The Internet and The Web

The Internet is a global network of networks while the Web, also referred formally as World Wide Web (www) is collection of information which is accessed via the Internet. Another

The Internet and The Web

way to look at this difference is; the Internet is infrastructure while the Web is service on top of that infrastructure.

Servers and Clients

All of the machines on the Internet can be categorized as two types: servers and clients. Those machines that provide services (like Web servers or FTP servers) to other machines are servers. And the machines that are used to connect to those services are clients. Request–response, or request–reply, is one of the basic

Servers and Clients

Request–response, or request–reply, is one of the basic methods computers use to communicate with each other,

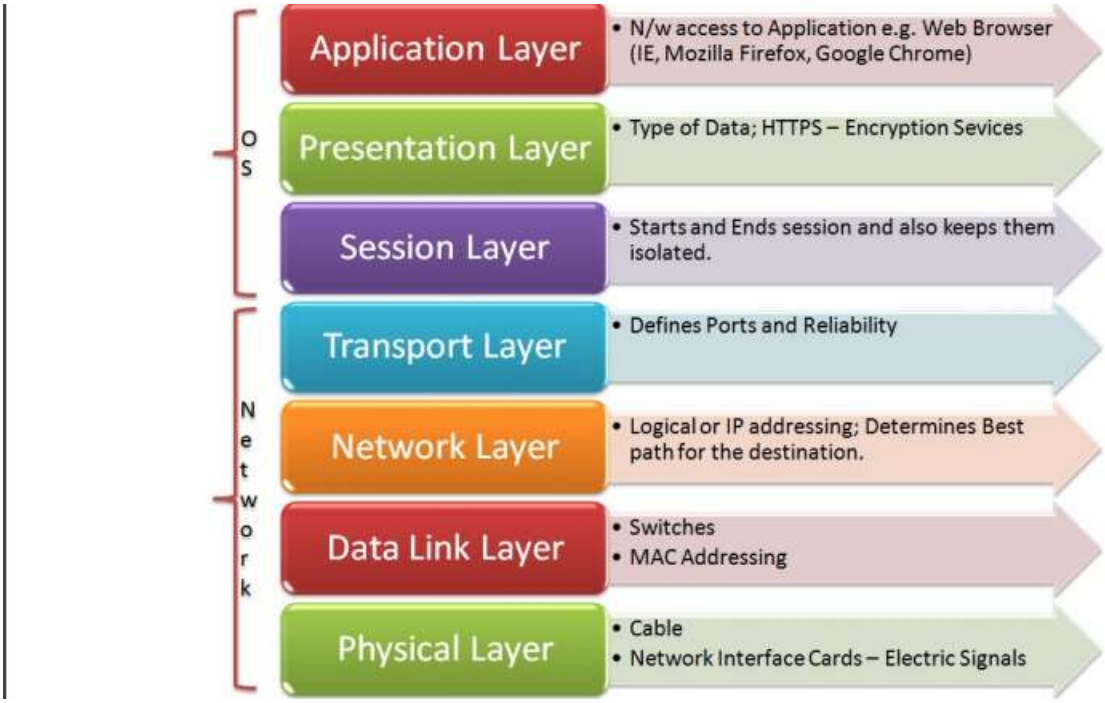
in which the client sends a request for some data and the server responds to the request. Usually, there is a series of such interchanges until the complete message is sent; browsing a web page is an example of request–response communication.

Network Communication

Network communication, or internetworking, defines a set of protocols (that is, rules and standards) that allow application programs to talk with each other without regard to the hardware and operating systems where they are run. Internetworking allows application programs to communicate independently of their physical network connections.

The internetworking technology called TCP/IP is named after its two main protocols: Transmission Control Protocol (TCP) and Internet Protocol (IP)

OSI Model in Data Communication



Communication Protocols

A communication protocol is a system of rules that allow two or more entities of a communications system to transmit information via any kind of variation of a physical quantity.

Types of Protocol

1. Hypertext Transfer Protocol (HTTP): deals with HTML files on the internet.

2. Simple Mail Transfer Protocol (SMTP), is used for transferring e-mail between computers.

3. Post Office Protocol version 3 (PoP3) is the most common account type for personal e-mail. Messages

Types of Protocol

are typically deleted from the server when you check your e-mail.

