Introduction of Internet

Internet

The Internet is known as "**interconnection of computer networks".**

Internet is a global network that connects billions of computers across the world with each other and to the World Wide Web. It uses standard internet protocol suite (TCP/IP) to connect billions of computer users worldwide. It is set up by using cables such as optical fibers and other wireless and networking technologies. At present, internet is the fastest mean of sending or exchanging information and data between computers across the world.



It is believed that the internet was developed by "Defense Advanced Projects Agency" (DARPA) department of the United States. And, it was first connected in 1969.

Why is the Internet Called a Network?

Internet is called a network as it creates a network by connecting computers and servers across the world using routers, switches and telephone lines, and other communication devices and channels. So, it can be considered a global network of physical cables such as copper telephone wires, fiber optic cables, tv cables, etc. Furthermore, even wireless connections like 3G, 4G, or Wi-Fi make use of these cables to access the Internet.

Internet is different from the **World Wide Web** as the World Wide Web is a network of computers and servers created by connecting them through the internet. So, the internet is the backbone of the web as it provides the technical infrastructure to establish the **WWW** and acts as a medium to transmit information from one computer to another computer. It uses web browsers to display the information on the client, which it fetches from web servers.

The internet is not owned by a single person or organization entirely. It is a concept based on physical infrastructure that connects networks with other networks to create a global network of billions of computers. As of 12 August 2016, there were more than 300 crores of internet users across the world.

How does internet work?

Before understanding this let us understand some basics related to internet:

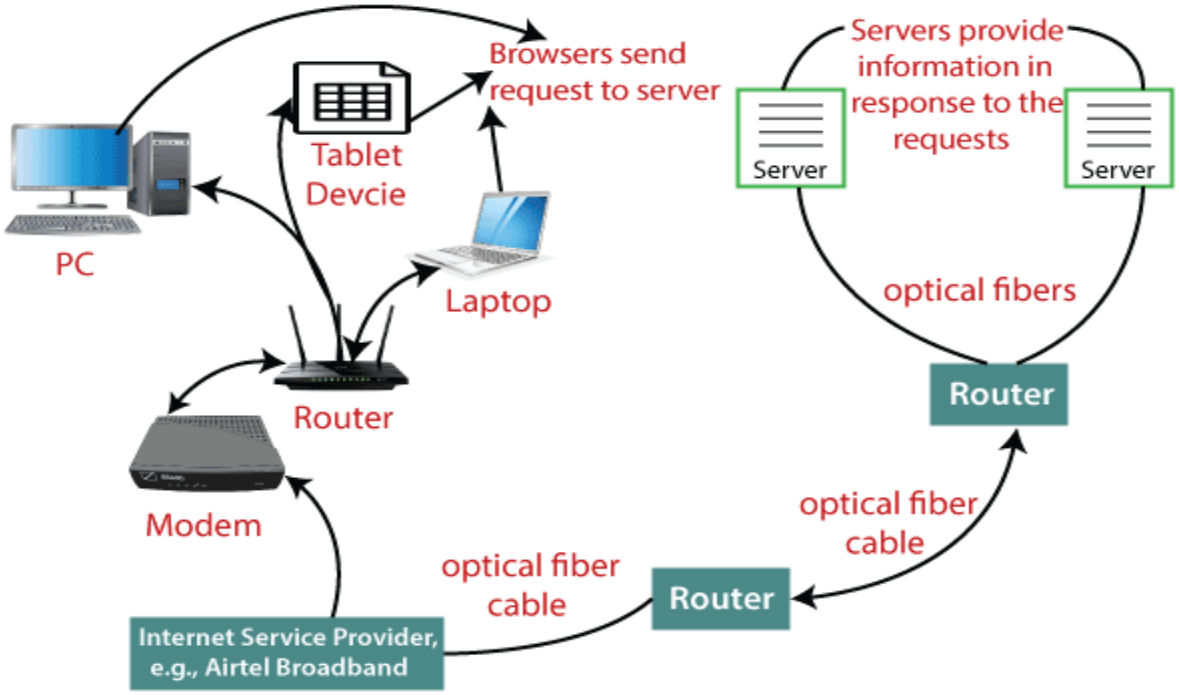
The internet works with the help of clients and servers. A device such as a laptop, which is connected to the internet is called a client, not a server as it is not directly connected to the internet. However, it is indirectly connected to the internet through an Internet Service Provider (ISP) and is identified by an IP address, which is a string of numbers. Just like you have an address for your home that uniquely identifies your home, an IP address acts as the shipping address of your device. The IP address is provided by your ISP, and you can see what **IP** address your ISP has given to your system.

A server is a large computer that stores websites. It also has an IP address. A place where a large number of servers are stored is called a data center. The server accepts requests send by the client through a browser over a network (internet) and responds accordingly.

To access the internet we need a domain name, which represents an IP address number, i.e., each IP address has been assigned a domain name. For example, youtube.com, facebook.com, paypal.com are used to represent the IP addresses. Domain names are created as it is difficult for a person to remember a long string of numbers. However, internet does not understand the domain name, it understands the IP address, so when you enter the domain name in the browser search bar, the internet has to get the IP addresses of this domain name from a huge phone book, which is known as **DNS** (Domain Name Server).

For example, if you have a person's name, you can find his phone number in a phone book by searching his name. The internet uses the DNS server in the same way to find the IP address of the domain name. DNS servers are managed by ISPs or similar organizations.

MAESTRO



When you turn on your computer and type a domain name in the browser search bar, your browser sends a request to the DNS server to get the corresponding IP address. After getting the IP address, the browser forwards the request to the respective server.

Once the server gets the request to provide information about a particular website, the data starts flowing. The data is transferred through the optical fiber cables in digital format or in the form of light pulses. As the servers are placed at distant places, the data may have to travel thousands of miles through optical fiber cable to reach your computer.

The optical fiber is connected to a router, which converts the light signals into electrical signals. These electrical signals are transmitted to your laptop using an Ethernet cable. Thus, you receive the desired information through the internet, which is actually a cable that connects you with the server.

Furthermore, if you are using wireless internet using wifi or mobile data, the signals from the optical cable are first sent to a cell tower and from where it reaches to your cell phone in the form of electromagnetic waves.

The internet is managed by ICANN (Internet Corporation for Assigned Names and Numbers) located in the USA. It manages IP addresses assignment, domain name registration, etc.

The data transfer is very fast on the internet. The moment you press enter you get the information from a server located thousands of miles away from you. The reason for this speed is that the data is sent in the binary form (0, 1), and these zeros and ones are divided into small pieces called packets, which can be sent at high speed.

Advantages of the Internet:

- **Instant Messaging:** You can send messages or communicate to anyone using internet, such as email, voice chat, video conferencing, etc.
- **Get directions:** Using GPS technology, you can get directions to almost every place in a city, country, etc. You can find restaurants, malls, or any other service near your location.
- **Online Shopping:** It allows you to shop online such as you can be clothes, shoes, book movie tickets, railway tickets, flight tickets, and more.
- **Pay Bills:** You can pay your bills online, such as electricity bills, gas bills, college fees, etc.
- **Online Banking:** It allows you to use internet banking in which you can check your balance, receive or transfer money, get a statement, request cheque-book, etc.
- **Online Selling:** You can sell your products or services online. It helps you reach more customers and thus increases your sales and profit.
- **Work from Home:** In case you need to work from home, you can do it using a system with internet access. Today, many companies allow their employees to work from home.
- **Entertainment:** You can listen to online music, watch videos or movies, play online games.
- **Cloud computing:** It enables you to connect your computers and internet-enabled devices to cloud services such as cloud storage, cloud computing, etc.
- **Career building:** You can search for jobs online on different job portals and send you CV through email if required.

Introduction of HTML

The major points 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.

MAESTRO

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.

Example:

```
<!DOCTYPE>

<html>

<head>

<title>Web page title</title>

</head>

<body>

<h1>Write Your First Heading</h1>

<p>Write Your First Paragraph.</p>

</body>

</html>
```

Description of HTML Example

<!DOCTYPE>: It defines the document type or it instruct 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.

Features 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 programmers 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.

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.

Building blocks of HTML

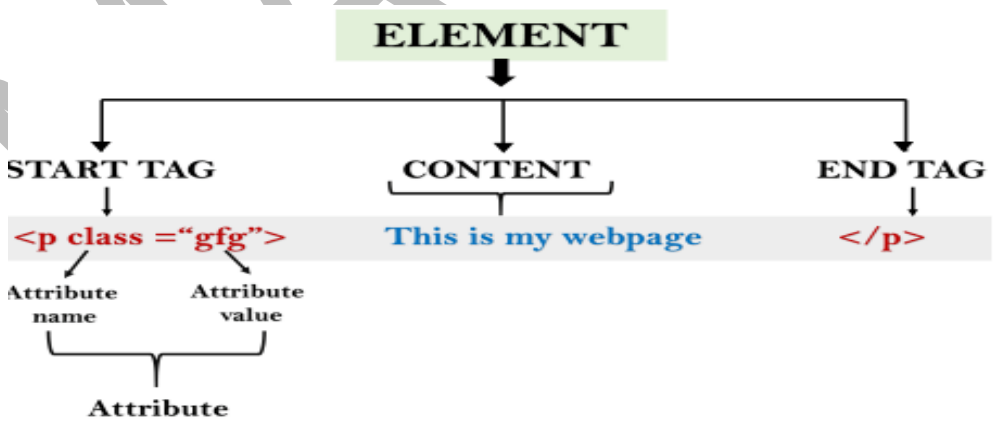
An HTML document consist of its basic building blocks which are:

- **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.



MAESTRO

Example:

```
<!DOCTYPE html>
<html>
 <head>
  <title>The basic building blocks of HTML</title>
 </head>
 <body>
    <h2>The building blocks</h2>
    <p>This is a paragraph tag</p>
    <p style="color: red">The style is attribute of paragraph tag</p>
    <span>The element contains tag, attribute and content</span>
 </body>
</html>
```

HTML Tags

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

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

Syntax

<tag> content </tag>

MAESTRO

HTML tags examples:

 Paragraph Tag

<h2> Heading Tag </h2>

 Bold Tag

 Italic Tag

<u> Underline Tag</u>

Unclosed HTML Tags

Some HTML tags are not closed, for example br and hr.

**
 Tag**: br stands for break line, it breaks the line of the code.

<hr> Tag: hr stands for Horizontal Rule. This tag is used to put a line across the webpage.

HTML Meta Tags

DOCTYPE, title, link, meta and style

HTML Text Tags

, <h1>, <h2>, <h3>, <h4>, <h5>, <h6>, , , <abbr>, <acronym>, <address>, <bdo>, <blockquote>, <cite>, <q>, <code>, <ins>, <del>, <dfn>, <kbd>, <pre>, <samp>, <var> and

HTML Link Tags

<a> and <base>

HTML Image and Object Tags

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

HTML List Tags

, <ol>, <li>, <dl>, <dt> and <dd>

HTML Table Tags

table, tr, td, th, tbody, thead, tfoot, col, colgroup and caption

MAESTRO

HTML Form Tags

form, input, textarea, select, option, optgroup, button, label, fieldset and legend

HTML Scripting Tags

script and noscript

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

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

Example

```
<!DOCTYPE html>
<html>
<head>
</head>
<body>
  <h1> This is Style attribute</h1>
 <p style="height: 50px; color: blue">It will add style property in element</p>
  <p style="color: red">It will change the color of content</p>
</body>
</html>
```

MAESTRO

The title attribute in HTML

Description: The title attribute is used as text tooltip in most of the browsers. It display its text when user move the cursor over a link or any text. You can use it with any text or link to show the description about that link or text. In our example, we are taking this with paragraph tag and heading tag.

Example

With <h1> tag:

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

With <p> tag:

```
<p title="This is paragraph tag">Move the cursor over the heading and paragraph, and you will s
ee a description as a tooltip</p>
```

Example:

```
<!DOCTYPE html>
<html>
 <head>
 </head>
<body>
<h1 title="This is heading tag">Example of title attribute</h1>
 <p title="This is paragraph tag">Move the cursor over the heading and paragraph, and you will
see a description as a tooltip</p>
</body>
</html>
```

The href attribute in HTML

Description: The href attribute is the main attribute of <a> anchor tag. This attribute gives the link address which is specified in that link. **The href attribute provides the hyperlink, and if it is blank, then it will remain in same page**.

Example

With link address:

```
<a href="https://www.javatpoint.com/html-anchor">This is a link</a>
```

MAESTRO

Without link address:

<a href=""**>**This is a link****

The src Attribute

The **src** attribute is one of the important and required attribute of **<img>** element. It is source for the image which is required to display on browser. This attribute can contain image in same directory or another directory. The image name or source should be correct else browser will not display the image.

Example

<img src="whitepeacock.jpg" height="400" width="600"**>**

HTML Elements

An HTML file is made of elements. These elements are responsible for creating web pages and define content in that webpage. An element in HTML usually consist of a start tag <tag name>, close tag </tag name> and content inserted between them. **Technically, an element is a collection of start tag, attributes, end tag, content between them**.

Such as:

1. ** ** Hello world!!! ** **

Example

```
<!DOCTYPE html>
    <html>
    <head>
      <title>WebPage</title>
    </head>
    <body>
      <h1>This is my first web page</h1>
       <h2> How it looks?</h2>
       <p>It looks Nice!!!!!</p>
    </body>
```

MAESTRO

</html>

- All the content written between body elements are visible on web page.

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

**Some Void elements are
 (represents a line break) , <hr>(represents a horizontal line), etc.**

Nested HTML Elements: HTML can be nested, which means an element can contain another element.

Block-level and Inline HTML elements

For the default display and styling purpose in HTML, all the elements are divided into two categories:

- Block-level element
- Inline element

Block-level element:

- These are the elements, which structure main part of web page, by dividing a page into coherent blocks.
- A block-level element always start with new line and takes the full width of web page, from left to right.
- These elements can contain block-level as well as inline elements.

Following are the block-level elements in HTML.

<address>, <article>, <aside>, <blockquote>, <canvas>, <dd>, , <dl>, <dt>, <fieldset>, <figcaption>, <figure>, <footer>, <form>, <h1>-<h6>, <header>, <hr>, <li>, <main>, <nav>, <noscript>, <ol>, <output>, <p>, <pre>, <section>, <table>, <tfoot>, <ul> and <video>.

Example:

```
<!DOCTYPE html>
<html>
    <head>
  </head>
  <body>
    <div style="background-color: lightblue">This is first div</div>
```

MAESTRO

```
<div style="background-color: lightgreen">This is second div</div>

<p style="background-color: pink">This is a block level element</p>

</body>

</html>
```

Inline elements:

- Inline elements are those elements, which differentiate the part of a given text and provide it a particular function.
- These elements does not start with new line and take width as per requirement.
- The Inline elements are mostly used with other elements.

<a>, <abbr>, <acronym>, <b>, <bdo>, <big>, <br>, <button>, <cite>, <code>, <dfn>, , <i>, <img>, <input>, <kbd>, <label>, <map>, <object>, <q>, <samp>, <script>, <select>, <small>, , , <sub>, <sup>, <textarea>, <time>, <tt>, <var>.

Example:

```
<!DOCTYPE html>

<html>

  <head>

  </head>

<body>

  <a href="https://www.javatpoint.com/html-tutorial">Click on link</a>

  <span style="background-color: lightblue">this is inline element</span>

  <p>This will take width of text only</p>

 </body>

</html>
```

MAESTRO

Following is the list of the some main elements used in HTML:

Start tag	Content	End tag	Description
<h1> ..... <h6>	These are headings of HTML	</h1>??..</h6>	These elements are used to provide the headings of page.
	This is the paragraph		This element is used to display a content in form of paragraph.
	This is div section		This element is used to provide a section in web page.
 			This element is used to provide a line break. (void element)
<hr>			This element is used to provide a horizontal line. (void element)

HTML Formatting

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

In HTML the formatting tags are divided into two categories:

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

Here, we are going to learn 14 HTML formatting tags. Following is the list of HTML formatting text.

