

E-content

CSE

2ndSemester

AdvancedIT(AIT)

Unit1:HTMLFundamentals

HTML provides basic and advanced concepts of HTML. If you are new in learning HTML, then you can learn HTML from basic to a professional level and after learning HTML with CSS and JavaScript you will be able to create your own interactive and dynamic website. But Now We will focus on HTML only from unit 1 to unit 4 and in unit 5 will cover JavaScript.

The major points or characteristics of HTML are given below:

- HTML stands for Hypertext Markup Language.
- HTML is used to create web pages and web applications.
- HTML is widely used language on the web.
- We can create a static website by HTML only.
- Technically, HTML is a Markup language rather than a programming language.

What is HTML

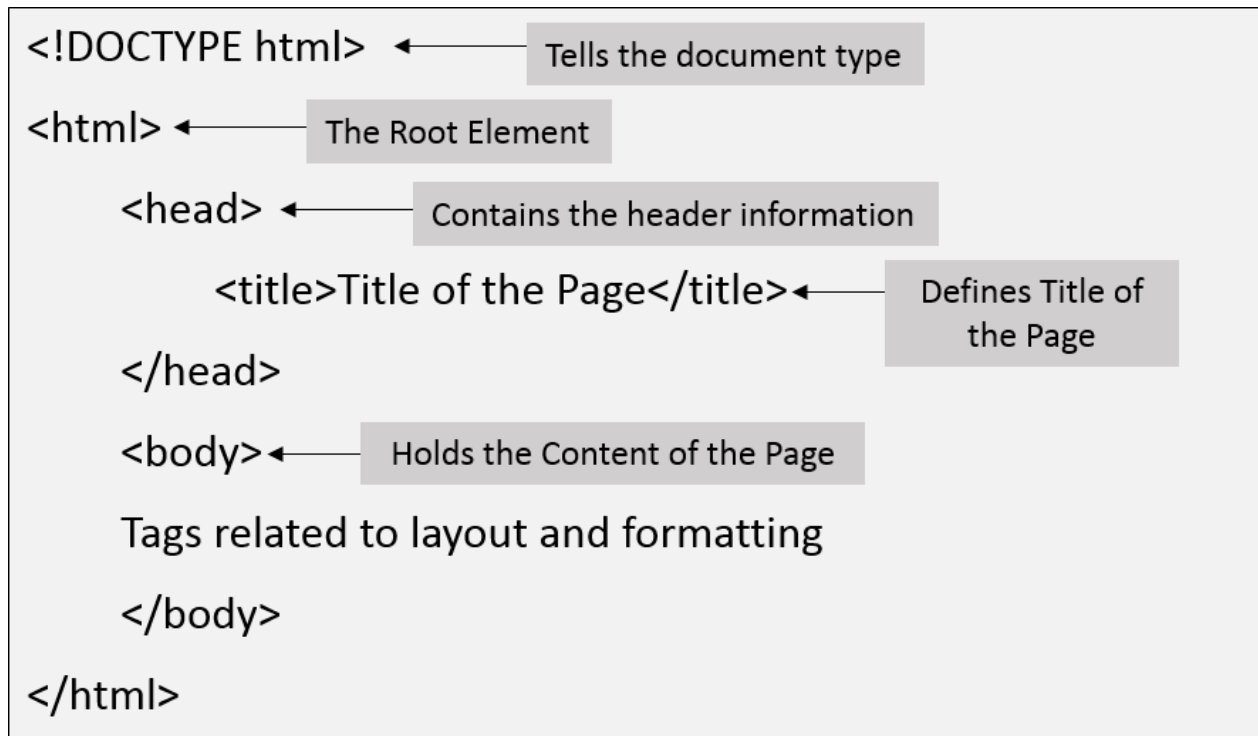
HTML is an acronym which stands for **Hyper Text Markup Language** which is used for creating web pages and web applications. Let's see what is meant by Hypertext Markup Language, and Web page.

Hyper Text: HyperText simply means "Text within Text." A text has a link within it, is a hypertext. Whenever you click on a link which brings you to a new webpage, you have clicked on a hypertext. HyperText is a way to link two or more web pages (HTML documents) with each other.

Markup language: A markup language is a computer language that is used to apply layout and formatting conventions to a text document. Markup language makes text more interactive and dynamic. It can turn text into images, tables, links, etc.

Web Page: A web page is a document which is commonly written in HTML and translated by a web browser. A web page can be identified by entering an URL. A Web page can be of the static or dynamic type. **With the help of HTML only, we can create static web pages.**

Hence, HTML is a markup language which is used for creating attractive web pages with the help of styling, and which looks in a nice format on a web browser. An HTML document is made of many HTML tags and each HTML tag contains different content.



Note: Inside `<body>` tag write `<h1>` and `<p>` tags

Description of HTML Example

<!DOCTYPE>: It defines the document type or it instructs the browser about the version of HTML.

<html >: This tag informs the browser that it is an HTML document. Text between html tag describes the web document. It is a container for all other elements of HTML except `<!DOCTYPE>`

<head>: It should be the first element inside the `<html>` element, which contains the metadata (information about the document). It must be closed before the body tag opens.

<title>: As its name suggested, it is used to add title of that HTML page which appears at the top of the browser window. It must be placed inside the head tag and should close immediately. (Optional)

<body>: Text between body tag describes the body content of the page that is visible to the end user. This tag contains the main content of the HTML document.

<h1>: Text between `<h1>` tag describes the first level heading of the webpage.

<p>: Text between `<p>` tag describes the paragraph of the webpage.

Brief History of HTML

In the late 1980's, a physicist, Tim Berners-Lee who was a contractor at CERN, proposed a system for CERN researchers. In 1989, he wrote a memo proposing an internet based hypertext system.

Tim Berners-Lee is known as the father of HTML. The first available description of HTML was a document called "HTML Tags" proposed by Tim in late 1991. The latest version of HTML is HTML5, which we will learn later in this tutorial.

HTML Versions

Since the time HTML was invented there are lots of HTML versions in market, the brief introduction about the HTML version is given below:

HTML 1.0: The first version of HTML was 1.0, which was the barebones version of HTML language, and it was released in 1991.

HTML 2.0: This was the next version which was released in 1995, and it was standard language version for website design. HTML 2.0 was able to support extra features such as form-based file upload, form elements such as text box, option button, etc.

HTML 3.2: HTML 3.2 version was published by W3C in early 1997. This version was capable of creating tables and providing support for extra options for form elements. It can also support a webpage with complex mathematical equations. It became an official standard for any browser till January 1997. Today it is practically supported by most of the browsers.

HTML 4.01: HTML 4.01 version was released on December 1999, and it is a very stable version of HTML language. This version is the current official standard, and it provides added support for stylesheets (CSS) and scripting ability for various multimedia elements.

HTML5 : HTML5 is the newest version of HyperText Markup language. The first draft of this version was announced in January 2008. There are two major organizations one is W3C (World Wide Web Consortium), and another one is WHATWG (Web Hypertext Application Technology Working Group) which are involved in the development of HTML 5 version, and still, it is under development.

Features or characteristics of HTML

1) It is a very easy and simple language. It can be easily understood and modified.

- 2) It is very easy to make an **effective presentation** with HTML because it has a lot of formatting tags.
- 3) It is a **markup language**, so it provides a flexible way to design web pages along with the text.
- 4) It facilitates programmer to add a **link** on the web pages (by HTML anchor tag), so it enhances the interest of browsing of the user.
- 5) It is **platform-independent** because it can be displayed on any platform like Windows, Linux, and Macintosh, etc.
- 6) It facilitates the programmer to add **Graphics, Videos, and Sound** to the web pages which makes it more attractive and interactive.
- 7) HTML is a case-insensitive language, which means we can use tags either in lower-case or upper-case.

NOTE: It is recommended to write all tags in lower-case for consistency, readability, etc.

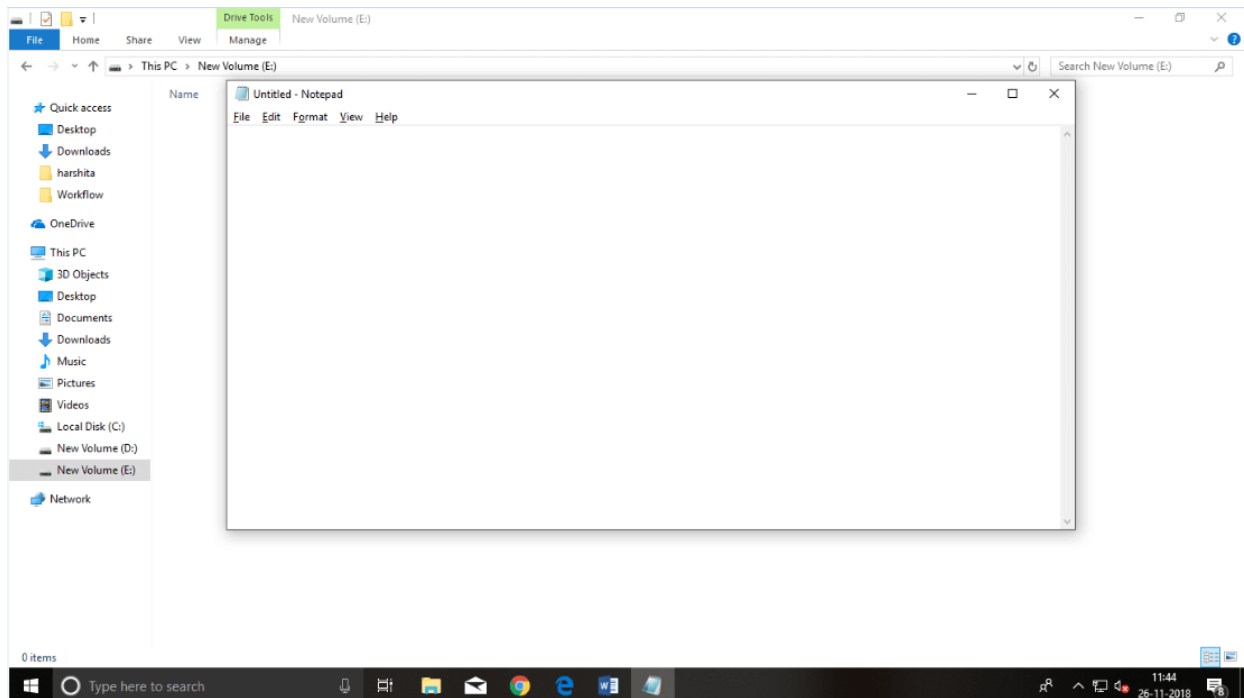
HTML text Editors:

- An HTML file is a text file, so to create an HTML file we can use any text editors.
- Text editors are the programs which allow editing in a written text, hence to create a web page we need to write our code in some text editor.
- There are various types of text editors available which you can directly download, but for a beginner, the best text editor is Notepad (Windows) or TextEdit (Mac).
- After learning the basics, you can easily use other professional text editors which are, **Notepad++, Sublime Text, Vim, etc.**
- In our tutorial, we will use Notepad and Sublime text editor. Following are some easy ways to create your first web page with Notepad, and Sublime text.

