="fname" name="fname"><br>
  <label for="lname">Last name:</label><br>
  <input type="text" id="lname" name="lname"><br><br>
  <input type="submit" value="Submit">
</form>
</body>
</html>
```

Output:



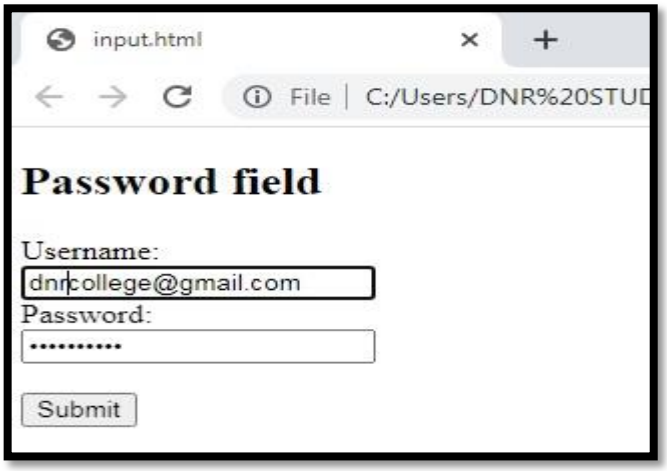
Input Type Password

<input type="password"> defines a password field

Ex: <!DOCTYPE html>

```
<html>
<body>
<h2>Password field</h2>
<form action="/action_page.php">
  <label for="username">Username:</label><br>
  <input type="text" id="username" name="username"><br>
  <label for="pwd">Password:</label><br>
  <input type="password" id="pwd" name="pwd"><br><br>
  <input type="submit" value="Submit">
</form>
</body>
</html>
```

Output:



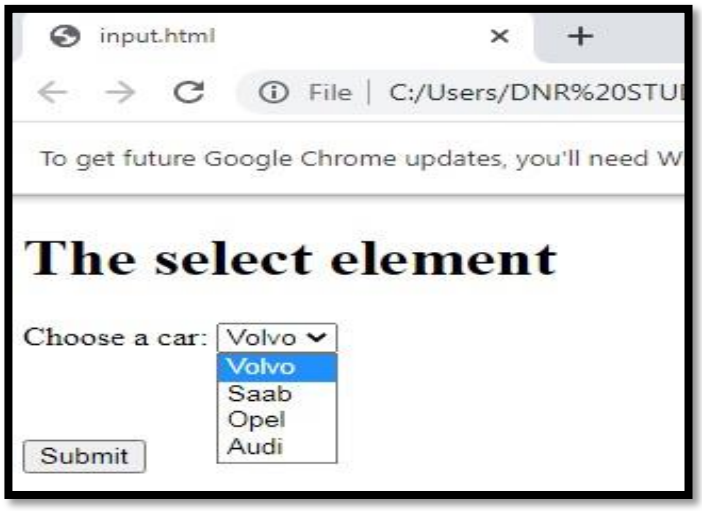
(ii)DROPDOWN

The following example shows how to create a simple dropdown list on a HTML page. A dropdown list is a selectable list.

Ex:

```
<!DOCTYPE html>
<html>
<body>
<h1>The select element</h1>
<form action="  ">
  <label for="cars">Choose a car:</label>
  <select name="cars" id="cars">
    <option value="volvo">Volvo</option>
    <option value="saab">Saab</option>
    <option value="opel">Opel</option>
    <option value="audi">Audi</option>
  </select>
  <br><br>
  <input type="submit" value="Submit">
</form>
</body>
</html>
```

Output:



(iii)CHECKBOXES

The HTML <checkbox> tag is used to define the square boxes. It is a form element which allows users to select one or more options from the given options.

It is created by the type attribute of the <input> element as shown in the following syntax:

```
<input type="checkbox" name="field name" value="Initial value">
```

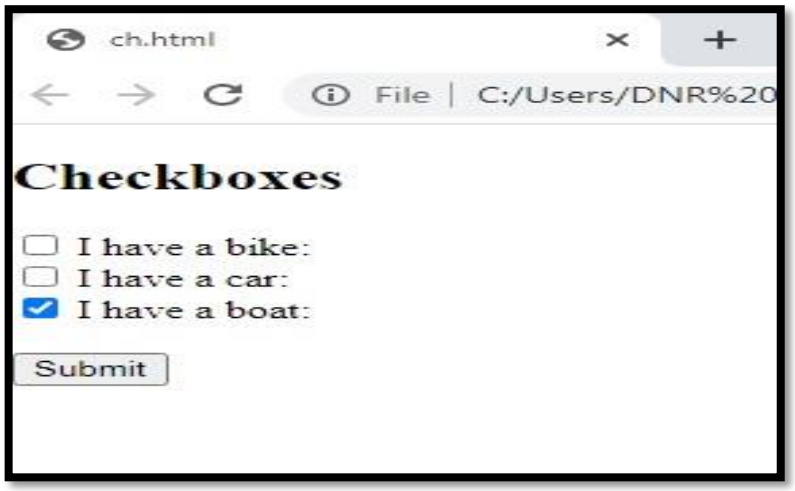
If we want to select any checkbox by default, then we have to set the checked attribute with the "yes" value as described in the following syntax:

<input type="checkbox" name="field name" value="Initial value" checked="yes">

Ex:

```
<!DOCTYPE html>
<html>
<body>
<h2>Checkboxes</h2>
<form action=" ">
  <input type="checkbox" id="vehicle1" name="vehicle1" value="Bike">
  <label for="vehicle1"> I have a bike:</label><br>
  <input type="checkbox" id="vehicle2" name="vehicle2" value="Car">
  <label for="vehicle2"> I have a car:</label><br>
  <input type="checkbox" id="vehicle3" name="vehicle3" value="Boat">
  <label for="vehicle3"> I have a boat:</label><br><br>
  <input type="submit" value="Submit">
</form>
</body>
</html>
```

Output:



(iv)RADIO BUTTONS

The HTML <Radio> button is used to define the small circles, which are highlighted when selected. It is a form element which allows the users to select only one option from the given set of options.

It is created by the type attribute of the <input> element as shown in the following syntax:

<input type="radio" name ="Any name"/>

Ex:

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

```
<body>
<h2>Radio Buttons</h2>
<form action=" ">
  <input type="radio" id="html" name="fav_language" value="HTML">
  <label for="html">HTML</label><br>
  <input type="radio" id="css" name="fav_language" value="CSS">
  <label for="css">CSS</label><br>
  <input type="radio" id="javascript" name="fav_language" value="JavaScript">
  <label for="javascript">JavaScript</label><br><br>
  <input type="submit" value="Submit">
</form>
</body>
</html>
```

Output:

HTML MEDIA:

Multimedia comes in many different formats. It can be almost anything you can hear or see, like images, music, sound, videos, records, films, animations, and more.

Web pages often contain multimedia elements of different types and formats.

Multimedia elements (like audio or video) are stored in media files.

The most common way to discover the type of a file, is to look at the file extension.

Multimedia files have formats and different extensions like: .wav, .mp3, .mp4, .mpg, .wmv, and .avi.

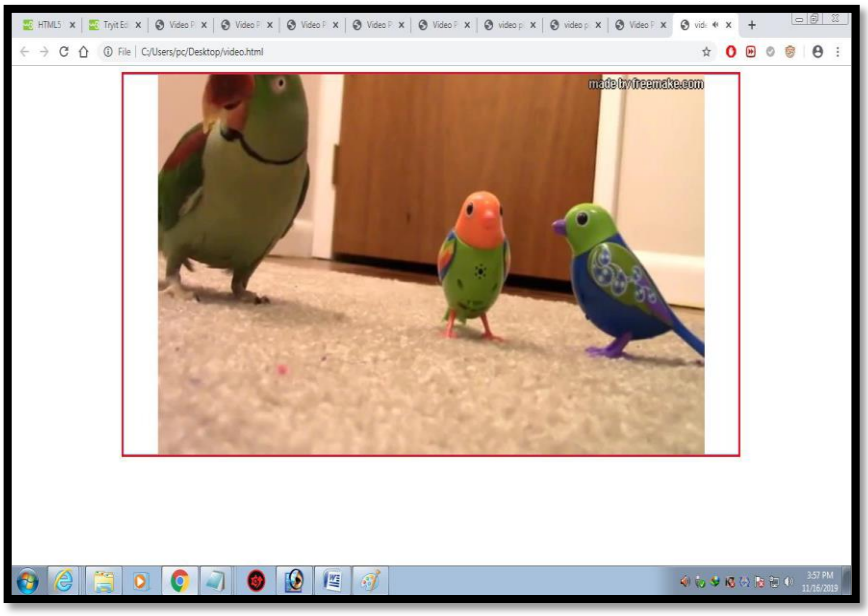
1.**VIDEO:**

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

```
<html>
<head>
```

```
<title>video player</title>
<style>
div
{
border-style:solid;
border-color:red;
width:1000;
}
</style>
</head>
<body>
<center>
<div>
<video width=1000 height=500 controls>
<source style="border:red" src="C:\Users\pc\Desktop\motivation.mp4"
type="video/mp4">
Your browser does not support the video tag.
</video>
</div>
</center>
</body>
</html>
```

Output:



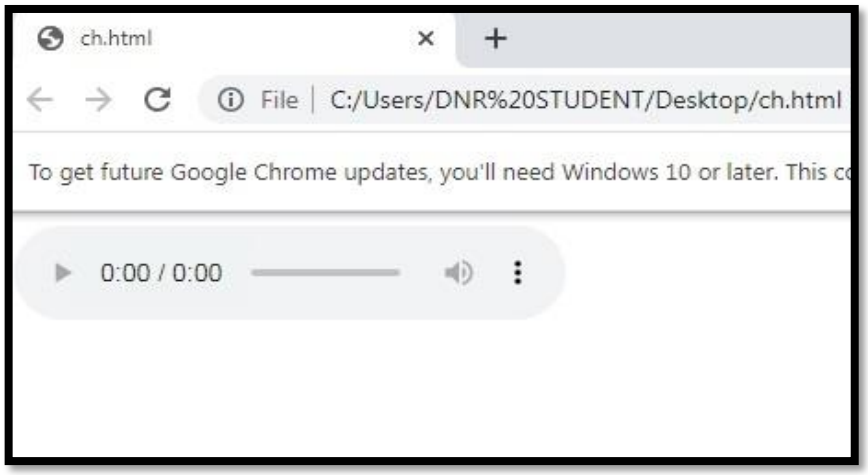
2.AUDIO:

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

Ex:

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

Output:



3.PLUG-INS:

Plug-ins are computer programs that extend the standard functionality of the browser.

Plug-ins were designed to be used for many different purposes:

To run Java applets

To run Microsoft ActiveX controls

To display Flash movies

To display maps

To scan for viruses

To verify a bank id

The <object> Element

The <object> element is supported by all browsers.

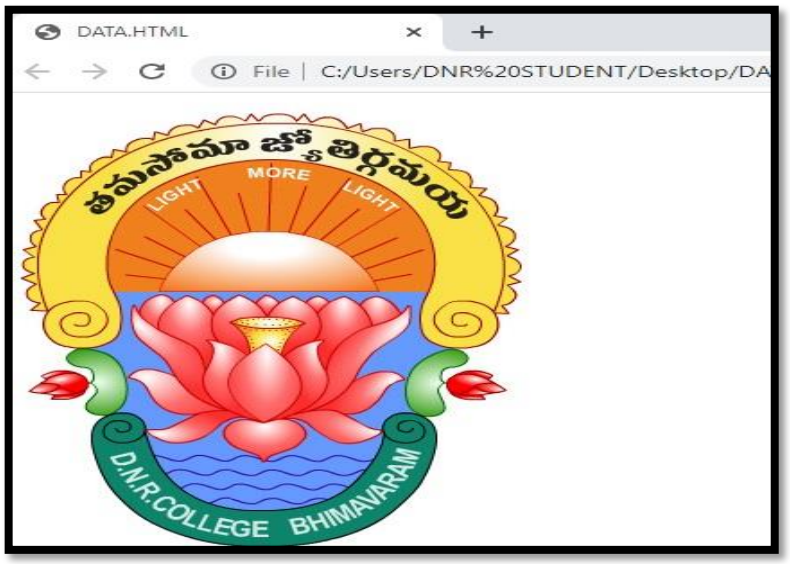
The <object> element defines an embedded object within an HTML document.

It was designed to embed plug-ins (like Java applets, PDF readers, and Flash Players) in web pages, but can also be used to include HTML in HTML:

Ex:

```
<!DOCTYPE html>
<html>
<body>
<object data="LOGO.jpg"></object>
</body>
</html>
```

Output:



4. YOUTUBE:

Playing a YouTube Video in HTML

To play your video on a web page, do the following:

Upload the video to YouTube

Take a note of the video id

Define an <iframe> element in your web page

Let the src attribute point to the video URL

Use the width and height attributes to specify the dimension of the player

Add any other parameters to the URL (see below)

Ex:

```
<!DOCTYPE html>
<html>
<body>
<iframe width="420" height="345" src="https://www.youtube.com/embed/tgbNymZ7vqY">
</iframe>
</body>
</html>
```

Output:

HTML GRAPHICS:

The HTML <canvas> element is used to draw graphics on a web page.

The graphic to the left is created with <canvas>. It shows four elements: a red rectangle, a gradient rectangle, a multicolor rectangle, and a multicolor text.

A canvas is a rectangular area on an HTML page. By default, a canvas has no border and no content.

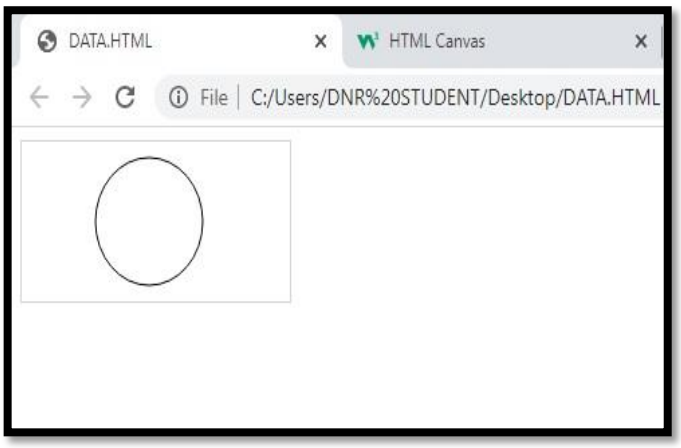
The markup looks like this:

<canvas id="myCanvas" width="200" height="100"></canvas>

Ex:

```
<!DOCTYPE html>
<html>
<body>
<canvas id="myCanvas" width="200" height="100" style="border:1px solid #d3d3d3;">
Your browser does not support the HTML canvas tag.</canvas>
<script>
var c = document.getElementById("myCanvas");
var ctx = c.getContext("2d");
ctx.beginPath();
ctx.arc(95,50,40,0,2*Math.PI);
ctx.stroke();
</script>
</body>
</html>
```

Output:



HTML SVG Graphics:

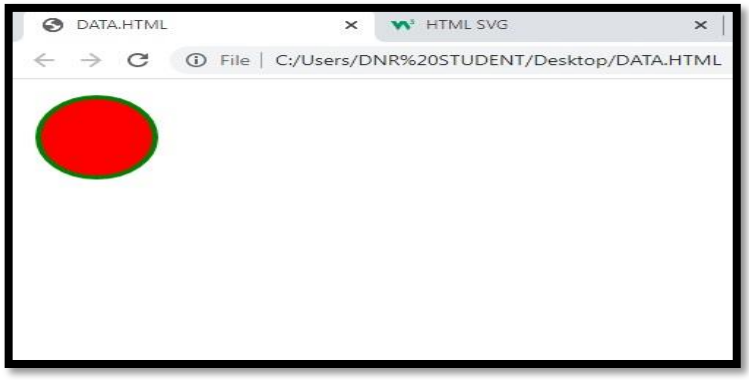
The HTML <svg> element is a container for SVG graphics.

SVG has several methods for drawing paths, boxes, circles, text, and graphic images.

Ex:1

```
!DOCTYPE html>
<html>
<body>
<svg width="100" height="100">
  <circle cx="50" cy="50" r="40"
  stroke="green" stroke-width="4" fill="red" />
Sorry, your browser does not support inline SVG.
</svg>
 </body>
</html>
```

Output:



Ex:2

```
<!DOCTYPE html>
<html>
<body>
<svg width="400" height="100">
```

```
  <rect width="400" height="100"
  style="fill:rgb(0,0,255);stroke-width:10;stroke:rgb(0,0,0)" />
Sorry, your browser does not support inline SVG.
</svg>
</body>
</html>
```

Output:

CSS AND STYLES.

"CSS is an acronym which means Cascading Style Sheets."

A style sheet is made up of style rules that tell a browser how to present a document. As the name suggests, CSS is a special purpose style sheet language and it helps in defining the format of presentation for any document that has been written in a markup language like the HTML.

"A STYLE is simply a set of formatting instructions that can be applied to a piece of text." There are three mechanisms by which we can apply styles to our HTML documents:

- The style can be defined within the basic HTML tag (This is also known as Inline Style).
- Styles can be defined in the <head> section and applied to the whole document (This is known as Internal Style).
- Styles can be defined in external files called styles sheets which can then be used in any document by including the style sheet via a URI

(This is also known as External Style).

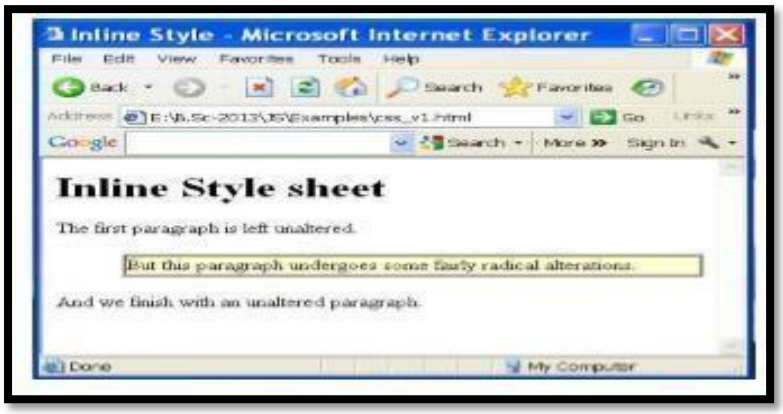
Styles can be cascaded i.e., the formats override any which were defined or included earlier in the document.

Ex:1 Inline Style:

```
<html>
<head>
<title> Inline Style </title>
</head>
```

<body>
<h1> Inline Style sheet </h1>
 The first paragraph is left unaltered.
<p style=”margin-left”: 10%; border: ridge; background: #ffffcc”> But this paragraph undergoes some fairly radical alterations.
 And we finish with an unaltered paragraph.
</body> </html>

Output:



Ex:2 Internal Style:

```
<html>
    <head>
        <title> Internal Style </title>
        <style>h1{
   color: red;
   border:  thin  groove; text-
   align:center;
      }
        </style>
      </head>
   <body>
        <h1> Internal Style Sheet </h1>
    </body>
</html>
```

Output:

Ex:3 External Style

Step:1 Open a new file in Notepad and type the following code:

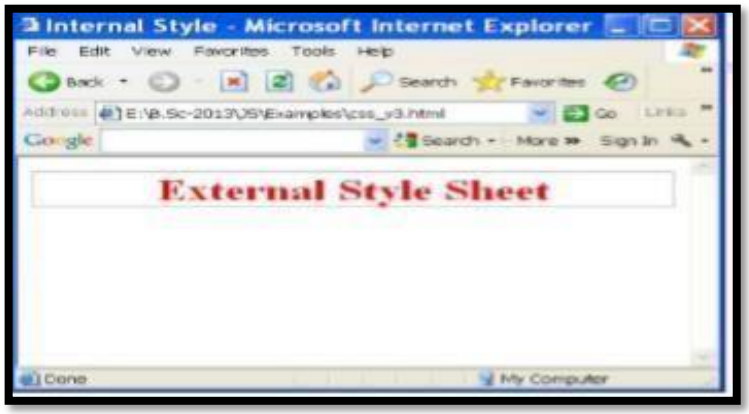
```
h1{
color: red;
border: thin groove; text-align:center;
}
```

**** Save the file as "css_v3.css"**

Step:2 Open a new file in Notepad and type the following code:

```
<html>
<head>
<title> Internal Style </title>
<link rel="stylesheet" href="css_v3.css" type="text/css">
</head>
<body>
<h1> External Style Sheet </h1>
</body>
</html>
```

Output:



FONT PROPERTY IN STYLE:

A number of properties of the text can be altered. The following properties can change the style of our work.

a. Fonts:

Fonts are identified by giving the name of a specific font. In CSS, there are two types of font family names:

- generic family - a group of font families with a similar look (like "Serif" or "Monospace")
- font family - a specific font family (like "Times New Roman" or "Arial") The general format is

<font-family>: <family name> [<generic family>]

Generic family	Font family	Description
Serif	Times New Roman	Serif fonts have small lines at the ends on some characters
	Georgia	
Sans-serif	Arial	"Sans" means without - these fonts do not have the lines at the ends of characters
	Verdana	
Monospace	Courier New	All monospace characters have the same width
	Lucida Console	

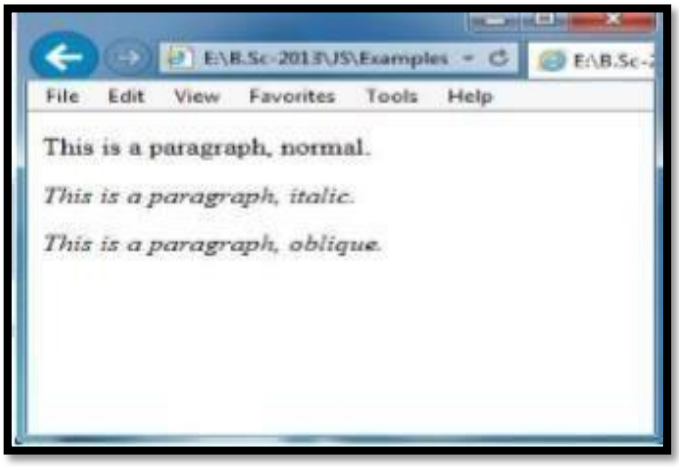
Font-style property:

- **Normal:** This is the default value of the font-style for which the browser will display normal font text.
- **Italic:** This font-style is used to display the text as italic in the browser.
- **Oblique:** This font-style specifies an angle for the slant of the text. The displayed font-style is oblique in the browser.

Ex:

```
<!DOCTYPE html>
<html>
 <head>
<style>
<! - -
p.normal {font-style:normal;}
p.italic {font-style:italic;}
p.oblique {font-style:oblique;}
//- - >
</style>
</head>
<body>
<p class="normal">This is a paragraph, normal.</p>
<p class="italic">This is a paragraph, italic.</p>
<p class="oblique">This is a paragraph, oblique.</p>
</body></html>
```

Output:



Font-weight Property:

The font-weight property sets how thick or thin characters in text should be displayed. The property values are:

Value	Description
Normal	Defines normal characters. This is default
Bold	Defines thick characters
Bolder	Defines thicker characters
Lighter	Defines lighter characters
100 200 300	Defines from thin to thick characters. 400 is the same as normal, and 700 is the same as bold

In the above table, the first four are relative and remaining (numbers) are absolute weights.

Ex:

p.normal {font-weight:normal;}

p.thick {font-weight:bold;}

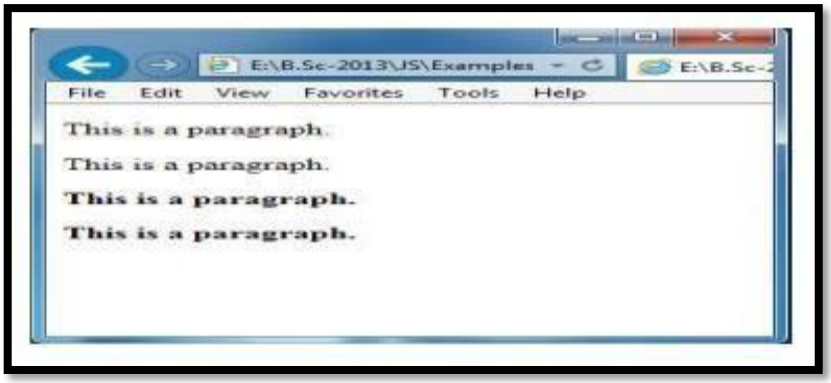
p.thicker {font-weight:900;}

Source Code:

```
<!DOCTYPE html>
<html>
<head>
<style>
<! - -
p.normal {font-weight:normal;}
 p.light {font-weight:lighter;}
p.thick {font-weight:bold;}
p.thicker {font-weight:900;}
 //- - >
 </style>
```

```
</head>
<body>
 <p class="normal">This is a paragraph.</p>
<p class="light">This is a paragraph.</p>
<p class="thick">This is a paragraph.</p>
<p class="thicker">This is a paragraph.</p>
</body>
</html>
```

Output:



Font-size Property:

The font-size property sets the size of a font. The property values are:

Value	Description
Small	Sets the font-size to a small size
Medium	Sets the font-size to a medium size. This is default
Large	Sets the font-size to a large size
Smaller	Sets the font-size to a smaller size than the parent element
Larger	Sets the font-size to a larger size than the parent element
Length	Sets the font-size to a fixed size in px (pixels), cm, etc.
%	Sets the font-size to a percent of the parent element's font size

Ex:

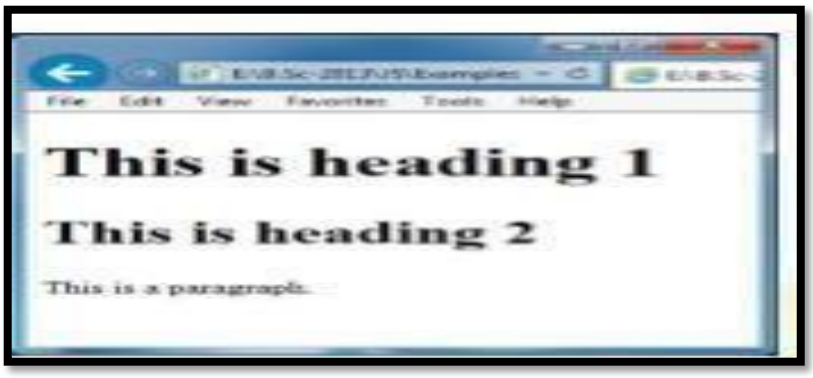
```
h1 {font-size:250%} h2 {font-size:200%} p {font-size:100%}
```

Source Code:

```
<!DOCTYPE html>
<html>
<head>
<style>
<! - -
h1 {font-size:250%;} h2 {font-size:200%;} p {font-size:100%;}
```

```
//- - > </style></head>
<body>
<h1>This is heading 1</h1>
<h2>This is heading 2</h2>
<p>This is a paragraph.</p>
</body>
</html>
```

Output:



SHORTS

Borders in CSS:

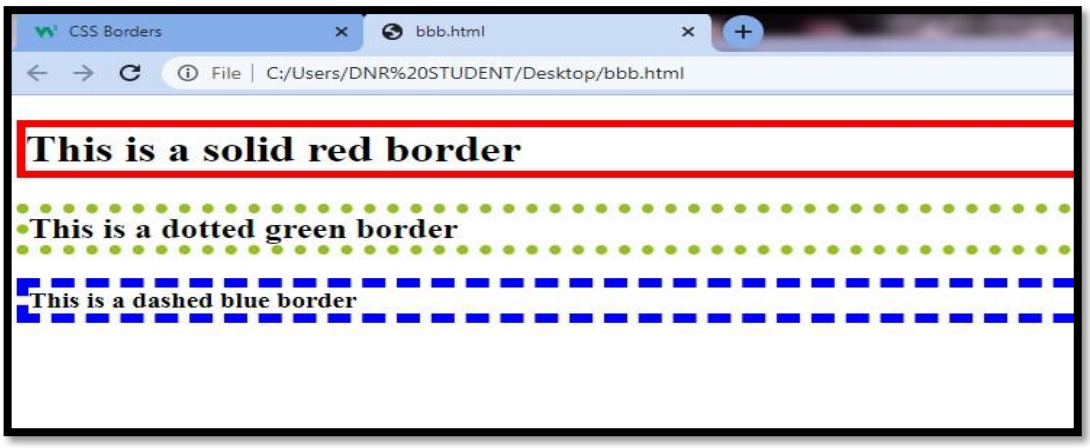
- The CSS border is a shorthand property used to set the border on an element.
- The CSS border properties are use to specify the style, color and size of the border of an element.

Value	Description
none	It doesn't define any border.
dotted	It is used to define a dotted border.
dashed	It is used to define a dashed border.
solid	It is used to define a solid border.
double	It defines two borders wlth the same border-width value.
groove	It defines a 3d grooved border. effect is generated according to border-color value.
ridge	It defines a 3d ridged border. effect is generated according to border-color value.
inset	It defines a 3d inset border. effect is generated according to border-color value.
outset	It defines a 3d outset border. effect is generated according to border-color value.

Ex:

```
<!DOCTYPE html>
<html>
<head>
<style>
h1 {
   border-style: solid;
   border-color: red;
   border-width:6px;
 }
h2 {
   border-style: dotted;
   border-color: #98bf21;
 border-width:8px;
}
h3 {
   border-style: dashed;
   border-color: blue;
 border-width:8px;
}
</style>
</head>
<body>
<h1>This is a solid red border</h1>
<h2>This is a dotted green border</h2>
<h3>This is a dashed  blue border</h3>
</body>
</html>
```

Output:



MARGINS IN CSS:

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

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

There are following CSS margin properties:

Property	Description
Margin	This property is used to set all the properties in one declaration.
margin-left	it is used to set left margin of an element.
margin-right	It is used to set right margin of an element.
margin-top	It is used to set top margin of an element.
margin-bottom	It is used to set bottom margin of an element.

All the margin properties can have the following values:

auto - the browser calculates the margin

length - specifies a margin in px, pt, cm, etc.

% - specifies a margin in % of the width of the containing element

inherit - specifies that the margin should be inherited from the parent element

Ex:

```
<!DOCTYPE html>
<html>
<head>
<style>
p{
  border: 2px solid black;
  margin-top: 80px;
  margin-bottom: 100px;
  margin-right: 50px;
  margin-left: 30px;
  background-color: lightblue;
}
</style>
</head>
<body>
```

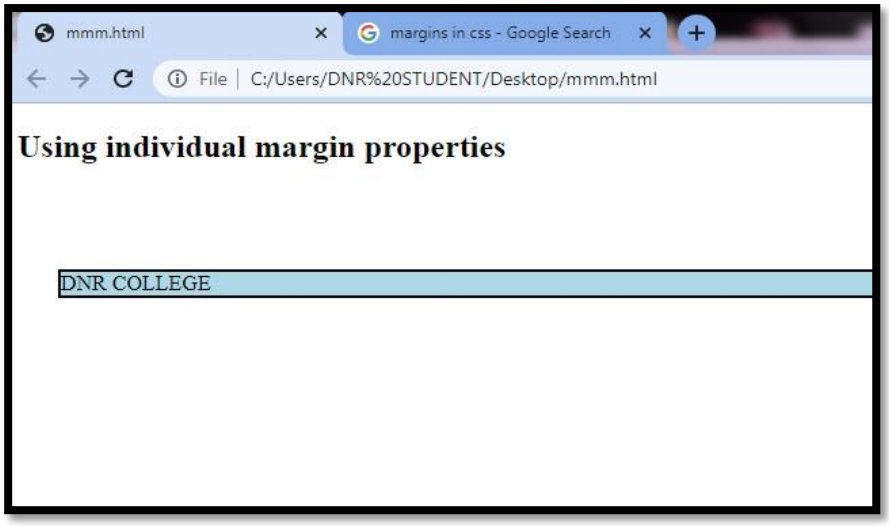
<h2>Using individual margin properties</h2>

DNR COLLEGE

</body>

</html>

Output:



TABLES IN CSS:

We can apply style on HTML tables for better look and feel. There are some CSS properties that are widely used in designing table using CSS:

- border
- border-collapse
- padding
- width
- height
- text-align
- color
- background-color

CSS Table Border

We can set border for the table, th and td tags using the CSS border property.

Ex:

<!DOCTYPE>

<html>

<head>

<style>

table, th, td {

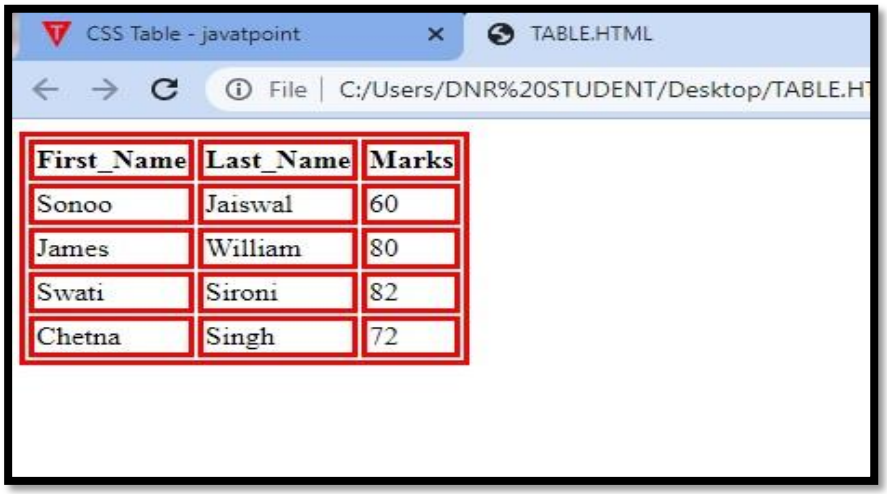
  border: 3px solid black;

border-color:red;

}

```
</style>
</head>
<body>
<table>
<tr><th>First_Name</th><th>Last_Name</th><th>Marks</th></tr>
<tr><td>Sonoo</td><td>Jaiswal</td><td>60</td></tr>
<tr><td>James</td><td>William</td><td>80</td></tr>
<tr><td>Swati</td><td>Sironi</td><td>82</td></tr>
<tr><td>Chetna</td><td>Singh</td><td>72</td></tr>
</table>
</body>
</html>
```

Output:



LISTS IN CSS:

The CSS list properties allow you to:

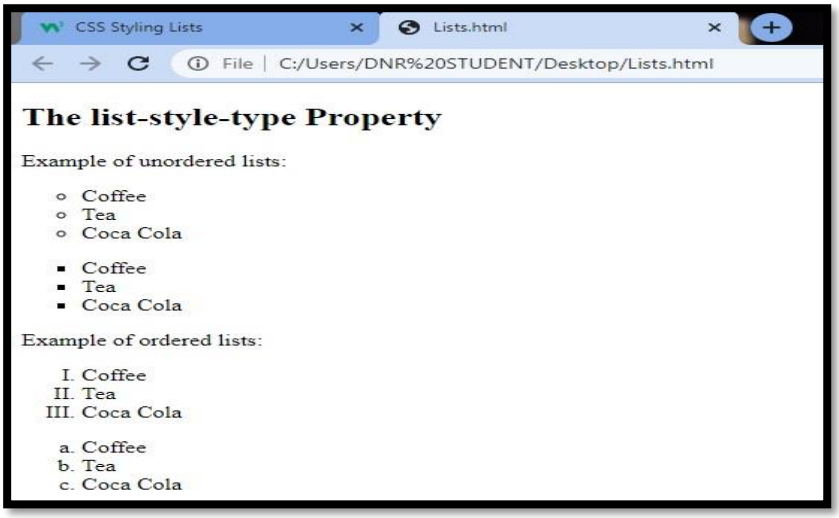
- Set different list item markers for ordered lists
- Set different list item markers for unordered lists
- Set an image as the list item marker
- Add background colors to lists and list items

Ex:

```
!DOCTYPE html>
<html>
<head>
<style>
ul.a {
  list-style-type: circle;
}
ul.b {
```

```
  list-style-type: square;
}
ol.c {
  list-style-type: upper-roman;
}
ol.d {
  list-style-type: lower-alpha;
}
</style>
</head>
<body>
<h2>The list-style-type Property</h2>
<p>Example of unordered lists:</p>
<ul class="a">
  <li>Coffee</li>
  <li>Tea</li>
  <li>Coca Cola</li>
</ul>
<ul class="b">
  <li>Coffee</li>
  <li>Tea</li>
  <li>Coca Cola</li>
</ul>
<p>Example of ordered lists:</p>
<ol class="c">
  <li>Coffee</li>
  <li>Tea</li>
  <li>Coca Cola</li>
</ol>
<ol class="d">
  <li>Coffee</li>
  <li>Tea</li>
  <li>Coca Cola</li>
</ol>
</body>
</html>
```

Output:



OPACITY IN CSS:

The CSS opacity property is used to specify the transparency of an element. In simple word, you can say that it specifies the clarity of the image.

In technical terms, Opacity is defined as degree in which light is allowed to travel through an object.

How to apply CSS opacity setting:

Opacity setting is applied uniformly across the entire object and the opacity value is defined in term of digital value less than 1. The lesser opacity value displays the greater opacity. Opacity is not inherited.

CSS Opacity Example: transparent image

Let's see a simple CSS opacity example of image transparency.

Ex:

```
<!DOCTYPE html>
<html>
<head>
<style>
img {
  opacity: 0.5;
}
</style>
</head>
<body>
<h1>Image Transparency</h1>
<p>The opacity property specifies the transparency of an element. The lower the value, the more transparent:</p>
<p>Image with 50% opacity:</p>
<img src="img_forest.jpg" alt="Forest" width="170" height="100">
</body>
</html>
```

Output:

BACKGROUND AND COLORS IN CSS:

The background-color property specifies the background color of an element.

The background color of a page is defined in the body selector. The general format is

color: <value>

background-color: <value> | transparent

background-image: URL | none

The values should be given in hexadecimal values. The background color should be transparent and is default.

Instead of a color, an image can be used, identified its URL.

Ex:

```
<!DOCTYPE html>
<html>
<head>
<style>
<! - -
 h1{
}
p{
 background-color:#6495ed;
background-color:#e0ffff;
 }
div{
background-color:#b0c4de;
}
//- - >
</style>
</head>
<body>
```

<h1>CSS background-color example!</h1>

 This is a text inside a div element.

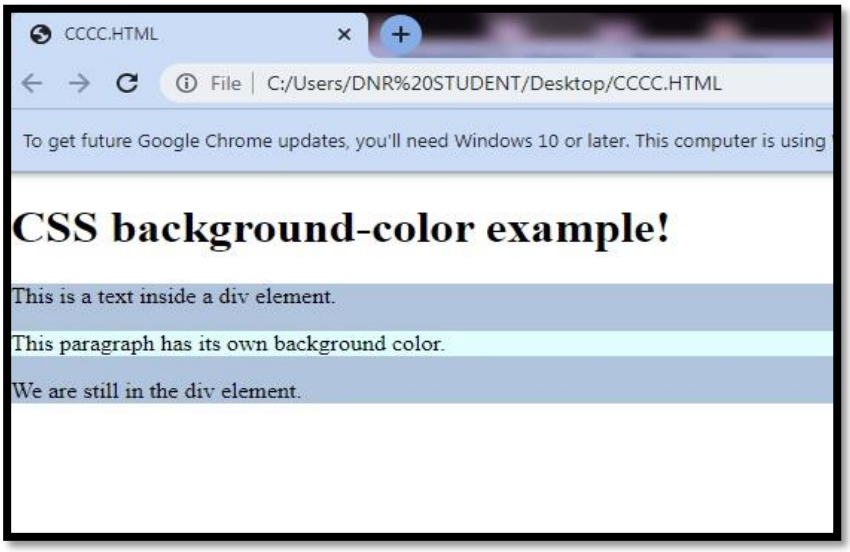
This paragraph has its own background color.

We are still in the div element.

</body>

</html>

Output:



TEXT - DECORATION:

The text-decoration property is used to set or remove decorations from text. The text decoration property is mostly used to remove underlines from links for design purposes. The general formatis text-decoration:

none | underline | overline | line-through

Ex:

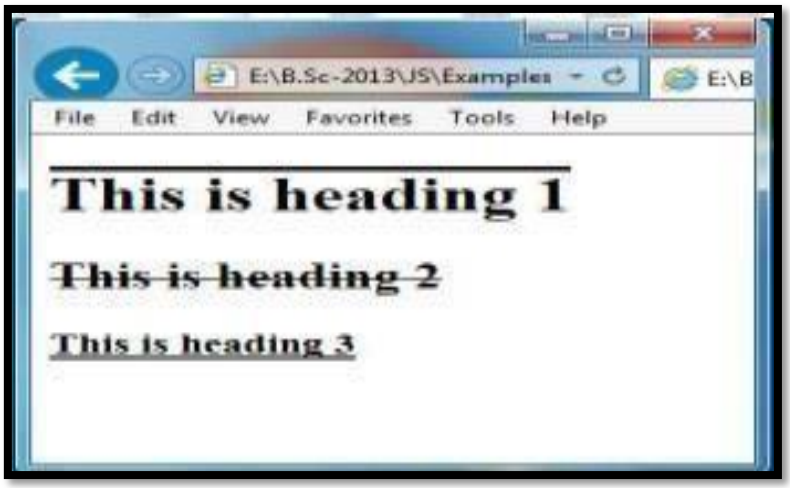
```
h1 {text-decoration:overline;}
h2 {text-decoration:line-through;}
h3 {text-decoration:underline;}
```

Source Code:

```
<!DOCTYPE html>
<html>
   <head>
      <style>
<! - - h1 {text-decoration:overline;}
      h2 {text-decoration:line-through;}h3 {text-
      decoration:underline;}
//- - > </style>
      </head>
      <body>
      <h1>This is heading 1</h1>
```