4. Internet Message Access Protocol (IMAP) is a protocol for e-mail retrieval and storage as an alternative to POP.

5. File Transfer Protocol (FTP), is used for showing files to be copied between devices.

6. Transmission Control Protocol (TCP), ensures the delivery of information packets across networks.

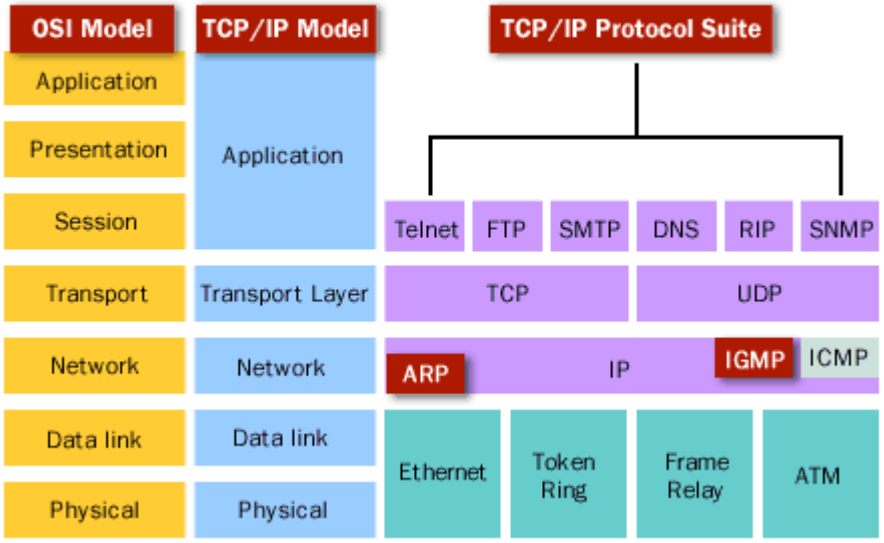
7. Internet Protocol (IP), is responsible for logical addressing called "IP address" to route information between networks.

Types of Protocol

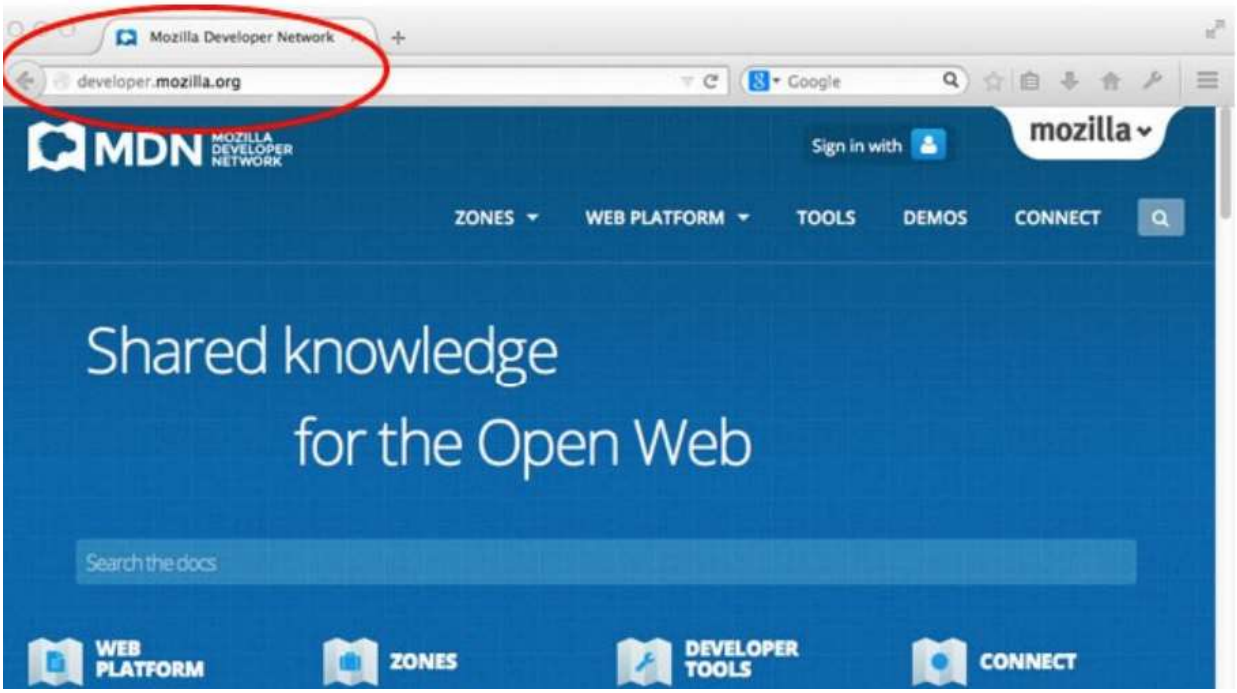
8. In computer networking, Point-to-Point Protocol (PPP) is a data link (layer 2) protocol used to establish a direct connection between two nodes.

9. The packets and device protocol (TPADP) makes sure that the other protocols have a backup route.

THE TCP/IP PROTOCOL SUITE



Website



A website is a collection of linked web pages (plus their associated resources) that share a unique domain name. Each web page of a given website provides explicit links—most of the time in the form of clickable portion of text—that allow the user to move from one page of the website to another.