A. HTML code with Notepad. (Recommended for Beginners)

Notepad is a simple text editor and is suitable for beginners to learn HTML. It is available in all versions of Windows, from where you easily access it.

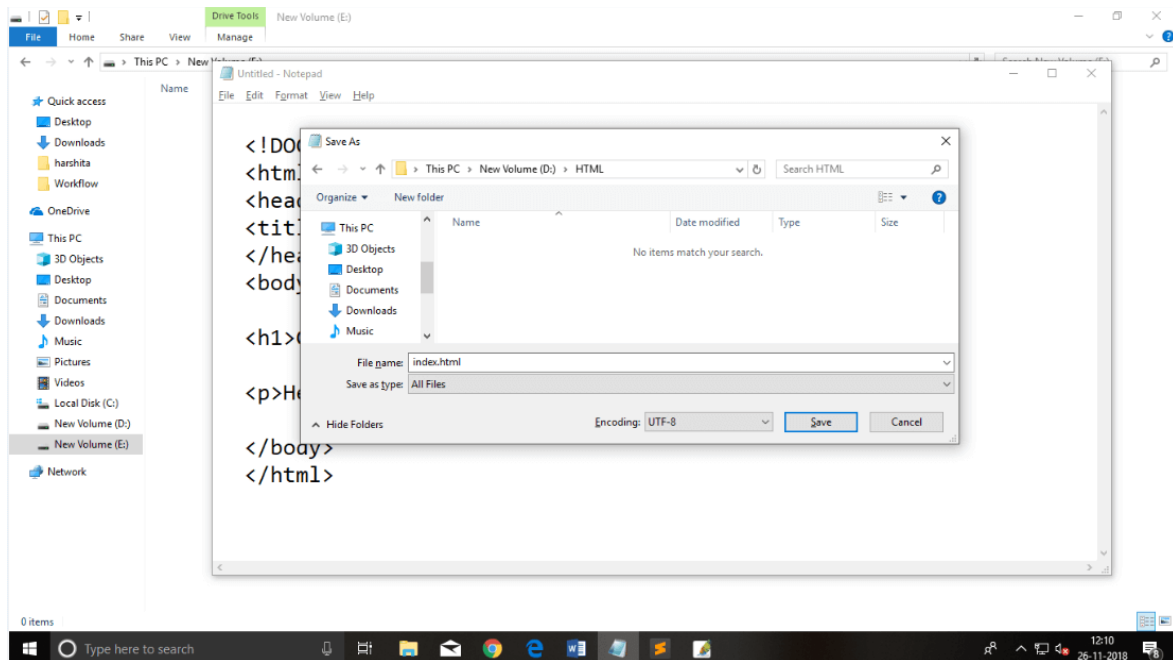
Step 1: Open Notepad (Windows)



Step2:WritecodeinHTML

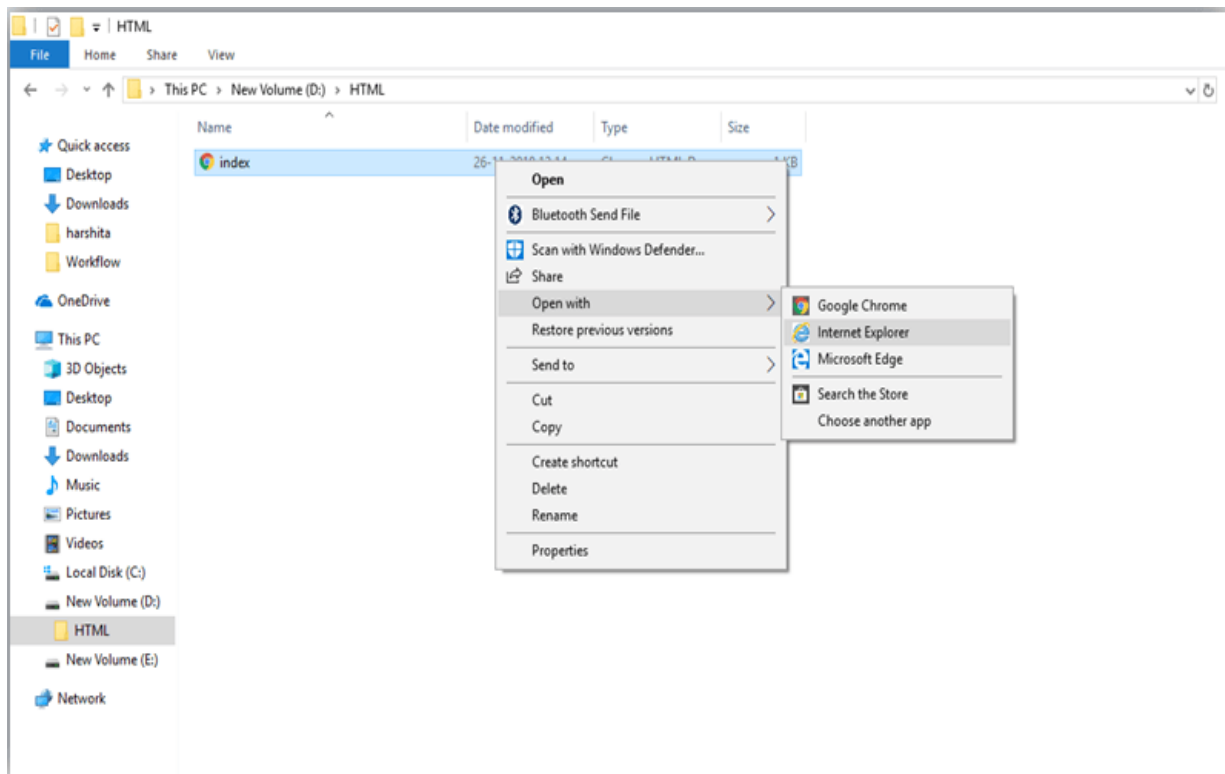


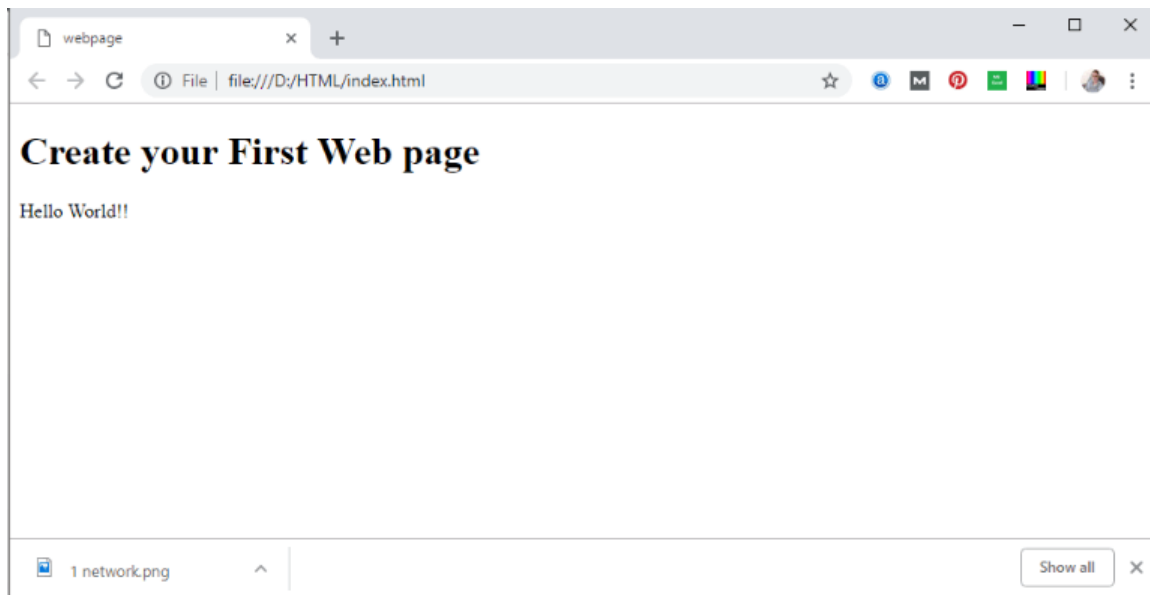
Step3:Save the HTML file with .htm or .html extension.



Step4:Open theHTMLpageinyourwebbrowser.

To run the HTMLpage, you need to open the file location, where you have saved the file and then either double-click on file or click on open with option





Note: You can execute HTML file in any browser, but there are some tags which are not supported by some web browser.

Commonly used web browsers:

Web Browsers are software installed on your PC. To access the Web, you need a web browser, such as Netscape Navigator, Microsoft Internet Explorer or Mozilla Firefox.

Currently you must be using any sort of Web browser while you are navigating through our site tutorialspoint.com. On the Web, when you navigate through pages of information, this is commonly known as web browsing or web surfing.

There are four leading web browsers – Explorer, Firefox, Netscape, and Safari, but there are many others browsers available. You might be interested in knowing Complete Browser Statistics. Now we will see these browsers in bit more detail.

While developing a site, we should try to make it compatible to as many browsers as possible. Especially sites should be compatible to major browsers like Explorer, Firefox, Chrome, Netscape, Opera, and Safari.



Internet Explorer

Internet Explorer (IE) is a product from software giant Microsoft. This is the most commonly used browser in the universe. This was introduced in 1995 along with Windows 95 launch and it has passed Netscape popularity in 1998.

You can download a latest version of this browser by clicking here → [Download Internet Explorer](#)



GoogleChrome

This web browser is developed by Google and its beta version was first released on September 2, 2008 for Microsoft Windows. Today, chrome is known to be one of the most popular web browser with its global share of more than 50%.

You can download a latest version of this browser by clicking here → [Download GoogleChrome](#)



Mozilla Firefox

Firefox is a new browser derived from Mozilla. It was released in 2004 and has grown to be the second most popular browser on the Internet.

You can download a latest version of this browser by clicking here → [Download Firefox](#)



Safari

Safari is a web browser developed by Apple Inc. and included in Mac OS X. It was first released as a public beta in January 2003. Safari has very good support for latest technologies like XHTML, CSS2 etc.

You can download a latest version of this browser by clicking here → [Download Safari](#)



Opera

Opera is smaller and faster than most other browsers, yet it is full-featured. Fast, user-friendly, with keyboard interface, multiple windows, zoom functions, and more. Java and non Java-enabled versions available. Ideal for newcomers to the Internet, school children, handicap and as a front-end for CD-Rom and kiosks.