```
        <h2>This is heading 2</h2>
        <h3>This is heading 3</h3>
    </body>
</html>
```

Output:



VIDEO TAG:

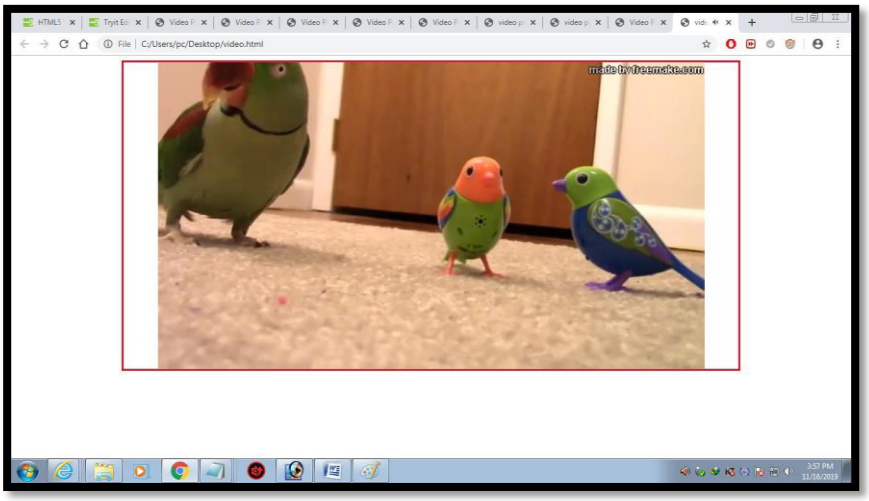
The HTML <video> element is used to show a video on a web page

```
<html>
<head>
<title>video player</title>
<style>
div
{
border-style:solid;
border-color:red;
width:1000;
}
</style>
</head>
<body>
<center>
<div>
<video width=1000 height=500 controls>
<source style="border:red" src="C:\Users\pc\Desktop\motivation.mp4"
type="video/mp4">
Your browser does not support the video tag.
</video>
</div>
```

</body>

</html>

Output:



UNIT-III

JAVASCRIPT

VARIABLES AND DATA TYPES IN JAVA SCRIPT:

JavaScript Data Types: JavaScript provides different data types to hold different types of values. There are two types of data types in JavaScript.

- **Primitive data type**
- **Non-primitive (reference) data type**

JavaScript is a dynamic type language, means you don't need to specify type of the variable because it is dynamically used by JavaScript engine. You need to use var here to specify the data type. It can hold any type of values such as numbers, strings etc.

JavaScript primitive data types

There are five types of primitive data types in JavaScript. They are as follows:

Data Type	Description
String	represents sequence of characters e.g. "hello"
Number	represents numeric values e.g. 100
Boolean	represents boolean value either false or true
Undefined	represents undefined value
Null	represents null i.e. no value at all

JavaScript non-primitive data types

The non-primitive data types are as follows:

Data Type	Description
Object	represents instance through which we can access members
Array	represents group of similar values
RegExp	represents regular expression

Examples:

// Numbers:

```
let length = 16;
let weight = 7.5;
```

// Strings:

```
let color = "Yellow";
let lastName = "Johnson";
```

// Booleans:

```
let x = true;
let y = false;
```

// Object:

```
const person = {firstName:"John", lastName:"Doe"};
```

// Array object:

```
const cars = ["Saab", "Volvo", "BMW"];
```

// Date object:

```
const date = new Date("2022-03-25");
```

VARIABLES IN JAVASCRIPT:

A JavaScript variable is simply a name of storage location. There are two types of variables in JavaScript: local variable and global variable.

There are some rules while declaring a JavaScript variable (also known as identifiers).

- Name must start with a letter (a to z or A to Z), underscore(_), or dollar($) sign.
- After first letter we can use digits (0 to 9), for example value1.
- JavaScript variables are case sensitive, for example x and X are different variables.

Syntax:

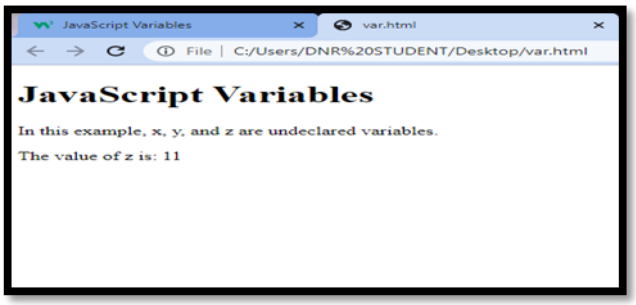
```
var x = 10;
var _value="sonoo";
```

Ex:

```
<!DOCTYPE html>
<html>
<body>
<h1>JavaScript Variables</h1>
<p>In this example, x, y, and z are undeclared variables.</p>
```

```
<p id="demo"></p>
<script>
x = 5;
y = 6;
z = x + y;
document.getElementById("demo").innerHTML =
"The value of z is: " + z;
</script>
</body>
</html>
```

Output:



OPERATORS IN JAVA SCRIPT:

Operators:-An operator is capable of manipulating a certain value or operand. Operators are used to perform specific mathematical and logical computations on operands. In JavaScript operators are used for compare values, perform arithmetic operations etc. Thereare various operators supported by JavaScript:

- Arithmetic Operators
- Assignment Operators
- Logical Operators
- Bitwise Operators

Arithmetic Operator: Arithmetic operators are used to perform arithmetic calculations suchas Addition, Subtraction, Multiplication and etc..., . For example,

Operator	Name	Example
+	Addition	x + y
-	Subtraction	x - y
*	Multiplication	x * y
/	Division	x / y
++	Increment (increments by 1)	++x or x++
--	Decrement (decrements by 1)	--x or x--

Assignment Operator: Assignment Operators are used to **assign** values to variables.

Operator	Name	Example
=	Assignment operator	a = 7; // 7
+=	Addition assignment	a += 5; // a = a + 5
-=	Subtraction Assignment	a -= 2; // a = a - 2
*=	Multiplication Assignment	a *= 3; // a = a * 3
/=	Division Assignment	a /= 2; // a = a / 2

Logical Operators: Logical operators perform logical operations and return a boolean value, either true or false.

Operator	Description	Example
&&	**Logical AND**: true if both the operands are true, else returns false	x && y
||	**Logical OR**: true if either of the operands is true; returns false if both are false	x || y
!	**Logical NOT**: true if the operand is false and vice-versa.	!x

Bitwise Operators: Bitwise operators perform operations on binary representations ofnumbers.

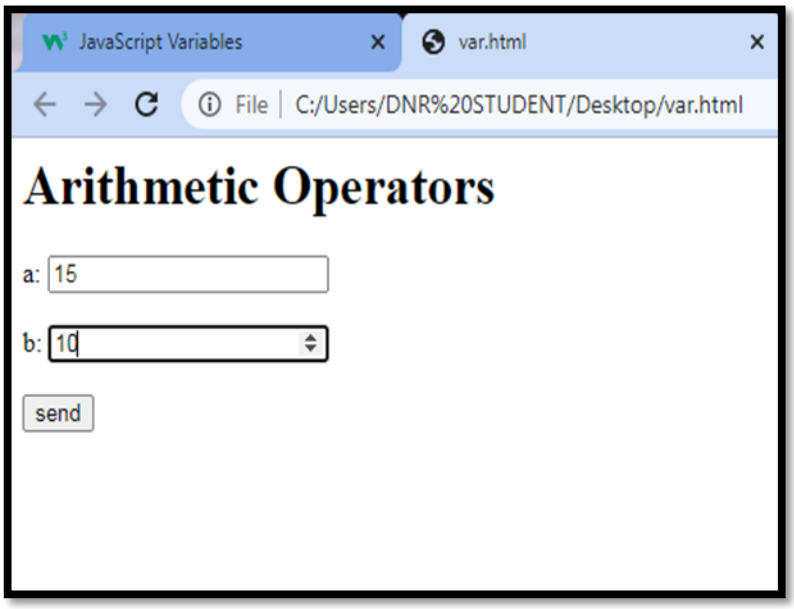
Operator	Description
&	Bitwise AND
|	Bitwise OR
^	Bitwise XOR
~	Bitwise NOT
<<	Left shift
>>	Sign-propagating right shift
>>>	Zero-fill right shift

Ex:

```
<html>
<body>
<h1>Arithmetic Operators</h1>
 a: <input type="number" name="a" id="a"><br><br>
 b:<input type="number" name="b" id="b"><br><br>
 <button onclick="arithematic(document.getElementById('a').value,
 document.getElementById('b').value)">send</button>
<script>
 function arithematic(a ,b)
{
 var sum = parseInt(a) + parseInt(b);
```

```
 document.write('Addition=',sum);
 document.write('<br/>');
 var sub=parseInt(a) - parseInt(b);
 document.write('Substration=',sub);
 document.write('<br/>');
 var mul=parseInt(a) * parseInt(b);
document.write('Multiplication=',mul);
document.write('<br/>');
var div=parseInt(a) / parseInt(b);
document.write('Division=',div);document.write('<br/>');
var mod=parseInt(a) %parseInt(b);
document.write(' Modulo Division=',mod);
 }
</script>
</body>
</html>
```

Output:



Addition=25

Substration=5

Multiplication=150

Division=1

Modulo Division=5

LOOPING STATEMENT IN JAVA SCRIPT:

The JavaScript loops are used to iterate the piece of code using for, while, do while or for-in loops. It makes the code compact. It is mostly used in array.

There are four types of loops in JavaScript.

- for loop
- while loop
- do-while loop
- for-in loop

for loop: The JavaScript for loop iterates the elements for the fixed number of times. It should be used if number of iteration is known. The syntax of for loop is given below.

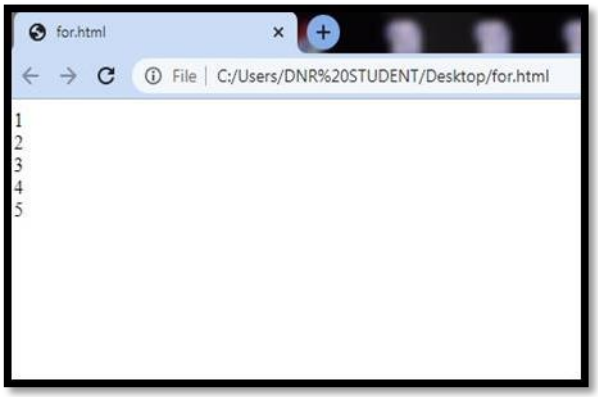
Syntax:

```
for (initialization; condition; increment)
{
   code to be executed
}
```

Ex:

```
<!DOCTYPE html>
<html>
<body>
<script>
for (i=1; i<=5; i++)
{
document.write(i + "<br/>")
}
</script>
</body>
</html>
```

Output:



While-loop: The JavaScript while loop iterates the elements for the infinite number of times. It should be used if number of iteration is not known. The syntax of while loop is given below.

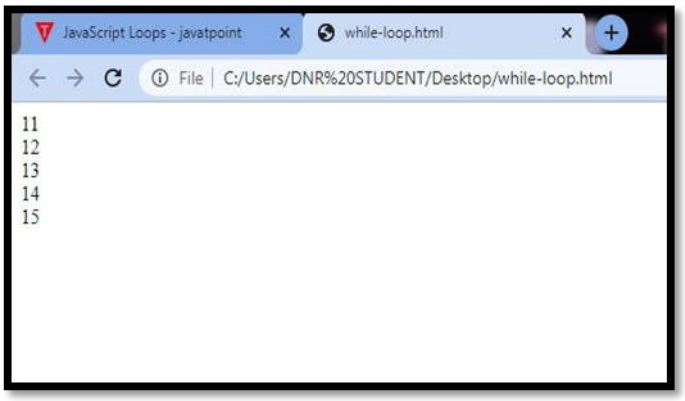
Syntax:

```
while (condition)

{

   code to be executed

}
```

Ex:

```
<!DOCTYPE html>
<html>
<body>
<script>
var i=11;
while (i<=15)
{
document.write(i + "<br/>");
i++;
}
</script>
</body>
</html>
```

Output:



Do-while loop: The JavaScript do while loop iterates the elements for the infinite number of times like while loop. But, code is executed at least once whether condition is true or false. The syntax of do while loop is given below.

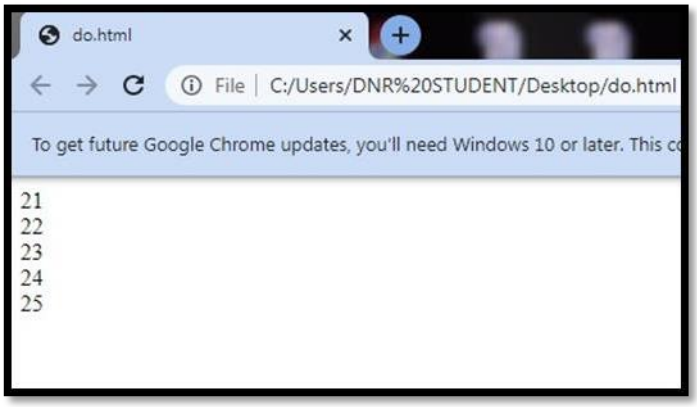
Syntax:

```
do{
   code to be executed
}while (condition);
```

Ex:

```
<!DOCTYPE html>
<html>
<body>
<script>
var i=21;
do{
document.write(i + "<br/>");
i++;
}while (i<=25);
</script>
</body>
</html>
```

Output:



for- in loop:

The JavaScript for in statement loops through the properties of an Object:

Syntax:

```
for (key in object) {
  // code block to be executed
}
```

DATA VALIDATION IN DHTML:

Validation is simply the process of ensuring that some data might be correct data for a particular application. Data validation is the process of ensuring that users submit only the set of characters which we require. It is not the process of ensuring that the data is any way accurate.

Ex: The following example has two text fields. One accepts names, the other accepts age.

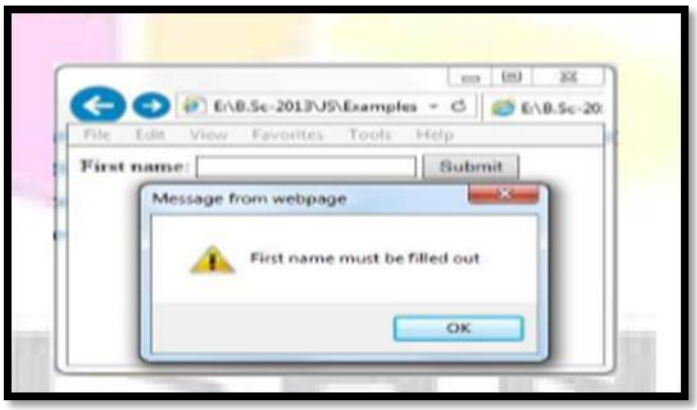
Both fields are validated using regular expressions and if the date is valid the contents of the form are transmitted in an email message.

Ex:1Required Fields: The function below checks if a field has been left empty.

If the field is blank, an alert box alerts a message, the function returns false, and the form will not be submitted:

```
<!DOCTYPE html>
<html>
  <head>
    <script>
      function validateForm(){
    var x=document.forms["myForm"]["fname"].value;
    if (x==null || x=="") {
    alert("First name must be filled out");
    return false;
   }
    }
</script>
    </head>
    <body>
<form name="myForm" action="demo_form.asp" onsubmit="return
validateForm()" method="post">
     First name: <input type="text" name="fname">

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

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

Output:

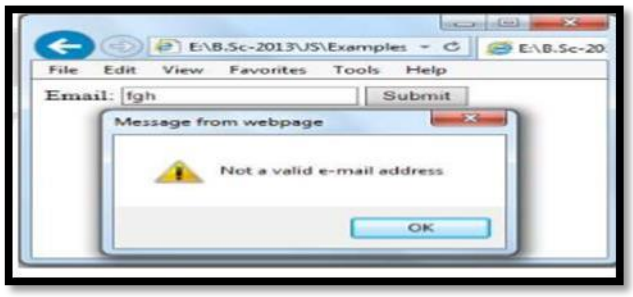


Ex:2 E-mail Validation:

The function below checks if the content has the general syntax of an email. This means that the input data must contain a @ sign and at least one dot (.). Also, the @ must not be the first character of the email address, and the last dot must be present after the @ sign, and minimum 2 characters before the end:

```
<!DOCTYPE html>
<html>
   <head>
      <script>
          function validateForm(){
 var x=document.forms["myForm"]["email"].value;
 var atpos=x.indexOf("@");
var dotpos=x.lastIndexOf(".");
 if (atpos<1 || dotpos<atpos+2 || dotpos+2>=x.length){
    alert("Not a valid e-mail address");
    return false;
   } }
      </script>
   </head>
   <body>
<form name="myForm" action="demo_form.asp" onsubmit="return
validateForm();" method="post">
  Email: <input type="text" name="email">
     <input type="submit" value="Submit">
      </form>
   </body>
    </html>
```

Output:



MESSAGE PASSING IN DHTML (JAVASCRIPT).

JavaScript provides three types of built-in window types that can be used from application code. These are useful when we need information from visitors to our site.

a. prompt() Method:

The prompt() method displays a dialog box that prompts the visitor for input. This method returns the string the visitor has entered. The general format is prompt(msg,defaultText);

Ex: Display a prompt box which asks the user for her/his name, and then write a greeting to the page:

```
<!DOCTYPE html>
<html>
  <body>
    <p>Click the button to demonstrate the prompt box.</p>
    <button onclick="myFunction()">Try it</button>
    <script>
function myFunction(){
  var x;
   var person=prompt("Please enter your name","Harry Potter");
   if (person!=null) {
  x="Hello " + person + "! How are you today?";
  document.write(x);
      }
}
</script>
  </body> </html>
```

Output:

b. confirm() Method:

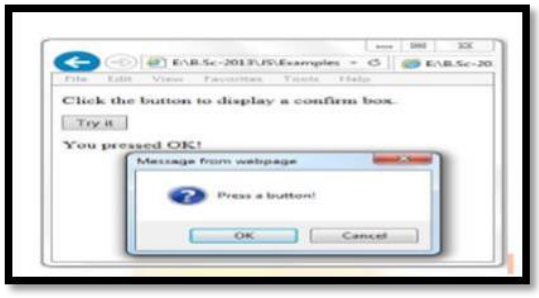
The confirm() method displays a dialog box with a specified message, along with an OK and a Cancel button. This method returns true if the visitor clicked "OK", and false otherwise. The general format is

confirm(message);

Ex: Display a confirm box, and alert what the visitor clicked:

```
<!DOCTYPE html>
<html>
  <body>
    <p>Click the button to display a confirm box.</p>
    <button onclick="myFunction()">Try it</button>
    <script>
    function myFunction(){
  var x;
            var r=confirm("Press a button!");
            if (r==true)
{
x="You pressed OK!";
            }
       else  {
               x="You pressed Cancel!";
            }
       document.write(x);
           }
     </script>
   </body>
</html>
```

Output:



c. alert() Method:

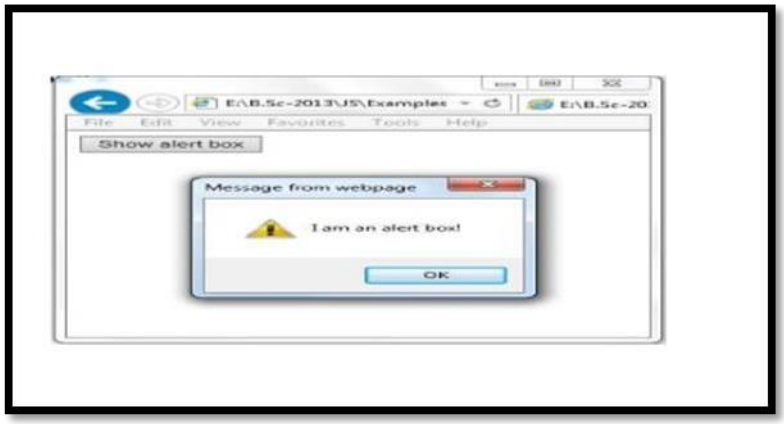
An alert box is often used if you want to make sure information comes through to the user. When an alert box pops up, the user will have to click "OK" to proceed. The general format is

window.alert("sometext");

Ex: The window.alert method can be written without the window prefix.

```
<!DOCTYPE html>
<html>
   <head>
      <script>
      function myFunction(){
   alert("I am an alert box!");
      }
      </script>
   </head>
   <body>
  <input type="button" onclick="myFunction()" value="Show alert box">
   </body>
</html>
```

Output:



HOW TO OPEN A NEW WINDOW:

JavaScript offers in-built methods to open and close the browser window to perform additional operations like robot window etc. These methods help to open or close the browser window pop-ups. Following are the window methods:

open()

close()

The window. Open method is used to open a new web page into a new window and window. Close method to close web page opened by window.open method. See the window. Open() method in

Window. open()

It is a pre-defined window method of JavaScript used to open the new tab or window in the browser. This will depend on your browser setting or parameters passed in the window.open() method that either a new window or tab will open.

This method is supported by almost all popular web browsers, like Chrome, Firefox, etc.