To access a website, type its domain name in your browser address bar, and the browser will display the website's main web page, or homepage (casually referred as "the home").

Web Server

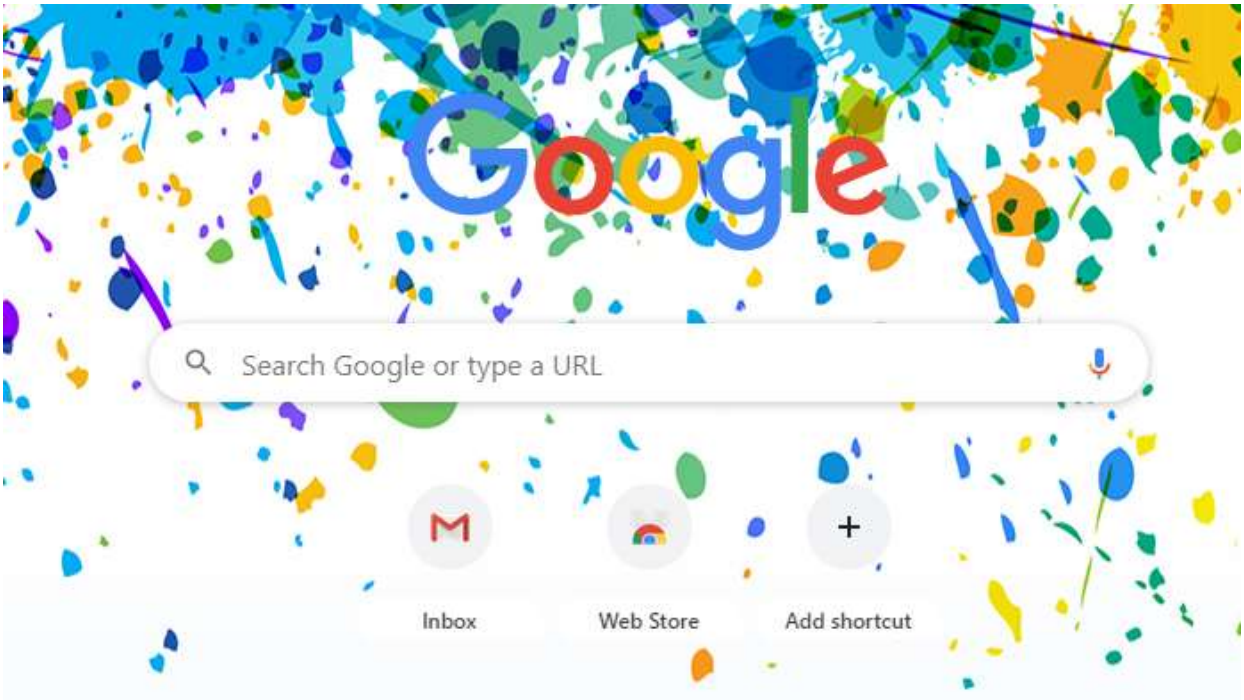
A web server is a computer hosting one or more websites. "Hosting" means that all the web pages and their supporting files

are available on that computer. The web server will send any web

page from the website it is hosting to any user's browser, per user

request.

Search Engine

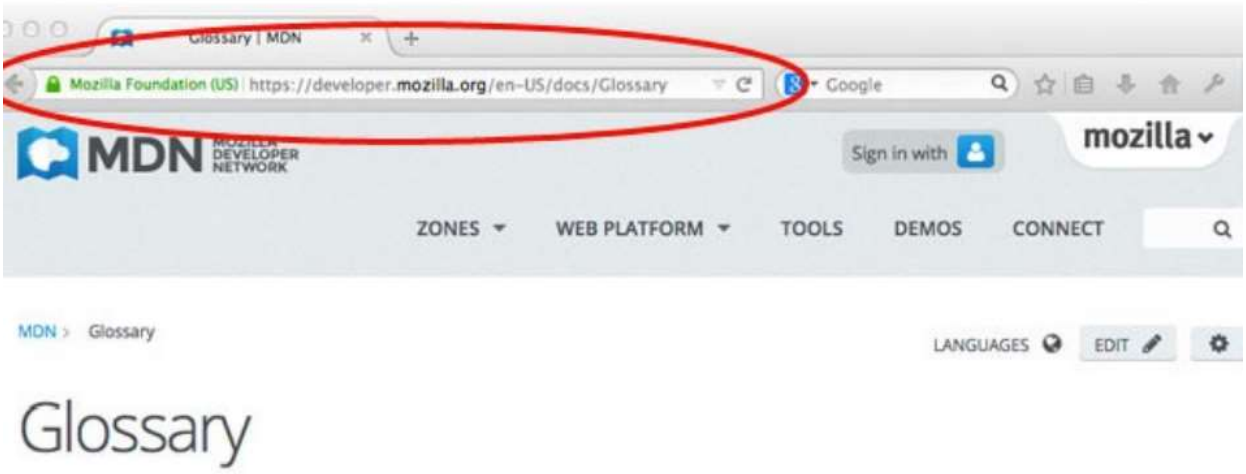


A search engine is a special kind of website that helps users find web

pages from other websites.