You can download a latest version of this browser by clicking here → [Download Opera](#)

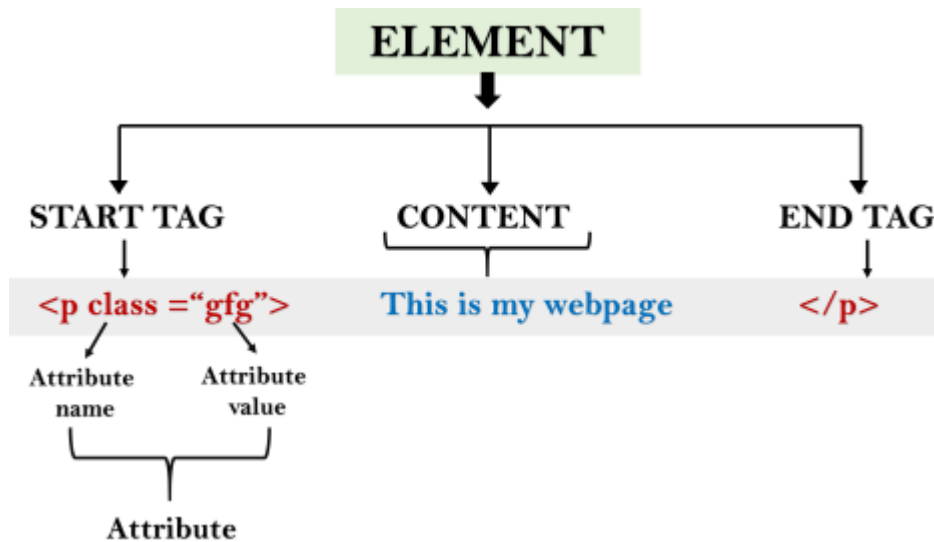
Building blocks of HTML

An HTML document consists of its basic building blocks which are:

- **Tags:** An HTML tag surrounds the content and applies meaning to it. It is written between < and > brackets.
- **Attribute:** An attribute in HTML provides extra information about the element, and it is applied within the start tag. An HTML attribute contains two fields: name & value.

Syntax

1. `<tagnameattribute_name="attr_value">content</tagname>`
 - **Elements:** An HTML element is an individual component of an HTML file. In an HTML file, everything written within tags are termed as HTML elements.



Example:

```
<!DOCTYPEhtml>
<html>
  <head>
    <title>ThebasicbuildingblocksofHTML</title>
  </head>
  <body>
    <h2>Thebuildingblocks</h2>
    <p>Thisisa paragraphtag</p>
    <pstyle="color:red">Thestyleisattributeofparagraphtag</p>
    <span>Theelementcontainstag,attributeandcontent</span>
  </body>
</html>
```

Output:

The building blocks

This is a paragraph tag

The style is attribute of paragraph tag

The element contains tag, attribute and content

HTML Tags

HTML tags are like keywords which define that how web browser will format and display the content. With the help of tags, a web browser can distinguish between an HTML content and a simple content. HTML tags contain three main parts: opening tag, content and closing tag. But some HTML tags are unclosed tags.

When a web browser reads an HTML document, browser reads it from top to bottom and left to right. HTML tags are used to create HTML documents and render their properties. Each HTML tag has different properties.

An HTML file must have some essential tags so that web browser can differentiate between a simple text and HTML text. You can use as many tags you want as per your code requirement.

- All HTML tags must be enclosed within `<>` these brackets.
- Every tag in HTML performs different tasks.
- If you have used an open tag `<tag>`, then you must use a close tag `</tag>` (except some tags)

Syntax

```
<tag>content</tag>
```

HTML Tag Examples

Note: HTML tags are always written in lowercase letters. The basic HTML tags are given below:

```
<p>ParagraphTag</p>
```

```
<h2>HeadingTag</h2>
```

`<b>BoldTag </b>`

`<i>ItalicTag</i>`

`<u>UnderlineTag</u>`

Testit Now

UnclosedHTMLTags

SomeHTMLtagsarenotclosed,forexamplebrandhr.

**
Tag:**brstandsforbreakline,itbreaksthelineof the code.

<hr>Tag:hrstandsforHorizontalRule.This tagisusedtoputalineacrossthe webpage.

HTMLMetaTags

DOCTYPE,title,link,metaandstyle

HTMLTextTags

`<p>,<h1>,<h2>,<h3>,<h4>,<h5>,<h6>,<strong>,<em>,<abbr>,<acronym>,<address>,<blockquote>,<cite>,<q>,<code>,<ins>,<del>,<pre>,<br>`

HTMLLink Tags

`<a>`

HTMLImageandObjectTags

`<img>,<area>,<map>,<param>and<object>`

HTMLListTags

,,,<dl>,<dt>and<dd>

HTMLTable Tags

table,tr,td,th, tbody,thead,tfoot,col,colgroupandcaption

HTMLFormTags

form,input,textarea,select,option,button,label,fieldsetandlegend

HTMLScriptingTags

script and noscript

Note: We will see examples using these tags in later chapters.

HTMLAttribute

- HTML attributes are special words which provide additional information about the elements or attributes are the modifier of the HTML element.
- Each element or tag can have attributes, which define the behaviour of that element.
- Attributes should always be applied with start tag.
- The attributes should always be applied with its name and value pair.
- The attributes name and values are case sensitive, and it is recommended by W3C that it should be written in Lowercase only.
- You can add multiple attributes in one HTML element, but need to give space between two attributes.

Syntax

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

HTMLHeading

A HTML heading or HTML htag can be defined as a title or a subtitle which you want to display on the webpage. When you place the text within the heading tags `<h1>.....</h1>`, it is displayed on the browser in the bold format and size of the text depends on the number of heading.

There are six different HTML headings which are defined with the `<h1>` to `<h6>` tags, from highest level h1 (main heading) to the least level h6 (least important heading).

h1 is the largest heading tag and h6 is the smallest one. So h1 is used for most important heading and h6 is used for least important.

Headings in HTML helps the search engine to understand and index the structure of webpage.

Note: The main keyword of the whole content of a webpage should be displayed by h1 heading tag.

See this example:

`<h1>Headingno. 1</h1>`

`<h2>Headingno. 2</h2>`

`<h3>Headingno. 3</h3>`

`<h4>Headingno. 4</h4>`

`<h5>Headingno. 5</h5>`

`<h6>Headingno. 6</h6>`

Output:

Headingno.1

Headingno.2

Headingno.3

Headingno.4

Headingno.5

Headingno.6

HTMLParagraph

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 <p> tag indicates starting of new paragraph.

Note: If we are using various <p> tags in one HTML file then browser automatically adds a single blank line between the two paragraphs.

See this example:

1. `<p>This is first paragraph.</p>`
2. `<p>This is second paragraph.</p>`
3. `<p>This is third paragraph.</p>` [Test it Now](#)

Output:

This is first paragraph.

This is second paragraph.

This is third paragraph.

Space inside HTML Paragraph

If you put a lot of spaces inside the HTML p tag, browser removes extra spaces and extra line while displaying the page. The browser counts number of spaces and lines as a single one.

`<p>`

I am

going to provide

you a tutorial on HTML

andhopethatitwill
beverybeneficialforyou.

</p>

<p>

Look,Iputherealot
ofspaces but I know,Browserwillignoreit.

</p>

<p>

Youcannotdeterminethe displayofHTML</p>

<p>because resized windows may createdifferentresult.

</p>

Output:

Iamgoingtoprovide youatutorialonHTML andhopethat itwillbevery beneficialforyou. Look, I put
here a lot of spaces but I know, Browser will ignore it.

Youcannot determinethedisplayofHTML

because resized windows may createdifferent result.

Asyoucansee,alltheextra linesandunnecessaryspacesareremovedbythe browser. [How to](#)

Use
 and <hr> tag with paragraph?

AnHTML
tagisusedforlinebreakanditcanbeusedwithparagraphelements.Following
istheexampletoshowhowtouse
with<p>element.

Example:

<!DOCTYPEhtml>

<html>

<head>

</head>

<body>

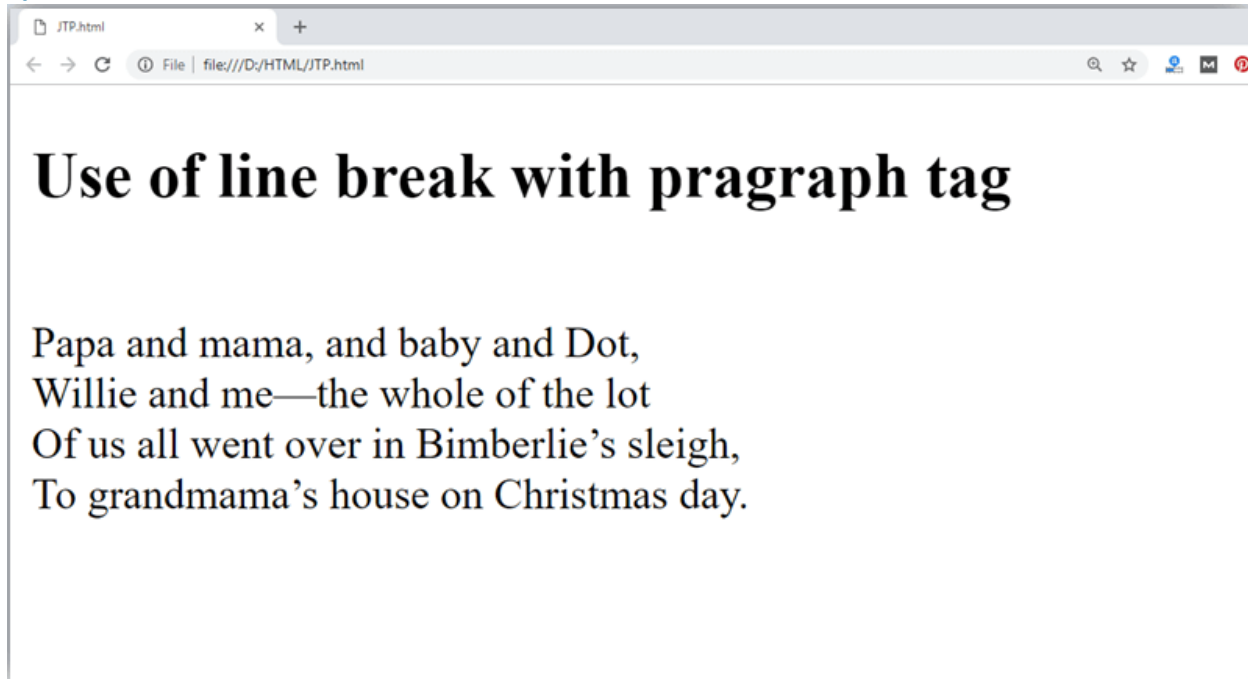
<h2>Useofflinebreakwithpragraph tag</h2>

<p>
Papaandmama,and babyand Dot,

Willieandme?thewholeofthelot

Ofusallwentover in Bimberlie'ssleigh,

```
<br>To grandmama's house on Christmas day.  
</p>  
</body>  
</html>
```

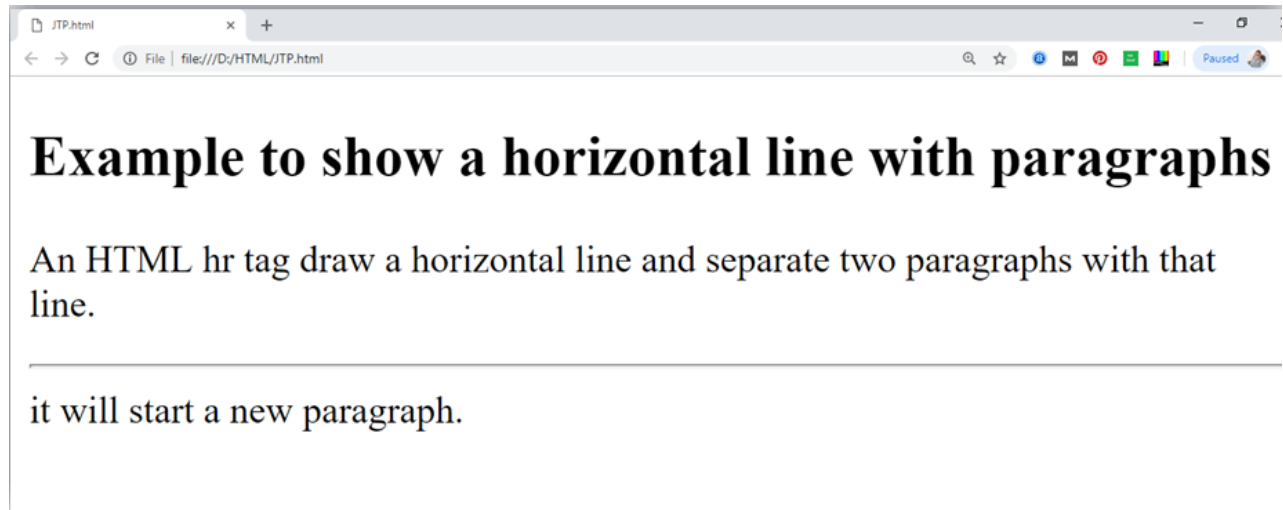


An HTML `<hr>` tag is used to apply a horizontal line between two statements or two paragraphs. Following is the example which is showing use of `<hr>` tag with paragraph.