Syntax:

This function accepts four parameters, but they are optional.

window. open(URL, name, specs, replace);

Parameters List

Below is the parameters list of window.open() method. Note that - all parameters of this method are optional and works differently.

URL: This optional parameter of the window.open() function contains the URL string of a webpage, which you want to open. If you do not specify any URL in this function, it will open a new blank window (about:blank).

_blank	Passed URL will load into a new tab/window.
_parent	URL will load into the parent window or frame that is already opened.
_self	By passing this parameter, the URL will replace the previous output and a new window will open in the same frame.
_top	URL replaces any frameset that can be loaded.
Name	Provide the name of the new window to show the text or any data on it. (Note - not the title of the window)

Name: Provide the name of the new window to show the text or any data on it. (Note - not the title of the window)

The above-specified values are passed inside a single or double quote to the window.open() function at the name parameter place.

specs: This parameter contains the settings that are separated by the comma. Element used in this parameter cannot have whitespaces, e.g., width=150,height=100.

It supports several values.

replace: Like the other parameters of window.open() method, this is also an optional parameter. It either creates a new entry or replaces the current entry in history list. It supports two Boolean values; this means that it returns either true or false:

Ex:

```
<html>
<body>

Click the button to open new window <br><br>
```

<button onclick="window.open('https://www.google.com')"> Open Window </button>

</body>

</html>

Output:

SHORTS

MATHEMATICAL FUNCTIONS IN JAVA SCRIPT:

Mathematical Function: The Math functions consist of methods and properties. Following is the list of methods used with the Math object:

1) Math. round():

This method provides the value of the given number to a rounded integer. It can be written as: Math. round(x), where x is a number.

Ex:- Math. round(4.3);

Math. round(4.7);

Round of 4.3 is: 4 Round of 4.7 is: 5

2) Math.sqrt():

It gives the square root of a given integer. It can be written as: Math.sqrt(x), where x is a number.

Ex:- Math.sqrt(17);

Square Root of 17 is: 4.123105625617661

3) Math. Ceil():

It gives a smaller number, which is greater or equal to the given integer. It can be written as: Math.ceil(x); where x is a number

Ex:- Math. Ceil(4.6);

Ceil of 4.6 is: 5

4) Math. Random():

It provides a random number between 0 and 1. It can be written as: **Ex:-** Math.random();

Math. Random ();

Random Number is: 0.6028976753720041

5) Math.pow ():

It provides the value of x to the power of y. It can be written as: Math.pow(x, y), where x is a base number and y is an exponent to the given base

Ex:- Math.pow(3,4);

3 to the power of 4 is: 81

ARRAYS:

JavaScript array is an object that represents a collection of similar type of elements.

There are 3 ways to construct array in JavaScript

- By array literal
- By creating instance of Array directly (using new keyword)
- By using an Array constructor (using new keyword)

1) JavaScript array literal

The syntax of creating array using array literal is given below:

var arrayname=[value1,value2.... valueN];

2) JavaScript Array directly (new keyword)

The syntax of creating array directly is given below:

var arrayname=new Array();

Here, new keyword is used to create instance of array.

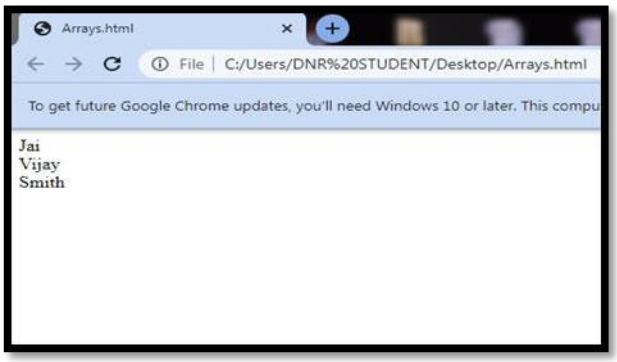
3) JavaScript array constructor (new keyword)

Here, you need to create instance of array by passing arguments in constructor so that we don't have to provide value explicitly.

Ex:

```
<html>
<body>
<script>
var emp=new Array("Jai","Vijay","Smith");
for (i=0;i<emp.length;i++){
document.write(emp[i] + "<br>");
}
</script>
</body>
</html>
```

Output:



STRING FUNCTIONS:

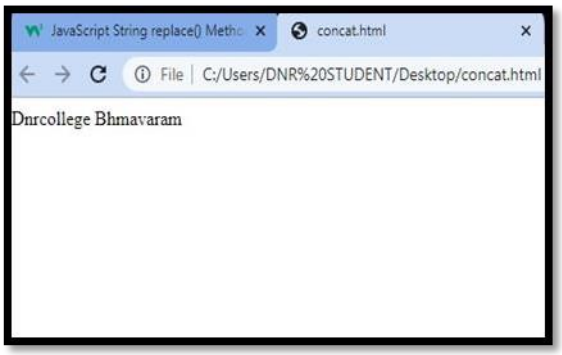
JavaScript String concat(str) Method:

The JavaScript String concat(str) method concatenates or joins two strings.

Ex:

```
<!DOCTYPE html>
<html>
<body>
<script>
var s1="Dnrcollege ";
var s2="Bhimavaram";
var s3=s1+s2;
document.write(s3);
</script>
</body>
</html>
```

Output:



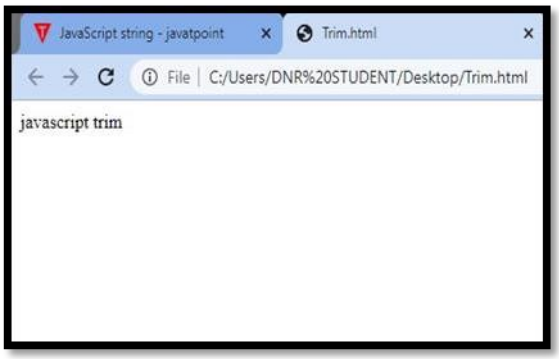
JavaScript String trim() Method:

The JavaScript String trim() method removes leading and trailing whitespaces from the string.

Ex:

```
<!DOCTYPE html>
<html>
<body>
<script>
var s1="     javascript trim    ";
var s2=s1.trim();
document.write(s2);
</script>
</body>
</html>
```

Output:



JavaScript String split() Method:

The split() method splits a string into an array of substrings.

The split() method returns the new array.

The split() method does not change the original string.

If (" ") is used as separator, the string is split between words.

Ex:

```
<!DOCTYPE html>
<html>
<body>
<script>
var str="Dnr College Bhimavram";
document.write(str.split(" ")); //splits the given string.
</script>
</body>
</html>
```

Output:

PROMT()METHOD:

The prompt() method displays a dialog box that prompts the visitor for input. This method returns the string the visitor has entered. The general format is

prompt (msg,defaultText);

Ex: Display a prompt box which asks the user for her/his name, and then write a greeting to The page:

```
<!DOCTYPE html>
<html>
  <body>
    <p>Click the button to demonstrate the prompt box.</p>
    <button onclick="myFunction()">Try it</button>
    <script>
      <!---
function myFunction(){
 var x;
    var person=prompt("Please enter your name","Harry Potter");
    if (person!=null) {
    x="Hello " + person + "! How are you today?";
    document.write(x);
      }
 }
  //-->
</script>
  </body>
</html>
```

Output:

CONFIRM()METHOD:

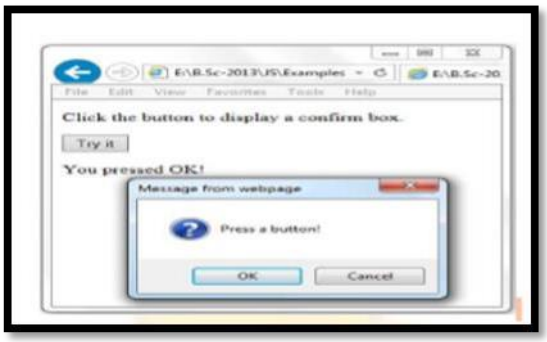
The confirm() method displays a dialog box with a specified message, along with an OK and a Cancel button. This method returns true if the visitor clicked "OK", and false otherwise. The general format is confirm(message);

Ex: Display a confirm box, and alert what the visitor clicked:

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

```
    <p>Click the button to display a confirm box.</p>
    <button onclick="myFunction()">Try it</button>
    <script>
    function myFunction(){
      var x;
           var r=confirm("Press a button!");
if (r==true)
{

 x="You pressed OK!";
           }
      else  {
              x="You pressed Cancel!";
           }
      document.write(x);
         }
     </script>
   </body>
</html>
```

Output:



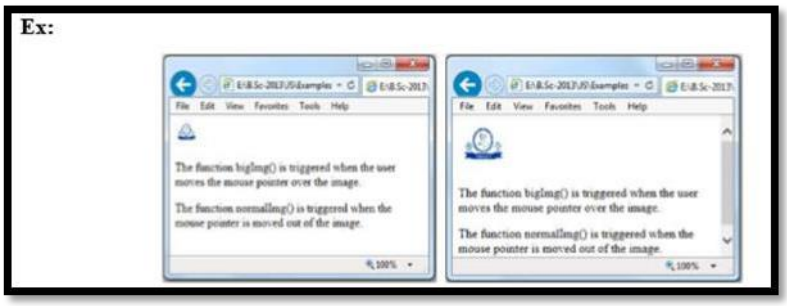
Rollover Buttons.

The most common usage of dynamic HTML is the image rollover. The technique is used to give visual feedback about the location of the mouse cursor by changing the images on the page as the mouse moves over them. This is a highly effective technique, especially where images are used as the hyperlinks in a menu or where an image map is being used.

The JavaScript code does not directly manipulate the image. If we need to change the actual image then we has to investigate ActiveX, flash or Java programming.

rollover is simpler because it uses two image files which it swaps between as the mouse is moved

Ex:



One image is created for the inactive static when the mouse is not over it. A second image is created for the active state when the mouse cursor is placed over it.

1. The onload event happens when the page is first loaded into the browser.
2. OnMouseover calls a JavaScript function when the cursor passes over the image.
3. OnMouseOut calls a function when the cursor moves away from the image.

MOVING IMAGES:

Images (layers) can around repeatedly but doing so takes up processor cycles. It is more user-friendly, if our images only move for a restricted amount of time such as when the page is first loaded (or) when the user specifically triggers the event.

Each layer can be positioned on the screen by changing the offset of the top left corner of the layer. The HTML code creates a division of the page named logo and positioned at pixel 5,100. The layer must be positioned absolutely so that the browser does not mess up the look of the page and must be visible. The sole content of this division is an image:

```
<div id=”logo” style=”top: 5; z-index: 4; left: 1000; visibility: visible; position: absolute;”>
  <img src=”./bullet.jpg”>
</div>
```

Use a division to place the image rather than a Netscape-specific layer. If we want create a browser-neural version then modify the code using the techniques.

```
var count=0;
function FlyLogo(){
if(count < 5){
  if(document.layers[“logo”].left == 200){
count++;
   document.layers[“logo”].left = 1000;
  }
  document.layers[“logo”].left - = 10;
  setTimeout(‘FlyLogo()’, 200);
```

```
 }
 else
document.layers[“logo”].left=200;
 }
```

When the function is called it checks the counter to make sure that it should run. If the counter is equal to 5 then the left edge of the logo is placed at pixel 200 and no more processing is performed by this routine.

If the counter is less than 5, the layer containing the logo will be moved. If the left hand side of the layer is at pixel 200 then the image has finished moving across the screen.

The counter is incremented and the layer is repositioned to pixel 1000. However, if the edge of the logo is not at position 200 is repositioned 10 pixels to the left of its current location.

The FlyLogo() routine then calls itself using the built-in setTimeOut() call. This takes the name of the function and a delay in milliseconds. It will not run the routine until after the delay has eased. In this case over image moves 10 pixels left every 200 ms.

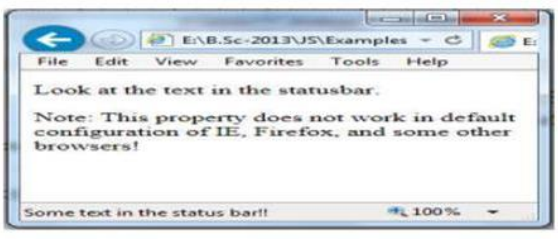
How to display a message on Status Bar:

The status property sets or returns the text in the status bar at the bottom of the Browser. The general format is window. Status

Ex:

```
<!DOCTYPE html>
<html>
   <body>
      <script>
        <!--
 window.status="Some text in the status bar!!";
        //-->
      </script>
 <p>Look at the text in the statusbar.</p>
 <p>Note: This property does not work in default configuration of IE, Firefox,
and some other browsers!</p>
   </body> </html>
```

Output:



UNIT-IV

WORDPRESS

WORD PRESS: Word press is an open source Content Management System (CMS), which allows the users to build dynamic websites and blog. 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. It is developed in 2003 by American blogger Matt Mullenweg and British blogger Mike Little.

FEATURES OF WORDPRESS: There are many reasons to use WordPress in today's scenario as it provides a great help to its users in all respect.

1. Creating a website without logical skill: The most interesting and best part about Word Press is that anyone can create his/her own website on this without any coding or designing skills. Out of million users on Word Press, most of them are neither designers nor programmers. To create a website on Word Press you only need an Internet connection and a web browser.

2. Provide a range of Themes: It gives thousand types of template options to give any type of feel and look to a website. They are very easy to customize as they come with their own option panel which allows users to select colors, background, sliders, fonts, logos and many more.

3. Plugins to add functionality: It provides both free and premium types of plugins. These plugins add extra functionality or may give a whole new platform to a website. It allows a user to add photo galleries, shopping carts and much more.

4. Free and Open-source platform: WordPress is a free and open-source software. It is free to install and use. As a new user you can easily create a website of your own that too absolutely free. It never has any type of hidden charges once a user started using it. Being an open-source, you can alter the source-code according to your need. Anyone can become a contributor to WordPress by answering questions, creating themes or writing plug-in.

5. Adding a blog is very easy: WordPress came into existence with blogging and still mostly consider it as a blogging site. It provides all type of blog solutions from designing, styling to the anti spam solutions. It will meet your demands in all possible ways.

6. More secure: It only takes 5 minutes to install, and is really fast in updating for security and new functionality. Upgrading to a new version is automatic and very fast as it provides a one link click upgradation. It constantly updates its site and software to prevent from hacking.

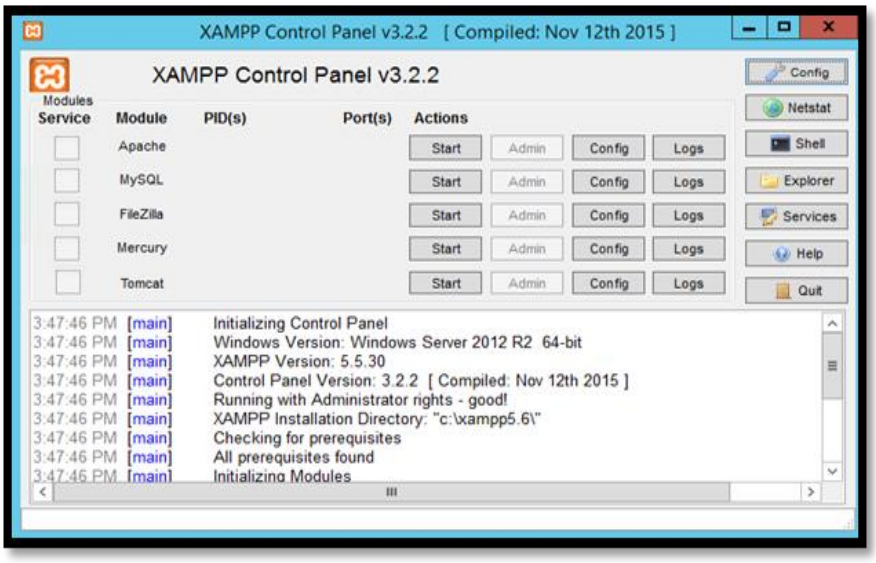
7. Google and WordPress: In a press conference in 2009, Matt Cutts, the head of Google's web spam team said that "Google loves WordPress". Even WordPress simple plugins and themes are also really attractive to Google. WordPress is a better option when doing SEO due to its framework which is easy to crawl.

8. Inserting Multimedia: A user whether writing a blog or creating a website, need to insert videos, pictures to make its content more powerful. WordPress makes it very easy to upload a multimedia file or to make some editing in files like images.

9. Mobile-Friendly: A website needs to be responsive for small screens like tablets, smartphones to reach all its users. WordPress is mobile-friendly as its themes are made responsive.

10. Easy to use: WordPress dashboard always look the same. You don't need to hire an expert to learn how to use WordPress.

XAMPP : XAMPP stands for X-operating system, Apache, Mysql, Php, and Perl. It is an open resource platform that is readily available absolutely free downloading. It includes attributes like supporting Perl, mercury mail, and also Filezilla, and so on. It is multiplatform and is supported through many operating systems such as Windows, MacOS, and Linux. It is supported by means of many report codecs that add to its robustness. It is easy to install and configure. In this the Control Panel makes it easy to manage and implement.



WAMP: WAMP stands for Windows, Apache, Mysql, Php. It is easy to download and to setup configuration in. Its server has a graphical user interface to switch on or off individual component software while it is running. Its server provide an option to swap among many versions of Apache, many versions of PHP and many versions of MySQL all installed which

provide more flexibility towards developing while XAMPP Server doesn't have such an option.

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

DIFFERENCE BETWEEN XAMP AND WAMP

XAMPP	WAMP
It is a cross-platform software package supported by platforms like Mac OS, Linux, and Windows.	Its local server is only supported by Windows Operating system.
It is easy to download and install but may differ for different platforms.	It is easy to download and install and also light- weighted.
It uses MariaDB, which is an RDBMS for storing and retrieving operations on data.	It uses MySQL, which is an RDBMS for storing and retrieving operations on data.
The programming or scripting languages used for development in XAMPP are Perl and PHP.	The programming or scripting languages used for development in WAMP is PHP.
It uses the Apache Web server.	It uses the Apache Web server.
It is more powerful and resource taking as compared to WAMP.	It is less powerful and resource taking than XAMPP.
It has SSL feature.	It does not have SSL feature.
It is available in 64-bit system only.	It is available in both 64 bit and 32-bit system.

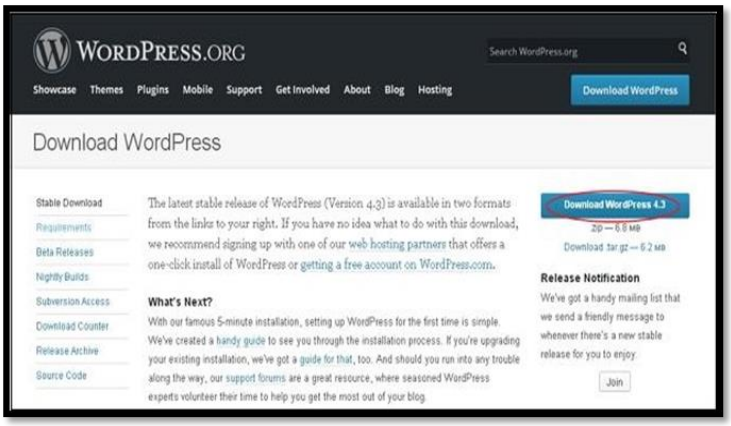
INSTALLING AND CONFIGURING WORDPRESS

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 –



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.

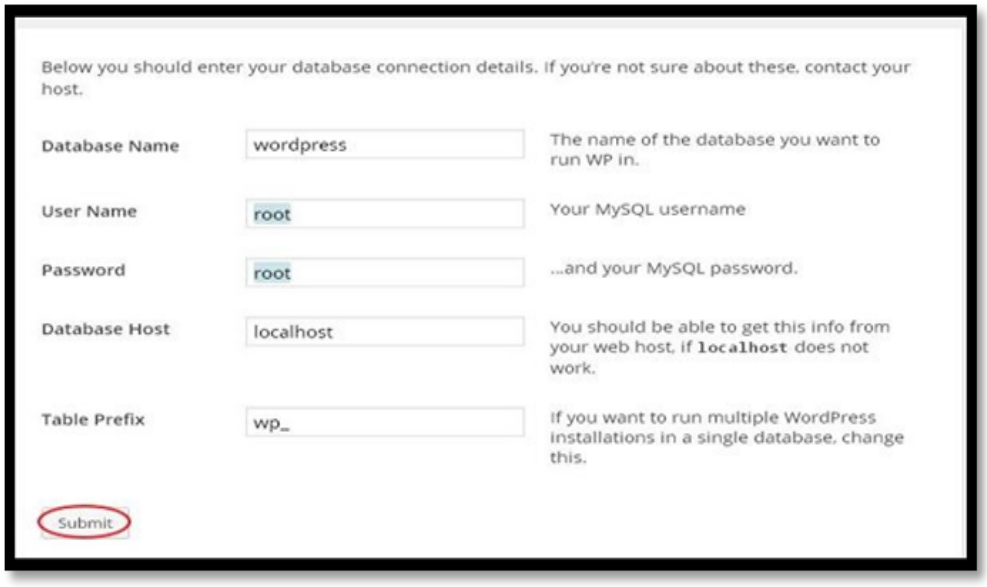
Step (2) − Open your browser and navigate to your WordPress file path, then you will get the first screen of the WordPress installer as shown in the following screen. In our case, the path is **localhost/< Your_wordpress_folder >**.