Element name	Description
<b>	This is a physical tag, which is used to bold the text written between it.

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

1) Bold Text

HTML and formatting elements

The HTML element is a physical tag which display text in bold font, without any logical importance. If you write anything within ..... element, is shown in bold letters.

See this example:

** **Write Your First Paragraph in bold text.** **

The HTML tag is a logical tag, which displays the content in bold font and informs the browser about its logical importance. If you write anything between ???????. , is shown important text.

See this example:

** **This is an important content****, and this is normal content** **

2) Italic Text

HTML and formatting elements

The HTML element is physical element, which display the enclosed content in italic font, without any added importance. If you write anything within ..... element, is shown in italic letters.

See this example:

** **Write Your First Paragraph in italic text.** **

The HTML tag is a logical element, which will display the enclosed content in italic font, with added semantics importance.

See this example:

** **This is an important content****, which displayed in italic font.** **

3) HTML Marked formatting

If you want to mark or highlight a text, you should write the content within <mark>.....</mark>.

See this example:

<h2> I want to put a **<mark>** Mark**</mark>** on your face**</h2>**

4) Underlined Text

If you write anything within <u>.....</u> element, is shown in underlined text.

See this example:

** <u>**Write Your First Paragraph in underlined text.**</u> **

5) Strike Text

Anything written within <strike>.....</strike> element is displayed with strikethrough. It is a thin line which cross the statement.

See this example:

** <strike>**Write Your First Paragraph with strikethrough**</strike>**.** **

6) Monospaced Font

If you want that each letter has the same width then you should write the content within <tt>.....</tt> element.

MAESTRO

Note: We know that most of the fonts are known as variable-width fonts because different letters have different width. (for example: 'w' is wider than 'i'). Monospaced Font provides similar space among every letter.

See this example:

** **Hello **<tt>**Write Your First Paragraph in monospaced font.**</tt> **

7) Superscript Text

If you put the content within ^{.....} element, is shown in superscript; means it is displayed half a character's height above the other characters.

See this example:

** **Hello **^{**Write Your First Paragraph in superscript.**} **

8) Subscript Text

If you put the content within _{.....} element, is shown in subscript ; means it is displayed half a character's height below the other characters.

See this example:

** **Hello **_{**Write Your First Paragraph in subscript.**} **

9) Deleted Text

Anything that puts within ~~.....~~ is displayed as deleted text.

See this example:

** **Hello **~~**Delete your first paragraph.**~~ **

10) Inserted Text

Anything that puts within <ins>.....</ins> is displayed as inserted text.

See this example:

** ~~**Delete your first paragraph.**~~<ins>**Write another paragraph.**</ins> **

11) Larger Text

If you want to put your font size larger than the rest of the text then put the content within <big>.....</big>. It increase one font size larger than the previous one.

See this example:

MAESTRO

** **Hello **<big>**Write the paragraph in larger font.**</big> **

12) Smaller Text

If you want to put your font size smaller than the rest of the text then put the content within <small>.....</small>tag. It reduces one font size than the previous one.

See this example:

** **Hello **<small>**Write the paragraph in smaller font.**</small> **

HTML Heading

A HTML heading or HTML h tag 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 web page.

See this example:

<h1>Heading no. 1**</h1>**

<h2>Heading no. 2**</h2>**

<h3>Heading no. 3**</h3>**

<h4>Heading no. 4**</h4>**

<h5>Heading no. 5**</h5>**

<h6>Heading no. 6**</h6>**

HTML Paragraph

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

See this example:

** **This is first paragraph.** **

MAESTRO

** **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.

How to Use <br> and <hr> tag with paragraph?

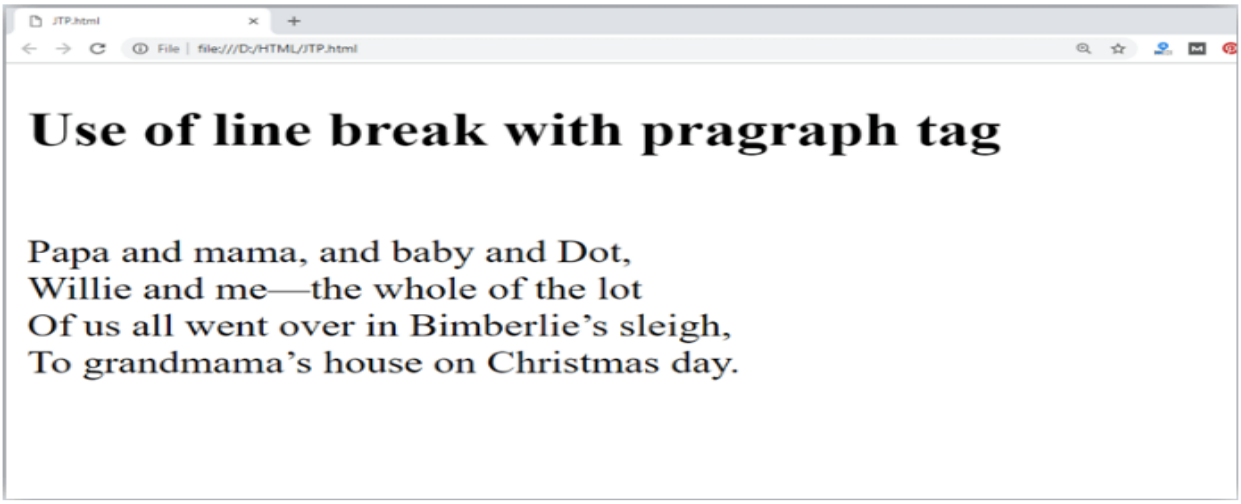
An HTML <br> tag is used for line break and it can be used with paragraph elements. Following is the example to show how to use <br> with element.

Example:

```
<!DOCTYPE html>
<html>
  <head>
  </head>
 <body>
   <h2> Use of line break with pragraph tag</h2>
     <p><br>Papa and mama, and baby and Dot,
   <br>Willie and me?the whole of the lot
       <br>Of us all went over in Bimberlie's sleigh,
        <br>To grandmama's house on Christmas day.
     </p>
  </body>
</html>
```

MAESTRO

Output:

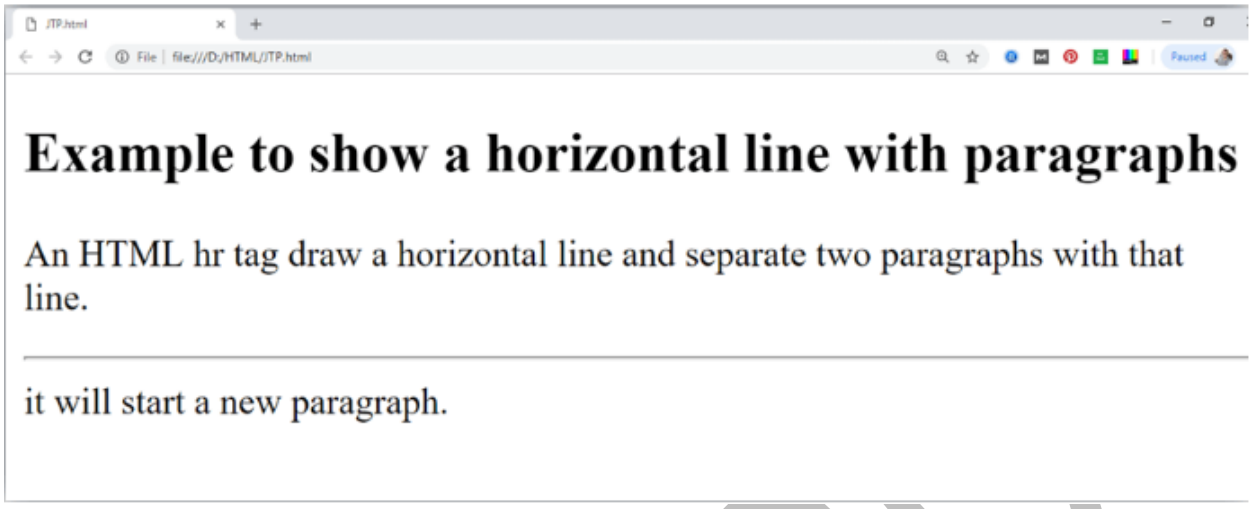


An HTML <hr> tag is used to apply a horizontal line between two statements or two paragraphs. Following is the example which is showing use of <hr> tag with paragraph.

Example:

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

MAESTRO



HTML Phrase tag

The HTML phrase tags are special purpose tags, which defines the structural meaning of a block of text or semantics of text. Following is the list of phrase tags, some of which we have already discussed in HTML formatting.

- Abbreviation tag : <abbr>
- Acronym tag: <acronym> (not supported in HTML5)
- Marked tag: <mark>
- Strong tag:
- Emphasized tag :
- Definition tag: <dfn>
- Quoting tag: <blockquote>
- Short quote tag : <q>
- Code tag: <code>
- Keyboard tag: <kbd>
- Address tag: <address>

1. Text Abbreviation tag

This tag is used to abbreviate a text. To abbreviate a text, write text between <abbr> and </abbr> tag.

Example

MAESTRO

** **An **<abbr** title = "Hypertext Markup language"**>**HTML **</abbr>**language is used to create web pa ges. ** **

2. Marked tag:

The content written between <mark> and </mark> tag will show as yellow mark on browser. This tag is used to highlight a particular text.

Example

** **This tag will **<mark>**highlight**</mark>** the text.** **

3. Strong text:

This tag is used to display the important text of the content. The text written between and will be displayed as important text.

Example

** **In HTML it is recommended to use ****lower-case****, while writing a code. ** **

4. Emphasized text

This tag is used to emphasize the text, and displayed the text in italic form. The text written between and tag will italicized the text.

Example

** **HTML is an ****easy ****to learn language.** **

5. Definition tag:

When you use the <dfn> and </dfn> tags, it allow to specify the keyword of the content. Following is the example to show how to definition element.

Example

** <dfn>**HTML **</dfn>** is a markup language. ** **

6. Quoting text:

The HTML <blockquote> element shows that the enclosed content is quoted from another source. The Source URL can be given using the cite attribute, and text representation of source can display using **<cite> ..... </cite>element**.

MAESTRO

Example

```
<blockquote cite="https://www.keepinspiring.me/famous-
quotes/"><p>?The first step toward success is taken when you refuse to be a captive of the envi
ronment in which you first find yourself.?</p></blockquote>

<cite>-Mark Caine</cite>
```

7. Short Quotations:

An HTML <q> ..... </q> element defines a short quotation. If you will put any content between <q> ..... </q>, then it will enclose the text in double quotes.

Example:

```
<p>Steve Jobs said: <q>If You Are Working On Something That You Really Care About, You Don?
t Have To Be Pushed. The Vision Pulls You.</q>?</p>
```

8. Code tags

The HTML <code> </code> element is used to display the part of computer code. It will display the content in monospaced font.

```
<p>First Java program</p>

  <p><code>class Simple{ public static void main(String args[]){

   System.out.println("Hello Java"); }} </code>

  </p>
```

9. Keyboard Tag

In HTML the keyboard tag, <kbd>, indicates that a section of content is a user input from keyboard.

```
<p>Please press <kbd>Ctrl</kbd> + <kbd>Shift</kbd> + t<kbd></kbd> to restore page on chro
me.</p>
```

10. Address tag

An HTML <address> tag defines the contact information about the author of the content. The content written between <address> and </address> tag, then it will be displayed in italic font.

```
<address> You can ask your queries by contact us on <a href="">example123@newdomain.com
</a>

 <br> You can also visit at: <br>58 S. Garfield Street. Villa Rica, GA 30187.

 </address>
```

MAESTRO

HTML Anchor

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

href attribute of HTML anchor tag

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.

 Link Text

Let's see an example of HTML anchor tag.

```
<a href="second.html">Click for Second Page</a>
```

Specify a location for Link using target attribute

If we want to open that link to another page then we can use target attribute of <a> tag. With the help of this link will be open in next page.

Example:

```
<!DOCTYPE html>
<html>
<head>
  <title></title>
</head>
<body>
<p>Click on <a href="https://www.javatpoint.com/" target="_blank"> this-
link </a>to go on home page of JavaTpoint.</p>
</body>
</html>
```

Note:

- The **target** attribute can only use with href attribute in anchor tag.

MAESTRO

- If we will not use target attribute then link will open in same page.

Appearance of HTML anchor tag

An **unvisited link** is displayed underlined and blue.

A **visited link** displayed underlined and purple.

An **active link** is underlined and red.

HTML Image

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

Let's see an example of HTML image.

```
<h2>HTML Image Example</h2>

<img src="good_morning.jpg" alt="Good Morning Friends"/>
```

Attributes of HTML img tag

The src and alt are important attributes of HTML img tag. All attributes of HTML image tag are given below.

1) 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 image may be on the same directory or another server.

2) alt

The alt attribute defines an alternate text for the image, if it can't be displayed. The value of the alt attribute describe the image in words. The alt attribute is considered good for SEO prospective.

3) width

It is an optional attribute which is used to specify the width to display the image. It is not recommended now. You should apply CSS in place of width attribute.

4) height

It h3 the height of the image. The HTML height attribute also supports iframe, image and object elements. It is not recommended now. You should apply CSS in place of height attribute.

MAESTRO

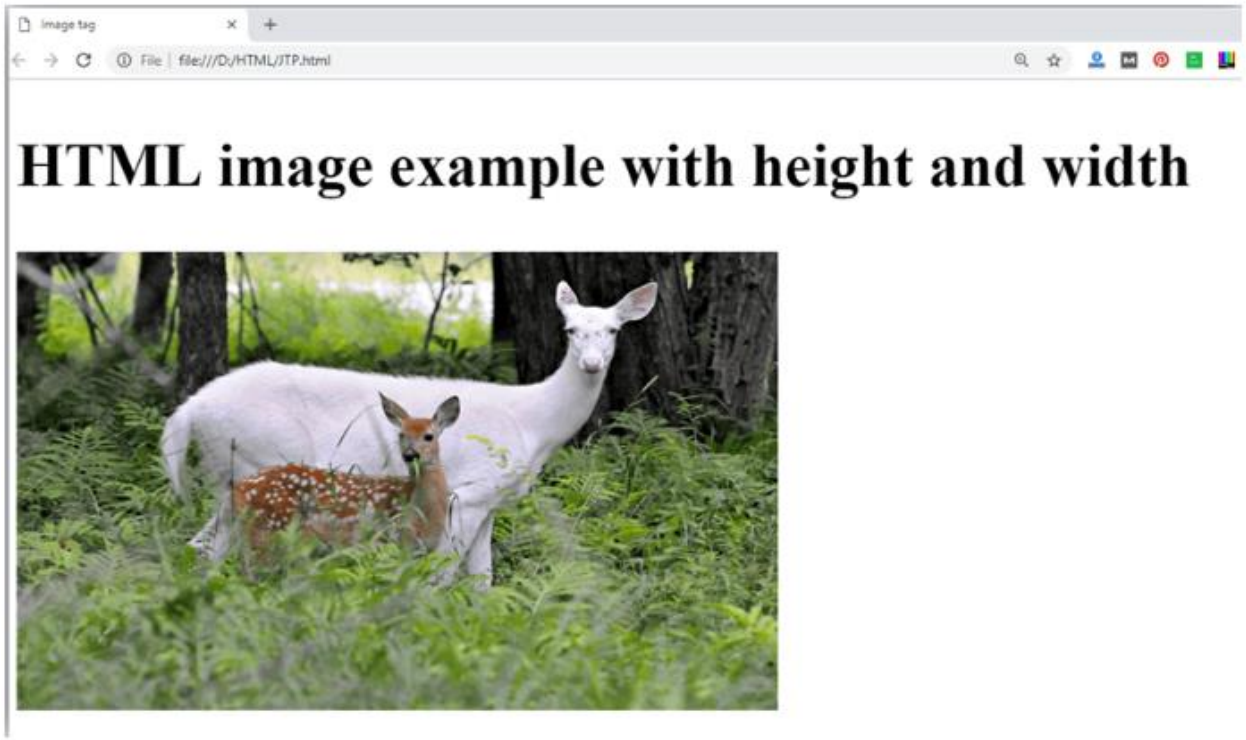
Use of height and width attribute with img tag

You have learnt about how to insert an image in your web page, now if we want to give some height and width to display image according to our requirement, then we can set it with height and width attributes of image.