Example:

```
<!DOCTYPE html>  
<html>  
  <head>  
  </head>  
  <body>  
    <h2>Example to show a horizontal line with paragraphs</h2>  
    <p>An HTML hr tag draws a horizontal line and separates two paragraphs with that line. <hr> it will  
start a new paragraph.  
  </p>  
</body>  
</html>
```

Output:



Changing appearance of Text tags:

Formatting elements were designed to display special types of text:

- `<b>`-Bold text
- `<strong>`-Important text
- `<i>`-Italic text
- `<em>`-Emphasized text
- `<mark>`-Marked text
- `<small>`-Smaller text
- `<del>`-Deleted text
- `<ins>`-Inserted text
- `<sub>`-Subscript text
- `<sup>`-Superscript text

HTML `<b>` and `<strong>` Elements

The HTML `<b>` element defines bold text, without any extra importance.

Example

`<b>`This text is bold`</b>`

The HTML `<strong>` element defines text with strong importance. The content inside is typically displayed in bold.

Example

****This text is important!****

HTML **<i>** and **** Elements

The HTML **<i>** element defines a part of text in an alternate voice or mood. The content inside is typically displayed in italic.

Tip: The **<i>** tag is often used to indicate a technical term, a phrase from another language, a thought, a ship name, etc.

Example

<i>This text is italic**</i>**

The HTML **** element defines emphasized text. The content inside is typically displayed in italic.

Tip: A screen reader will pronounce the words in **** with an emphasis, using verbal stress.

Example

****This text is emphasized****

HTML **<small>** Element

The HTML **<small>** element defines small text:

Example

<small>This is some small text.**</small>**

HTML **<mark>** Element

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

Example

<p>Do not forget to buy **<mark>**milk**</mark>** today.**</p>**

HTML **** Element

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

Example

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

HTML <ins> Element

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

Example

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

HTML <sub> Element

The HTML <sub> element defines subscript text. Subscript text appears half a character below the normal line, and is sometimes rendered in a smaller font. Subscript text can be used for chemical formulas, like H₂O:

Example

```
<p>This is <sub>subscripted</sub> text.</p>
```

HTML <sup> Element

The HTML <sup> element defines superscript text. Superscript text appears half a character above the normal line, and is sometimes rendered in a smaller font. Superscript text can be used for footnotes, like WWW^[1]:

Example

```
<p>This is <sup>superscripted</sup> text.</p>
```

HTML Comments

HTML comments are not displayed in the browser, but they can help document your HTML source code.

HTML Comment Tag

You can add comments to your HTML source by using the following syntax:

```
<!-- Write your comments here -->
```

Notice that there is an exclamation point (!) in the start tag, but not in the end tag.

Note:Comments are not displayed by the browser, but they can help document your HTML source code.

Add Comments

With comments you can place notifications and reminders in your HTML code:

Example

```
<!--This is a comment-->
```

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

```
<!--Remember to add more information here -->
```

```
>Hide Content
```

Comments can be used to hide content.

This can be helpful if you hide content temporarily:

Example

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

```
<!--<p>This is another paragraph</p>-->
```

```
<p>This is a paragraph too.</p>
```

You can also hide more than one line. Everything between the `<!--` and the `-->` will be hidden from the display.

BackgroundColor

You can set the background color for HTML elements:

HelloWorld

BackgroundColor feature

Example

```
<h1style="background-color:DodgerBlue;">HelloWorld</h1>  
<pstyle="background-color:Tomato;">Loremipsum...</p>Text
```

Color

You can set the color of text:

Hello World

Welcome to HTML How

are you...

Example

```
<h1style="color:Tomato;">HelloWorld</h1>  
<pstyle="color:DodgerBlue;">Welcome to HTML</p>  
<pstyle="color:MediumSeaGreen;">How are you...</p>
```

Non-breaking Space

A commonly used entity in HTML is the non-breaking space: ** **

A non-breaking space is a space that will not break into a new line.

Two words separated by a non-breaking space will stick together (not break into a new line). This is handy when breaking the words might be disruptive.

Examples:

- §10
- 10km/h
- 10PM

Another common use of the non-breaking space is to prevent browsers from truncating spaces in HTML pages.

If you write 10 spaces in your text, the browser will remove 9 of them. To add real space to your text, you can use the ** ** character entity.

Tip: The non-breaking hyphen ([‑](#)) is used to define a hyphen character (-) that does not break into a new line.

HTML <hr> Tag

Example

Use the <hr> tag to define thematic changes in the content:

<h1>The Main Languages of the Web</h1>

<p>HTML is the standard markup language for creating Web pages. HTML describes the structure of a Web page, and consists of a series of elements. HTML elements tell the browser how to display the content.</p>

<hr>

<p>CSS is a language that describes how HTML elements are to be displayed on screen, paper, or in other media. CSS saves a lot of work, because it can control the layout of multiple web pages all at once.</p>

<hr>

<p>JavaScript is the programming language of HTML and the Web. JavaScript can change HTML content and attribute values. JavaScript can change CSS. JavaScript can hide and show HTML elements, and more.</p>

Definition and Usage

The <hr> tag defines a thematic break in an HTML page (e.g. a shift of topic).

The <hr> element is most often displayed as a horizontal rule that is used to separate content (or define a change) in an HTML page.

Example

Align an <hr> element (with CSS):

<hr style="width:50%;text-align:left;margin-left:0">

Example

An unshaded <hr> (with CSS):

`<hrstyle="height:2px;border-width:0;color:gray;background-color:gray">`

Example

Set the height of a `<hr>` element (with CSS):

`<hrstyle="height:30px">`

Example

Set the width of a `<hr>` element (with CSS):

`<hrstyle="width:50%">`

UNIT2

WorkingwithHTML

- HTML Lists are used to specify lists of information. All lists may contain one or more list elements. There are three different types of HTML lists:
- Ordered List or Numbered List()
- Unordered List or Bulleted List()
- Description List or Definition List(<dl>)

- In the ordered HTML lists, all the list items are marked with numbers by default. It is known as numbered list also. The ordered list starts with `<ol>` tag and the list items start with `<li>` tag.
- `<ol>`
- `<li>HTML</li>`
- `<li>CSS</li>`
- `<li>JS</li>`
- `<li>JAVA</li>`
- `</ol>`

We can use ordered list to represent items either in numerical order format or alphabetical order format, or any format where an order is emphasized. There can be different types of numbered list:

- NumericNumber (1,2,3)
- CapitalRomanNumber(IIIII)
- SmallRomanNumber(iiiiii)
- CapitalAlphabet(AB C)
- SmallAlphabet(abc)

To represent different ordered lists, there are 5 types of attributes in tag

Type	Description
Type"1"	This is the default type. In this type, the list items are numbered with numbers.
Type"I"	In this type, the list items are numbered with upper case roman numbers.
Type"i"	In this type, the list items are numbered with lower case roman numbers.
Type"A"	In this type, the list items are numbered with upper case letters.
Type"a"	In this type, the list items are numbered with lower case letters.

```
<!DOCTYPEhtml>
<html>
<body>
<ol>
  <li>HTML</li>
  <li>Java</li>
  <li>JavaScript</li>
  <li>SQL</li>
</ol>
</body>
</html>
```

1. HTML
2. Java
3. JavaScript
4. SQL

StartAttribute: The start attribute is used with `<ol>` tag to specify from where to start the list items.

- **`<ol type="1" start="5">`**: It will show numeric values starting with "5".
- **`<ol type="A" start="5">`**: It will show capital alphabets starting with "E".
- **`<ol type="a" start="5">`**: It will show lowercase alphabets starting with "e".
- **`<ol type="I" start="5">`**: It will show Roman upper case value starting with "V".

reversedAttribute:

This is a Boolean attribute of HTML `<ol>` tag, and it is new in HTML5 version. If you use the reversed attribute with tag then it will number the list in descending order (7, 6, 5, 4.....1). Here is example:

```
<!DOCTYPEhtml>
```

```
<html>
```

```
<head>
```

```
</head>
```

```
<body>
```

```
<olreversed>
```

```
<li>HTML</li>
```

```
<li>Java</li>
```

```
<li>JavaScript</li>
```

```
<li>SQL</li>
```

```
</ol>
```

```
</body>
```

```
</html>
```

4. HTML

3. Java

2. JavaScript

1. SQL

UnorderedListorBulletedList

- **HTML Unordered List** or Bulleted List displays elements in bulleted format . We can use unorderedlistwherewedonotneedtodisplay items in any particular order. The HTML tag is used for the unordered list. There can be 4 types of bulleted list:
 - disc
 - circle
 - square
 - none

- To represent different ordered lists, there are 4 types of attributes in tag.

Type	Description
Type"disc"	This is the default style. In this style, the list items are marked with bullets.
Type"circle"	In this style, the list items are marked with circles.
Type"square"	In this style, the list items are marked with squares.
Type"none"	In this style, the list items are not marked.

Example:

```
<!DOCTYPEhtml>
<html>
<body>
<h1>Hereisanexampleof
unordered list</h1>
<ul>
<li>HTML</li>
<li>Java</li>
<li>JavaScript</li>
<li>SQL</li>
</ul>
</body>
</html>
```

Output:

Hereisanexampleof unordered list

- HTML
- Java
- JavaScript
- SQL

Note:bydefaultunorderedlist type is disc.

Otheryoucantryusingtype attribute in tag

- **HTML Description List** or Definition List displays elements in definition form like in dictionary. The `<dl>`, `<dt>` and `<dd>` tags are used to define description list.
- The 3 HTML description list tags are given below:
- **`<dl>` tag** defines the description list.
- **`<dt>` tag** defines a term.
- **`<dd>` tag** defines a definition (description).