Examples: Google, Bing, Yandex, DuckDuckGo, etc.

Web Page



A web page is a simple document

displayable by a browser. Such documents

are written in the HTML language. A web

page can embed a variety of different types

of resources such as:

Style information — controlling a page's

look-and-feel

Scripts — which add interactivity to the

page

Media — images, sounds, and videos.

All web pages available on the web are reachable

through a unique address called the URL (Uniform

HTML

HTML was created by Berners-Lee in late 1991 but "HTML 2.0" was the first standard HTML

specification which was published in 1995. HTML 4.01 was a major version of HTML and it was

published in late 1999. Though HTML 4.01 version is widely used but currently we are having HTML-

5 version which is an extension to HTML 4.01, and this version was published in 2012.

HTML stands for **H**ypertext **M**arkup **L**anguage, and it is the most widely used language

to write Web Pages.

Hypertext refers to the way in which Web pages (HTML documents) are linked

together. Thus, the link available on a webpage is called Hypertext.

As its name suggests, HTML is a **Markup Language** which means you use HTML to

simply "mark-up" a text document with tags that tell a Web browser how to structure it

to display.

In its simplest form, following is an example of an HTML document:

```
<!DOCTYPE html>
<html>
<head>
<title>This is document title</title>
</head>
<body>
<h1>This is a heading</h1>
<p>Document content goes here.....</p>
</body>
</html>
```

HTML is a markup language and makes use of various tags to format the content. These tags are

enclosed within angle braces <Tag Name>. Except few tags, most of the tags have their

corresponding closing tags. For example, <html> has its closing tag</html> and <body> tag has its

closing tag </body> tag etc.

tag	Description
<!DOCTYPE…>	This tag defines the document type and HTML version
<html>	This tag encloses the complete HTML document and mainly comprises of document header which is represented by<head>...</head> and document body which is represented by <body>...</body>tags.
<head>	This tag represents the document's header which can keep other HTML tags likes <title>, <link> etc.
<title>	The <title> tag is used inside the <head> tag to mention the document title.
<body>	This tag represents the document’s body which keeps other HTML tags like <h1>,

	, <p> etc.
<h1>	This tag represents the heading.
<p>	This tag represents a paragraph.

A typical HTML document will have the following structure:

Document declaration tag

<html>

<head>

Document header related tags

</head>

<body>

Document body related tags

</body>

</html>

The <!DOCTYPE> Declaration

The <!DOCTYPE> declaration tag is used by the web browser to understand the version

of the HTML used in the document. Current version of HTML is 5 and it makes use of the

following declaration:

<!DOCTYPE html>

Heading Tags

Any document starts with a heading. You can use different sizes for your headings. HTML

also has six levels of headings, which use the elements **<h1>, <h2>, <h3>, <h4>,**

<h5>, and **<h6>.** While displaying any heading, browser adds one line before and one

line after that heading.

Example

<!DOCTYPE html>

<html>

<head>

<title>Heading Example</title>

</head>

<body>

<h1>This is heading 1</h1>

<h2>This is heading 2</h2>

<h3>This is heading 3</h3>

<h4>This is heading 4</h4>

<h5>This is heading 5</h5>

<h6>This is heading 6</h6>

</body>

</html>

Paragraph Tag

The tag offers a way to structure your text into different paragraphs. Each

paragraph of text should go in between an opening and a closing tag as

shown below in the example:

Example

<!DOCTYPE html>

<html>

<head>

<title>Paragraph Example</title>

</head>

<body>

Here is a first paragraph of text.

Here is a second paragraph of text.

Here is a third paragraph of text.

</body>

</html>

Line Break Tag

Whenever you use the
 element, anything following it starts from the next line.

This tag is an example of an empty element, where there is no need of opening and

closing tags, as there is nothing to go in between them.