Example:

```
<img src="animal.jpg" height="180" width="300" alt="animal image">
```

Output:

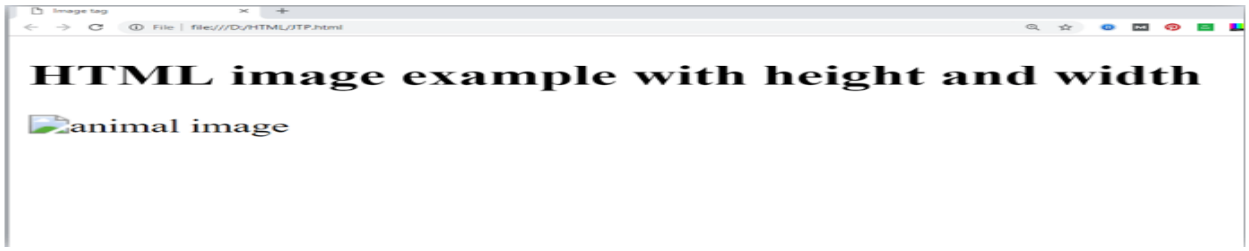


Use of alt attribute

We can use alt attribute with tag. It will display an alternative text in case if image cannot be displayed on browser. Following is the example for alt attribute:

```
<img src="animal.png" height="180" width="300" alt="animal image">
```

Output:



How to get image from another directory/folder?

To insert an image in your web, that image must be present in your same folder where you have put the HTML file. But if in some case image is available in some other directory then you can access the image like this:

```
<img src="E:/images/animal.png" height="180" width="300" alt="animal image">
```

In above statement we have put image in local disk E------>images folder------>animal.png.

Use <img> tag as a link

We can also link an image with other page or we can use an image as a link. To do this, put <img> tag inside the <a> tag.

Example:

```
<a href="https://www.javatpoint.com/what-is-
robotics"><img src="robot.jpg" height="100" width="100"></a>
```

HTML Table

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.

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

HTML Table Tags

MAESTRO

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.
<colgroup>	It specifies a group of one or more columns in a table for formatting.
<col>	It is used with <colgroup> element to specify column properties for each column.
<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.

HTML Table Example

Let's see the example of HTML table tag. It output is shown above.

```
<table>
<tr><th>First_Name</th><th>Last_Name</th><th>Marks</th></tr>
<tr><td>Sonoo</td><td>Jaiswal</td><td>60</td></tr>
<tr><td>James</td><td>William</td><td>80</td></tr>
</table>
```

Output:

First_Name	Last_Name	Marks
Sonoo	Jaiswal	60
James	William	80

In the above html table, there are 5 rows and 3 columns = 5 * 3 = 15 values.

HTML Table with Border

There are two ways to specify border for HTML tables.

1. By border attribute of table in HTML
2. By border property in CSS

1) HTML Border attribute

You can use border attribute of table tag in HTML to specify border. But it is not recommended now.

```
<table border="1">
<tr><th>First_Name</th><th>Last_Name</th><th>Marks</th></tr>
<tr><td>Sonoo</td><td>Jaiswal</td><td>60</td></tr>
<tr><td>James</td><td>William</td><td>80</td></tr>
<tr><td>Swati</td><td>Sironi</td><td>82</td></tr>
<tr><td>Chetna</td><td>Singh</td><td>72</td></tr>
</table>
```

Output:

First_Name	Last_Name	Marks
Sonoo	Jaiswal	60
James	William	80
Swati	Sironi	82
Chetna	Singh	72

MAESTRO

2) CSS Border property

It is now recommended to use border property of CSS to specify border in table.

```
<style>
table, th, td {
  border: 1px solid black;  }
</style>
```

You can collapse all the borders in one border by border-collapse property. It will collapse the border into one.

```
<style>
table, th, td {
  border: 2px solid black;
  border-collapse: collapse;
}
</style>
```

Output:

Name	**Last Name**	**Marks**
Sonoo	Jaiswal	60
James	William	80
Swati	Sironi	82
Chetna	Singh	72

HTML Table with cell padding

You can specify padding for table header and table data by two ways:

1. By cellpadding attribute of table in HTML
2. By padding property in CSS

MAESTRO

The cellpadding attribute of HTML table tag is obselete now. It is recommended to use CSS. So let's see the code of CSS.

```
<style>
table, th, td {
  border: 1px solid pink;
  border-collapse: collapse;  }
th, td {
  padding: 10px;
}
</style>
```

Output:

Name	**Last Name**	**Marks**
Sonoo	Jaiswal	60
James	William	80
Swati	Sironi	82
Chetna	Singh	72

HTML Table width:

We can specify the HTML table width using the **CSS width** property. It can be specify in pixels or percentage.

We can adjust our table width as per our requirement. Following is the example to display table with width.

```
table{
```

```
    width: 100%;
   }
```

Example:

```
<!DOCTYPE html>
<html>
<head>
  <title>table</title>
  <style>
    table{
      border-collapse: collapse;
      width: 100%;
    }
  th,td{
    border: 2px solid green;
    padding: 15px;
  }
 </style>
</head>
<body>
 <table>
  <tr>
    <th>1 header</th>
    <th>1 header</th>
    <th>1 header</th>
  </tr>
```

MAESTRO

```
    <tr>
      <td>1data</td>
      <td>1data</td>
      <td>1data</td>
    </tr>
    <tr>
      <td>2 data</td>
      <td>2 data</td>
      <td>2 data</td>
    </tr>
    <tr>
      <td>3 data</td>
      <td>3 data</td>
      <td>3 data</td>
    </tr>
  </table>
  </body>
  </html>
```

HTML Table with colspan

If you want to make a cell span more than one column, you can use the colspan attribute.

It will divide one cell/row into multiple columns, and the number of columns depend on the value of colspan attribute.

Let's see the example that span two columns.

CSS code:

```
    <style>
    table, th, td {
```

MAESTRO

```
  border: 1px solid black;
  border-collapse: collapse;
}
th, td {
  padding: 5px;
}
</style>
```

HTML code:

```
<table style="width:100%">
  <tr>
  <th>Name</th>
  <th colspan="2">Mobile No.</th>
  </tr>
  <tr>
  <td>Ajeet Maurya</td>
  <td>7503520801</td>
  <td>9555879135</td>
  </tr>
</table>
```

Output:

Name	**Mobile No.**	
Ajeet Maurya	7503520801	9555879135

HTML Table with rowspan

If you want to make a cell span more than one row, you can use the rowspan attribute.

MAESTRO

It will divide a cell into multiple rows. The number of divided rows will depend on rowspan values.

Let's see the example that span two rows.

CSS code:

```
<style>
table, th, td {
  border: 1px solid black;
  border-collapse: collapse;
}
th, td {
  padding: 10px;
}
</style>
```

HTML code:

```
<table>
<tr><th>Name</th><td>Ajeet Maurya</td></tr>
<tr><th rowspan="2">Mobile No.</th><td>7503520801</td></tr>
<tr><td>9555879135</td></tr>
</table>
```

Output:

Name	Ajeet Maurya
Mobile No.	7503520801
	9555879135

MAESTRO

HTML table with caption

HTML caption is diplayed above the table. It must be used after table tag only.

```
<table>
<caption>Student Records</caption>
<tr><th>First_Name</th><th>Last_Name</th><th>Marks</th></tr>
<tr><td>Vimal</td><td>Jaiswal</td><td>70</td></tr>
<tr><td>Mike</td><td>Warn</td><td>60</td></tr>
<tr><td>Shane</td><td>Warn</td><td>42</td></tr>
<tr><td>Jai</td><td>Malhotra</td><td>62</td></tr>
</table>
```

Styling HTML table even and odd cells

CSS code:

```
<style>
table, th, td {
  border: 1px solid black;
  border-collapse: collapse;
}
th, td {
  padding: 10px;
}
table#alter tr:nth-child(even) {
  background-color: #eee;
}
table#alter tr:nth-child(odd) {
  background-color: #fff;
```

MAESTRO

```
}
table#alter th {
 color: white;
 background-color: gray;
}
</style>
```

HTML Lists

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

1. Ordered List or Numbered List (ol)
2. Unordered List or Bulleted List (ul)
3. Description List or Definition List (dl)

HTML Ordered List or Numbered List

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

Output:

1. Aries
2. Bingo
3. Leo
4. Oracle

MAESTRO

HTML Unordered List or Bulleted List

In HTML Unordered list, all the list items are marked with bullets. It is also known as bulleted list also. The Unordered list starts with <ul> tag and list items start with the <li> tag.

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

Output:

- Aries
- Bingo
- Leo
- Oracle

HTML Description List or Definition List

HTML Description list is also a list style which is supported by HTML and XHTML. It is also known as definition list where entries are listed like a dictionary or encyclopedia.

The definition list is very appropriate when you want to present glossary, list of terms or other name-value list.

The HTML definition list contains following three tags:

<dl> tag defines the start of the list.

<dt> tag defines a term.

<dd> tag defines the term definition (description).

```
<dl>
 <dt>Aries</dt>
 <dd>-One of the 12 horoscope sign.</dd>
 <dt>Bingo</dt>
```

MAESTRO

<dd>-One of my evening snacks**</dd>**

<dt>Leo**</dt>**

<dd>-It is also an one of the 12 horoscope sign.**</dd>**

<dt>Oracle**</dt>**

<dd>-It is a multinational technology corporation.**</dd>**

</dl>

Output:

Aries

-One of the 12 horoscope sign.

Bingo

-One of my evening snacks

Leo

-It is also an one of the 12 horoscope sign.

Oracle

-It is a multinational technology corporation.

HTML Nested List

A list within another list is termed as nested list. If you want a bullet list inside a numbered list then such type of list will called as nested list.

Code:

<!DOCTYPE html**>**

<html>

<head>

<title>Nested list**</title>**

</head>

<body>

** **List of Indian States with thier capital** **

MAESTRO

```
<ol>
  <li>Delhi
    <ul>
      <li>NewDelhi</li>
    </ul>
  </li>
  <li>Haryana
    <ul>
      <li>Chandigarh</li>
    </ul>
  </li>
  <li>Gujarat
    <ul>
      <li>Gandhinagar</li>
    </ul>
  </li>
  <li>Rajasthan
    <ul>
      <li>Jaipur</li>
    </ul>
  </li>
  <li>Maharashtra
    <ul>
      <li>Mumbai</li>
    </ul>
```

MAESTRO

```
    </li>
    <li>Uttarpradesh
      <ul>
        <li>Lucknow</li></ul>
    </li>
  </ol>
  </body>
  </html>
```

HTML Form

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

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

Why use HTML Form

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

For example: If a user want to purchase some items on internet, he/she must fill the form such as shipping address and credit/debit card details so that item can be sent to the given address.

HTML Form Syntax

```
<form action="server url" method="get|post">
  //input controls e.g. textfield, textarea, radiobutton, button
</form>
```

MAESTRO

HTML Form Tags

Let's see the list of HTML 5 form tags.

Tag	Description
<form>	It defines an HTML form to enter inputs by the used side.
<input>	It defines an input control.
<textarea>	It defines a multi-line input control.
<label>	It defines a label for an input element.
<fieldset>	It groups the related element in a form.
<legend>	It defines a caption for a <fieldset> element.
<select>	It defines a drop-down list.
<optgroup>	It defines a group of related options in a drop-down list.
<option>	It defines an option in a drop-down list.
<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.

Syntax:

<form>

//Form elements

</form>

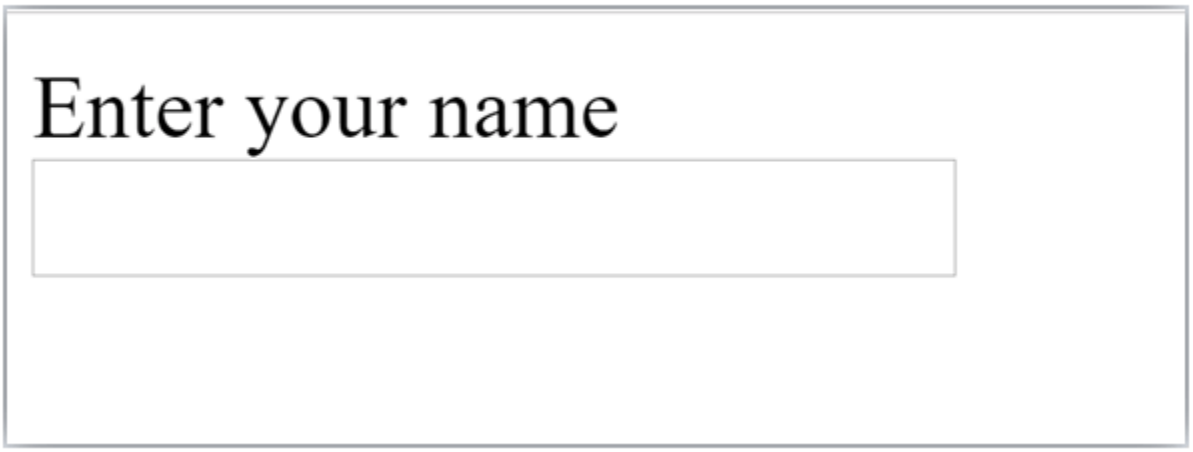
HTML <input> element

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 filed to gather different information form user. Following is the example to show the simple text input.