Example

```
<!DOCTYPEhtml>
<html>
<body>
<p><b>HereisanexampleofDescription/DefinitionListinHTML</b></p>
<dl>
  <dt>HTML</dt>
  <dd>isamarkuplanguageanddevelpoedbyTimBernersleein1992</dd>
  <dt>Java</dt>
  <dd>isaprogramminglanguageandplatformIndependent</dd>
  <dt>JavaScript</dt>
  <dd>isascriptinglanguage</dd>
  <dt>SQL</dt>
  <dd>isaquerylanguage</dd>
</dl>
</body>
</html>
```

Here is an example of Description/ Definition List in HTML

HTML

is a markup language and develpoed by Tim Berners lee in 1992

Java

is a programming language and platform Independent

JavaScript

is a scripting language

SQL

is a query language

WorkingwithImage

- **HTML tag** is used to display image on the web page. HTML tag is an empty tag that contains attributes only, closing tags are not used in HTML image element.

Attributes of HTML img tag

- The src and alt are important attributes of HTML img tag. All attributes of HTML image tag are given in next slide:

src: It is a necessary attribute that describes the source or path of the image. It instructs the browser where to look for the image on the server. The location of the image may be on the same directory or another server.

alt: The alt attribute defines an alternate text for the image, if it can't be displayed.

ImageSize-WidthandHeight:usestyleattribute to specify the width and height of an image.

Example

```
<imgsrc="img_flower.jpg"alt="Pink
```

```
Flower"style="width:500px;height:600px;">
```

Alternatively,the`width`and`height`attributescan be used as given below:

Example

- `<imgsrc="img_flower.jpg"alt="Pink Flower"width="500"height="600">`
- The`width`and`height`attributesalwaysdefine the width and height of the image in pixels.

- Other attributes are:
 1. Border
 2. Hspace
 3. Vspace
 4. Align(left,center,right)

```
<!DOCTYPEhtml>
```

```
<html>
```

```
<body>
```

```
<p>UseimagetaginHTMLalongwithallthe attributes</p>
```

```

```

```
</body>
```

```
</html>
```

Use image tag in HTML along with all the attributes



WorkingwithLinks

- **HTML anchor tag** defines *a hyperlink that links one page to another page*. It can create hyperlink to other web pages as well as files, location, or any URL. The "href" attribute is the most important attribute of the HTML <a> tag which links to destination page or URL.

href attribute: The href attribute is used to define the address of the file to be linked. In other words, it points out the destination page.

The syntax of HTML anchor tag is given below.

- `<a href=".....">LinkText</a>`

- **target attribute:** By default, the linked page will be displayed in the current browser window. To change this, you must specify another target for the link.
- The `target` attribute specifies where to open the linked document.
- The `target` attribute can have one of the following values:

`_self`: Default. Open the document in the same window/tab as it was clicked

`_blank`: Open the document in a new window or tab

`_parent`: Open the document in the parent frame

`_top`: Open the document in the full body of the window

```
<!DOCTYPEhtml>
```

```
<html>
```

```
<body>
```

```
<h2>TheanchortagalongwithhrefandtargetAttribute</h2>
```

```
<a href="https://www.youtube.com/" target="_blank">Visit  
youtube!</a>
```

```
<p>If target="_blank", the link will open in a new browser window  
or tab.</p>
```

```
</body>
```

```
</html>
```

The anchor tag along with href and target Attribute

[Visit youtube!](https://www.youtube.com/)

If target="_blank", the link will open in a new browser window or tab.

AbsoluteURLsvs.RelativeURLs

A local link (a link to a page within the same website) is specified with a **relativeURL** (without the "https://www" part):

Example:

- External Linking

```
<h2>AbsoluteURLs</h2>
```

```
<p><a href="https://www.w3.org/">W3C</a></p>
```

```
<p><a href="https://www.google.com/">Google</a></p>
```

- Internal Linking

```
<h2>RelativeURLs</h2>
```

```
<p><a href="html_images.asp">HTMLImages</a></p>
```

```
<p><a href="/css/default.asp">CSSTutorial</a></p>
```

Unit 3

DesigningwithHTML

Creating Tables

- **HTML table tag** is used to display data in tabular form (row * column). There can be many columns in a row.
- We can create a table to display data in tabular form, using `<table>` element, with the help of `<tr>`, `<td>`, and `<th>` elements.
- In Each table, table row is defined by `<tr>` tag, table header is defined by `<th>`, and table data is defined by `<td>` tags.

Tag	Description
<table>	It defines a table.
<tr>	It defines a row in a table.
<th>	It defines a header cell in a table.
<td>	It defines a cell in a table.
<caption>	It defines the table caption.
<tbody>	It is used to group the body content in a table.
<thead>	It is used to group the header content in a table.
<tfooter>	It is used to group the footer content in a table.

Important table tags

<table>: It defines a table.

<tr>: It defines a row in a table.

<th>: It defines a header cell in a table.

<td>: It defines a cell in a table.

<caption>: It defines the table caption.

<tbody>: It is used to group the body content in a table.

<thead>: It is used to group the header content in a table.

<tfooter>: It is used to group the footer content in a table.

```

<!DOCTYPE>
<html>
<body>
<!--hereisstep1oftableformation-->
<table>
<caption><b>Student_Info</b></caption>
<tr><th>First_Name</th><th>Last_Name</th><th>Marks</th></tr>
<tr><td>Abay</td><td>Singh</td><td>95</td></tr>
<tr><td>Jatin</td><td>Singh</td><td>85</td></tr>
<tr><td>Ankush</td><td>Kumar</td><td>87</td></tr>
<tr><td>Mohan</td><td>Kumar</td><td>90</td></tr>
</table>
</body>
</html>

```

Student_Info		
First_Name	Last_Name	Marks
Abay	Singh	95
Jatin	Singh	85
Ankush	Kumar	87
Mohan	Kumar	90

There are two ways to specify border for HTML tables.

a. By border attribute of table in HTML

b. By border property in CSS

1) HTML Border attribute: Use border attribute of table tag in HTML to specify border. But it is not recommended now.

2) CSS Border property: It is now recommended to use border property of CSS to specify border in table.

```
<!DOCTYPE>
<html>
<body>
<!--Step2-->
<tableborder="1">
<caption><b>Student_Info</b></caption>
<tr><th>First_Name</th><th>Last_Name</th><th>Marks</th></tr>
<tr><td>Abay</td><td>Singh</td><td>95</td></tr>
<tr><td>Jatin</td><td>Singh</td><td>85</td></tr>
<tr><td>Ankush</td><td>Kumar</td><td>87</td></tr>
<tr><td>Mohan</td><td>Kumar</td><td>90</td></tr>
</table>
</body>
</html>
```

Student_Info		
First_Name	Last_Name	Marks
Abay	Singh	95
Jatin	Singh	85
Ankush	Kumar	87
Mohan	Kumar	90

```
<!DOCTYPE>
```

```
<html>
```

```
<head>
```

```
<style>table,
```

```
th,td{
```

```
    border: 1px solid black;
```

```
    border-collapse:collapse;
```

```
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<table border="1">
```

```
<caption><b>Student_Info</b></caption>
```

```
<tr><th>First_Name</th><th>Last_Name</th><th>Marks</th></tr>
```

```
<tr><td>Abay</td><td>Singh</td><td>95</td></tr>
```

```
<tr><td>Jatin</td><td>Singh</td><td>85</td></tr>
```

```
<tr><td>Ankush</td><td>Kumar</td><td>87</td></tr>
```

```
<tr><td>Mohan</td><td>Kumar</td><td>90</td></tr>
```

```
</table>
```

```
</body>
```

```
</html>
```

Student_Info

First_Name	Last_Name	Marks
Abay	Singh	95
Jatin	Singh	85
Ankush	Kumar	87
Mohan	Kumar	90

HTML CellPadding & Cell Spacing:

HTML tables can adjust the padding inside the cells, and also the space between the cells using cellspacing and cellpadding attributes of <table> tag.

WithPadding		
hello	hello	hello
hello	hello	hello
hello	hello	hello

WithSpacing		
hello	hello	hello
hello	hello	hello
hello	hello	hello

Other attributes of <table> tag are given below:

- Bordercolor
- Bgcolor
- Align
- Height
- Weight

rowspanandcolspan

HTML tables can have cells that span over multiple rows and/or columns.

colspan: To make a cell span over multiple columns, use the colspan attribute. The value of the colspan attribute represents the number of columns to span.

rowspan: To make a cell span over multiple rows, use the rowspan attribute. The value of the rowspan attribute represents the number of rows to span.

Example of colspan

```
<!DOCTYPEhtml>
<html>
<head>
<style>
table,th,td{
  border:1pxsolidblack;

  border-collapse:collapse;
}
</style>
</head>
<body>
<h2>Cellthatspanstwocolumns</h2>
<p>Tomakeacellspanmorethanonecolumn,
usethecolspanattribute.</p>
```

```
<tablestyle="width:100%">
<tr>
  <thcolspan="2">Name</th>
  <th>Age</th>
</tr>
```

```
<tr>
  <td>Abay</td>
  <td>Singh</td>
  <td>47</td>
</tr>
<tr>
  <td>Mohan</td>
  <td>Singh</td>
  <td>67</td>
</tr>
</table>
</body>
</html>
```

Output:

Cell that spans two columns

To make a cell span more than one column, use the colspan attribute.

Name		Age
Abay	Singh	47
Mohan	Singh	67

Example of rowspan

```
<!DOCTYPEhtml>
<html>
<head>
<style>
table,th,td{
    border: 1px solid black;
    border-collapse:collapse;
}
</style>
</head>
<body>

<h2>Cell thatspanstworows</h2>
<p>Tomakeacellspanmorethanone
row, use the rowspan attribute.</p>
```

```
<tablestyle="width:100%">
```

```
<tr>
    <th>Name</th>
    <td>Mohan</td>
</tr>
<tr>
    <th>Phone</th>
    <td>9876543210</td>
</tr>
<tr>
    <td>8907654321</td>
</tr>
</table>
</body>
</html>
```

Cell that spans two rows

To make a cell span more than one row, use the rowspan attribute.

Name	Mohan
Phone	9876543210
	8907654321

Workingwithframes

HTML frames are used to divide your browser window into multiple sections where each section can load a separate HTML document.

A collection of frames in the browser window is knownasaframeset.Thewindowisdivided into frames in a similar way the tables are organized: into rows and columns.

To use frames on a page we use `<frameset>` tag

instead of `<body>` tag.

The `<frameset>` tag defines, how to divide the window into frames.

The **rows** attribute of `<frameset>` tag defines horizontal frames and **cols** attribute defines vertical frames. Each frame is indicated by `<frame>` tag and it defines which HTML document shall open into the frame.

```
<!DOCTYPEhtml>
<html>

  <head>
    <title>HTML Frames</title>
  </head>

  <framesetrows="10%,80%,10%">
<!--attheplaceofrowswecanusecolsarewell...TrythisinLab-->
    <frame name="top"src="/html/top_frame.htm"/>
    <frame name="main"src="/html/main_frame.htm"/>
    <frame name="bottom"src="/html/bottom_frame.htm"/>

    <noframes>
      <body>Yourbrowserdoesnotsupportframes.</body>
    </noframes>

  </frameset>

</html>
```

Note:The<frame>tag deprecated in HTML5. So instead of using <frame> we can use the concept of <iframe >.