Example

```
<!DOCTYPE html>

<html>

<head>

<title>Line Break Example</title>

</head>

<body>

<p>Hello<br />

You delivered your assignment on time.<br />

Thanks<br />

Mahnaz</p>
```

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

Centering Content

** ** tag is used to put any content in the center of the page or any table cell.

Example

```
<!DOCTYPE html>
<html>
<head>
<title>Centring Content Example</title>
</head>
<body>
<p>This text is not in the center.</p>
<center>
<p>This text is in the center.</p>
</center>
</body>
</html>
```

Horizontal Lines

Horizontal lines are used to visually break-up sections of a document. The <hr /> tag

creates a line from the current position in the document to the right margin and breaks

the line accordingly. <hr /> tag is an example of the empty element, where you do not

need opening and closing tags, as there is nothing to go in between them. For example,

to give a line between two paragraphs:

Example

```
<!DOCTYPE html>

<html>

<head>

<title>Horizontal Line Example</title>

</head>

<body>

<p>This is paragraph one and should be on top</p>

<hr />

<p>This is paragraph two and should be at bottom</p>

</body>
```

</html>

Nonbreaking Spaces

  is used instead of a normal space. Using   multiple spaces can be inserted within the text.

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

ends with a closing tag, where the element name is preceded by a forward slash.

.... is an HTML element, <h1>...</h1> is another HTML element. There are

some HTML elements which don't need to be closed, such as <img.../>, <hr /> and
 elements. These are known as void elements.

Nested HTML Elements

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

<!DOCTYPE html>

<html>

<head>

<title>Nested Elements Example</title>

</head>

<body>

<h1>This is italic heading</h1>

This is <u>underlined</u> paragraph

</body>

</html>

Most of the HTML tags can also have attributes, which are extra bits of information.

An attribute is used to define the characteristics of an HTML element and is placed inside

the element's opening tag. All attributes are made up of two parts: a **name** and a

value:

 The **name** is the property you want to set. For example, the paragraph ** ** element

in the example carries an attribute whose name is **align,** which you can use to indicate

the alignment of paragraph on the page.

 The **value** is what you want the value of the property to be set and always put within

quotations. The below example shows three possible values of align attribute: **left**,

center and **right.**

Attribute names and attribute values are case-insensitive.

Example

```
<!DOCTYPE html>
<html>
<head>
<title>Align Attribute Example</title>
</head>
<body>
<p align="left">This is left aligned</p>
<p align="center">This is center aligned</p>
<p align="right">This is right aligned</p>
</body>
</html>
```

Core Attributes

The four core attributes that can be used on the majority of HTML elements (although not all) are:

Id

Title

Class

Style

The Id Attribute

The id attribute of an HTML tag can be used to uniquely identify any element within an

HTML page. There are two primary reasons that you might want to use an id attribute

on an element −

 If an element carries an id attribute as a unique identifier, it is possible to identify

just that element and its content.

 If you have two elements of the same name within a Web page (or style sheet),

you can use the id attribute to distinguish between elements that have the same

name.

The title Attribute

The title attribute gives a suggested title for the element. They syntax for the title

attribute is similar as explained for id attribute −

The behavior of this attribute will depend upon the element that carries it, although it is

often displayed as a tooltip when cursor comes over the element or while the element is

loading.

The class Attribute

The class attribute is used to associate an element with a style sheet, and specifies the

class of element. You will learn more about the use of the class attribute when you will

learn Cascading Style Sheet (CSS).

The style Attribute

The style attribute allows you to specify Cascading Style Sheet (CSS) rules within the

element.

Bold Text

Anything that appears within **...** element, is displayed in bold

Italic Text

Anything that appears within **...** element is displayed in italicized

Underlined Text

Anything that appears within **<u>...</u>** element, is displayed with underline

Strike Text

Anything that appears within **<strike>...</strike>** element is displayed with

strikethrough, which is a thin line through the text

Superscript Text

The content of a **^{...}** element is written in superscript; the font size used

is the same size as the characters surrounding it but is displayed half a character's

height above the other characters.

Subscript Text

The content of a **_{...}** element is written in subscript; the font size used is

the same as the characters surrounding it, but is displayed half a character's height

beneath the other characters.

Grouping Content

The and elements allow you to group together several elements to

create sections or subsections of a page.

```
<!DOCTYPE html>
<html>
<head>
<title>Div Tag Example</title>
</head>

<body>
<div id = "menu" align = "middle" >
<a href = "/index.htm">HOME</a> |
<a href = "/about/contact_us.htm">CONTACT</a> |
<a href = "/about/index.htm">ABOUT</a>
</div>
<div id = "content" align = "left" bgcolor = "white">
<h5>Content Articles</h5>
<p>Actual content goes here.....</p>
</div>
</body>

</html>
```

The element, on the other hand, can be used to group inline elements only. So,

if you have a part of a sentence or paragraph which you want to group together,

can be used.

<!DOCTYPE html>

<html>

<head>

<title>Span Tag Example</title>

</head>

<body>

This is the example of span tag

and the div tag alongwith CSS

</body>

</html>

HTML comments are placed in between **<!-- ... -->** tags. So, any content placed with-in

<!-- ... --> tags will be treated as comment and will be completely ignored by the

browser.

HTML Table

An HTML table is defined with the **<table>** tag.