Example:

```
<body>
 <form>
   Enter your name <br>
  <input type="text" name="username">
 </form>
</body>
```

Output:

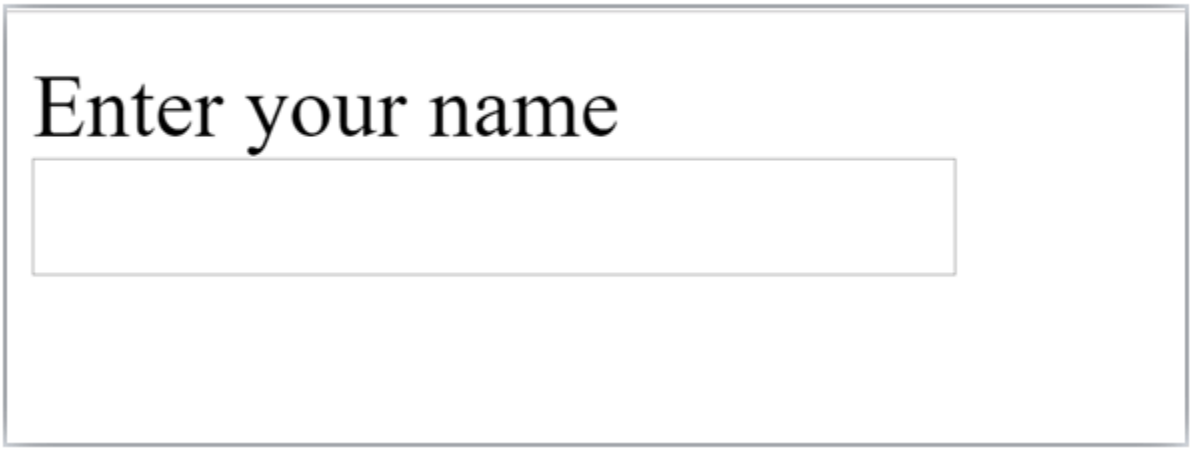


HTML TextField Control

The type="text" attribute of input tag creates textfield control also known as single line textfield control. The name attribute is optional, but it is required for the server side component such as JSP, ASP, PHP etc.

```
<form>
   First Name: <input type="text" name="firstname"/> <br/>
   Last Name:  <input type="text" name="lastname"/> <br/>
</form>
```

Output:



HTML <textarea> tag in form

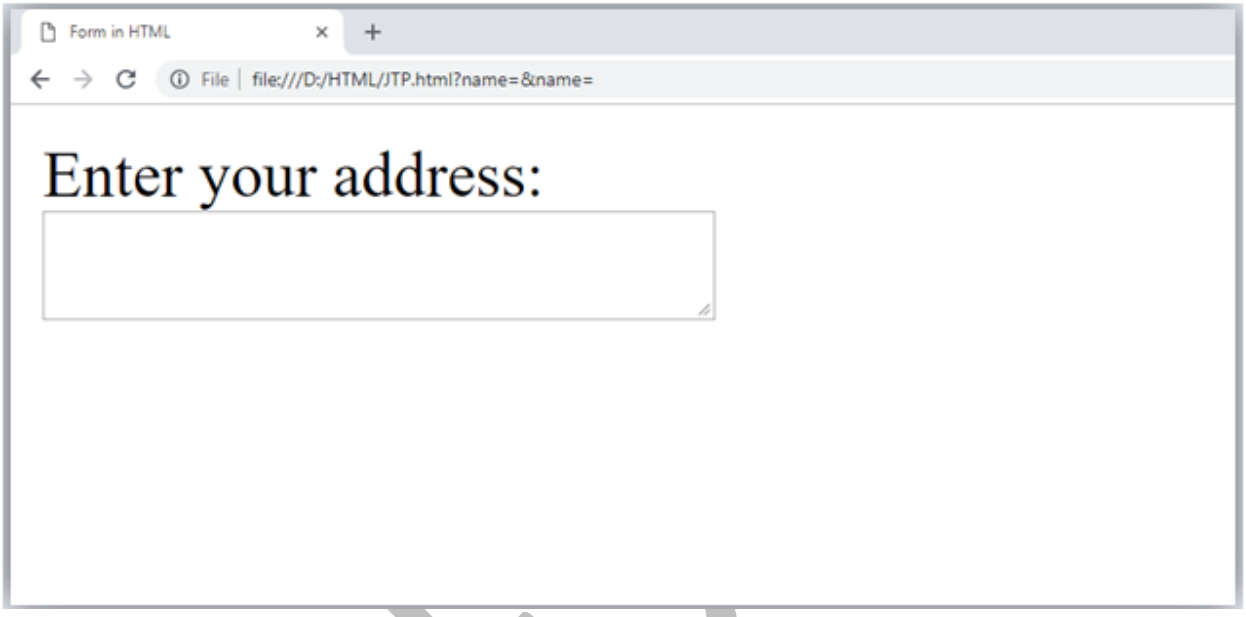
The <textarea> tag in HTML is used to insert multiple-line text in a form. The size of <textarea> can be specify either using "rows" or "cols" attribute or by CSS.

Example:

```
<!DOCTYPE html>
<html>
<head>
  <title>Form in HTML</title>
</head>
<body>
 <form>
    Enter your address:<br>
   <textarea rows="2" cols="20"></textarea>
```

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

Output:



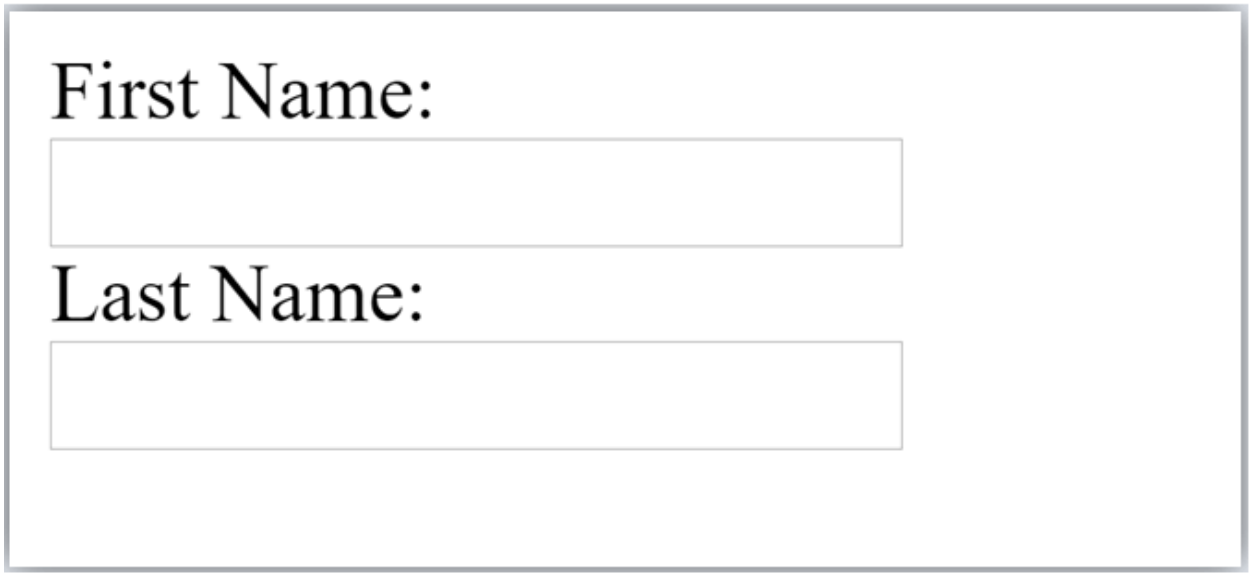
Label Tag in Form

It is considered better to have label in form. As it makes the code parser/browser/user friendly.

If you click on the label tag, it will focus on the text control. To do so, you need to have for attribute in label tag that must be same as id attribute of input tag.

```
<form>
  <label for="firstname">First Name: </label> <br/>
        <input type="text" id="firstname" name="firstname"/> <br/>
  <label for="lastname">Last Name: </label>
        <input type="text" id="lastname" name="lastname"/> <br/>
</form>
```

Output:



HTML Password Field Control

The password is not visible to the user in password field control.

```
<form>
  <label for="password">Password: </label>
    <input type="password" id="password" name="password"/> <br/>
</form>
```



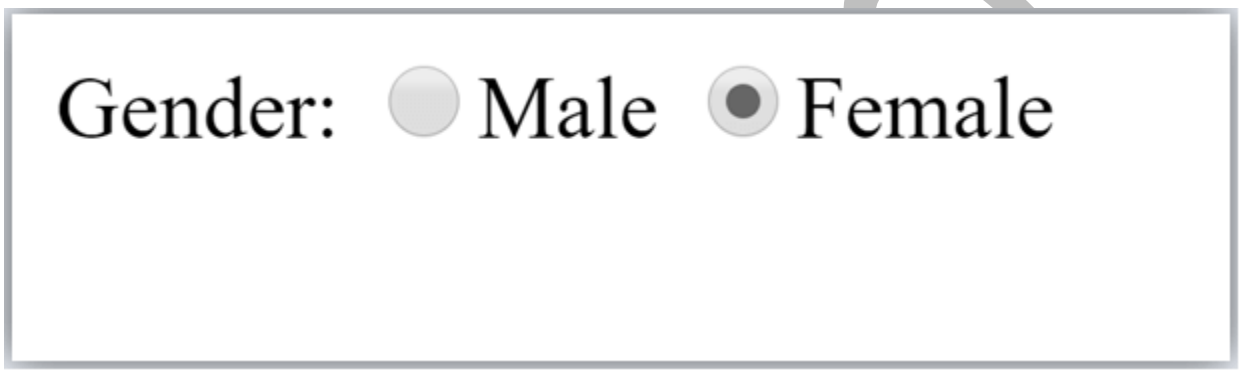
Radio Button Control

The radio button is used to select one option from multiple options. It is used for selection of gender, quiz questions etc.

If you use one name for all the radio buttons, only one radio button can be selected at a time.

Using radio buttons for multiple options, you can only choose a single option at a time.

```
<form>
  <label for="gender">Gender: </label>
      <input type="radio" id="gender" name="gender" value="male"/>Male
      <input type="radio" id="gender" name="gender" value="female"/>Female <br/>
</form>
```

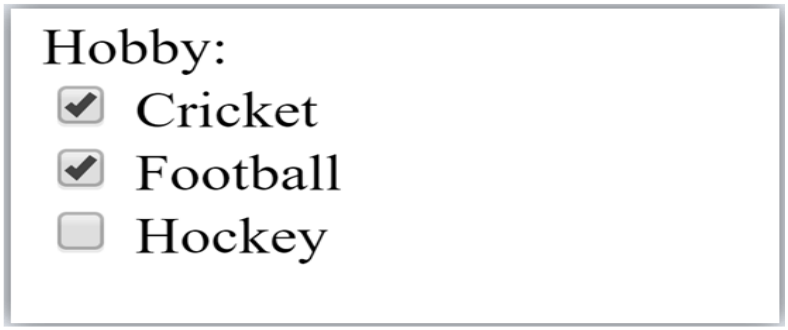


Checkbox Control

The checkbox control is used to check multiple options from given checkboxes.

```
<form>
Hobby:<br>
      <input type="checkbox" id="cricket" name="cricket" value="cricket"/>
        <label for="cricket">Cricket</label> <br>
      <input type="checkbox" id="football" name="football" value="football"/>
        <label for="football">Football</label> <br>
      <input type="checkbox" id="hockey" name="hockey" value="hockey"/>
        <label for="hockey">Hockey</label>
</form>
```

Output:



Submit button control

HTML **<input type="submit">** are used to add a submit button on web page. When user clicks on submit button, then form get submit to the server.

Syntax:

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

The type = submit , specifying that it is a submit button

The value attribute can be anything which we write on button on web page.

The name attribute can be omit here.

Example:

```
<form>
  <label for="name">Enter name</label><br>
  <input type="text" id="name" name="name"><br>
  <label for="pass">Enter Password</label><br>
  <input type="Password" id="pass" name="pass"><br>
  <input type="submit" value="submit">
</form>
```

MAESTRO

Output:



HTML <fieldset> element:

The <fieldset> element in HTML is used to group the related information of a form. This element is used with <legend> element which provide caption for the grouped elements.

Example:

```
<form>
  <fieldset>
   <legend>User Information:</legend>
  <label for="name">Enter name</label><br>
<input type="text" id="name" name="name"><br>
<label for="pass">Enter Password</label><br>
<input type="Password" id="pass" name="pass"><br>
<input type="submit" value="submit">
</fieldset>
lt;/form>
```

Output:



MAESTRO

HTML Form Input Types

In HTML <input type=" "> is an important element of HTML form. The "type" attribute of input element can be various types, which defines information field. Such as <input type="text" name="name"> gives a text box.

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

type=" "	Description
Text	Defines a one-line text input field
Password	Defines a one-line password input field
Submit	Defines a submit button to submit the form to server
Reset	Defines a reset button to reset all values in the form.
Radio	Defines a radio button which allows select one option.
Checkbox	Defines checkboxes which allow select multiple options form.
Button	Defines a simple push button, which can be programmed to perform a task on an event.
File	Defines to select the file from device storage.
Image	Defines a graphical submit button.

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

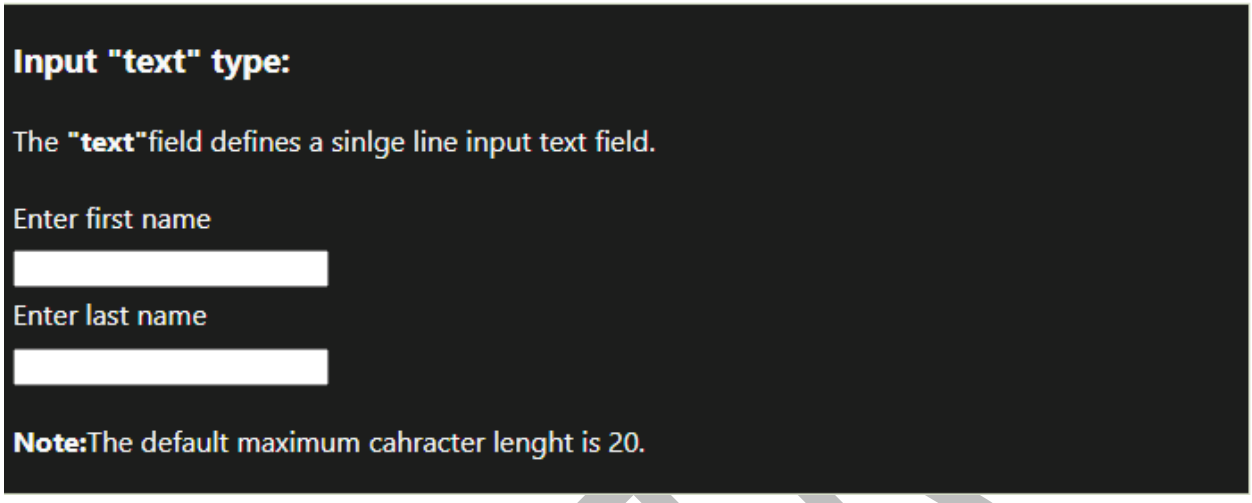
1. <input type="text">:

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

Example:

```
<form>
  <label>Enter first name</label><br>
  <input type="text" name="firstname"><br>
  <label>Enter last name</label><br>
  <input type="text" name="lastname"><br>
```

```
    <p><strong>Note:</strong>The default maximum cahracter lenght is 20.</p>
</form>
```

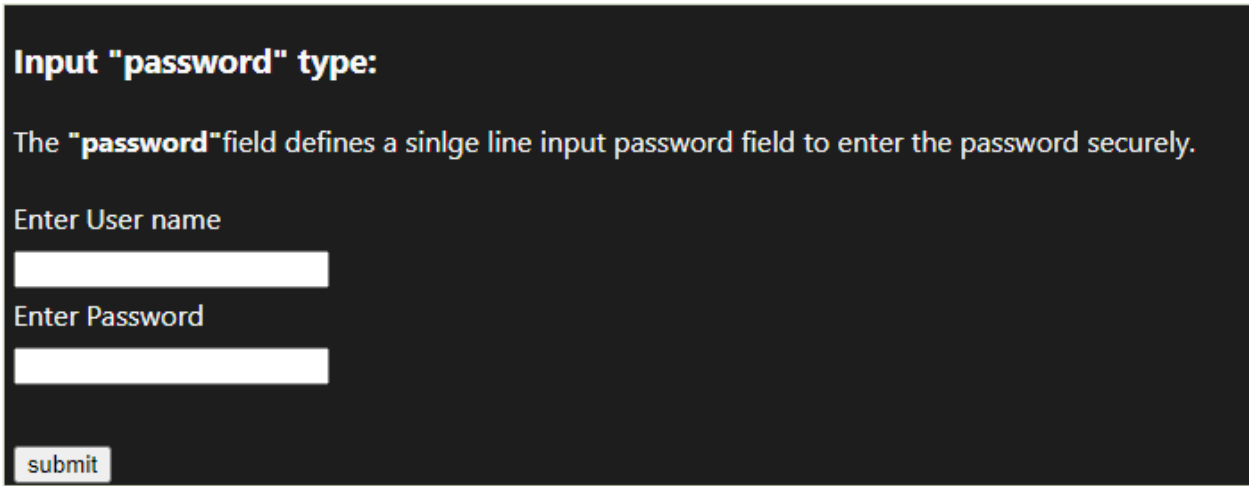


2. <input type="password">:

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

Example:

```
<form>
   <label>Enter User name</label><br>
   <input type="text" name="firstname"><br>
   <label>Enter Password</label><br>
   <input type="Password" name="password"><br>
   <br><input type="submit" value="submit">
</form>
```

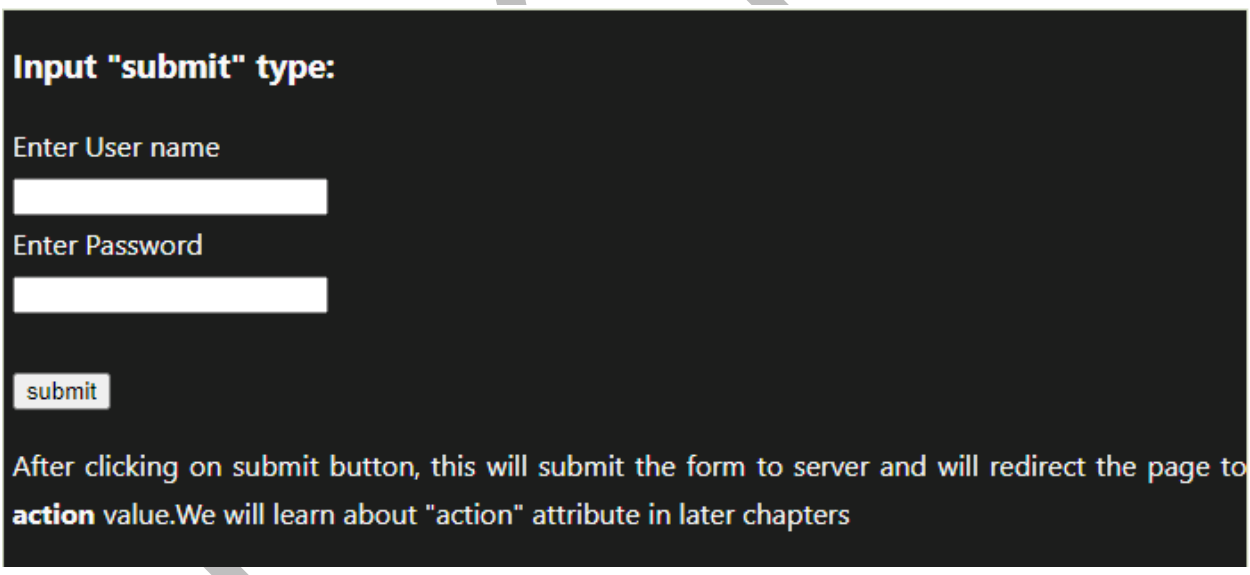


3. <input type="submit">:

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

Example:

```
<form action="https://www.javatpoint.com/html-tutorial">
   <label>Enter User name</label><br>
   <input type="text" name="firstname"><br>
   <label>Enter Password</label><br>
   <input type="Password" name="password"><br>
   <br><input type="submit" value="submit">
</form>
```



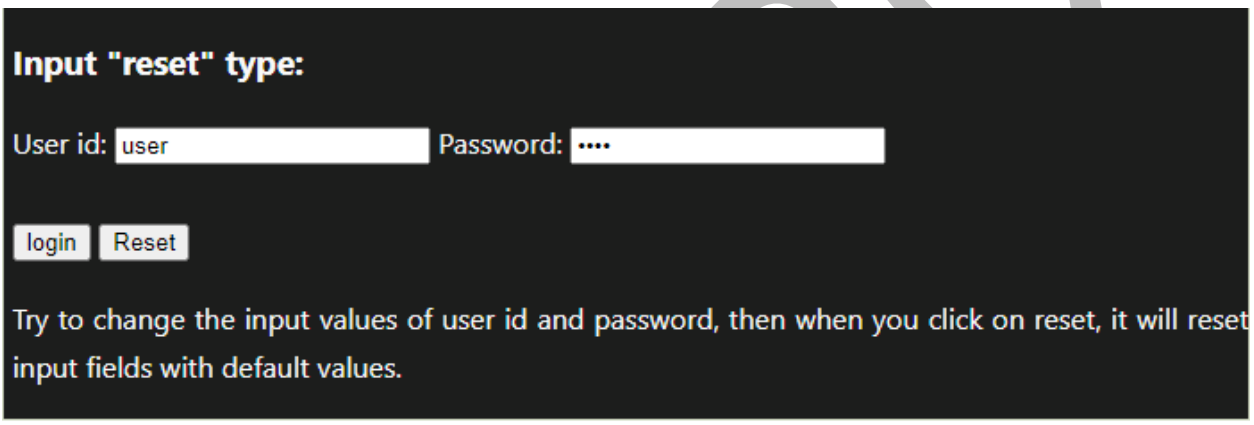
4. <input type="reset">:

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

Example:

```
<form>
```

```
<label>User id: </label>
<input type="text" name="user-id" value="user">
<label>Password: </label>
<input type="password" name="pass" value="pass"><br><br>
<input type="submit" value="login">
<input type="reset" value="Reset">
</form>
```



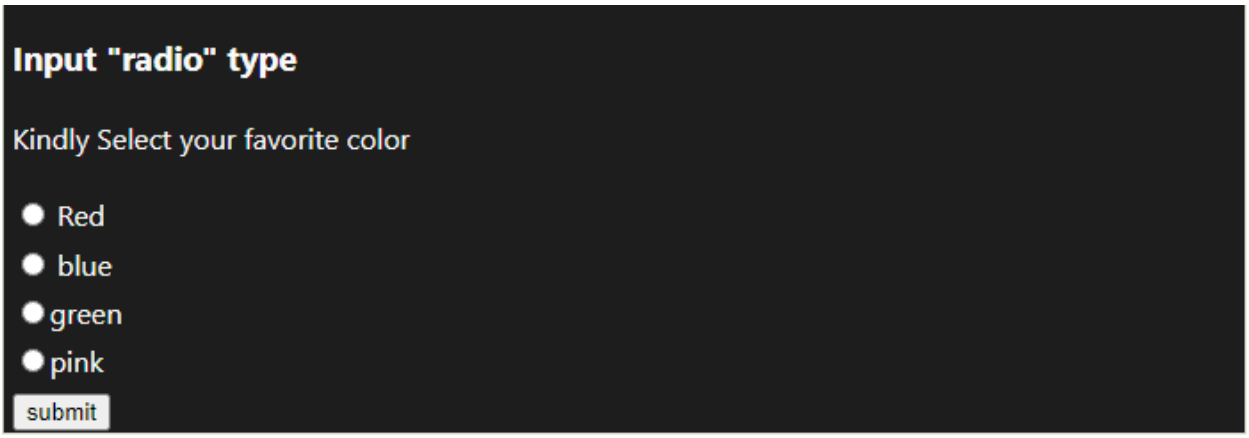
5. <input type="radio">:

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

Example:

```
<form>
<p>Kindly Select your favorite color</p>
<input type="radio" name="color" value="red"> Red <br>
<input type="radio" name="color" value="blue"> blue <br>
<input type="radio" name="color" value="green">green <br>
<input type="radio" name="color" value="pink">pink <br>
<input type="submit" value="submit">
```

</form>

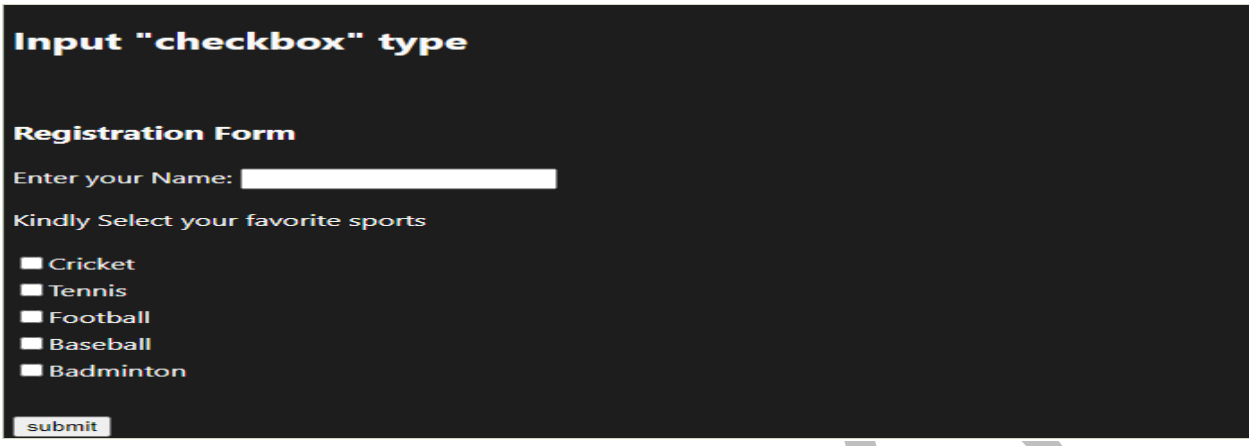


6. <input type="checkbox">:

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

Example:

```
<form>
    <label>Enter your Name:</label>
    <input type="text" name="name">
    <p>Kindly Select your favourite sports</p>
    <input type="checkbox" name="sport1" value="cricket">Cricket<br>
    <input type="checkbox" name="sport2" value="tennis">Tennis<br>
    <input type="checkbox" name="sport3" value="football">Football<br>
    <input type="checkbox" name="sport4" value="baseball">Baseball<br>
    <input type="checkbox" name="sport5" value="badminton">Badminton<br><br>
    <input type="submit" value="submit">
</form>
```



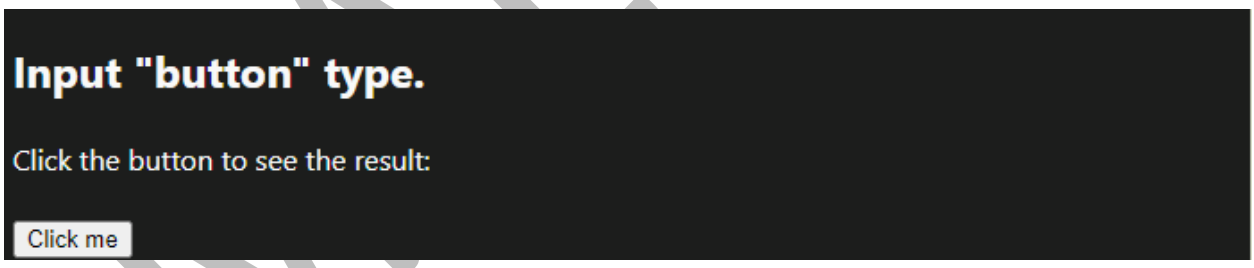
7. <input type="button">:

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

Note: It mainly works with JavaScript.

Example:

```
<form>
    <input type="button" value="Clcik me " onclick="alert('you are learning HTML')">
</form>
```



8. <input type="file">:

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

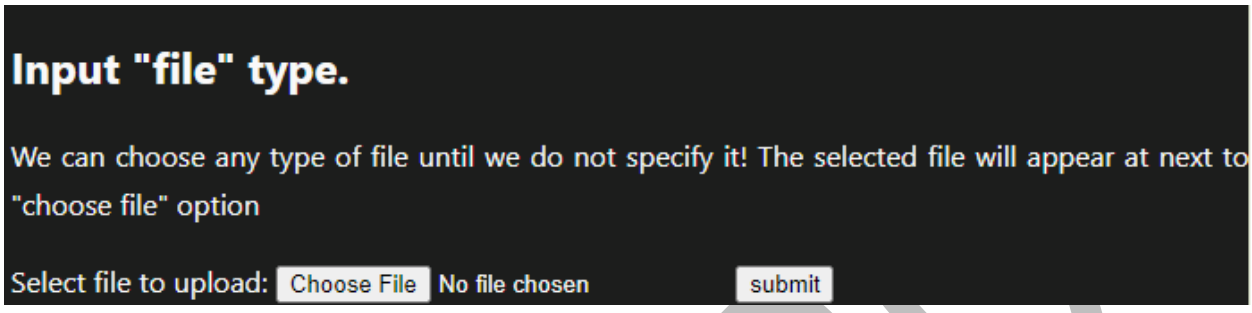
Example:

```
<form>
    <label>Select file to upload:</label>
```

```
    <input type="file" name="newfile">

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

</form>
```



9. <input type="image">:

The <input> type "image" is used to represent a submit button in the form of image.

Example:

```
<!DOCTYPE html>

<html>

<body>

<h2>Input "image" type.</h2>

<p>We can create an image as submit button</p>

 <form>

  <label>User id:</label><br>

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

   <input type="image" alt="Submit" src="login.png"  width="100px">

 </form>

</body>

</html>
```

HTML form Attribute

HTML <form> element attributes

In HTML there are various attributes available for <form> element which are given below:

MAESTRO

HTML action attribute

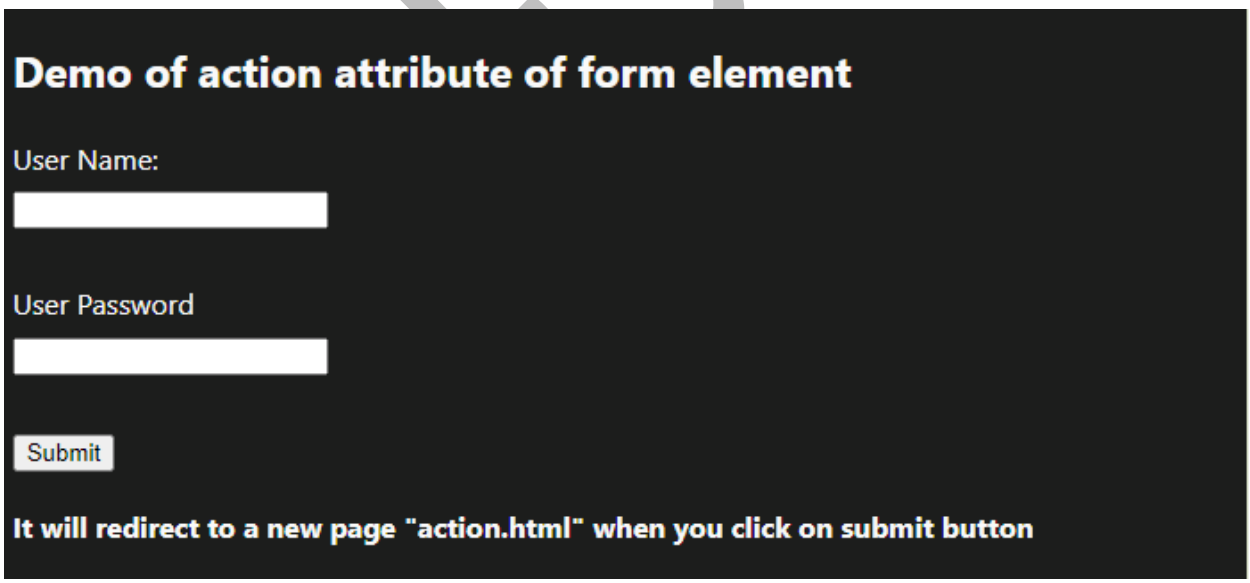
The action attribute of <form> element defines the process to be performed on form when form is submitted, or it is a URI to process the form information.

The action attribute value defines the web page where information proceed. It can be .php, .jsp, .asp, etc. or any URL where you want to process your form.

Note: If action attribute value is blank then form will be processed to the same page.