HTMLForm

HTMLformsarerequiredifyouwanttocollect some data from of the site visitor.

For example: If a user want to purchase some items on internet, want to fill the form, shipping addressandcredit/debitcarddetailssothatitem can be sent to the given address.

Tag	Description
<form>	It defines an HTML form to enter inputs by the user side.
<input>	It defines an input control.
<textarea>	It defines a multi-line input control.
<label>	It defines a label for an input element.
<legend>	It defines a caption for a <fieldset> element.
<button>	It defines a clickable button.

- HTML<form>element
- The HTML <form> element provide a document section to take input from user. It provides various interactive controls for submitting information to web server such as text field, text area, password field, etc.

Note: The <form> element does not itself create a form but it is container to contain all required form elements, such as <input>, <label>, etc.

Syntax:

<form>

//Form elements

</form>

HTML<input> element

- The HTML <input> element is fundamental form element. It is used to create form fields, to take input from user. We can apply different input field to gather different information from user. Following is the example to show the simple text input.

Example:

```
<body>
```

```
  <form>
```

```
    Enteryourname<br>
```

```
    <input type="text" name="username">
```

```
  </form>
```

```
</body>
```

An<input>element can be displayed in many ways, depending on the type attribute given in next slide.

Type	Description
<code><input type="text"></code>	Displays a single-line text input field
<code><input type="radio"></code>	Displays a radio button (for selecting one of many choices)
<code><input type="checkbox"></code>	Displays a checkbox (for selecting zero or more of many choices)
<code><input type="submit"></code>	Displays a submit button (for submitting the form)
<code><input type="button"></code>	Displays a clickable button

Other permissible values in type attribute are given below:

- `<input type="color">`
- `<input type="date">`
- `<input type="email">`
- `<input type="file">`
- `<input type="number">`
- `<input type="password">`
- `<input type="search">`
- `<input type="time">`
- `<input type="url">`
- `<input type="week">`

The<label> Element

- The <label> element is useful for screen-reader users, because the screen-reader will read out loud the label when the user focuses on the input element.
- The <label> element also helps users who have difficulty clicking on very small regions (such as radio buttons or checkboxes) - because when the user clicks the text within the <label> element, it toggles the radio button/checkbox.
- The **for** attribute of the <label>tag should be equal to the **id** attribute of the <input>element to bind them together.

```
<!DOCTYPEhtml>
```

```
<html>
```

```
<head>
```

```
<body>
```

```
<h2>UseHtmlformandcreateaformwithFirstNameandLastName</h2>
```

```
<p>Forthisuselabelninputtagsalongwithsuitableattributesandmakeaform more  
attractive. </p>
```

```
<form>
```

```
  <labelfor="firstname">FirstName:</label>
```

```
    <inputtype="text"id="firstname"name="firstname"/>
```

```
    <br/>
```

```
    <br/>
```

```
  <labelfor="lastname">LastName:</label>
```

```
    <inputtype="text"id="lastname"name="lastname"/><br/>
```

```
</form>
```

```
</body>
```

```
</html>
```

Use Html form and create a form with First Name and Last Name

For this use label n input tags along with suitable attributes and make a form more attractive.

First Name:

Last Name:

```
<!DOCTYPEhtml>
```

```
<html>
```

```
<body>
```

```
<h2>RadioButtons</h2>
```

```
<p>ChooseyourfavoriteWeblanguage:</p>
```

```
<form>
```

```
<inputtype="radio"id="html"name="fav_language"value="HTML">
```

```
<labelfor="html">HTML</label><br>
```

```
<inputtype="radio"id="css"name="fav_language"value="CSS">
```

```
<labelfor="css">CSS</label><br>
```

```
<inputtype="radio"id="javascript"name="fav_language"value="JavaScri  
pt">
```

```
<labelfor="javascript">JavaScript</label>
```

```
</form>
```

```
</body>
```

```
</html>
```

Radio Buttons

Choose your favorite Web language

- ☐ HTML
- ☐ CSS
- ☐ JavaScript

Note:TrycheckboxesinLab

UNIT4

What is JavaScript?

JavaScript (JS) is a light-weight object-oriented programming language which is used by several websites for scripting the webpages. It is an interpreted, full-fledged programming language that enables dynamic interactivity on websites when applied to an HTML document.

It was introduced in the year 1995 for adding programs to the webpages in the Netscape

Navigator browser. Since then, it has been adopted by all other graphical web browsers. With JavaScript, users can build modern web applications to interact directly without reloading the page every time.

Although, JavaScript has no connectivity with Java programming language. The name was suggested and provided in the times when Java was gaining popularity in the market.

Another popular use of JS: In addition to web browsers, databases such as CouchDB and Mongo DB use JavaScript as their scripting and query language.

Features of JavaScript

There are following features of JavaScript:

1. All popular web browsers support JavaScript as they provide built-in execution environments.
2. JavaScript follows the syntax and structure of the C programming language. Thus, it is a structured programming language.
3. JavaScript is a weakly typed language, where certain types are on implicitly cast (depending the operation).
4. JavaScript is an object-oriented programming language that uses prototypes rather than using classes for inheritance.
5. It is a light-weighted and interpreted language.
6. It is a case-sensitive language.
7. JavaScript is supportable in several operating systems including, Windows, macOS, etc.
8. It provides good control to the users over the web browsers.

JavaScript History

JavaScript was invented by **Brendan Eich** in 1995.

It was developed for **Netscape 2**, and became the **ECMA-262** standard in 1997.

After Netscape handed JavaScript over to ECMA, the Mozilla foundation continued to develop JavaScript for the Firefox browser. Mozilla's latest version was 1.8.5. (Identical to ES5).

Internet Explorer (IE4) was the first browser to support ECMA-262 Edition 1 (ES1).

Year	ECMA	Browser
1995		JavaScript was invented by Brendan Eich
1997		JavaScript became an ECMA standard (ECMA-262)
1997	ES1	ECMAScript 1 was released

1998	ES2	ECMAScript2wasreleased
2009	ES5	ECMAScript5wasreleased
2015	ES6	ECMAScript6wasreleased
2017	ES6	FullsupportforES6inFirefox54
2017	ES6	FullsupportforES6 inEdge15
2018	ES6	FullsupportforES6inallbrowsers**

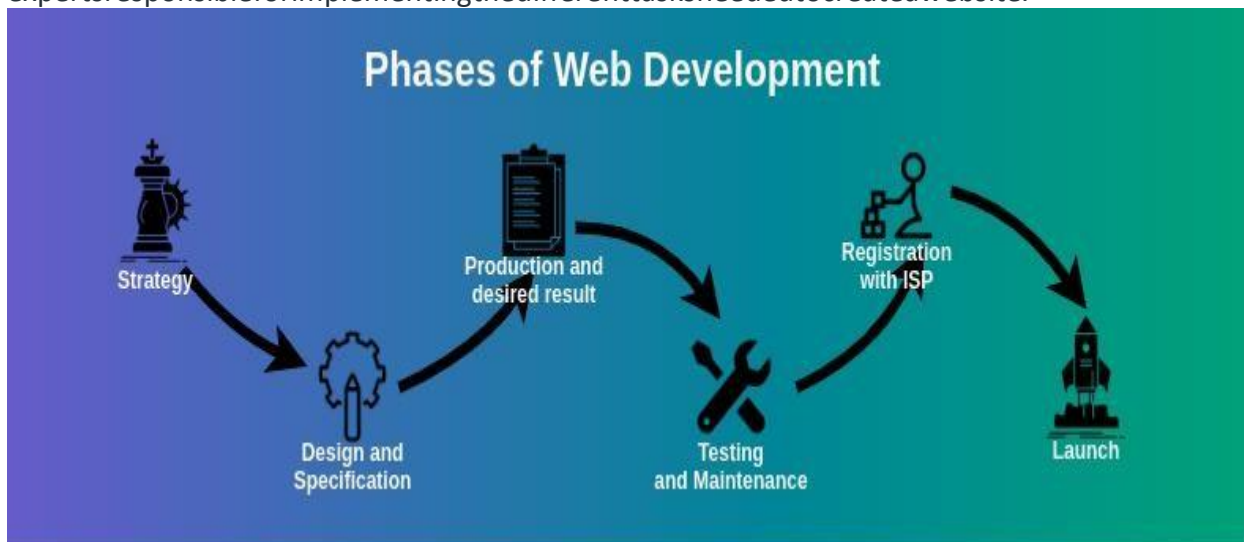
The main difference between client-side and server-side scripting is given below.

Basis	Client-side Scripting	Server-side Scripting
Primary Function	The primary function of client-side scripting is to provide the requested output to the end-user	The primary function of server-side scripting is to manipulate and give access to the required database as per request.
Uses	The client side is used as the front end, where the user gets to see what we have browsed.	The server side is used as a backend where data is processed and is not visible to the client user.
Code allowance	On the client side, the user is allowed to access the code written after verifying the user's need.	Server-side scripting allows the back-end developer to hide the source code from the user.
Processing	The client-side does not need any interaction with the server.	Server-side scripting on the other hand is all about communicating with the servers.
Function	Used for the visibility and getting out the required data from servers' database	Used for the customization or modification of the data to change the website dynamically.
Dependability	Client-side scripting depends upon the user's browser version.	Server-side does not depend on the client.

Basis	Client-side Scripting	Server-side Scripting
Security	This way of scripting is less secure than Server-side scripting because of the accessibility of code provided to the client.	Server-side scripting is considered a more secure way of working on a web application
Connectivity	The client-side does not connect to the database at the webserver.	The server side helps connect with the database, which is already stored in the server database.
File Access	It does have any access to the files present on the web servers. But we have the option to upload files from the front end	It has total access to the files which are stored in the web database server.
Occurrence	It occurs when the browser processes all the codes, and then it reacts according to the client's query.	It only acts after the client initiates the browsing request.
Running	It runs on the end-user's system.	It runs on the webserver.
Languages	HTML, JavaScript, and CSS are used to display the request	PHP, Python, Ruby, Node.js are some of the programming languages used on server-side

Phases of web Development

Web Development refers to a term that includes all the processes involved in developing a web project or website. It contains the various phases such as planning, designing, Implementation & testing, and launching of the web project. The web development process requires a team of experts responsible for implementing the different tasks needed to create a website.



The various stages that are needed in order to develop a web project in web development are as following:

Strategy: The first step in the web development process for a developer is to make a strategy for developing a webpage or website. In the strategy phase, web developer has to do the following:

- Deciding goals and objectives
- Developing team

- Make the appropriate analysis associated with problem and review the analysis
- Formulate a list of tasks
- Proposal of project to web team for developing

Specification and Design: After the strategy-making, the next step in the web development process is to develop a planned work. Web developer has to determine the schedule and the specifications. The tasks in this phase are as follows.