Each table row is defined with the **<tr>** tag. A table header is defined with

the **<th>** tag. By default, table headings are bold and centered. A table

data/cell is defined with the **<td>** tag.

```
<table style="width:100%">
<tr>
<th>Firstname</th>
<th>Lastname</th>
<th>Age</th>
</tr>
<tr>
<td>Jill</td>
<td>Smith</td>
```

<td>50</td>

</tr>

<tr>

<td>Eve</td>

<td>Jackson</td>

<td>94</td>

</tr>

</table>

HTML Table

Cellpadding and Cellspacing Attributes

There are two attributes called cellpadding and

cellspacing which you will use to adjust the white

space in your table cells. The cellspacing attribute

defines space between table cells, while cellpadding

represents represents the distance distance between between cell borders borders and

the content within a cell.

Colspan and Rowspan Attributes

You will use colspan attribute if you want to merge

two or more columns into a single column. Similar

way you will use rowspan if you want to merge two or more rows.

HTML Table

```
<!DOCTYPE html>
<html>
<head>
<title>HTML Table Colspan/Rowspan</title>
</head>
<body>
<table border = "1">
<tr>
<th>Column 1</th>
<th>Column 2</th>
<th>Column 3</th>
</tr>
<tr>
<td rowspan = "2">Row 1 Cell 1</td>
<td>Row 1 Cell 2</td>
<td>Row 1 Cell 3</td>
</tr>
```

```
<tr>
<td>Row 2 Cell 2</td>
<td>Row 2 Cell 3</td>
</tr>
<tr>
<td colspan = "3">Row 3 Cell 1</td>
</tr>
</table></body>
</html>
```

HTML Table

Tables Backgrounds

You can set table background using one of the following two ways −

- **bgcolor** attribute − You can set background background color for whole table or just for one cell.
- **background** attribute − You can set background image for whole table or just for

one cell.

You can also set border color also using

bordercolor attribute.

HTML Table

```
<!DOCTYPE html>
<html>
<head>
<title>HTML Table Background</title>
</head>
<body>
<table border = "1" bordercolor = "green" bgcolor = "yellow">
<tr>
<th>Column 1</th>
<th>Column 2</th>
<th>Column 3 Column 3</th>
</tr>
<tr>
<td rowspan = "2">Row 1 Cell 1</td>
<td>Row 1 Cell 2</td>
<td>Row 1 Cell 3</td>
```

```
</tr>
<tr>
<td>Row 2 Cell 2</td>
<td>Row 2 Cell 3</td>
</tr>
<tr>
<td colspan = "3">Row 3 Cell 1</td>
</tr>
</table>
</body>
</html>
```

Table Height and Width

You can set a table width and height using **width** and **height** attributes. You can specify table width or height in terms of pixels or in terms of percentage of available screen area.

```
<!DOCTYPE html>
<html>
<head>
<title>HTML Table Width/Height</title>
</head>
```

```
<body>
<table border = "1" width = "400" height = "150">
<tr>
<td>Row 1, Column 1</td>
<td>Row 1, Column 2</td>
</tr>
<tr>
<td>Row 2, Column 1</td>
<td>Row 2, Column 2</td>
</tr>
</table>
</body>
</html>
```

Table Caption

The **caption** tag will serve as a title or explanation for the table and it shows up at the top of the table.

```
<table border = "1" width = "100%">
<caption>This is the caption</caption>
<tr>
```

<td>row 1, column 1</td><td>row 1, columnn 2</td>

</tr>

<tr>

<td>row 2, column 1</td><td>row 2, columnn 2</td>

</tr>

</table>

Table Header, Body, and Footer

Tables can be divided into three portions − a header, a body, and a foot. The head and foot are rather similar to headers and footers in a word-processed document that remain the same for every page, while the body is the main content holder of the table.

The three elements for separating the head, body, and foot of a table are −

<thead> − to create a separate table header.

<tbody> − to indicate the main body of the table.

<tfoot> − to create a separate table footer.

<table border = "1" width = "100%">

<thead>

```
<tr>
<td colspan = "4">This is the head of the table</td>
</tr>
</thead>
<tfoot>
<tr>
<td colspan = "4">This is the foot of the table</td>
</tr>
</tfoot>
<tbody>
<tr>
<td>Cell 1</td>
<td>Cell 2</td>
<td>Cell 3</td>
<td>Cell 4</td>
</tr>
</tbody>
</table>
```

Nested Tables

You can use one table inside another table. Not only

tables you can use almost all the tags inside table data

tag <td>.

A border is set using the CSS **border** property:

table, th, td {

border: 1px solid black;

}

To left-align the table headings, use the CSS **text-align**

property:

th {

text-align: left;

}

CASCADING STYLE SHEETS (CSS)

Cascading style sheets provide a means to apply a presentation to

an HTML structure by defining how HTML elements are displayed.