Example:

```
<form action="action.html" method="post">
<label>User Name:</label><br>
<input type="text" name="name"><br><br>
<label>User Password</label><br>
<input type="password" name="pass"><br><br>
 <input type="submit">
  </form>
```



HTML method attribute

The method attribute defines the HTTP method which browser used to submit the form. The possible values of method attribute can be:

- post: We can use the post value of method attribute when we want to process the sensitive data as it does not display the submitted data in URL.

Example:

```
<form action="action.html" method="post">
```

- get: The get value of method attribute is default value while submitting the form. But this is not secure as it displays data in URL after submitting the form.

Example:

```
<form action="action.html" method="get">
```

When submitting the data, it will display the entered data in the form of:

```
file:///D:/HTML/action.html?name=JavaTPoint&pass=123
```

HTML target attribute

The target attribute defines where to open the response after submitting the form. The following are the keywords used with the target attribute.

- **_self:** If we use _self as an attribute value, then the response will display in current page only.

Example:

```
<form action="action.html" method="get" target="_self">
```

- **_blank:** If we use _blank as an attribute it will load the response in a new page.

Example:

```
<form action="action.html" method="get" target="_blank">
```

HTML autocomplete attribute

The HTML autocomplete attribute is a newly added attribute of HTML5 which enables an input field to complete automatically. It can have two values "on" and "off" which enables autocomplete either ON or OFF. The default value of autocomplete attribute is "on".

Example:

```
<form action="action.html" method="get" autocomplete="on">
```

MAESTRO

Example:

```
<form action="action.html" method="get" autocomplete="off">
```

Note: it can be used with <form> element and <input> element both.

HTML enctype attribute

The HTML enctype attribute defines the encoding type of form-content while submitting the form to the server. The possible values of enctype can be:

- **application/x-www-form-urlencoded:** It is default encoding type if the enctype attribute is not included in the form. All characters are encoded before submitting the form.

Example:

```
<form action="action.html" method="post" enctype="application/x-www-form-urlencoded" >
```

- **multipart/form-data:** It does not encode any character. It is used when our form contains file-upload controls.

Example:

```
<form action="action.html" method="post" enctype="multipart/form-data">
```

- **text/plain (HTML5):** In this encoding type only space are encoded into + symbol and no any other special character encoded.

Example:

```
<form action="action.html" method="post" enctype="text/plain" >
```

HTML <input> element attribute

HTML name attribute

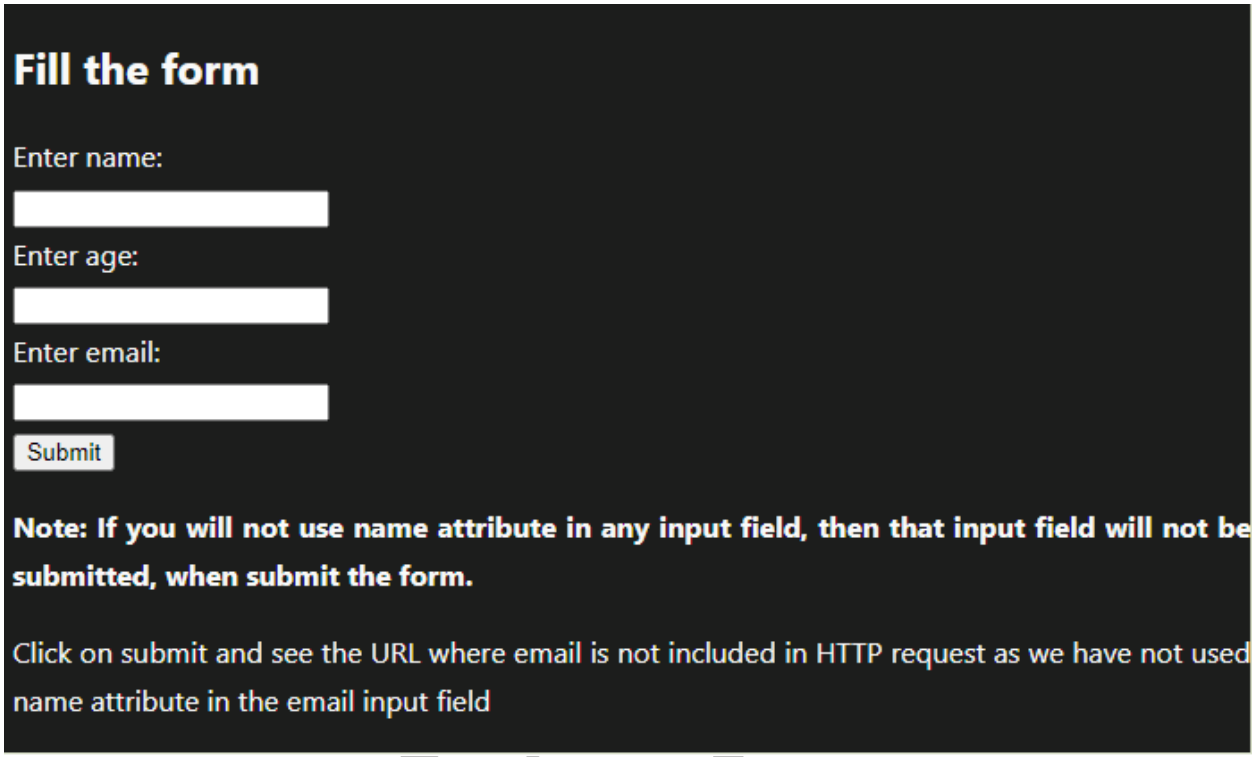
The HTML name attribute defines the name of an input element. The name and value attribute are included in HTTP request when we submit the form.

Example:

```
<form action = "action.html" method = "get">
    Enter name:<br><input type="name" name="uname"><br>
    Enter age:<br><input type="number" name="age"><br>
    Enter email:<br><input type="email"><br>
```

MAESTRO

```
    <input type="submit" value="Submit">
</form>
```

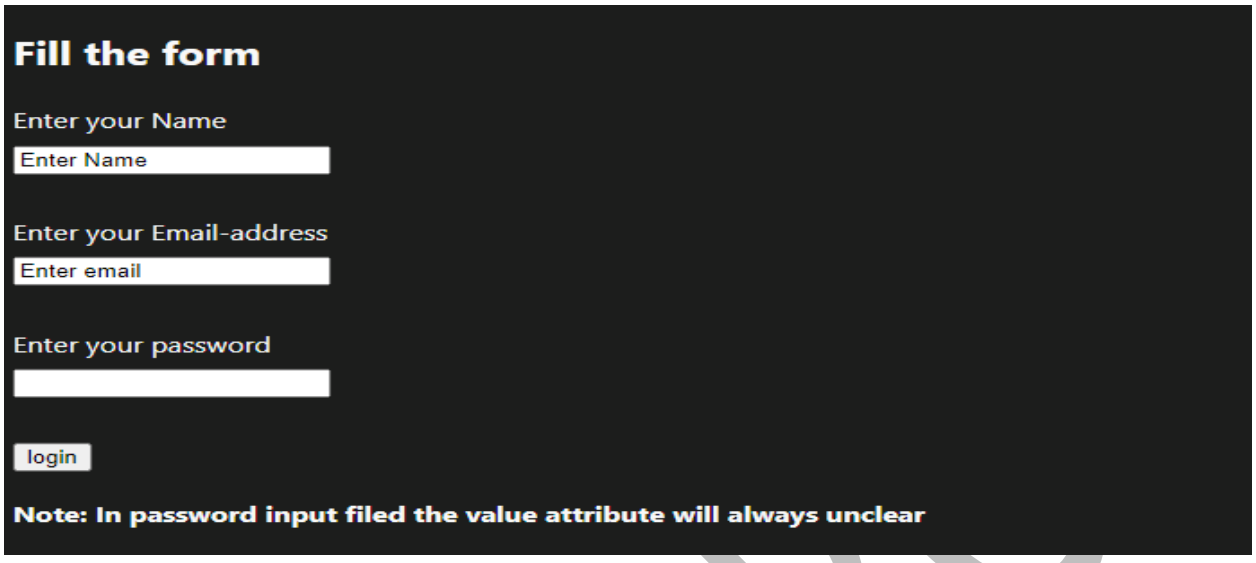


HTML value attribute

The HTML value attribute defines the initial value or default value of an input field.

Example:

```
    <form>
        <label>Enter your Name</label><br>
        <input type="text" name="uname" value="Enter Name"><br><br>
        <label>Enter your Email-address</label><br>
        <input type="text" name="uname" value="Enter email"><br><br>
        <label>Enter your password</label><br>
        <input type="password" name="pass" value=""><br><br>
<input type="submit" value="login">
    </form>
```



HTML disabled attribute

The HTML disabled attribute when applied then it disable that input field. The disabled field does not allow the user to interact with that field.

The disabled input filed does not receive click events, and these input value will not be sent to the server when submitting the form.

Example:

```
<input type="text" name="uname" disabled><br><br>
```

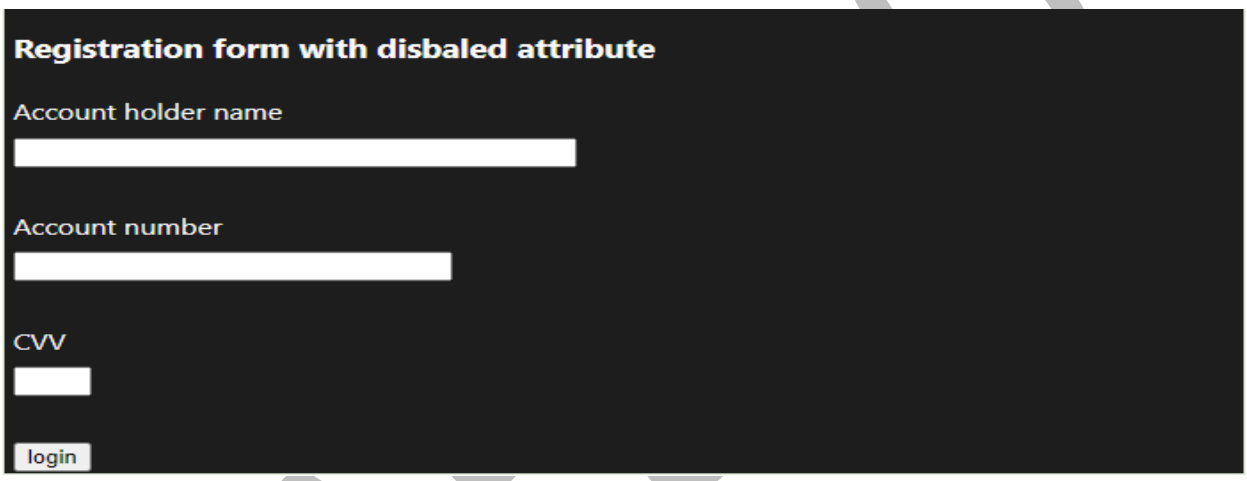


HTML size attribute

The size attribute controls the size of the input field in typed characters.

Example:

```
<label>Account holder name</label><br>
    <input type="text" name="uname" size="40" required><br><br>
    <label>Account number</label><br>
    <input type="text" name="an" size="30" required><br><br>
    <label>CVV</label><br>
    <input type="text" name="cvv"  size="1" required><br><br>
```

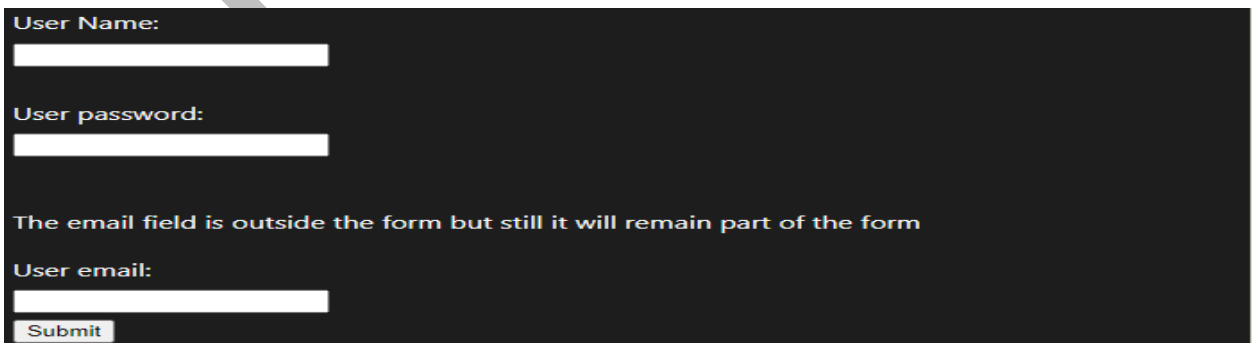


HTML form attribute

HTML form attribute allows a user to specify an input filed outside the form but remains the part of the parent form.

Example:

```
User email: <br><input type="email" name="email"  form="fcontrol"  required><br>
    <input type="submit" form="fcontrol">
```



HTML style using CSS

Let's suppose we have created our web page using a simple HTML code, and we want something which can present our page in a correct format, and visibly attractive. So to do this, we can style our web page with CSS (Cascading Stylesheet) properties.

CSS is used to apply the style in the web page which is made up of HTML elements. It describes the look of the webpage.

CSS provides various style properties such as background color, padding, margin, border-color, and many more, to style a webpage.

Each property in CSS has a name-value pair, and each property is separated by a semicolon (;).

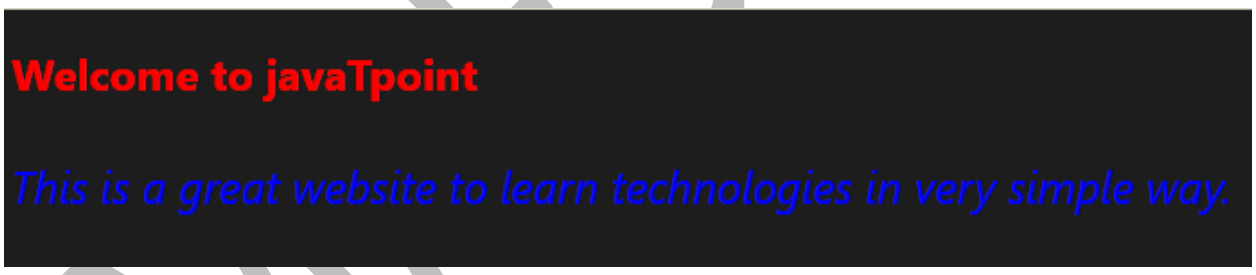
Example:

```
<body style="text-align: center;">

    <h2 style="color: red;">Welcome to javaTpoint</h2>

    <p style="color: blue; font-size: 25px; font-
style: italic ;">This is a great website to learn technologies in very simple way. </p>

</body>
```



Three ways to apply CSS

To use CSS with HTML document, there are three ways:

- **Inline CSS:** Define CSS properties using style attribute in the HTML elements.
- **Internal or Embedded CSS:** Define CSS using <style> tag in <head> section.
- **External CSS:** Define all CSS property in a separate .css file, and then include the file with HTML file using tag in section.

Inline CSS:

Inline CSS is used to apply CSS in a single element. It can apply style uniquely in each element.

To apply inline CSS, you need to use style attribute within HTML element. We can use as many properties as we want, but each property should be separated by a semicolon (;).

Example:

```
<h3 style="color: red;
    font-style: italic;
    text-align: center;
    font-size: 50px;
    padding-top: 25px;">Learning HTML using Inline CSS</h3>
```



Internal CSS:

An Internal stylesheets contains the CSS properties for a webpage in <head> section of HTML document. To use Internal CSS, we can use class and id attributes.

We can use internal CSS to apply a style for a single HTML page.