- Developing approach
- Selection of frontend (client side) or backend (server side) programming languages
- Planning of contents needed for use
- Making of rough design for project
- Making of final design from rough design, if there is no considerable modification in rough design.
- Frame a prototype (dummy website) of project for developing
- Test the prototype

If prototype is accomplished, then go to next phase phase-3 otherwise repeat the phase 2 until prototype is accomplished.

Production of desired result: In this phase of the web development process the actual functional site is built. After the proper testing of the prototype, the developer has to work on developing the actual live web project. The actual live web project is built according to the requirements of the client. Web developer has to consider all the situations from the design phase to create all the features in the web project.

This phase involves both front end development and back end development of the website.

Front end development comprises of the writing codes with the basic technologies like HTML, CSS, JavaScript etc. according to the web standards. It generally starts by developing the home page first and then other pages.

Back end development is also completed in this phase by installing and configuring the content management systems, databases, and frameworks using PHP, Python, Ruby, SQL etc.

After completing all the steps that were finalized in the strategy and design phase by which the original website becomes functional, it is tested in the next phase.

Testing and Maintenance: Testing is an important phase in the web development process. Testing is performed by the developers and testers to ensure the client's requirements after completion of the web project.

Registration with ISP: After completion of the Testing and Maintenance and removing all the bugs from the project, the next step or phase is to register the web project with the regional ISP to make the web project legal.

The client has to select and decide the ISP which provides domain name registration and web hosting services.

Launch: This is the last phase of the web development process. Project is launched after getting registered with ISP. After launching, web project is publicly accessed by the users of the particular domain. The tasks performed in the launch phase are as follows.

- Migration of data
- Launching of server
- Merging of code
- Redirecting domain name

Application of JavaScript

JavaScript is used to create interactive websites. It is mainly used for:

- Client-side validation,
- Dynamic drop-down menus,
- Displaying date and time,
- Displaying pop-up windows and dialog boxes (like an alert dialog box, confirm dialog box and prompt dialog box),
- Displaying clock etc.

NOTE: JavaScript example is easy to code. JavaScript provides 3 places to put the JavaScript code:

a. within body tag

b. within head tag

c. external JavaScript file (.js file)

Example 1 (within body Tag)

```
<html>
<head><title>MyfirstJSprogram</title></head>
  <body>
    <h2>WelcometoJavaScript</h2>
    <script>
      document.write("JavaScriptis a simple language for learners");
    </script>
  </body>
</html>
```

Output:

WelcometoJavaScript

JavaScript is a simple language for learners

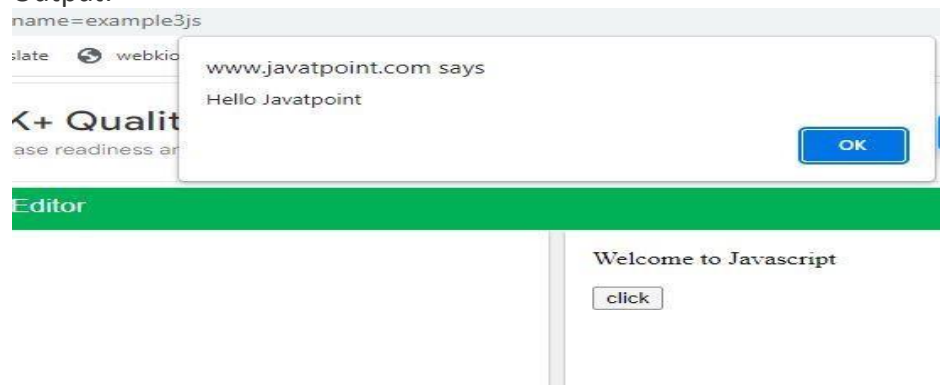
Highlight of Example 1:

The `document.write()` function is used to display dynamic content through JavaScript. We will learn about document object in detail later.

Example 2: within head section

```
<html>
<head>
  <script>function msg()
  {
    alert("Hello Javatpoint");
  }
</script>
</head>
<body>
<p>Welcome to Javascript</p>
<form>
<input type="button" value="click" onclick="msg()" />
</form>
</body>
</html>
```

Output:



How to add comments in JS: The **JavaScript comments** are a meaningful way to deliver a message. It is used to **add information about the code, warnings or suggestions** so that end users can easily interpret the code.

The JavaScript comment is ignored by the JavaScript engine i.e. embedded in the browser.

Advantages of JavaScript comments:

There are mainly two advantages of JavaScript comments.

To make code easy to understand: It can be used to elaborate the codes so that end users can easily understand the code.

To avoid the unnecessary code: It can also be used to avoid the code being executed. Sometimes, we add the code to perform some action. But after some time, there may be a need to disable the code. In such case, it is better to use comments.

Types of JavaScript Comments

There are two types of comments in JavaScript.

Single-line Comment

Multi-line Comment

JavaScript Single-line Comment

It is represented by double forward slashes (//). It can be used before and after the statement.

Let's see the example of single-line comment i.e. added before the statement.

<script>

// It is single line comment

document.write("hello javascript");

</script>

JavaScript Multiline Comment

It can be used to add single as well as multiline comments. So, it is more convenient.

It is represented by forward slash with asterisk then asterisk with forward slash at the end. For example:

/*your code here*/

It can be used before, after and middle of the statement.

<script>

/* It is multiline comment.

It will not be displayed */

document.write("example of javascript multiline comment");

</script>

JavaScript Variable:

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

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

1. Name must start with a letter (a to z or A to Z), underscore (_), or dollar (\$) sign.

2. After first letter we can use digits (0 to 9), for example value1.
3. JavaScript variables are case sensitive, for example x and X are different variables.

Example:

```
<html>
<head><title>Variable in JS program</title></head>
<body>
<script>var
x=10;
y=20;
z=x+y;
document.write(z);
</script>
</body>
</html>
```

Output: 30

Operators in JavaScript:

There are following types of operators in JavaScript.

1. Arithmetic Operators
2. Comparison (Relational) Operators
3. Bitwise Operators
4. Logical Operators
5. Assignment Operators
6. Special Operators

JavaScript Arithmetic Operators

Arithmetic operators are used to perform arithmetic operations on the operands. The following operators are known as JavaScript arithmetic operators.

Operator	Description	Example
+	Addition	10+20=30
-	Subtraction	20-10=10
*	Multiplication	10*20=200
/	Division	20/10=2

%	Modulus(Remainder)	20%10=0
++	Increment	vara=10;a++;Nowa= 11
--	Decrement	vara=10;a--; Nowa=9

JavaScriptComparisonOperators

TheJavaScriptcomparisonoperatorcomparesthetwooperands.Thecomparisonoperatorsare as follows:

Operator	Description	Example
==	Isequal to	10==20= false
!=	Notequal to	10!=20=true
!==	NotIdentical	20!==20=false
>	Greaterthan	20>10=true
>=	Greaterthanorequal to	20>=10= true
<	Less than	20<10=false
<=	Lessthanorequal to	20<=10= false

JavaScriptBitwiseOperators

The bitwise operators perform bitwise operations on operands. The bitwise operators are asfollows:

Operator	Description	Example
&	BitwiseAND	(10==20&20==33)= false
	BitwiseOR	(10==20 20==33)= false
~	BitwiseNOT	(~10)=-10

JavaScriptLogicalOperators

ThefollowingoperatorsareknownasJavaScript logical operators.

Operator	Description	Example
&&	LogicalAND	(10==20&&20==33)=false
	LogicalOR	(10==20 20==33)= false
!	LogicalNot	!(10==20)=true

JavaScriptAssignmentOperators

ThefollowingoperatorsareknownasJavaScriptassignmentoperators.

Operator	Description	Example
=	Assign	10+10=20
+=	Addand assign	vara=10;a+=20; Nowa=30
-=	Subtractandassign	vara=20;a-=10;Nowa=10
=	Multiplyandassign	vara=10;a=20; Nowa=200
/=	Divideandassign	vara=10;a/=2; Nowa=5
%=	Modulusandassign	vara=10;a%=2;Nowa=0

JavaScriptSpecialOperators

ThefollowingoperatorsareknownasJavaScriptspecial operators.

Operator	Description
(?:)	ConditionalOperatorreturnsvalue basedonthecondition.Itislikeif-else.
,	CommaOperatorallowsmultipleexpressionstobeevaluatedassinglestatement.
delete	DeleteOperatordeletesapropertyfromthe object.
in	InOperatorchecksifobjecthasthegivenproperty
new	createsaninstance(object)
typeof	checksthetypeofobject.

void	it discards the expression's return value.
------	--

Ex1:

Program of addition, subtraction, multiply, and division in JavaScript

```
<!doctypehtml>
```

```
<html>
```

```
<body>
```

```
<script>
```

```
var numOne=12,numTwo=10,res; res
```

```
= numOne + numTwo;
```

```
document.write("Add="+res+"<br/>"); res
```

```
= numOne - numTwo;
```

```
document.write("Subtract="+res+"<br/>"); res =
```

```
numOne * numTwo;
```

```
document.write("Multiply="+res+"<br/>"); res =
```

```
numOne/numTwo; document.write("Divide = "
```

```
+ res + "<br/>");
```

```
</script>
```

```
</body>
```

```
</html>
```

Output:

Add=22

Subtract=2

Multiply=120

Divide=1.2

Different ways to declare Variables in JavaScript:

- Using **var**
- Using **let**

- Using **const**
- Using **nothing**

```
<!DOCTYPEhtml>
<html>
<body>
<h2>JavaScriptVariables</h2>
<p>Inthisexample,x,y,andzarevariables.</p>
<pid="demo"></p>
<script>|
  etx=5;
  lety=6;
  letz=x+y;
  document.getElementById("demo").innerHTML="Thevalueofzis:"+z; document.write("The
  value of z is:" + z);
</script>
</body>
</html>
```

JavaScriptVariables

Inthisexample,x,y,andzarevariables. The

value of z is: 11

Thevalue ofz1is:11

WhentoUseJavaScript var?

AlwaysdeclareJavaScriptvariableswith**var**,**let**,or**const**.

The**var**keywordisusedinalJavaScriptcodefrom1995to2015. The **let**

and **const** keywords were added to JavaScript in 2015.

Ifyouwantyour codetorun in olderbrowsers,youmustuse**var**.

WhentoUseJavaScriptconst?

Ifyouwantageneralrule:alwaysdeclarevariableswith**const**. If you

think the value of the variable can change, use **let**.

Inthisexample, **price1**,**price2**,and**total**,arevariables:

Example

```
const price1=5;  
const price2=6;  
let total=price1+price2;
```

The two variables `price1` and `price2` are declared with the `const` keyword. These are constant values and cannot be changed.

The variable `total` is declared with the `let` keyword. This is a value that can be changed.

Different Kinds of Loops

JavaScript supports different kinds of loops:

- `for` - loop through a block of code a number of times
- `while` - loop through a block of code while a specified condition is true
- `do/while` - also loop through a block of code while a specified condition is true

For Loop

The `for` statement creates a loop with 3 optional expressions:

```
for(expression1; expression2; expression3)  
{  
  //code block to be executed  
}
```

Expression1 (Initialization) is executed (one time) before the execution of the code block.

Expression2 (Condition) defines the condition for executing the code block.