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.

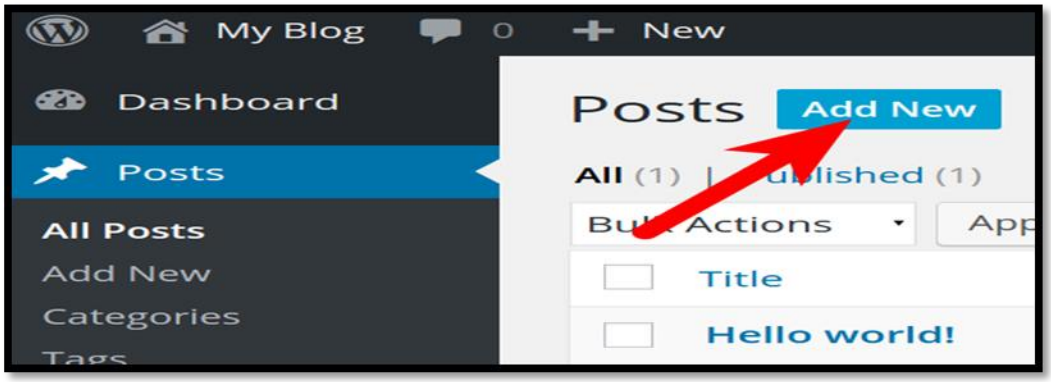
Enter the username and password which you had mentioned during installation as shown in step 6 and click on the **Log In** button.

POSTS AND PAGES: Posts allow you to write a blog and post it on your site. They are listed in reverse chronological order in front page of your blog. Posts are the site content which are published on a site with an exact date and time. They can be categorized in a systematic way on the basis of category and tags. If you are a blogger, then you'll write posts to update your new blog. They are listed in a reverse chronological order on a site.

Working with posts:

ADDING NEW POST:

Go to **Posts** > **Add New.**



Look at the above snapshot, editor page will appear in front of you. There are two fields,

Title - Enter the title of your post

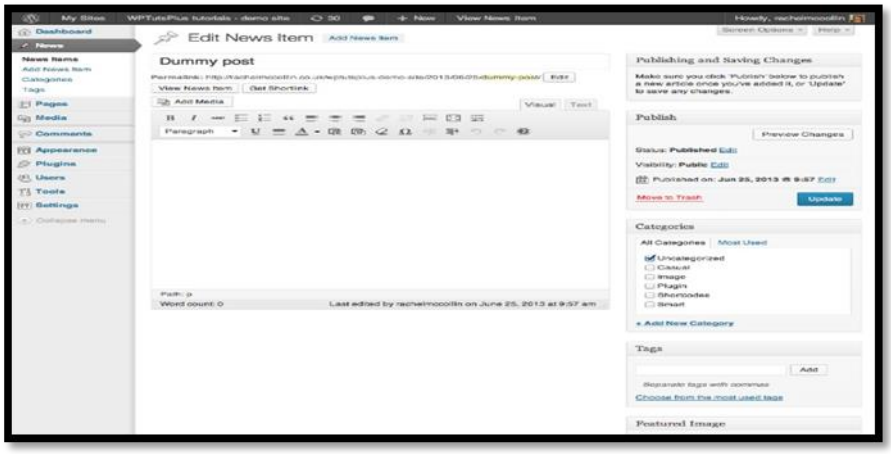
Content - Here you have to write content of your post.

Preview - This button allows you to have a look on your post before publishing it.

EDITING POSTS: Following are The simple steps to Edit posts in Word press.

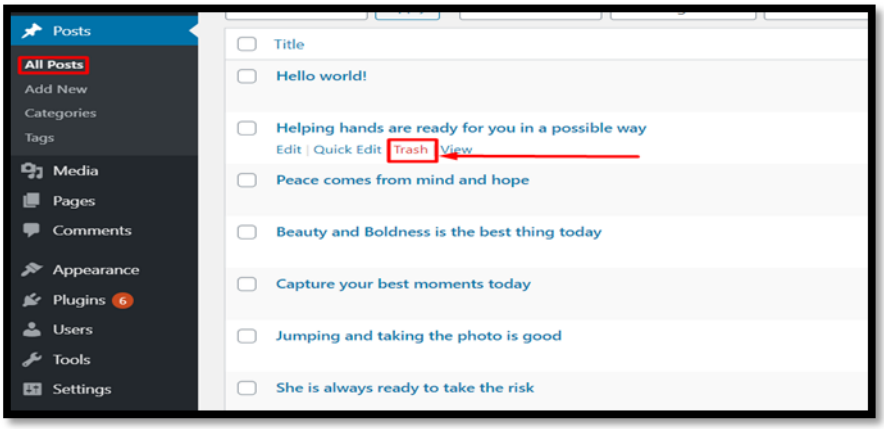
(i) Click on posts→All posts in Word Press.

(ii) When the cursor hovers on the post, then a few options get displayed below the post name. There are two ways to edit the post i.e., **Edit and Quick Edit.** Choose any one edit or change the content or title of the post as per your needs, and then click on update button.



DELETING A POST: Following are the steps to delete Posts in Word press.

1. Click on posts→All posts in Word Press.
2. When the cursor hovers on the Post, then a few options get displayed below Post name Click on **Trash** option to delete the Post.
3. You can check your Post lists to confirm if the Post is deleted or not.

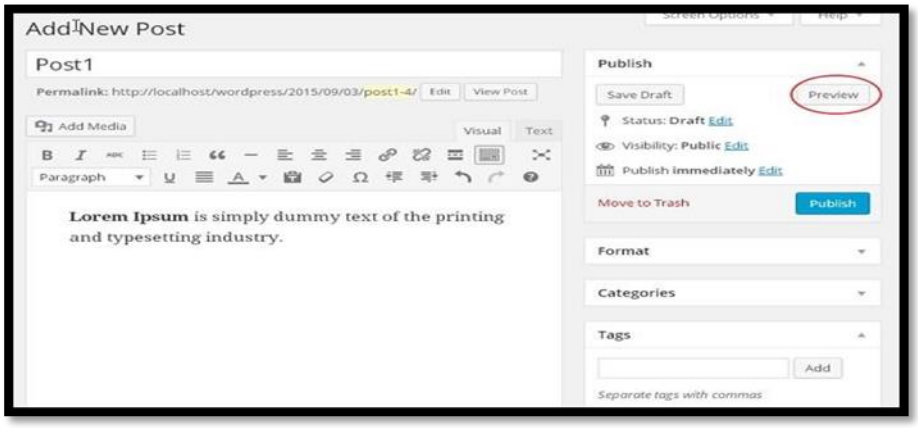


<u>PREVIEW POST</u>: Preview option helps to see you your post before publishing it to the public. It is better to have a look on your post that how it looks before publishing it to the world.

Following are the steps to preview posts in word press

1. Click on posts→All posts in Word Press.
2. When the cursor hovers on the Post, then a few options get displayed below Post name. Click on the View or else you can view your post directly while editing or adding post by clicking on the Preview button.

3.You can view your post, when you click on View or Preview.

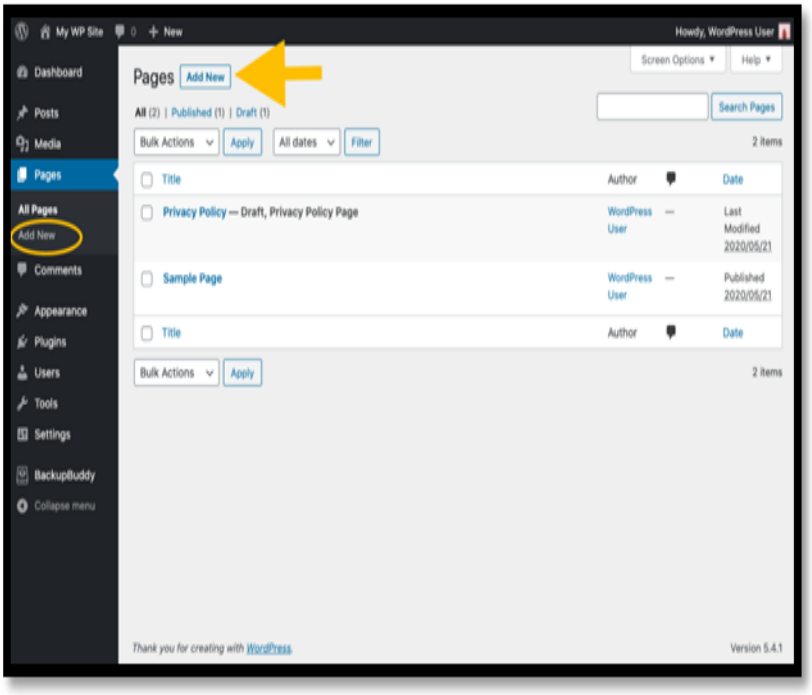


PAGES: Pages are different from Posts. They are static and they do not change often. Some examples of pages are About, Contact, etc. you can add pages containing information about you and your site.

WORKING WITH PAGES:

ADDING A NEW PAGE:

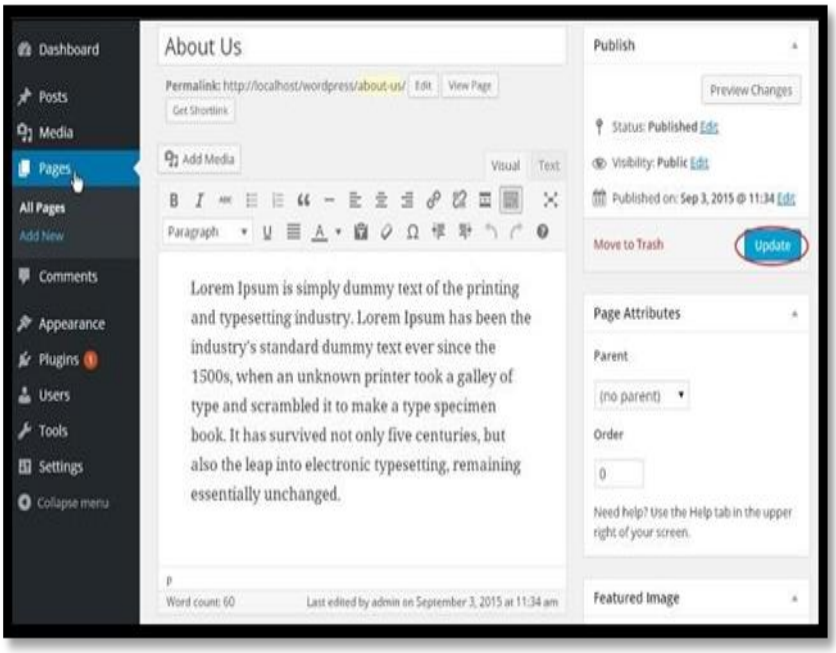
Go to **Pages** > **Add New**



Look at the above snapshot, this is the editorial page where you can give a title and content to your page. Here you can upload media, write content, add a title to your page. Click on Publish button once you have completed the page.

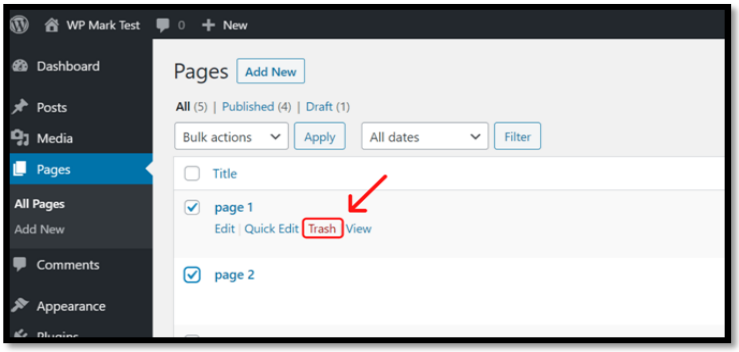
EDIT PAGE:

There are two options to edit. One is Edit and other one is Quick Edit. **Edit and Quick Edit.** Choose any one edit or change the content or title of the Page as per your needs, and then click on update button.



DELETING A PAGE: To delete a page, go to Pages > All Pages

Click on the trash option to delete a page.



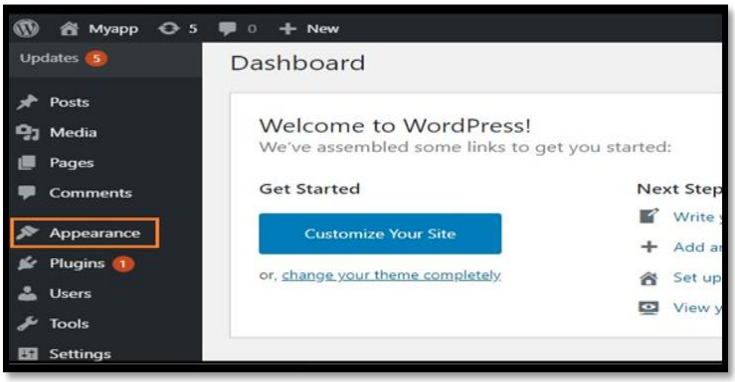
MENUS IN WORDPRESS:

The Menu is a list of options presented to the user. They are mostly used as a navigation building method on WordPress websites. Usually, the common content management system's installation is followed by a default menu that provides links to pages at the top level. These menus can be used at the top of the sites with links. It is generally used as a horizontal menu with links like home, email, contact, or pages.

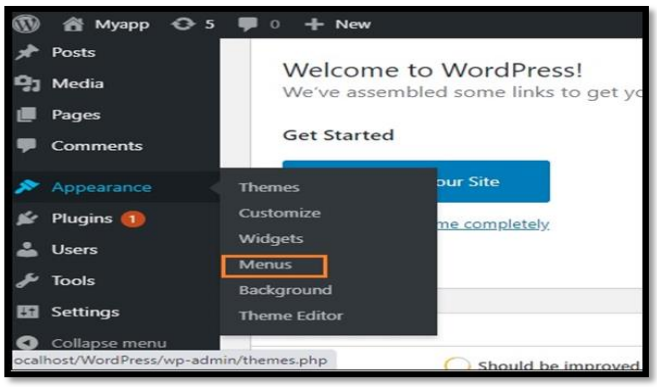
Defining a WordPress Menu

Firstly, we need to define a WordPress menu before adding items to it or use it. We can use the following steps to define a WordPress menu.

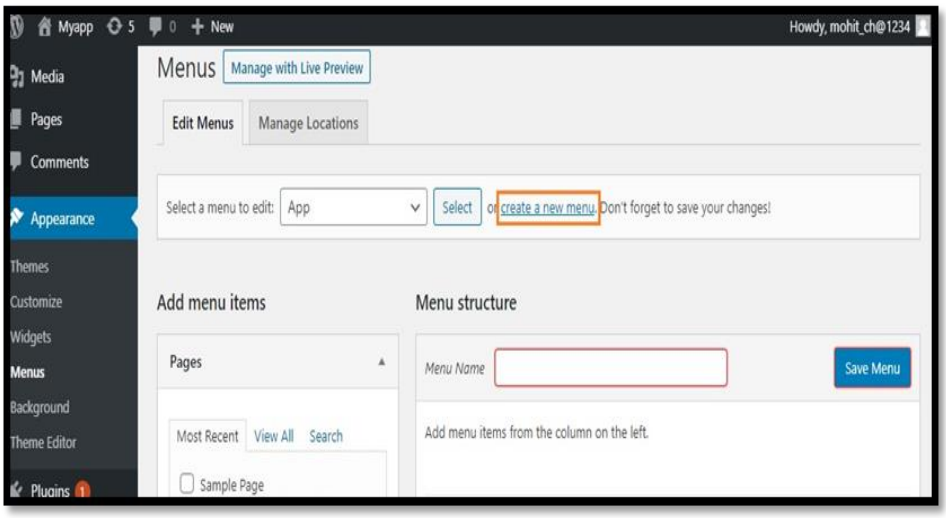
Step 1: First, log in to the WordPress Dashboard and click on the "Appearance" menu option, as shown in the below screenshot.



Step 2: The "Appearance" menu option opens a sub-menu where we choose the "Menus" option to bring up the Menu Editor.

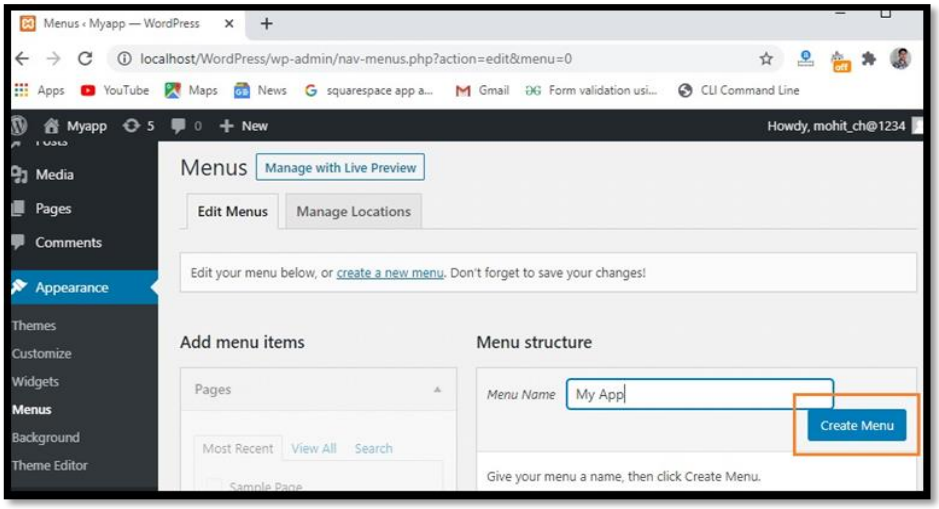


Step 3: After selecting the "Menus" option, choose the "Create a new menu" option shown in the red rectangular box at the top of the page.



Step 4: Now, enter the name to create a new menu in the "Menu Name" box.

Step 5: After that, click on the "Create Menu" button to create a menu.

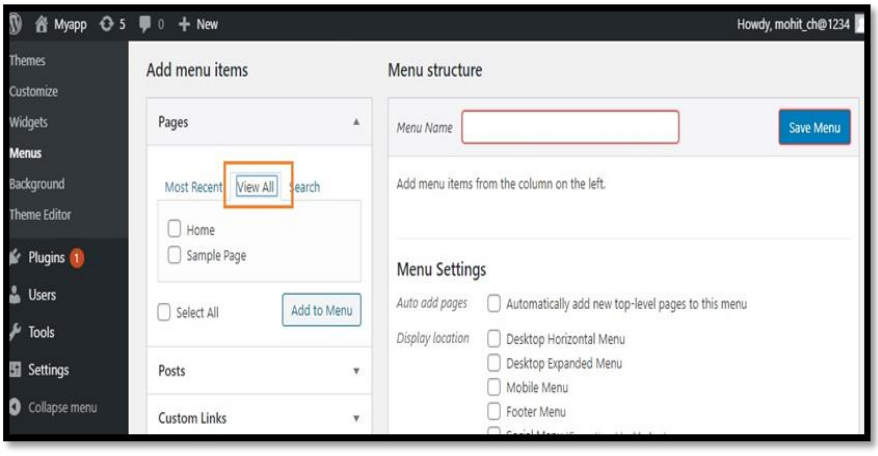


Adding Menu Items

In this section, we can add several link type options to the menu like Pages, Posts, Custom Links, and Categories. The following are the steps to add items in the menu.

Step 1: Firstly, Find the pane entitled Pages.

Step 2: Choose the "View All" link to see a list of all recently published pages on the WordPress site.



Step 3: Now, we can add Pages by clicking the checkbox corresponding to each page title.

Step 4: To add our selection to the menu that we created in the previous step, click on the "Add to Menu" button shown in the red rectangular box at the bottom of this page.

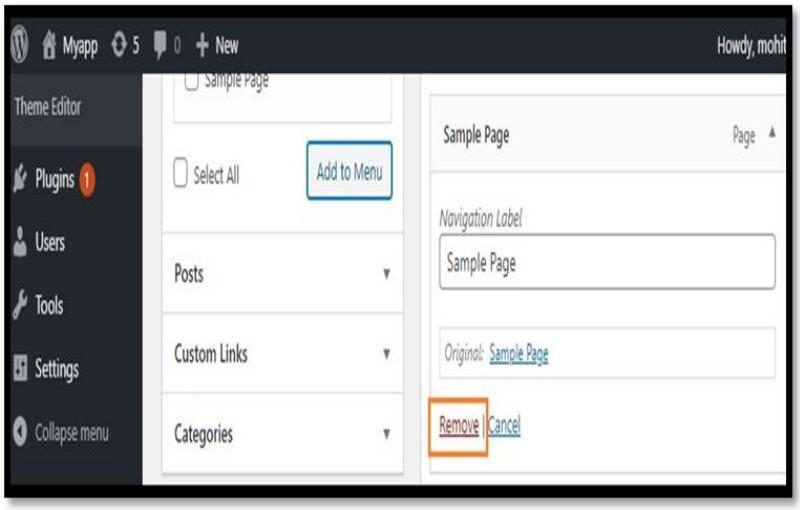
Step 5: After that, click on the "Save Menu" button.