Example:

```
<!DOCTYPE html>
<html>
<head>
    <style>
  /*Internal CSS using element name*/
```

```
        body{background-color:lavender;
         text-align: center;}
         h2{font-style: italic;
          font-size: 30px;
          color: #f08080;}
        p{font-size: 20px;}
      /*Internal CSS using class name*/
        .blue{color: blue;}
        .red{color: red;}
        .green{color: green;}
    </style>
   </head>
  <body>
   <h2>Learning HTML with internal CSS</h2>
    <p class="blue">This is a blue color paragraph</p>
    <p class="red">This is a red color paragraph</p>
    <p class="green">This is a green color paragraph</p>
  </body>
 </html>
```

HTML Classes

Class Attribute in HTML

The HTML class attribute is used to specify a single or multiple class names for an HTML element. The class name can be used by CSS and JavaScript to do some tasks for HTML elements. You can use this class in CSS with a specific class, write a period (.) character, followed by the name of the class for selecting elements.

A class attribute can be defined within <style> tag or in separate file using the (.) character.

In an HTML document, we can use the same class attribute name with different elements.

MAESTRO

Defining an HTML class

To create an HTML class, firstly define style for HTML class using <style> tag within <head> section as following example:

Example:

```
<head>
  <style>
    .headings{
      color: lightgreen;
      font-family: cursive;
      background-color: black; }
  </style>
</head>
```

We have define style for a class name "headings", and we can use this class name with any of HTML element in which we want to provide such styling. We just need to follow the following syntax to use it.

```
<tag class="ghf"> content </tag>
```

Example 1:

```
<!DOCTYPE html>
<html>
<head>
  <style>
    .headings{
      color: lightgreen;
      font-family: cursive;
      background-color: black; }
  </style>
</head>
```

MAESTRO

```
<body>

<h1 class="headings">This is first heading</h1>

<h2 class="headings">This is Second heading</h2>

<h3 class="headings">This is third heading</h3>

<h4 class="headings">This is fourth heading</h4>

</body>

</html>
```

Another Example with different class name

Example:

Let's use a class name "Fruit" with CSS to style all elements.

```
<style>

.fruit {

   background-color: orange;

   color: white;

   padding: 10px;

}

</style>

<h2 class="fruit">Mango</h2>

<p>Mango is king of all fruits.</p>

<h2 class="fruit">Orange</h2>

<p>Oranges are full of Vitamin C.</p>

<h2 class="fruit">Apple</h2>

<p>An apple a day, keeps the Doctor away.</p>
```

Here you can see that we have used the class name "fruit" with (.) to use all its elements.

Note: You can use class attribute on any HTML element. The class name is case-sensitive.

MAESTRO

Multiple Classes

You can use multiple class names (more than one) with HTML elements. These class names must be separated by a space.

Example:

Let's style elements with class name "fruit" and also with a class name "center".

```
<!DOCTYPE html>
<html>
<style>
.fruit {
  background-color: orange;
  color: white;
  padding: 10px;
}
.center {
  text-align: center;
}
</style>
<body>
<h2>Multiple Classes</h2>
<p>All three elements have the class name "fruit". In addition, Mango also have the class name "center", which center-aligns the text.</p>
<h2 class="fruit center">Mango</h2>
<h2 class="fruit">Orange</h2>
<h2 class="fruit">Apple</h2>
</body>
</html>
```

MAESTRO

You can see that the first element <h2> belongs to both the "fruit" class and the "center" class.

Same class with Different Tag

You can use the same class name with different tags like <h2> and <p> etc. to share the same style.

Example:

```
<!DOCTYPE html>
<html>
<style>
.fruit {
  background-color: orange;
  color: white;
  padding: 10px;
}
</style>
<body>
<h2>Same Class with Different Tag</h2>
<h2 class="fruit">Mango</h2>
<p class="fruit">Mango is the king of all fruits.</p>
</body>
</html>
```

External CSS:

An external CSS contains a separate CSS file which only contains style code using the class name, id name, tag name, etc. We can use this CSS file in any HTML file by including it in HTML file using <link> tag.

If we have multiple HTML pages for an application and which use similar CSS, then we can use external CSS.

There are two files need to create to apply external CSS

- First, create the HTML file

MAESTRO

- Create a CSS file and save it using the .css extension (This file only will only contain the styling code.)
- Link the CSS file in your HTML file using tag in header section of HTML document.

Example:

```
<!DOCTYPE html>
<html>
<head>
  <link rel="stylesheet" type="text/css" href="style.css">
  </head>
 <body>
 <h2>Learning HTML with External CSS</h2>
  <p class="blue">This is a blue color paragraph</p>
  <p class="red">This is a red color paragraph</p>
  <p class="green">This is a green color paragraph</p>
 </body>
</html>
```

CSS file:

```
body{
background-color:lavender;
text-align: center;
}
h2{
font-style: italic;
size: 30px;
color: #f08080;
}
p{
font-size: 20px;
}

.blue{
```

MAESTRO

```
color: blue;
}
.red{
color: red;
}
.green{
color: green;
}
```

Commonly used CSS properties:

Properties-name	Syntax	Description
background-color	background-color:red;	It defines the background color of that element.
Color	color: lightgreen;	It defines the color of text of an element
Padding	padding: 20px;	It defines the space between content and the border.
Margin	margin: 30px; margin-left:	It creates space around an element.
font-family	font-family: cursive;	Font-family defines a font for a particular element.
Font-size	font-size: 50px;	Font-size defines a font size for a particular element.
text-align	text-align: left;	It is used to align the text in a selected position.

HTML Id Attribute

The **id attribute** is used to specify the unique ID for an element of the HTML document. It allocates the unique identifier which is used by the **CSS** and the **JavaScript** for performing certain tasks.

Note: In the Cascading Style sheet (CSS), we can easily select an element with the specific id by using the # symbol followed by id.

Note: JavaScript can access an element with the given ID by using the getElementById() method.

Syntax

<tag id="value"**>**

Example 1: The following example describes how to use the id attribute in CSS document:

```
<!DOCTYPE html>
<html>
<head>
<title>
Example of Id attribute in CSS
</title>
<style>
#Cars {
padding: 40px;
background-color: lightblue;
color: black;
text-align: center;
}
#Bikes
{
padding: 50px;
background-color: lightGreen;
text-align: center;
}
</style>
</head>
<body>
```

MAESTRO

** ** Use CSS to style an element with the id: ** **

<h1 id="Cars"**>** Cars **</h1>**

<h1 id="Bikes"**>** Bikes **</h1>**

</body>

</html>

Example 2:

The following example describes how to use the ID attribute in JavaScript.

<!DOCTYPE html**>**

<html>

<head>

<title> Date Attribute **</title>**

<script>

function viewdate() {

var x = document.getElementById("dob").value;

document.getElementById("demo").innerHTML = x;

</script>

</head>

<body>

Employee Name: **<input** type="text" placeholder="Your Good name"**/>**

**
**

**
**

Date of Joining:

<input type="date" id="dob"**>**

**
**

<button onclick="viewdate()"**>** Submit

MAESTRO

```
</button>

<br>

<h2 id="demo"> </h2>

</body>

</html>
```

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. The traditional website uses js to provide several forms of interactivity and simplicity.

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. In addition to web browsers, databases such as CouchDB and MongoDB uses 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 implicitly cast (depending on 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.

MAESTRO

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 clocks etc.

JavaScript Example

```
<script>
document.write("Hello JavaScript by JavaScript");
</script>
```

3 Places to put JavaScript code

1. Between the body tag of html
2. Between the head tag of html
3. In .js file (external javaScript)

1) JavaScript Example : code between the body tag

In the above example, we have displayed the dynamic content using JavaScript. Let's see the simple example of JavaScript that displays alert dialog box.

```
<script type="text/javascript">
 alert("Hello Javatpoint");
</script>
```

2) JavaScript Example : code between the head tag

Let's see the same example of displaying alert dialog box of JavaScript that is contained inside the head tag.

In this example, we are creating a function msg(). To create function in JavaScript, you need to write function with function_name as given below.

MAESTRO

To call function, you need to work on event. Here we are using onclick event to call msg() function.

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

External JavaScript file

We can create external JavaScript file and embed it in many html page.

It provides **code re usability** because single JavaScript file can be used in several html pages.

An external JavaScript file must be saved by .js extension. It is recommended to embed all JavaScript files into a single file. It increases the speed of the webpage.

message.js

```
function msg(){
 alert("Hello Javatpoint");
}
```

MAESTRO

index.html

```
<html>
<head>
<script type="text/javascript" src="message.js"></script>
</head>
<body>
<p>Welcome to JavaScript</p>
<form>
<input type="button" value="click" onclick="msg()"/>
</form>
</body>
</html>
```

Advantages of External JavaScript

There will be following benefits if a user creates an external javascript:

1. It helps in the reusability of code in more than one HTML file.
2. It allows easy code readability.
3. It is time-efficient as web browsers cache the external js files, which further reduces the page loading time.
4. It enables both web designers and coders to work with html and js files parallelly and separately, i.e., without facing any code conflictions.
5. The length of the code reduces as only we need to specify the location of the js file.

Disadvantages of External JavaScript

There are the following disadvantages of external files:

1. The stealer may download the coder's code using the url of the js file.
2. If two js files are dependent on one another, then a failure in one file may affect the execution of the other dependent file.
3. The web browser needs to make an additional http request to get the js code.

MAESTRO

4. A tiny to a large change in the js code may cause unexpected results in all its dependent files.
5. We need to check each file that depends on the commonly created external javascript file.
6. If it is a few lines of code, then better to implement the internal javascript code.

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.

Javascript Data Types

JavaScript provides different **data types** to hold different types of values. There are two types of data types in JavaScript.

1. Primitive data type
2. Non-primitive (reference) data type

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

```
1. var a=40;//holding number
2. var b="Rahul";//holding string
```

JavaScript primitive data types

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

Data Type	Description
String	represents sequence of characters e.g. "hello"
Number	represents numeric values e.g. 100

MAESTRO

Boolean	represents boolean value either false or true
Undefined	represents undefined value
Null	represents null i.e. no value at all

JavaScript non-primitive data types

The non-primitive data types are as follows:

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

JavaScript Operators

JavaScript operators are symbols that are used to perform operations on operands. For example:

1. var sum=10+20;

Here, + is the arithmetic operator and = is the assignment operator.

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 If-else

The **JavaScript if-else statement** is used *to execute the code whether condition is true or false*. There are three forms of if statement in JavaScript.

1. If Statement
2. If else statement
3. if else if statement

JavaScript If statement

It evaluates the content only if expression is true. The signature of JavaScript if statement is given below.

```
if(expression){
//content to be evaluated
}
```

JavaScript If...else Statement

It evaluates the content whether condition is true of false. The syntax of JavaScript if-else statement is given below

```
if(expression){
//content to be evaluated if condition is true
}
else{
//content to be evaluated if condition is false
}
```

JavaScript If...else if statement

It evaluates the content only if expression is true from several expressions. The signature of JavaScript if else if statement is given below.

```
if(expression1){
//content to be evaluated if expression1 is true
}
else if(expression2){
```

MAESTRO

```
//content to be evaluated if expression2 is true

}

else if(expression3){

//content to be evaluated if expression3 is true

}

else{

//content to be evaluated if no expression is true

}
```

JavaScript Switch

The **JavaScript switch statement** is used *to execute one code from multiple expressions*. It is just like else if statement that we have learned in previous page. But it is convenient than *if..else..if* because it can be used with numbers, characters etc.

The signature of JavaScript switch statement is given below.

```
switch(expression){

case value1:

 code to be executed;

 break;

case value2:

 code to be executed;

 break;

.....

default:

 code to be executed if above values are not matched;

}
```

JavaScript Loops

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

MAESTRO

There are four types of loops in JavaScript.

1. for loop
2. while loop
3. do-while loop
4. for-in loop

---

1) JavaScript For loop

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

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

2) JavaScript while loop

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

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

3) JavaScript do while loop

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

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

MAESTRO

4) JavaScript for in loop

The **JavaScript for in loop** is used *to iterate the properties of an object*.

JavaScript Functions

JavaScript functions are used to perform operations. We can call JavaScript function many times to reuse the code.

Advantage of JavaScript function

There are mainly two advantages of JavaScript functions.

1. **Code reusability**: We can call a function several times so it save coding.
2. **Less coding**: It makes our program compact. We don't need to write many lines of code each time to perform a common task.

JavaScript Function Syntax

The syntax of declaring function is given below.

```
function functionName([arg1, arg2, ...argN]){
 //code to be executed
}
```

JavaScript Function Methods

Let's see function methods with description.

Method	Description
apply()	It is used to call a function contains this value and a single array of arguments.
bind()	It is used to create a new function.
call()	It is used to call a function contains this value and an argument list.
toString()	It returns the result in a form of a string.

JAVAESTRO

What is xml?

- **Xml** (eXtensible Markup Language) is a markup language.
- XML is designed to store and transport data.
- Xml was released in late 90's. it was created to provide an easy to use and store self-describing data.
- XML became a W3C Recommendation on February 10, 1998.
- XML is not a replacement for HTML.
- XML is designed to be self-descriptive.
- XML is designed to carry data, not to display data.
- XML tags are not predefined. You must define your own tags.
- XML is platform independent and language independent.

What is mark-up language

A **markup language** is a modern system for highlight or underline a document.

Students often underline or highlight a passage to revise easily, same in the sense of modern markup language highlighting or underlining is replaced by tags.

Why xml

Platform Independent and Language Independent: The main benefit of xml is that you can use it to take data from a program like Microsoft SQL, convert it into XML then share that XML with other programs and platforms. You can communicate between two platforms which are generally very difficult.

Features and Advantages of XML

XML is widely used in the era of web development. It is also used to simplify data storage and data sharing.

The main features or advantages of XML are given below.

1) XML separates data from HTML

If you need to display dynamic data in your HTML document, it will take a lot of work to edit the HTML each time the data changes.

With XML, data can be stored in separate XML files. This way you can focus on using HTML/CSS for display and layout, and be sure that changes in the underlying data will not require any changes to the HTML.

With a few lines of JavaScript code, you can read an external XML file and update the data content of your web page.

2) XML simplifies data sharing

In the real world, computer systems and databases contain data in incompatible formats.

XML data is stored in plain text format. This provides a software- and hardware-independent way of storing data.

This makes it much easier to create data that can be shared by different applications.

3) XML simplifies data transport

One of the most time-consuming challenges for developers is to exchange data between incompatible systems over the Internet.

Exchanging data as XML greatly reduces this complexity, since the data can be read by different incompatible applications.

4) XML simplifies Platform change

Upgrading to new systems (hardware or software platforms), is always time consuming. Large amounts of data must be converted and incompatible data is often lost.

XML data is stored in text format. This makes it easier to expand or upgrade to new operating systems, new applications, or new browsers, without losing data.

5) XML increases data availability

Different applications can access your data, not only in HTML pages, but also from XML data sources.

With XML, your data can be available to all kinds of "reading machines" (Handheld computers, voice machines, news feeds, etc), and make it more available for blind people, or people with other disabilities.

6) XML can be used to create new internet languages

A lot of new Internet languages are created with XML.

MAESTRO

Here are some examples:

- **XHTML**
- **WSDL** for describing available web services
- **WAP** and **WML** as markup languages for handheld devices
- **RSS** languages for news feeds
- **RDF** and **OWL** for describing resources and ontology
- **SMIL** for describing multimedia for the web

XML documents must contain a **root element.** This element is "the parent" of all other elements.

The elements in an XML document form a document tree. The tree starts at the root and branches to the lowest level of the tree.

All elements can have sub elements (child elements).

```
<root>
 <child>
  <subchild>.....</subchild>
 </child>
</root>
```

The terms parent, child, and sibling are used to describe the relationships between elements. Parent elements have children. Children on the same level are called siblings (brothers or sisters).

All elements can have text content and attributes (just like in HTML).

XML Attributes

XML elements can have attributes. By the use of attributes we can add the information about the element.

XML attributes enhance the properties of the elements.