Expression3 (Increment/Decrement) is executed (every time) after the code block has been executed.

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

```
<h2>JavaScript For Loop</h2>
```

```
<pid="demo"></p>
```

```
<script>
lettext="";
for(let i=0;i<5; i++)
{
  text+="Thenumber is"+i+"<br>";
}
document.getElementById("demo").innerHTML=text;
</script>
</body>
</html>
```

Fromtheexample above,youcan read:

Expression1setsavariablenbeforetheloopstarts(leti=0).

Expression2definestheconditionforthelooptorun(imustbelessthan 5).

Expression3increasesavalue(i++)eachthetimethe codeblock inthe loophas beenexecuted. Another

Example:

```
<!DOCTYPEhtml>
<html>
<body>

<h2>JavaScriptForLoop</h2>

<pid="demo"></p>

<script>
constp_lang=["html","css","javascript","java","c","c++"];

lettext="";
for(leti=0;i<p_lang.length;i++){ text +=
  p_lang[i] + "<br>";
}

document.getElementById("demo").innerHTML=text;
</script>

</body>
</html>
```

JavaScriptWhileLoop

Loops can execute a block of code as long as a specified condition is true.

The While Loop: The **while** loop loops through a block of code as long as a specified condition is true.

Syntax

```
while(condition){  
    //code block to be executed  
}
```

Example

In the following example, the code in the loop will run, over and over again, as long as a variable (i) is less than 10:

```
<!DOCTYPE html>  
<html>  
<body>  
  
<h2>JavaScriptWhileLoop</h2>  
  
<p id="demo"></p>  
  
<script>  
let text="";  
let i = 0;  
while(i < 10){  
    text+="<br>The number is "+i; i++;  
}  
document.getElementById("demo").innerHTML=text;  
</script>  
  
</body>  
</html>
```

JavaScriptWhileLoop

The number is 0
The number is 1
The number is 2
The number is 3
The number is 4

Thenumberis5
Thenumberis6
Thenumberis7
Thenumberis8
Thenumberis9

If you forget to increase the variable used in the condition, the loop will never end. This will crash your browser.

The Do While Loop

The **do while** loop is a variant of the while loop. This loop will execute the code block once, before checking if the condition is true, then it will repeat the loop as long as the condition is true.

Syntax

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

Example

The example below uses a **dowhile** loop. The loop will always be executed at least once, even if the condition is false, because the code block is executed before the condition is tested:

```
<!DOCTYPEhtml>  
<html>  
<body>  
  
<h2>JavaScriptDoWhileLoop</h2>  
  
<pid="demo"></p>  
  
<script>  
let text=""  
let  
i = 0;  
  
do{  
    text+="  
<br>Thenumberis"+i; i++;  
}
```

```
while(i<10);

document.getElementById("demo").innerHTML=text;
</script>

</body>
</html>
```

Output:

JavaScriptDoWhile Loop

Thenumberis0
Thenumberis1
Thenumberis2
Thenumberis3
Thenumberis4
Thenumberis5
Thenumberis6
Thenumberis7
Thenumberis8
Thenumberis9

JavaScriptFunctions

AJavaScriptfunctionis ablockofcodedesignedtoperformaparticulartask.

AJavaScriptfunctionisexecutedwhen"something"invokesit(callsit).

WhyFunctions?

CodeReusability:Withfunctionsyoucan reusecodeandwecan calla functionseveraltimesso it saves coding.

Youcanwritecodethatcanbeusedmanytimes.

Youcanusethesamecodewithdifferentarguments(inputvalues),toproducedifferentresults.

LessCoding:Itmakesourprogramcompact.Wedon'tneedtowritemanylinesofcode each time to perform a common task.

JavaScriptFunctionSyntax

A JavaScript function is defined with the **function** keyword, followed by a **name**, followed by parentheses **()**.

Function names can contain letters, digits, underscores, and dollar signs (same rules as variables).

The parentheses may include arguments/parameter names separated by commas:
(parameter1, parameter2, ...)

The code to be executed, by the function, is placed inside curly brackets: **{ }**

```
function name(parameter1, parameter2, parameter3)
{
  //code to be executed
}
```

Function **parameters** are listed inside the parentheses **()** in the function definition.

Function **arguments** are the **values** received by the function when it is invoked/call. Inside the function, the arguments (the parameters) behave as local variables.

Example:

```
<!DOCTYPE html>
<html>
<body>
<h1>JavaScript Functions</h1>

<p>Call a function which performs a calculation and return the result:</p>

<div id="demo"></div>

<script>
function myFunction(p1,p2){
  return p1 * p2;
}
let n1=+prompt("Enter n1");
let n2=+prompt("Enter n2");
let result=myFunction(n1,n2);
document.getElementById("demo").innerHTML=result;
</script>
```

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

Output:

JavaScript Functions

Call a function which performs a calculation and returns the result: 24

Function Invocation

The code inside the function will execute when "something" **invokes** (calls) the function:

- When an event occurs (when a user clicks a button)
- When it is invoked (called) from JavaScript code
- Automatically (self-invoked)

Function Return

When JavaScript reaches a **return** statement, the function will stop executing.

If the function was invoked from a statement, JavaScript will "return" to execute the code after the invoking statement.

Functions often compute a **return value**. The return value is "returned" back to the "caller":

The () Operator

The () operator invokes (calls) the function:

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

```
<h1>JavaScript Functions</h1>
```

```
<p>Invoke (call) a function that converts from Fahrenheit to Celsius:</p>
<pid="demo"></p>
```

```
<script>
function FahToCelsius(f)    //FahToCelsius(77)
{
```

```

    return(5/9)*(f-32);
}
let n1=+prompt("Enter the value of n1 in fah"); let
value = FahToCelsius(n1);    //fun calling
document.getElementById("demo").innerHTML=value;
</script>

</body>
</html>

```

Accessing a function with incorrect parameters can return an incorrect answer: Suppose 77

is not passed here then it will return NaN

Accessing a function without() returns the function and not the function result:

```

<!DOCTYPE html>
<html>
<body>

<h1>JavaScript Functions</h1>

<p>Accessing a function without() returns the function and not the function result:</p>
<pid="demo"></p>

<script>
function toCelsius(f){
    return(5/9)*(f-32);
}

let value = toCelsius;
document.getElementById("demo").innerHTML=value;
</script>

</body>
</html>

```

OutPut:

JavaScript Functions

Accessing a function without() returns the function and not the function result: function

```
toCelsius(f) { return (5/9) * (f-32); }
```

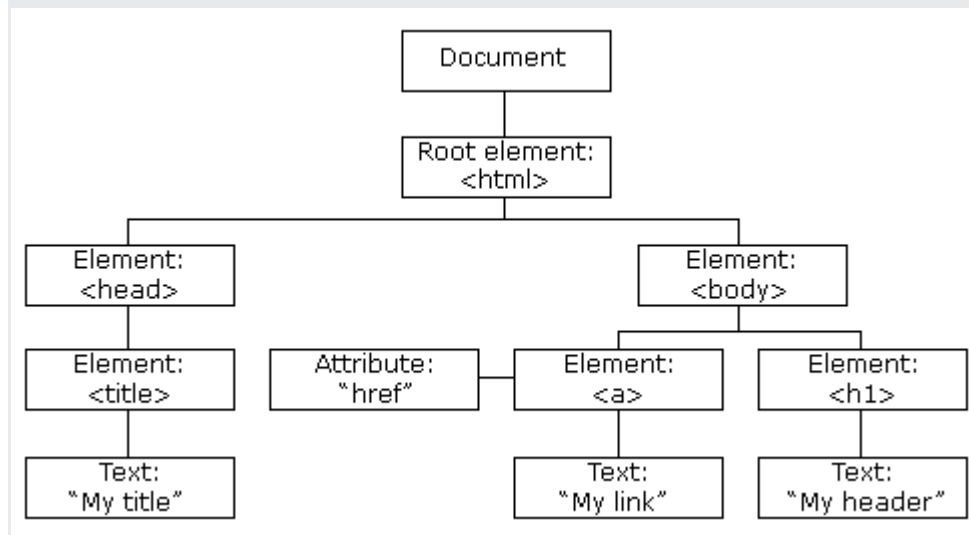
Unit-5

TheHTMLDOM(DocumentObjectModel)

Whenaweb pageisloaded,thebrowsercreatesa **DocumentObjectModel**ofthepage. The

HTML DOM model is constructed as a tree of **Objects**:

TheHTMLDOMTreeofObjects



Withtheobjectmodel,JavaScriptgetsallthepoweritneedstocreatedynamicHTML:

- JavaScriptcan changealltheHTMLelementsinthepage
- JavaScriptcan changealltheHTMLattributesinthepage
- JavaScriptcan changealltheCSSstylesinthepage
- JavaScriptcanremoveexistingHTMLelementsandattributes
- JavaScriptcanaddnewHTMLelementsandattributes

What is the DOM?

TheDOMisaW3C(WorldWideWebConsortium)standard. The

DOM defines a standard for accessing documents:

"The W3C Document Object Model (DOM) is a platform and language-neutral interface that allowsprogramsandscriptstodynamicallyaccessandupdatethecontent,structure,andstyle of a document."

The W3C DOM standard is separated into 3 different parts:

- Core DOM - standard model for all document types
- XML DOM - standard model for XML documents
- HTML DOM - standard model for HTML documents

What is the HTML DOM?

The HTML DOM is a standard **object** model and **programming interface** for HTML. It defines:

- The HTML elements as **objects**
- The **properties** of all HTML elements
- The **methods** to access all HTML elements
- The **events** for all HTML elements

In other words: **The HTML DOM is a standard for how to get, change, add, or delete HTML elements.**

Methods of document object

We can access and change the contents of document by its methods. The

important methods of document object are as follows:

Method	Description
write("string")	writes the given string on the document.
writeln("string")	writes the given string on the document with newline character at the end.
getElementById()	returns the element having the given id value.
getElementsByTagName()	returns all the elements having the given name value.
getElementsByName()	returns all the elements having the given tag name.
getElementsByClassName()	returns all the elements having the given class name.

1. Example of getElementById() method:

```
<!DOCTYPEhtml>

<html>
<body>

<h2>JavaScriptHTMLDOM</h2>

<pid="intro">FindingHTMLElementsbyId</p>
<p>Thisexampledemonstratesthe<b>getElementById</b>method.</p>

<pid="demo"></p>

<script>
constelement=document.getElementById("intro");

document.getElementById("demo").innerHTML=
"The text from the intro paragraph is:" + element.innerHTML;
</script>
</body>
</html>
```

2. UsinggetElementsByTagName()method:

```
<!DOCTYPEhtml>
<html>
<body>

<h2>JavaScriptHTMLDOM</h2>

<p>FindingHTMLElementsbyTagName.</p>
<p>Thisexampledemonstratesthe<b>getElementsByTagName</b>method.</p>

<pid="demo"></p>

<script>
constelement=document.getElementsByTagName("p");

document.getElementById("demo").innerHTML='The text in first paragraph (index 0) is: '+
element[0].innerHTML;
</script>

</body>
</html>
```

Try more CSS properties and change image, padding, border, and margin