Deleting a Menu Item: If we want to delete an item form the menu, we can use the following steps:

Step 1: First, find the menu items that we want to delete in the menu editor window.

Step 2: Next, click the arrow icon at the top right corner of the menu item.

Step 3: Click on the "Remove" option. The menu item or box will be removed immediately.



Step 4: Finally, click on the **"Save Menu"** button to save the changes.

WIDGETS IN WORDPRESS:

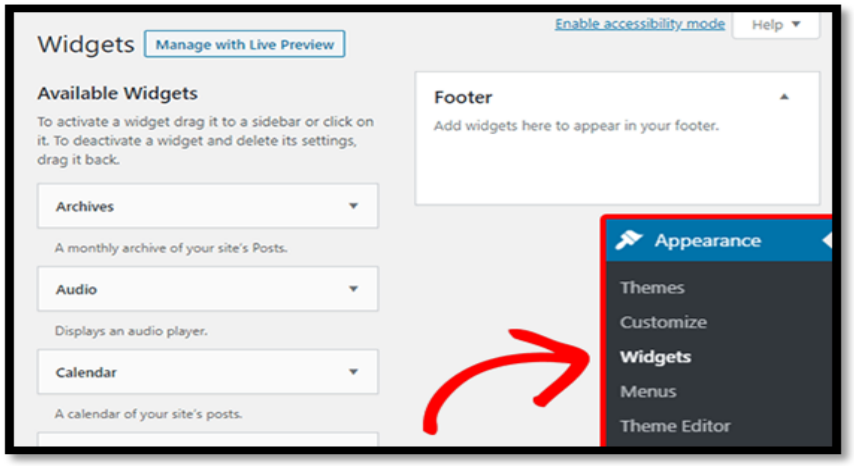
- In WordPress, widgets are blocks of content that you can add to your site's sidebars, footers, and other areas.
- Ever visit someone's blog and see a photo, signup form, or menu in the sidebar.
- Each widget can add a feature or function to your site, without having to write any code.
- Common widgets to add to your site are menus, popular post lists, calendars, banner ads, social icons, and more.
- Where can you add these widgets on your site? It all depends on your WordPress theme. Many themes have sidebar and footer widget areas. Some also allow you to place widgets in the header, homepage, or other areas of your site.
- WordPress widgets were created to provide a simple and easy way for WordPress users to control the design and content of their site without having to code.
- Most WordPress themes support widgets. Depending on your WordPress theme, widget areas may be in the header, footer, sidebar, right below your blog post content, and almost any other area.
- WordPress by default comes with several widgets including categories, tag cloud, navigation menu, calendar, search bar, recent posts, and more.
- Many WordPress plugins also add their own widgets which you can use to add plugin functionality to different sections. For example, you can add an email newsletter form to your sidebar, or add a

contact form to your footer with a plugin like WPForms.

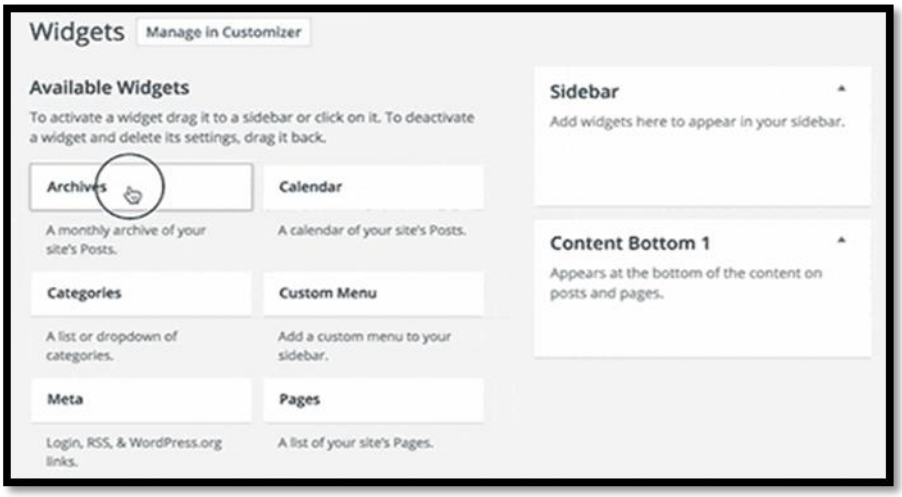
- Widgets can be really helpful in helping your visitors to navigate your site and encouraging them to take actions like signing up for your newsletter or following you on social media.

How to Add Widgets in WordPress

- When you first start a blog, setting up your widgets is an important step.
- Widgets are easy to use and can be added by dragging and dropping them into one of your theme's widget areas.
- You can find the list of available widgets on your own site by going to the Appearance » Widgets page from the admin sidebar.
- Once you're there, you'll see all the available widgets on the left. Your theme's widget areas will be listed on the right side.



- Under each of your installed widgets, you see a brief description that helps to explain its purpose and use.
- To add a widget to a widget area on your site, simply drag and drop the it from the list on the left to an area on the right.



- You can also move widgets up and down to rearrange them.
- Each widget comes with its own settings which you can see after adding the widget.
- Don't forget to click on the Save button to store your widget settings.

SHORTS

SYSTEM REQUIREMENTS FOR WORDPRESS:

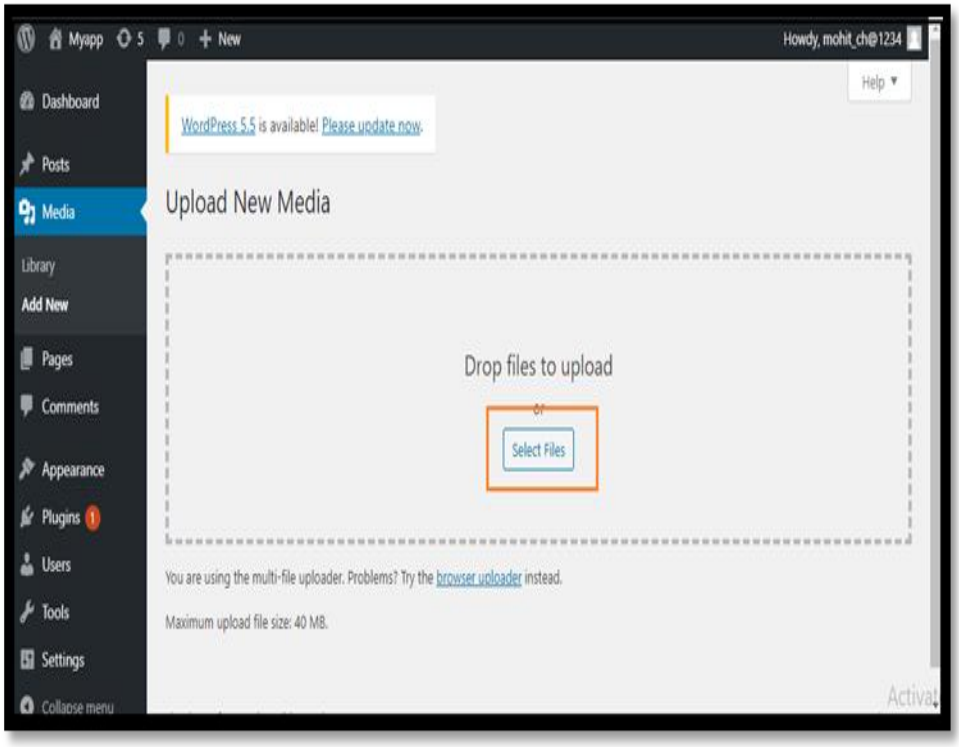
System Requirements for WordPress

- **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+

ADDING THE MEDIA FILES:: Now, we are going to know how to add the Media Files in WordPress. WordPress permits us to add various media files, such as **images, videos,** and **audios**. The following are the steps to add Media Files in WordPress.

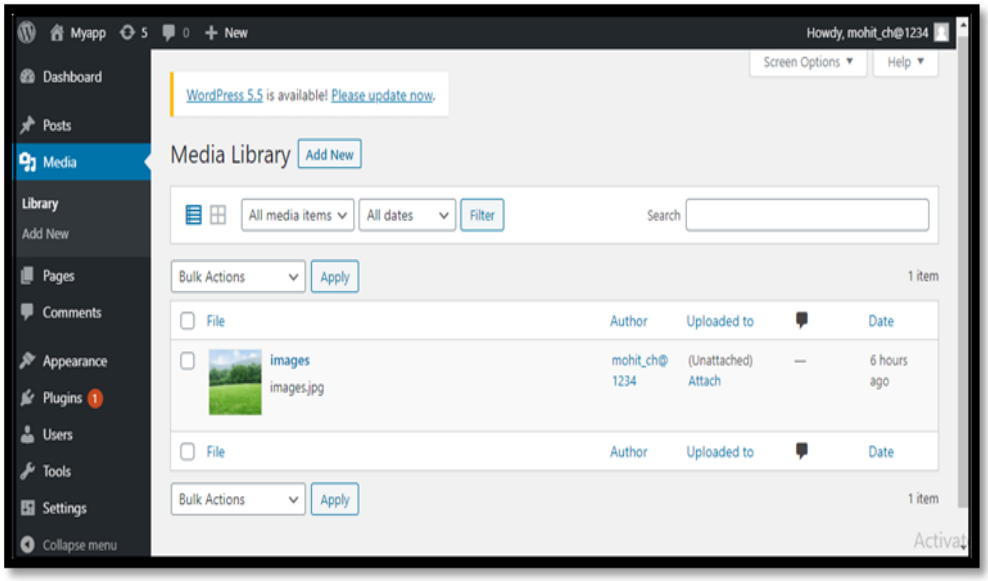
Step 1: Firstly, go to the **WordPress dashboard** and select the **"Media"** option, and then click on the **"Add New"** option.

Step 2: After clicking on the **"Add New"** option, click on the **"Select Files"** option to select the files from our local storage, as shown in the following screenshot.



Step 3: After clicking on the Select Files, add the media files by selecting them and then click on the open button.

Step 4: Now, we will see the media files list that we added in the list.



ADMIN PANEL:

- 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.
- Dashboard can be categorized as following.

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:

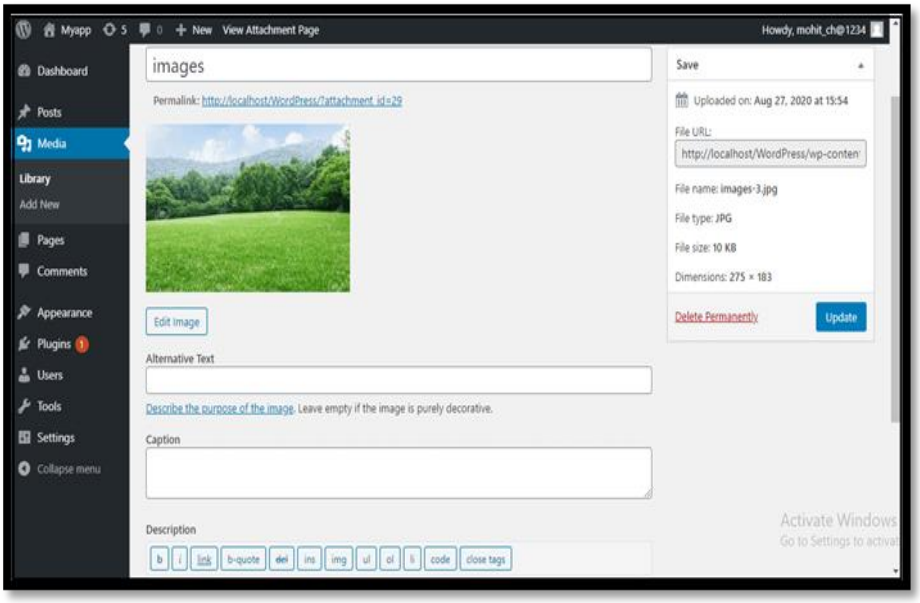
This section gives an overview of your blog's posts, number of published posts and pages, and number of comments.

EDIT MEDIA FILES:: Here, we are going to know how to Edit Media Files in WordPress. We can handle information about the Media File, which is stored in the Media Library. The following are the steps to edit the Media Files in WordPress.

Step 1: Firstly, go to the WordPress dashboard and select the "Media" option, and then click on the "Library" option. After that, click on the media file name or the edit link.

Step 2: Here, we will see the **Media Files** list. Now, we can select an image that we want to edit.

Step 3: Finally, we can see the media page with some options shown on the site. The options are as follows:



URL: It helps us to read only a link from the media file.

Title: It shows the media name. If the themes and plugins are designed to be displayed, the title will often show up in the galleries and attachment pages.

Permalink: It is the media attachment page URL. It is also a link to see the attachment page of the media file.

Edit Image button: It helps to edit the position of the image like rotate anti-clockwise, rotate clockwise, flip vertically, horizontally, crop, flip, and more.

Caption: It allows us to provide a detailed explanation of the media file.

Alternate Text: The alternate text for the image is used to describe media and is also used for the availability of images.

Description: It allows us to explain the media file.

WORDPRESS THEMES

THEMES: With the help of WordPress theme, you can design the layout and appearance of your website in the front-end. When WordPress is installed, it has a pre-installed theme which is very simple and unattractive. But no one wants a website to be that much simple. To make it more attractive one can install and apply different themes from WordPress.

Themes provide a style (including font styling), colors, page layouts, widget positions, etc. You can change the theme of your site without changing the content of the site. Themes provide a unique look to your site.

Types of WordPress theme: There are thousands of themes available on the WordPress. Some are free and some are paid. You can choose a theme according to your choice.

Free: WordPress provides many good looking themes absolutely free in the WordPress theme directory. If you are using your site for a small audience or yourself, go for free one. Because there are some chances that same theme is being used by another site.

Premium: Premium themes may cost you starting from $1,000. Prize varies depending upon the design and requirement of your site. One advantage in premium themes is that they provide a reliable code base and offer support. It gives you a more enhanced security.

How to make a WordPress theme: Building a WordPress theme is somewhat technical work. To build a theme you should have a little knowledge about the following development languages.

HTML

CSS

JavaScript

PHP

Database

You should know how to use a web server.

Selecting WordPress Theme: WordPress theme should be selected very wisely, so that it perfectly fits on your site. You can select any one either paid or free. It should be complemented to your site. Based on the following points you can select theme for your website.

Simplicity: Your website should look good, but for that you shouldn't compromise with your site's functionality. Theme styling should not be complicated.

Responsive: Your theme should be compatible with all type of screen sizes. A large number of traffic generates from devices other than laptops and PCs. Google also gives better ranking to the responsive sites. Lot of themes is by default responsive but still there are some themes which are not responsive. So look for a theme which is responsive to make your website accessible by all device users.

Browser Support: Make sure the theme you select is supported by all the browsers. Before selecting it, test it on different browsers for mobiles as well as laptops like Chrome, Internet Explorer, and Firefox.

Plugin Support: Plugins are very important for a WordPress site. They enhance the functionality of your site. Make sure that your selected theme must support all the popular plugins.

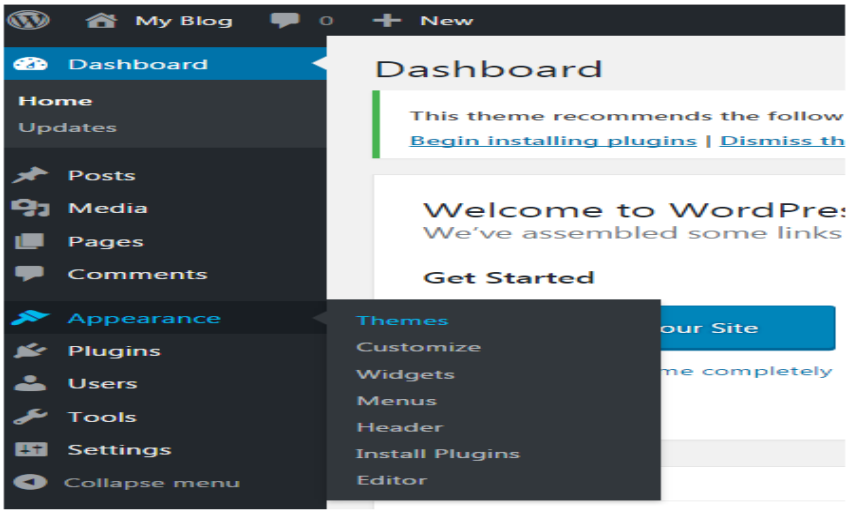
Multilingual Support: Your theme should support multilingual to support different languages for users.

Help Option: Only some developers provide support option with free themes, not all. So, if you messed up with your site then you may have to solve it on your own. So rather than facing all these problems, you should better choose a theme which has a good documentation and support help option.

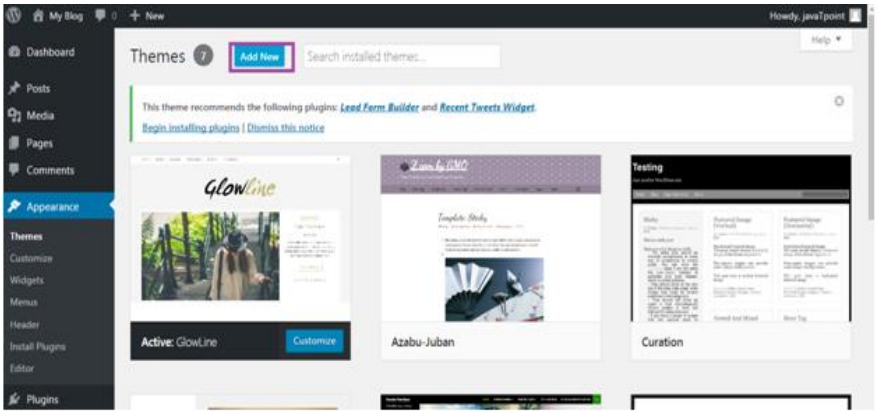
Ratings and Reviews: Before downloading a theme, have a look on its rating and reviews. A theme has both good and bad reviews, but if there are more negative reviews then read it carefully and consider it once again.

HOW TO INSTALL WORDPRESS THEMES:: For theme installation, first you need to select a theme either free or a premium one. There are a lot of sites from where you can download a theme. Theme will be downloaded in zip format. The next process is installation process. Installation has some steps to be followed.

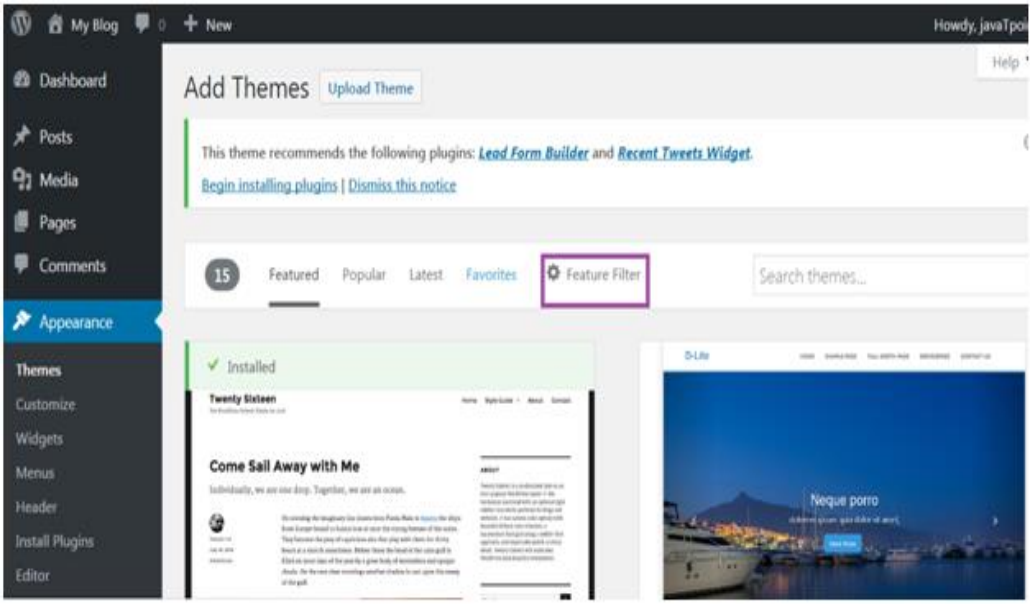
Installing WordPress using admin theme search: To install free themes from **WordPress.org** directory, login into your account and click **Appearance > Themes**.