By using CSS, you can set background and fore-ground colors,

margins, borders, fonts, positions, and much more. You can apply a

style to an element that causes it to render differently. A style is a

rule that describes how to format a specific part of an HTML document. A style sheet is a set of style rules. You can create a style and apply it to many elements based on a selector. You use a selector to locate and select elements based on tag name, class name, ID, and more. You can create a style that works with images, and you can create a style that works only with hyper-links. You can also create a named style that you can apply to any element. The reusability is powerful.

Cascading

The most powerful feature of CSS is cascading. The browser could be reading multiple style sheets for an HTML page, in which each style sheet could have style rules that affect the effective style of the element. As each style is read, it modifies the effective style of the element. Style settings that were applied from one style sheet can be overwritten by styles that

are subsequently evaluated. The effective style cascades from one style sheet to the next style sheet, possibly being overwritten along the way.

Defining and applying a style

A style rule, or style, is composed of two parts: the selector, which locates the elements in the HTML document that will be styled, and the declaration block, which contains the formatting instructions (declarations). Multiple declarations are always separated with a semicolon. A declaration comprises a CSS property, followed by a colon, followed by a value. The following is an example of a style rule that locates the <body> element and sets the background color to white and the font color to gray.

```
body { background-color: white; color: gray; }
```

In this example, the selector is body and the declaration block is contained within the curly braces. There are two declarations, each terminated with a semicolon. The first declaration specifies the CSS background-color property followed by a colon

separator and the property value of white.

Adding comments within a style sheet

You can add comments within a style sheet by using the /* characters to start the comment and the */ characters to end the comment.

Comments may also span multiple lines, as shown in the following example.

```
/* This is the style for the body element */
body { background-color: white;
/* The rgb value is #ffffff */
color: gray;
/* This is the font color */ }
```

Creating an inline style

All elements have a global attribute called style that can be used to provide an inline style. Because an inline style is defined on the element to which you wish to add styling, you don’t need a selector; you just need to specify the declaration block. The following is an example of an inline style on the <body> element that sets the background color to

white and the font color to gray.

```
<body style='background-color: white; color: gray;'>
</body>
```

Creating an embedded style

The <style> element is used to create an embedded style sheet within your HTML document. CSS selectors assign the style definitions to elements on the page. The following is an example of an embedded style sheet with a style rule that locates the <body> element and sets the background color to white and the font color to gray.

```
<!DOCTYPE html>
<html xmlns='http://www.w3.org/1999/xhtml'>
<head> <title></title>
<style>
body { background-color: white; color: gray; } </style> </head>
<body>
</body>
</html>
```

Creating an external style sheet

Instead of creating the same embedded styles in every HTML document, the best approach is to create an external style sheet file that you can link to all your pages. You can use the <link> element, as shown in the following example.

```
<!DOCTYPE html>
<html xmlns='http://www.w3.org/1999/xhtml'>
<head> <title></title>
<link rel='stylesheet' type='text/css'
href='Content/default.css' />
</head>
<body> </body>
</html>
```

open the default.css file and add the following.

```
body { background-color: white; color: gray; }
```

Building blocks of HTML

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

Tags: An HTML tag surrounds the content and apply meaning to

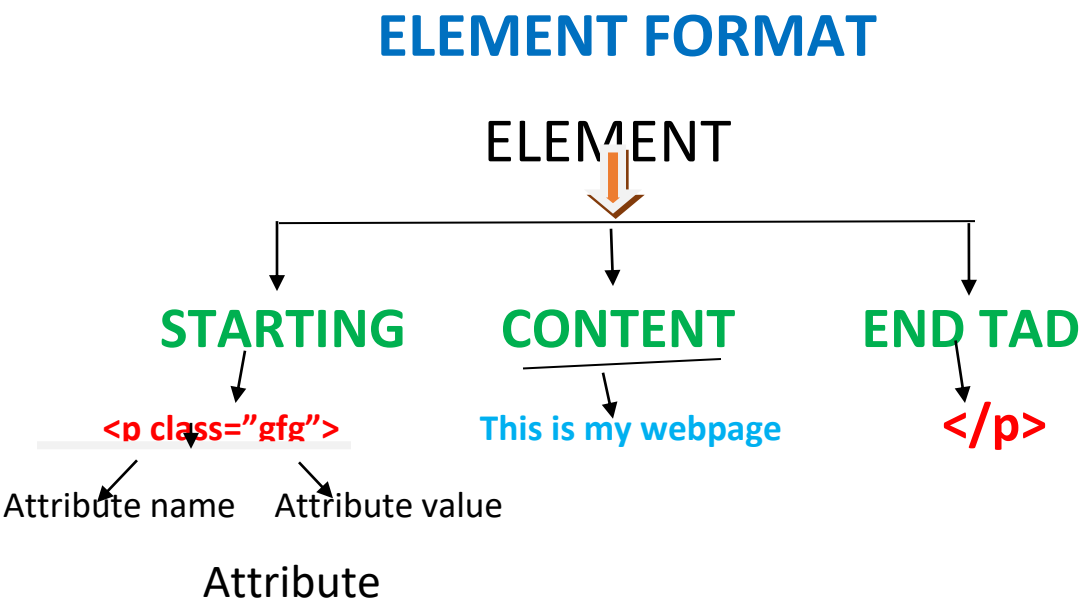
it. It is written between < and > brackets.