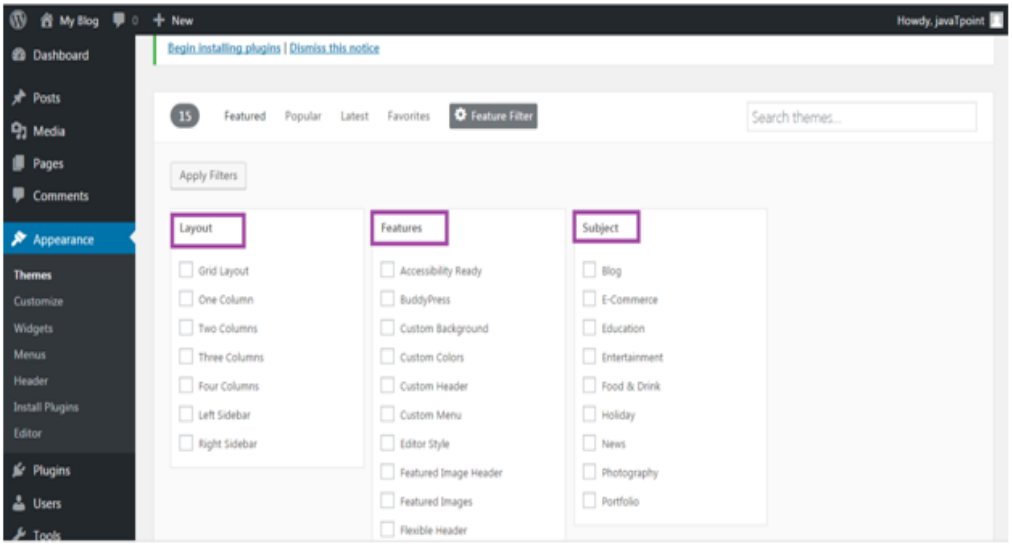
Look at the above snapshot, click on **Themes** option.



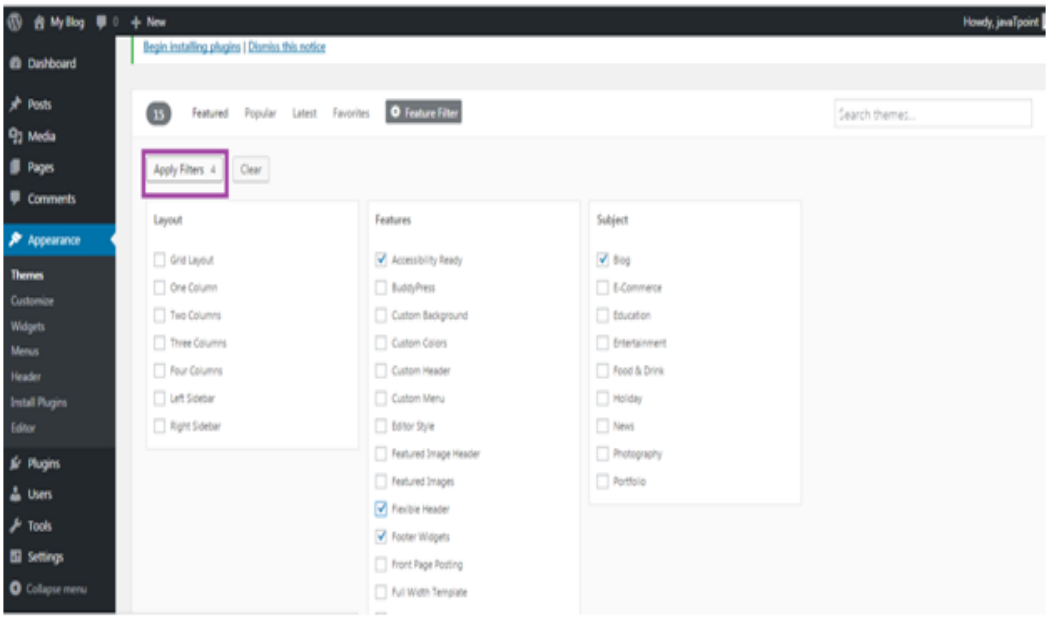
Click on Add New button to **Add New themes**.



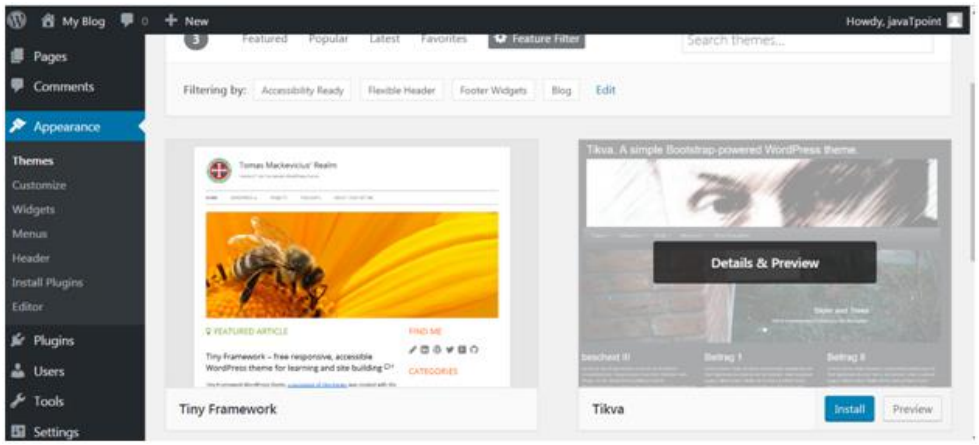
Look at the above snapshot, here you'll see an option for **Feature Filter.**



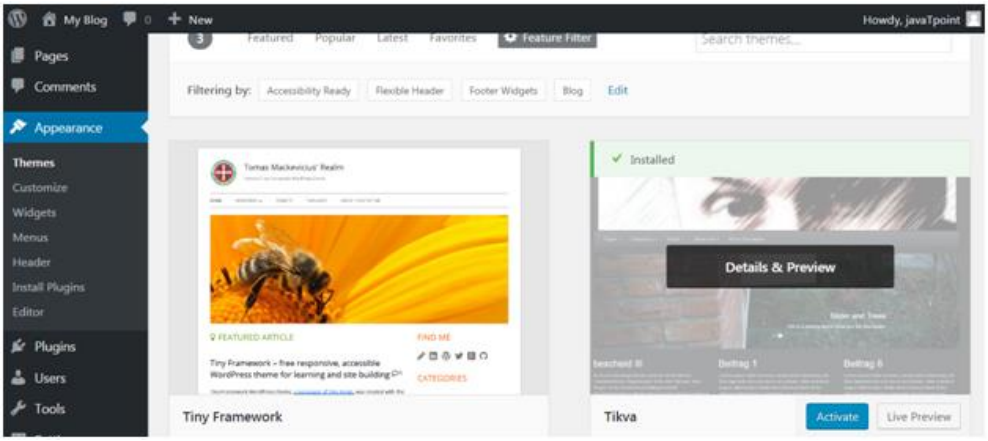
Look at the above snapshot, here you can filter your search for themes by applying your choice for Layout, Features and Subject.



Look at the above snapshot, we have selected 4 filter criteria, now click on button Apply Filters 4 to apply the filters.



Look at the above snapshot, select a theme and take your mouse above it. Click on **Install button.**



Look at the above snapshot, our theme is installed. To activate it click on **Activate** button. And you have successfully activated your WordPress theme.

USER AND USER ROLES AND PROFILES:: There are six default WordPress roles you can give to your user- administrator, editor, author, contributor, Subscriber and super admin.

Administrator: Administrators have full control over every aspect of the website. They can add, edit and delete plugins,as well as shape how website looks. They also have access to the site's setting and content management, including posts, pages and comments.

Additionally administrators are the only role that can manage other user roles**.** Thus if you are an administrator yourself you can add new users, delete existing ones or change their WordPress roles.

Editor: Editors can manage comments, pages, and posts. However, they have no access to the setting panel, installing new plugins, customizing the site's theme or organizing other users.

Author: Unlike editors, authors role is limited to their own content management. Tats why they have no permission to organize other user's posts approve and delete comments other than that they have the same limited WordPress roles and permissions as the editors.

Contributor: The contributor role in WordPress allows users to add, edit, and delete their own content.

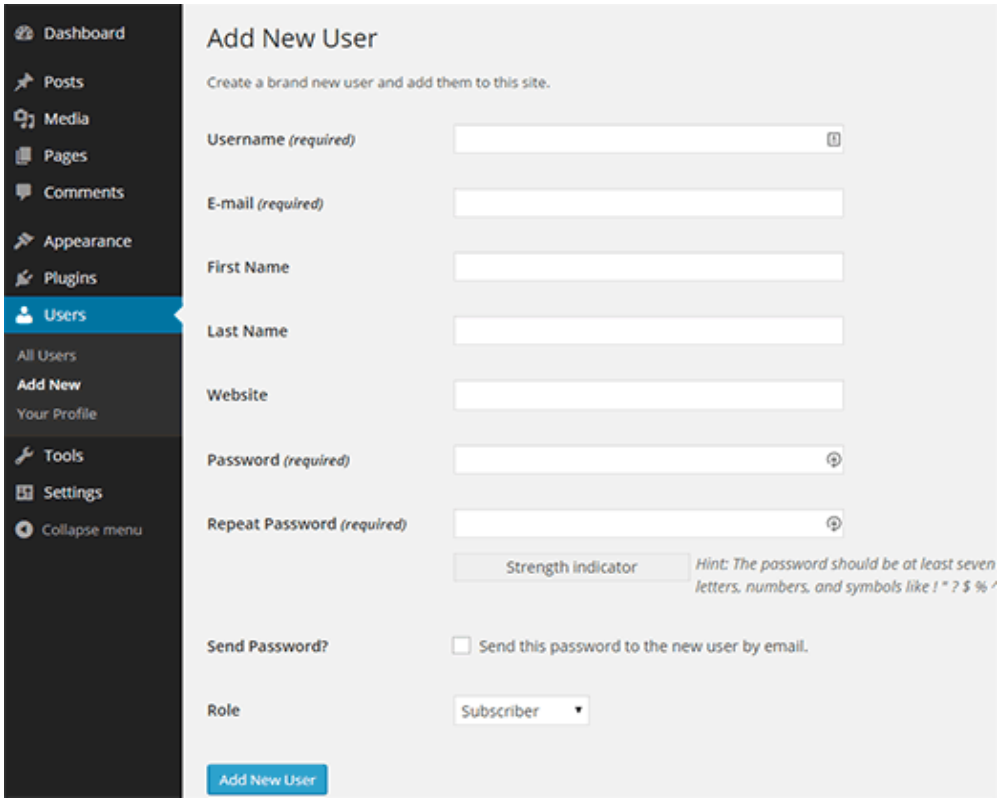
Contributors can't publish, upload media files and their posts once published.

Subscriber: Subscribers only have access to view published posts or comments and manage their profile section on the dashboard.

Super Admin: A super admin role can perform any administration task within the network, such as add or delete websites, install a wordpress theme or plugin,organize content and its settings. They have full control over the network's users.

Adding a New User:

Follow these steps to create and assign a new WordPress user role:



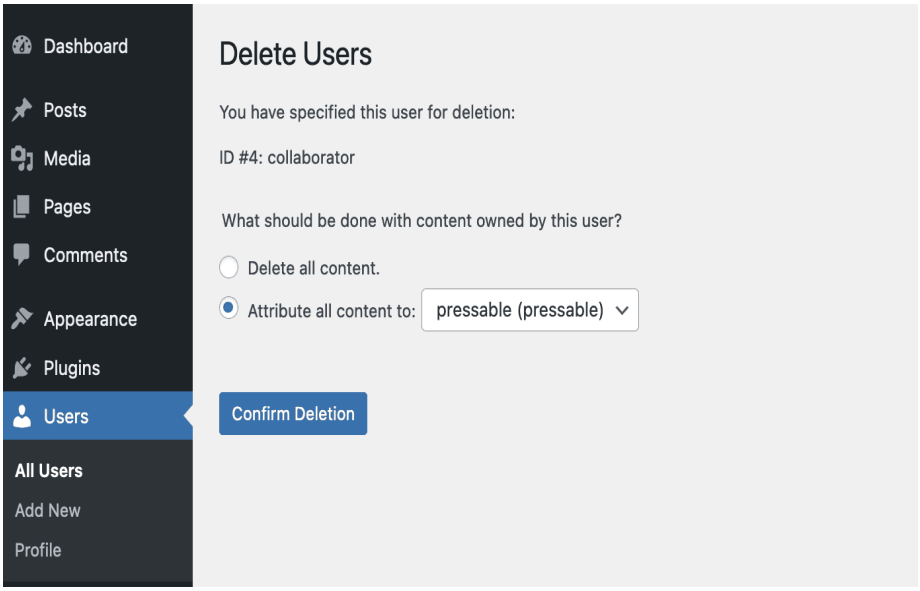
1. Log in to your WordPress Dashboard-→Users-→Add New.
2. Fill out the form with the user's personal details.
3. Create a new password by clicking the Generate password button.
4. From the drop-down menu, choose the preferred role.
5. Click on Add New User.

Deleting WordPress User:

If you want to delete a user account:

1. Log in to your WordPress→All Users.
2. Click Delete next to the user's name.
3. On the WordPress Delete user's page, press Confirm Deletion button. If users had any content on your site

You need to choose whether to delete or attribute their content to another user.

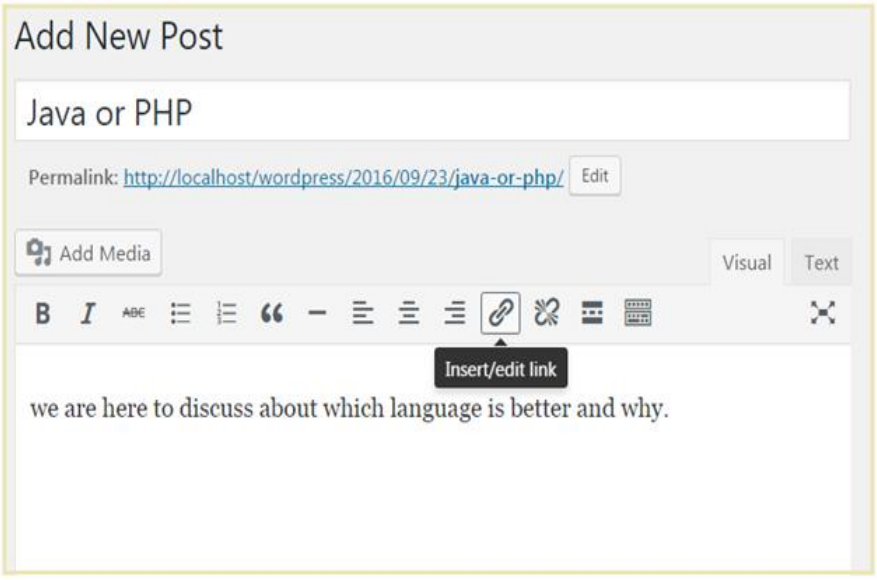


WORDPRESS LINKS: We can add/edit/delete links on WordPress posts, pages, text widgets and navigation menus.

- Add a link in WordPress posts and pages
- Add a link in WordPress text widgets
- Add a link in WordPress navigation menus
- How to Delete WordPress Links

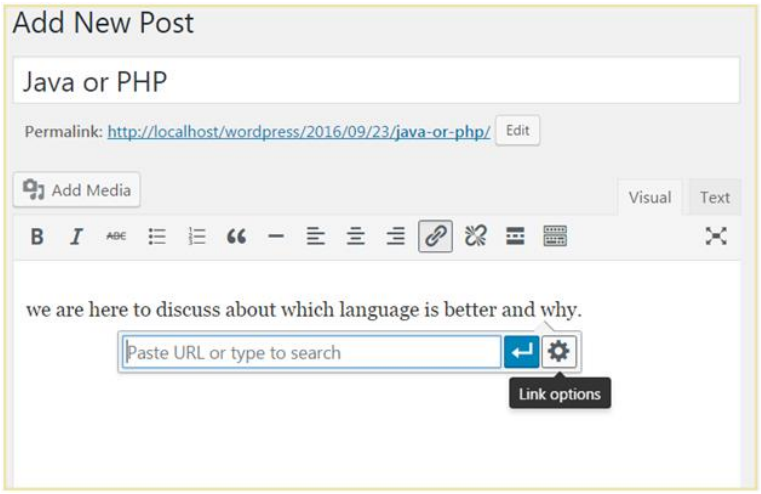
Add a New Link in WordPress Posts and Pages

- To add a link, go to **Posts > Add New**
- You can add a link either in a new post or in the existing post.

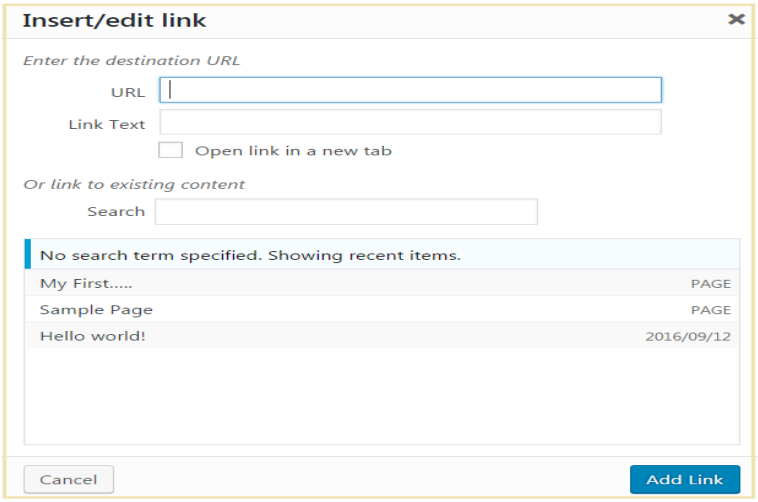


Look at the above snapshot, to create a link, click on the link icon.

On clicking the link icon, following page will appear.



Look at the above snapshot, next step is to click on Link Options button.



- Look at the above snapshot, this pop-up will come on clicking Link Options.
- In URL, enter URL you want to link.
- In Link Text, insert text you want to enter into the link.
- Tick on check-box Open link in new tab.
- You can also link it to the existing content.
- Now click on Add Link button and your link will be added.

Add a New Link in WordPress Text Widgets: In WordPress you can add links in widgets. Different widgets possess different functions like recent post widget will show links to your recent posts. In text widgets you can add text and basic HTML. There are two ways to add links in text widgets.

Adding a link in HTML

This method requires writing some of very basic HTML. In href tag you have to add URL. Remember to add http:// before domain. Between the closing and opening anchor tag add any text that you want to link (here we have written javaTpoint).

For example, javaTpoint

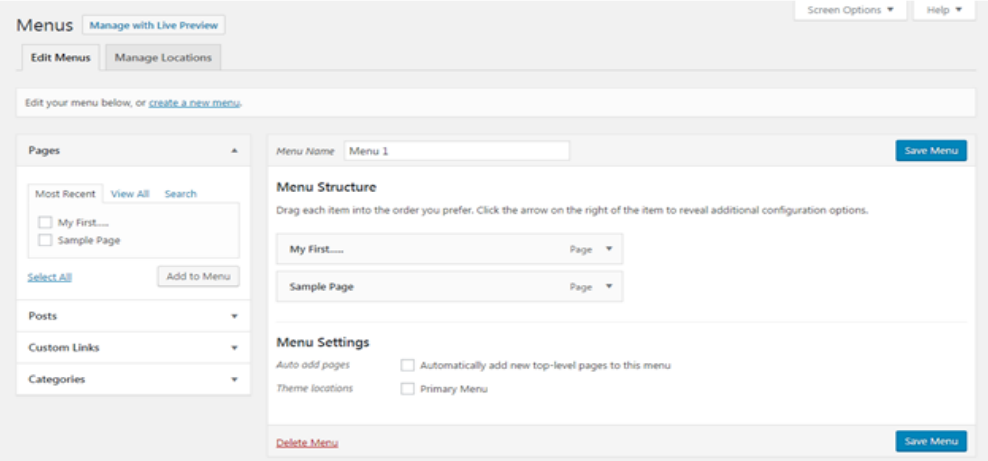
Using a plugin

Here you have to install and activate a plugin to enable visual editor for your text widgets.

Add a New Link in WordPress Navigation Menus

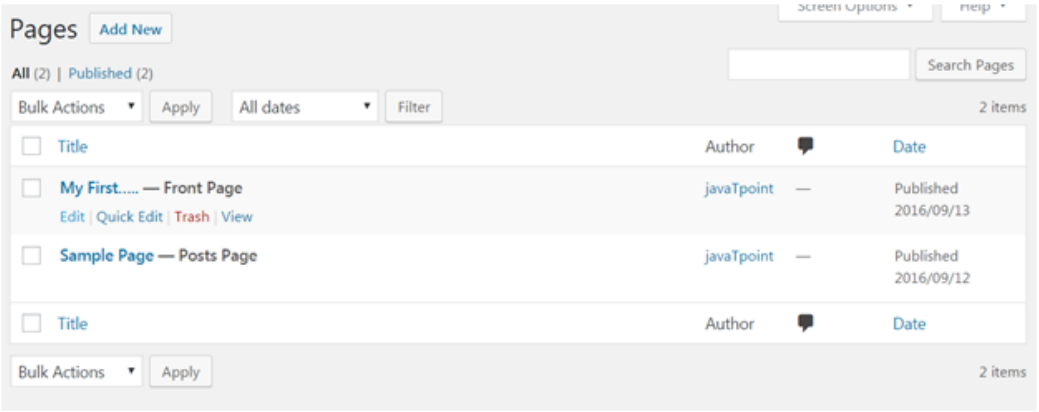
There is an editor in WordPress which allows you to create and manage navigation menus.

For navigation menus, go to Appearance > Menus

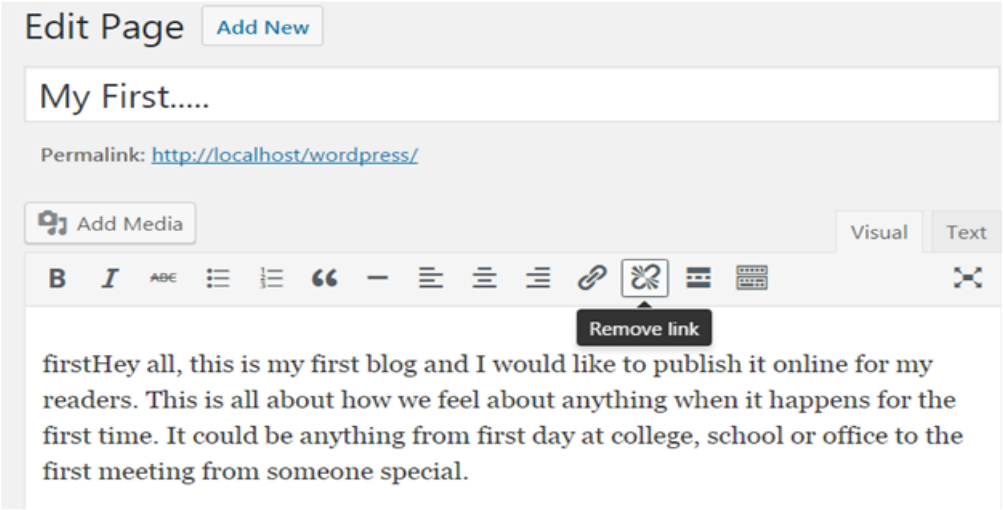


Delete Links

To edit links, go to Pages > All Pages



Look at the above snapshot, click on Edit button.



Look at the above snapshot this is the Remove Link option. Click on this and your link will be removed.

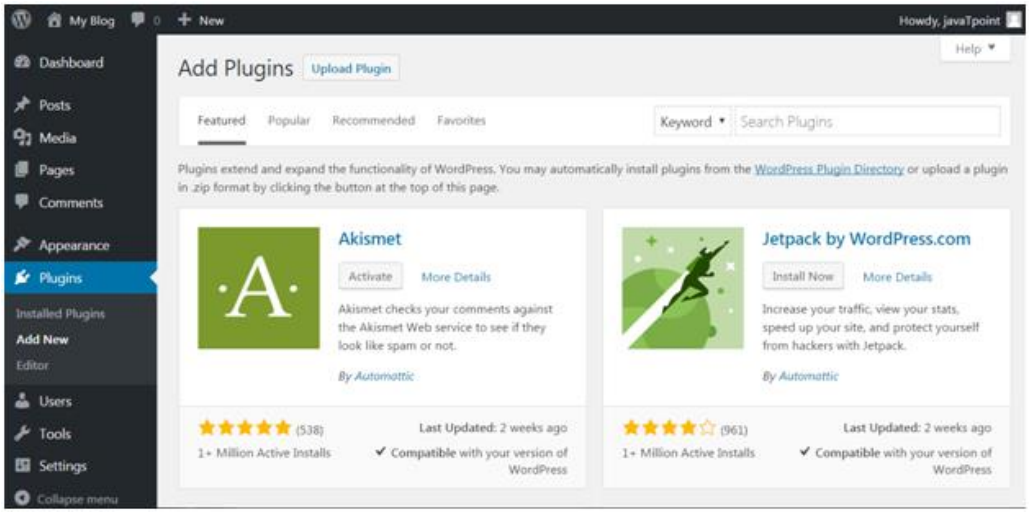
HOW TO INSTALL WORDPRESS PLUGINS: Installing plugins will provide extra functionality to your site. To install a plugin, you just need to put the plugin file into your wordpress file. After installation, you may activate it or deactivate it.

There are two methods from which you can install plugins.

- Installing plugin using search
- Installing plugin using uploading

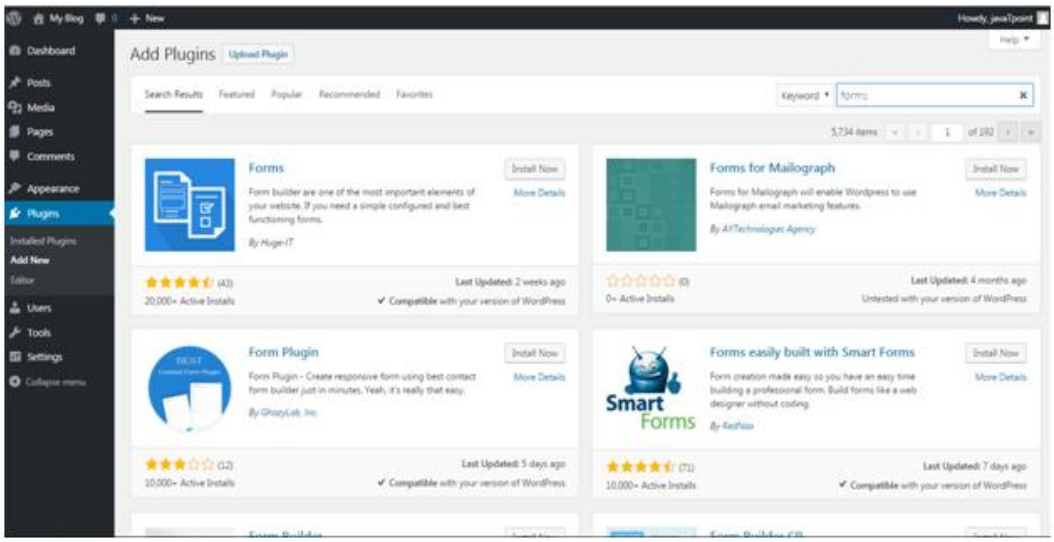
Installing plugin using search: This method is only available for the free plugins because search is applicable for the WordPress directory only.

Login to your WordPress, and go to **Plugins > Add New.**



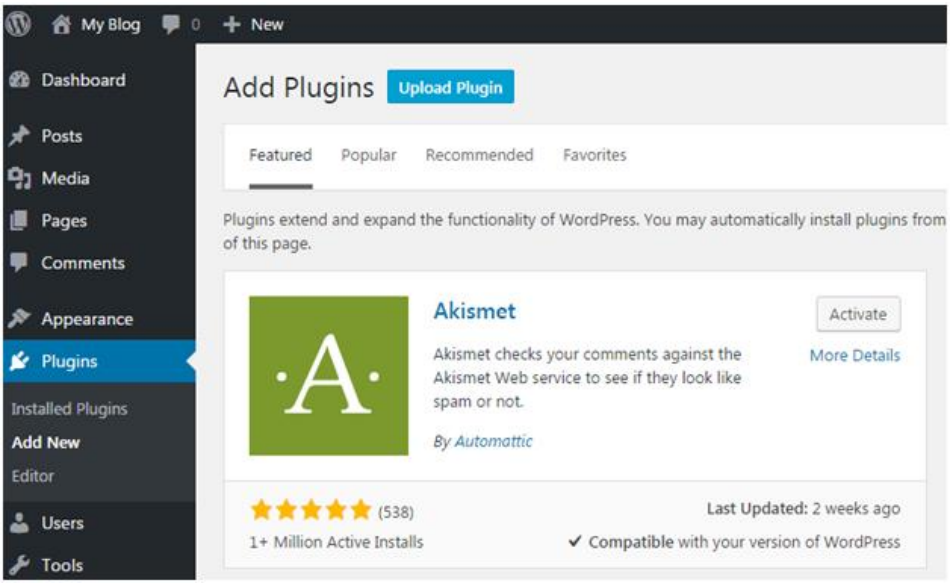
Look at the above snapshot, this page will be displayed. Here in the search option, you can give a plugin name the function name for which you want that plugin.

For example, we are searching for form plugin.

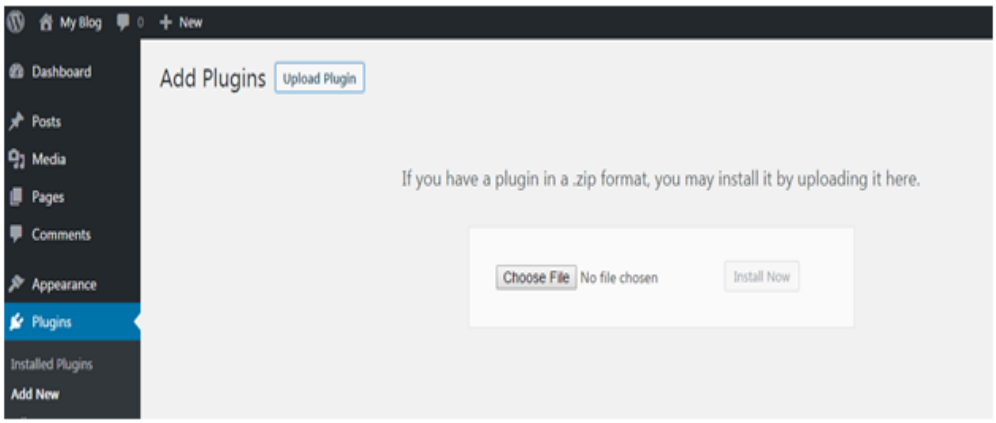


Look at the above snapshot, results related to our search is displayed with a list of matching plugins. We can select a plugin matching our criteria. Once you have selected a plugin, click on Install Now button. Your plugin will be installed but to activate it you have to click on Activate button.

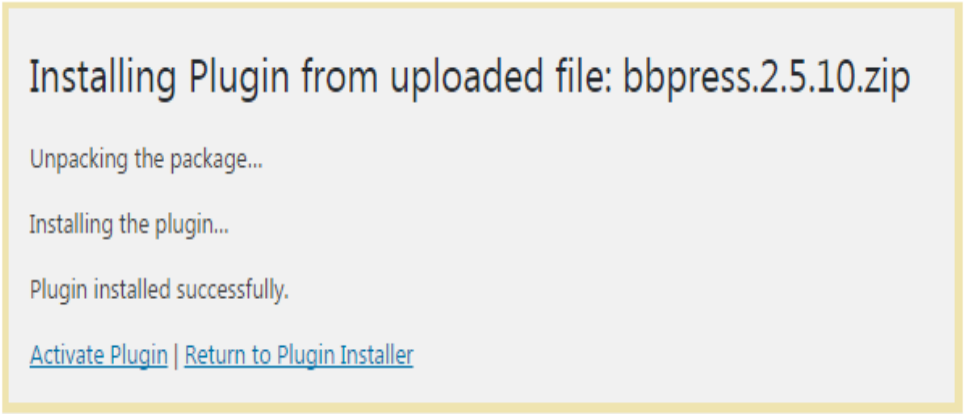
Installing plugin using uploading: From this method paid plugins can also be installed. First of all, download a plugin file from the source which will be in zip format. Now go to **Plugins > Add New page.**



Look at the above snapshot, click on Upload Plugin.



Look at the above snapshot, here you have to choose the downloaded plugin file and click on Install Now.



Look at the above snapshot, plugin has successfully installed showing a message. To activate this plugin you need to activate it.

WORDPRESS PLUGINS: WordPress plugins are programs written in PHP scripting language that extends the functionality or add some new features to our WordPress site. They provide additional functions to an application.

Users can add some more functions in their site without any coding skills by adding plugins. There are thousands of freely and commercially plugins available. For each and every function there is a plugin available in the WordPress.

How to choose a plugin

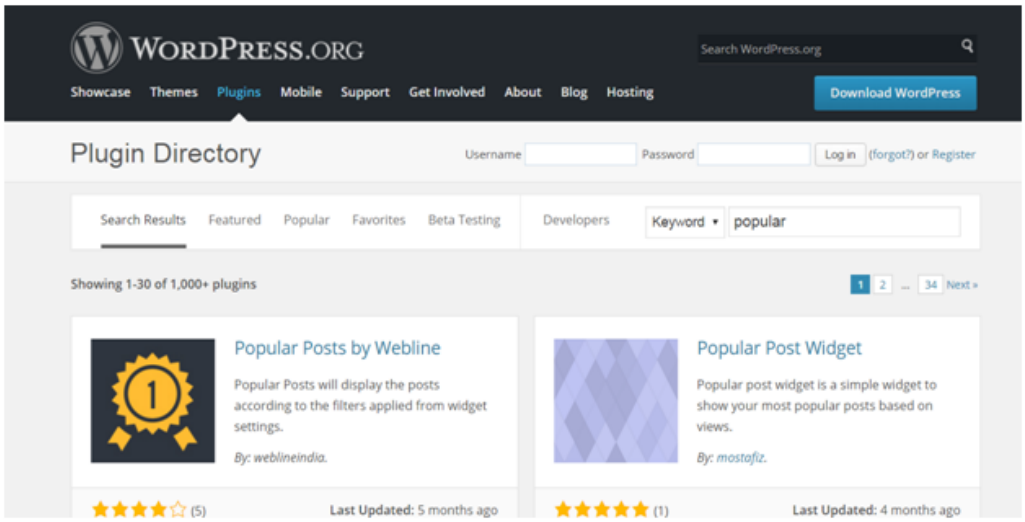
It is difficult for the users to find out the best suited plugin for them, especially for beginners. As plugins are open source hence every plugin is maintained by different owners. Some of them may be a kind of busy or inexperienced which lead to outdated version of plugin. According to your need you may have to install more than one plugin, but that?s not a problem. You can install more than one plugin.

Here we'll show a list which may help you to choose a right plugin.

Make a list of your requirements

First of all, you need to know what you want in your site. Make a list of all the features you want to implement in your site. Remember to make them in descending order of your need, it will make decision making easier. It's not necessary that you'll find all of it in one plugin. Feel free to install more than one plugin.

Search plugins: It is recommended to search plugins through Google search rather than WordPress directory. wordpress.org search is not that much helpful.



Look at the above snapshot, this is WordPress directory search. It shows the search result by relevancy which shows a lot of options. Use relevant words in your search otherwise you may not get the desired result.

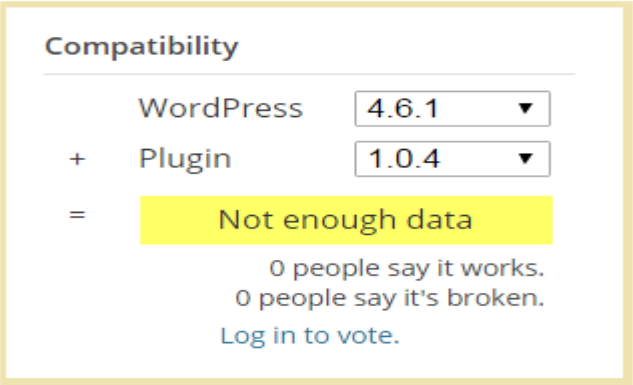
Enter a search, for example we entered Map as our search. It displays all map plugins from the directory.

Select a plugin to see its description and information.

Compare them: Choose 4 or 5 plugins and compare them by reading their description. Description contains information about that plugin, why it is used, what are its functions, supported versions, etc.

Support Overview: Support gives an overview of number of support threads opened for last 2 months. It also tells how many support threads have been resolved. To view the details of support threads click on the support tab.

Compatibility: Compatibility displays the latest version of WordPress with version of plugin.



Look at the above snapshot, WordPress latest version is 4.6.1 and plugin current version is 1.0.4

Below this people's vote for the given combination of WordPress and plugin is displayed. If more number of people say it works then the number of people saying it is broken then this combination will work. Here no one has voted for this.

Screenshots: Plugins screenshots shows a preview of how the plugin will look like from front-end and back-end. Screenshots can be more helpful than plugin descriptions.

FAQ: Frequently Asked Questions help in a lot of way. Before start using a plugin in your website, go through all these questions. It will help you in the smoother use of plugin.

Testing speed: A d\site's speed depends upon the plugin installed in it. Check your site's speed before and after installing a plugin. If speed slows down then better watch for a new plugin.

SHORTS

THEME MANAGEMENT: It includes images 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 assets 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.

CUSTOMIZING THE SITE:

- WordPress Website Customization is the best thing because custom made website are always well-coded and bug free.
- Your custom WordPress website will have a unique appearance and feel as well as the up-to-date features for your visitors to enjoy.
- It will also be safer and load faster, and your potential clients will be able to find it easily in the vast web space.

Configure WordPress Settings

- There are a couple of different spots where you will find wordpress setting.
- The first set will help you customize the look of wordpress itself.
- The second will help you customize the look of your website.

Customize Screen Options:

Click on “Screen Options” from WordPress dashboard to open the settings.

CHANGING THE APPEARANCE OF SITE USING CSS:

- Adding custom CSS helps customize the design and appearance of your site, Which is not possible through the default options.
- It gives you more control, and you can easily make changes to your WordPress theme with a few lines of code.
- For example, you want to change the background color of each individual post instead of using the same color throughout the website.
- By adding a custom CSS code, you can personalize the background of a specific post or page.
- Similarly, you can change the style and appearance of product categories on your ecommerce store using additional CSS.
- There are Different ways you can add custom CSS to your WordPress website.

They are

- ❖ Adding Custom CSS Using Theme Customizer.
- ❖ Adding Custom CSS Using a Plug-in.
- ❖ Adding Additional CSS WITH Full Site Editor(FSE).

PROTECTING WORDPRESS WEBSITE FROM HACKERS: WordPress is one of the most popular content management systems(CMS).That's why it is vital to prevent Wordpress hacking.

1. Choose a secure hosting provider
2. Get a security plugin
3. Choose a secure theme
4. Keep WordPress updated
5. Use secure login details

1. Choose a secure hosting provider: All good hosting providers will include security protection to ensure your website information is kept safe on their servers. When choosing a **hosting provider**, make sure to check what security measures they have (such as firewalls and secure FTP), how they monitor their server network, and how they respond to any security breaches.

2.Get a security plugin: Having a high-quality security plugin is a must-have to prevent your WordPress site getting hacked. We recommend Wordfence – an excellent, free security plugin. Once installed, 'Wordfence' will appear in the left-hand menu of your WordPress dashboard. You can click here at any time to scan your site, see the latest notifications and get recommendations to improve site security.

3.Choose a secure theme: Choosing the right theme for your site is crucial. Of course, it needs to have the right look and features for your organisation. But it also needs to be robust and secure.

A secure theme will:

- Be updated and patched regularly
- Follow good coding standards
- Not be associated with bugs or compatibility errors

4.Keep WordPress updated: Keeping WordPress up-to-date is another important security measure. WordPress software updates are made regularly to optimise performance and patch any security issues as they are discovered.

5.Use secure login details: To prevent hacking, make sure to choose an atypical username. This basically means not using "admin", which is so common it's usually the first username hackers will try.

Secondly, go for a secure password including a mix of letters, symbols and numbers. For maximum security, this should be at least 12 characters and not include any dictionary words.

Additional Questions

FLOAT PROPERTY IN CSS: The float property is used for positioning and formatting content e.g. let an image float left to the text in a container.

The float property can have one of the following values:

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

In its simplest use, the **float** property can be used to wrap text around images.

Syntax:

Right:

```
img {
  float: right;
}
```

Left:

```
img {
  float: left;
}
```

No-float:

```
img {
  float: none;
}
```

CSS COUNTERS:

CSS counters are "variables" maintained by CSS whose values can be incremented by CSS rules (to track how many times they are used). Counters let you adjust the appearance of content based on its placement in the document.

Automatic Numbering With Counters

CSS counters are like "variables". The variable values can be incremented by CSS rules (which will track how many times they are used).

To work with CSS counters we will use the following properties:

- counter-reset - Creates or resets a counter
- counter-increment - Increments a counter value
- content - Inserts generated content
- counter() or counters() function - Adds the value of a counter to an element

EXCEPTION HANDELLING:

- In programming, exception handling is a process or method used for handling the abnormal statements in the code and executing them.
- It also enables to handle the flow control of the code/program.
- For handling the code, various handlers are used that process the exception and execute the code.
- For example, the Division of a non-zero value with zero will result into infinity always, and it is an exception.
- Thus, with the help of exception handling, it can be executed and handled.
- A throw statement is used to raise an exception. It means when an abnormal condition occurs, an exception is thrown using throw.
- The thrown exception is handled by wrapping the code into the try…catch block.
- If an error is present, the catch block will execute, else only the try block statements will get executed.
- Thus, in a programming language, there can be different types of errors which may disturb the proper execution of the program.

Syntax:

```
try{
throw exception; // user can define their own exception
}
catch(error){
expression; }  // code for handling exception.
```