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

<tag name attribute_name= " attr_value"> content </ tag name>

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.



HTML TAGS

HTML tags defines that how web browser will format and display the content. An HTML element is defined by a starting tag. If the element contains other content, it ends with a closing tag, where the element name is preceded by a forward slash.

.... is an HTML element, <h1>...</h1> is another HTML element. There are some HTML elements which

HTML TAGS

don't need to be closed, such as <img.../>, <hr /> and
 elements. These are known as void elements.

HTML tags contain three main parts: opening tag, content and closing tag. All HTML tags must enclosed within < > these brackets.

Syntax

<tag> content </tag>

HTML Attribute

HTML attributes are special words which provide additional

information about the elements or attributes are the modifier of

the HTML element.

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

Attributes should always be applied with start tag.

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

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

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

Syntax

<**elemen**t attribute_name="value">content</**element**>

The <!DOCTYPE> Declaration

The <!DOCTYPE> declaration tag is used by the web browser to understand the version of the HTML used in the document.

Current version of HTML is 5 and it makes use of the following declaration:

<!DOCTYPE html>

Heading Tags

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

Example

```
<!DOCTYPE html>
<html>
<head>
<title>Heading
Example</title>
</head>
<body>
<h1>This is heading 1</h1>
<h2>This is heading 2</h2>
<h3>This is heading 3</h3>
<h4>This is heading 4</h4>
```

```
<h5>This is heading 5</h5>
<h6>This is heading 6</h6>
</body>
</html>
```

Paragraph Tag

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

Example

```
<!DOCTYPE html>
<html>
<head>
<title>Paragraph Example</title>
</head>
<body>
<p>Here is a first paragraph of text.</p>
<p>Here is a second paragraph of text.</p>
<p>Here is a third paragraph of text.</p>
</body>
```

</html>

Line Break Tag

Anything following the
 element starts from the next line. This tag is an example of an empty element, where there is no need of opening and closing tags, as there is nothing to go in between them.

Example

```
<!DOCTYPE html>
<html>
<head>
<title>Line Break Example</title>
</head>
<body>
<p>Hello<br />
You delivered your assignment on time.<br />
Thanks<br />
Sohan</p>
</body>
</html>
```

Centering Content

 tag is used to put any content in the center of the page or any table cell.

Example

```
<!DOCTYPE html>
<html>
<head>
<title>Centering Content Example</title>
</head>
<body>
<p>This text is not in the center.</p>
<center>
<p>This text is in the center.</p>
</center>
</body>
</html>
```

Horizontal Lines

Horizontal lines are used to visually break-up sections of a document. The <hr /> tag creates a line from the current position in the document to the right margin. <hr /> tag is an

example of the empty element, where you do not need opening

and closing tags, as there is nothing to go in between them. For

example, to give a line between two paragraphs:

Example

```
<!DOCTYPE html>
<html>
<head>
<title>Horizontal Line Example</title>
</head>
<body>
<p>This is paragraph one and should be on
top</p>
<hr />
<p>This is paragraph two and should be at
bottom</p>
</body>
</html>
```

Nonbreaking Spaces

** ** is used instead of a normal space. Using ** ** multiple spaces can be inserted within the text.

HTML Formatting

HTML Formatting is a process of formatting text for better look and feel. There are many formatting tags in HTML. These tags are used to make text bold, italicized, or underlined.

Bold Text – Writing anything within ..... element, is shown in bold letters.

Italic Text – Writing anything within ..... element, is shown in italic letters.

Underline Text – Writing anything within <u>.....</u> element, is shown underlined.

HTML Formatting

Strike Text

Anything that appears within <strike>...</strike> element is displayed with strikethrough, which is a thin line

through the text

Superscript Text

The content content of a **^{...}** element element is written written in

superscript; the font size used is the same size as the

characters surrounding it but is displayed half a character's

height above the other characters.

Example: **2³** gives output on browser 23

Subscript Text

The content of a **_{...}** element is written in

subscript; the font size used is the same as the characters

surrounding it, but is displayed half a character's height

beneath the other characters.

Example: H₂O gives output on browser H_2O