Why should we avoid XML ATTRIBUTES?

- Attributes cannot contain multiple values but child elements can have multiple values.
- Attributes cannot contain tree structure but child element can.

MAESTRO

- Attributes are not easily expandable. If you want to change in attribute's vales in future, it may be complicated.
- Attributes cannot describe structure but child elements can.
- Attributes are more difficult to be manipulated by program code.
- Attributes values are not easy to test against a DTD, which is used to define the legal elements of an XML document.

XML Tree Structure

An XML document has a self-descriptive structure. It forms a tree structure which is referred as an XML tree. The tree structure makes easy to describe an XML document.

A tree structure contains root element (as parent), child element and so on. It is very easy to traverse all succeeding branches and sub-branches and leaf nodes starting from the root.

Example of an XML document

```
<?xml version="1.0"?>

<college>

 <student>

   <firstname>Tamanna</firstname>

   <lastname>Bhatia</lastname>

   <contact>09990449935</contact>

   <email>tammanabhatia@abc.com</email>

   <address>

     <city>Ghaziabad</city>

     <state>Uttar Pradesh</state>

     <pin>201007</pin>

   </address>

 </student>

</college>
```

MAESTRO

XML Tree Rules

These rules are used to figure out the relationship of the elements. It shows if an element is a child or a parent of the other element.

Descendants: If element A is contained by element B, then A is known as descendant of B. In the above example "College" is the root element and all the other elements are the descendants of "College".

Ancestors: The containing element which contains other elements is called "Ancestor" of other element. In the above example Root element (College) is ancestor of all other elements.

XML Validation

A well formed XML document can be validated against DTD or Schema.

A well-formed XML document is an XML document with correct syntax. It is very necessary to know about valid XML document before knowing XML validation.

Valid XML document

It must be well formed (satisfy all the basic syntax condition)

It should be behave according to predefined DTD or XML schema

Rules for well formed XML

- It must begin with the XML declaration.
- It must have one unique root element.
- All start tags of XML documents must match end tags.
- XML tags are case sensitive.
- All elements must be closed.
- All elements must be properly nested.
- All attributes values must be quoted.
- XML entities must be used for special characters.

XML DTD

A DTD defines the legal elements of an XML document

In simple words we can say that a DTD defines the document structure with a list of legal elements and attributes.

XML schema is a XML based alternative to DTD.

MAESTRO

Actually DTD and XML schema both are used to form a well formed XML document.

We should avoid errors in XML documents because they will stop the XML programs.

XML schema

It is defined as an XML language

Uses namespaces to allow for reuses of existing definitions

It supports a large number of built in data types and definition of derived data types

XML DTD

What is DTD

DTD stands for **Document Type Definition**. It defines the legal building blocks of an XML document. It is used to define document structure with a list of legal elements and attributes.

Purpose of DTD

Its main purpose is to define the structure of an XML document. It contains a list of legal elements and define the structure with the help of them.

Checking Validation

Before proceeding with XML DTD, you must check the validation. An XML document is called "well-formed" if it contains the correct syntax.

A well-formed and valid XML document is one which have been validated against DTD.

Valid and well-formed XML document with DTD

Let's take an example of well-formed and valid XML document. It follows all the rules of DTD.

employee.xml

```
<?xml version="1.0"?>

<!DOCTYPE employee SYSTEM "employee.dtd">

<employee>

 <firstname>vimal</firstname>
```

MAESTRO

<lastname>jaiswal**</lastname>**

<email>vimal@javatpoint.com**</email>**

</employee>

In the above example, the DOCTYPE declaration refers to an external DTD file. The content of the file is shown in below paragraph.

employee.dtd

<!ELEMENT employee (firstname,lastname,email)**>**

<!ELEMENT firstname (#PCDATA)**>**

<!ELEMENT lastname (#PCDATA)**>**

<!ELEMENT email (#PCDATA)**>**

Description of DTD

<!DOCTYPE employee : It defines that the root element of the document is employee.

<!ELEMENT employee: It defines that the employee element contains 3 elements "firstname, lastname and email".

<!ELEMENT firstname: It defines that the firstname element is #PCDATA typed. (parse-able data type).

<!ELEMENT lastname: It defines that the lastname element is #PCDATA typed. (parse-able data type).

<!ELEMENT email: It defines that the email element is #PCDATA typed. (parse-able data type).

XML DTD with entity declaration

A doctype declaration can also define special strings that can be used in the XML file.

An entity has three parts:

1. An ampersand (&)
2. An entity name
3. A semicolon (;)

Syntax to declare entity:

MAESTRO

<!ENTITY entity-name "entity-value">

Let's see a code to define the ENTITY in doctype declaration.

author.xml

```
<?xml version="1.0" standalone="yes" ?>

<!DOCTYPE author [

 <!ELEMENT author (#PCDATA)>

 <!ENTITY sj "Sonoo Jaiswal">

]>

<author>&sj;</author>
```

XML CSS

Purpose of CSS in XML

CSS (Cascading Style Sheets) can be used to add style and display information to an XML document. It can format the whole XML document.

How to link XML file with CSS

To link XML files with CSS, you should use the following syntax:

```
<?xml-stylesheet type="text/css" href="cssemployee.css"?>
```

PHP

PHP is an open-source, interpreted, and object-oriented scripting language that can be executed at the server-side. PHP is well suited for web development. Therefore, it is used to develop web applications (an application that executes on the server and generates the dynamic page.).

PHP was created by Rasmus Lerdorf in 1994 but appeared in the market in 1995. PHP 7.4.0 is the latest version of PHP, which was released on 28 November. Some important points need to be noticed about PHP are as followed:

- PHP stands for Hypertext Preprocessor.
- PHP is an interpreted language, i.e., there is no need for compilation.

MAESTRO

- PHP is faster than other scripting languages, for example, ASP and JSP.
- PHP is a server-side scripting language, which is used to manage the dynamic content of the website.
- PHP can be embedded into HTML.
- PHP is an object-oriented language.
- PHP is an open-source scripting language.
- PHP is simple and easy to learn language.

PHP Features

PHP is very popular language because of its simplicity and open source. There are some important features of PHP given below:

Performance:

PHP script is executed much faster than those scripts which are written in other languages such as JSP and ASP. PHP uses its own memory, so the server workload and loading time is automatically reduced, which results in faster processing speed and better performance.

Open Source:

PHP source code and software are freely available on the web. You can develop all the versions of PHP according to your requirement without paying any cost. All its components are free to download and use.

Familiarity with syntax:

PHP has easily understandable syntax. Programmers are comfortable coding with it.

Embedded:

PHP code can be easily embedded within HTML tags and script.

Platform Independent:

PHP is available for WINDOWS, MAC, LINUX & UNIX operating system. A PHP application developed in one OS can be easily executed in other OS also.

Database Support:

PHP supports all the leading databases such as MySQL, SQLite, ODBC, etc.

MAESTRO

Error Reporting -

PHP has predefined error reporting constants to generate an error notice or warning at runtime. E.g., E_ERROR, E_WARNING, E_STRICT, E_PARSE.

Loosely Typed Language:

PHP allows us to use a variable without declaring its datatype. It will be taken automatically at the time of execution based on the type of data it contains on its value.

Web servers Support:

PHP is compatible with almost all local servers used today like Apache, Netscape, Microsoft IIS, etc.

Security:

PHP is a secure language to develop the website. It consists of multiple layers of security to prevent threads and malicious attacks.

Control:

Different programming languages require long script or code, whereas PHP can do the same work in a few lines of code. It has maximum control over the websites like you can make changes easily whenever you want.

A Helpful PHP Community:

It has a large community of developers who regularly updates documentation, tutorials, online help, and FAQs. Learning PHP from the communities is one of the significant benefits.

How to run PHP code in XAMPP

Generally, a PHP file contains HTML tags and some PHP scripting code. It is very easy to create a simple PHP example. To do so, create a file and write HTML tags + PHP code and save this file with .php extension.

Note: PHP statements ends with semicolon (;)

MAESTRO

All PHP code goes between the php tag. It starts with <?php and ends with ?>. The syntax of PHP tag is given below:

```
<?php

//your code here

?>
```

File: first.php

```
<!DOCTYPE>

<html>

<body>

<?php

echo "<h2>Hello First PHP</h2>";

?>

</body>

</html>
```

Output:

Hello First PHP

PHP Case Sensitivity

In PHP, keyword (e.g., echo, if, else, while), functions, user-defined functions, classes are not case-sensitive. However, all variable names are case-sensitive.

PHP Echo

PHP echo is a language construct, not a function. Therefore, you don't need to use parenthesis with it. But if you want to use more than one parameter, it is required to use parenthesis.

MAESTRO

The syntax of PHP echo is given below:

void echo (string $arg1 [, string $...])

PHP echo statement can be used to print the string, multi-line strings, escaping characters, variable, array, etc. Some important points that you must know about the echo statement are:

- Echo is a statement, which is used to display the output.
- Echo can be used with or without parentheses: echo(), and echo.
- Echo does not return any value.
- We can pass multiple strings separated by a comma (,) in echo.
- Echo is faster than the print statement.

PHP Print

Like PHP echo, PHP print is a language construct, so you don't need to use parenthesis with the argument list. Print statement can be used with or without parentheses: print and print(). Unlike echo, it always returns 1.

The syntax of PHP print is given below:

int print(string $arg)

PHP print statement can be used to print the string, multi-line strings, escaping characters, variable, array, etc. Some important points that you must know about the echo statement are:

- Print is a statement, used as an alternative to echo at many times to display the output.
- Print can be used with or without parentheses.
- Print always returns an integer value, which is 1.
- Using print, we cannot pass multiple arguments.
- Print is slower than the echo statement.

PHP echo and print Statements

We frequently use the echo statement to display the output. There are two basic ways to get the output in PHP:

- Echo
- print

Echo and print are language constructs, and they never behave like a function. Therefore, there is no requirement for parentheses. However, both the statements can be used with or without parentheses. We can use these statements to output variables or strings.

MAESTRO

Difference between echo and print

Echo

- Echo is a statement, which is used to display the output.
- Echo can be used with or without parentheses.
- Echo does not return any value.
- We can pass multiple strings separated by comma (,) in echo.
- Echo is faster than print statement.

Print

- Print is also a statement, used as an alternative to echo at many times to display the output.
- Print can be used with or without parentheses.
- Print always returns an integer value, which is 1.
- Using print, we cannot pass multiple arguments.
- Print is slower than echo statement.

PHP Variables

In PHP, a variable is declared using a $ sign followed by the variable name. Here, some important points to know about variables:

- As PHP is a loosely typed language, so we do not need to declare the data types of the variables. It automatically analyzes the values and makes conversions to its correct datatype.
- After declaring a variable, it can be reused throughout the code.
- Assignment Operator (=) is used to assign the value to a variable.

Syntax of declaring a variable in PHP is given below:

```
$variablename=value;
```

Rules for declaring PHP variable:

- A variable must start with a dollar ($) sign, followed by the variable name.
- It can only contain alpha-numeric character and underscore (A-z, 0-9, _).
- A variable name must start with a letter or underscore (_) character.
- A PHP variable name cannot contain spaces.
- One thing to be kept in mind that the variable name cannot start with a number or special symbols.

MAESTRO

- PHP variables are case-sensitive, so $name and $NAME both are treated as different variable.

PHP Variable Scope

The scope of a variable is defined as its range in the program under which it can be accessed. In other words, "The scope of a variable is the portion of the program within which it is defined and can be accessed."

PHP has three types of variable scopes:

1. Local variable
2. Global variable
3. Static variable

PHP $ and $$ Variables

The $var (single dollar) is a normal variable with the name var that stores any value like string, integer, float, etc.

The $$var (double dollar) is a reference variable that stores the value of the $variable inside it.

PHP Constants

PHP constants are name or identifier that can't be changed during the execution of the script except for magic constants, which are not really constants. PHP constants can be defined by 2 ways:

1. Using define() function
2. Using const keyword

Constants are similar to the variable except once they defined, they can never be undefined or changed. They remain constant across the entire program. PHP constants follow the same PHP variable rules. For example, it can be started with a letter or underscore only.

PHP constant: define()

Use the define() function to create a constant. It defines constant at run time. Let's see the syntax of define() function in PHP.

```
define(name, value, case-insensitive)
```

1. Name: It specifies the constant name.

2. Value: It specifies the constant value.
3. Case-insensitive: Specifies whether a constant is case-insensitive. Default value is false. It means it is case sensitive by default.

PHP constant: const keyword

PHP introduced a keyword const to create a constant. The const keyword defines constants at compile time. It is a language construct, not a function. The constant defined using const keyword are case-sensitive.

PHP Data Types

PHP data types are used to hold different types of data or values. PHP supports 8 primitive data types that can be categorized further in 3 types:

1. Scalar Types (predefined)
2. Compound Types (user-defined)
3. Special Types

PHP Data Types: Scalar Types

It holds only single value. There are 4 scalar data types in PHP.

1. boolean
2. integer
3. float
4. string

PHP Data Types: Compound Types

It can hold multiple values. There are 2 compound data types in PHP.

1. array
2. object

PHP Data Types: Special Types

There are 2 special data types in PHP.

1. resource
2. NULL

MAESTRO

PHP Operators

PHP Operator is a symbol i.e used to perform operations on operands. In simple words, operators are used to perform operations on variables or values.

- Arithmetic Operators
- Assignment Operators
- Bitwise Operators
- Comparison Operators
- Incrementing/Decrementing Operators
- Logical Operators
- String Operators
- Array Operators
- Type Operators
- Execution Operators
- Error Control Operators

We can also categorize operators on behalf of operands. They can be categorized in 3 forms:

- Unary Operators: works on single operands such as ++, -- etc.
- Binary Operators: works on two operands such as binary +, -, *, / etc.
- Ternary Operators: works on three operands such as "?:".

PHP Comments

PHP comments can be used to describe any line of code so that other developer can understand the code easily. It can also be used to hide any code.

PHP supports single line and multi-line comments. These comments are similar to C/C++ and Perl style (UNIX shell style) comments.

PHP Single Line Comments

There are two ways to use single line comments in PHP.

- // (C++ style single line comment)
- # (Unix Shell style single line comment)

PHP Multi Line Comments

In PHP, we can comments multiple lines also. To do so, we need to enclose all lines within /* */.

PHP If Else

PHP if else statement is used to test condition. There are various ways to use if statement in PHP.

- if
- if-else
- if-else-if
- nested if

PHP Switch

PHP switch statement is used to execute one statement from multiple conditions. It works like PHP if-else-if statement.

Syntax

```
switch(expression){

case value1:

 //code to be executed

 break;

case value2:

 //code to be executed

 break;

.....

default:

 code to be executed if all cases are not matched;

}
```

MAESTRO

Important points to be noticed about switch case:

1. The default is an optional statement. Even it is not important, that default must always be the last statement.
2. There can be only one default in a switch statement. More than one default may lead to a Fatal error.
3. Each case can have a break statement, which is used to terminate the sequence of statement.
4. The break statement is optional to use in switch. If break is not used, all the statements will execute after finding matched case value.
5. PHP allows you to use number, character, string, as well as functions in switch expression.
6. Nesting of switch statements is allowed, but it makes the program more complex and less readable.
7. You can use semicolon (;) instead of colon (:). It will not generate any error.

PHP For Loop

PHP for loop can be used to traverse set of code for the specified number of times.

It should be used if the number of iterations is known otherwise use while loop. This means for loop is used when you already know how many times you want to execute a block of code.

It allows users to put all the loop related statements in one place. See in the syntax given below:

Syntax

```
for(initialization; condition; increment/decrement)

{

//code to be executed

}
```

Parameters

The php for loop is similar to the java/C/C++ for loop. The parameters of for loop have the following meanings:

Initialization - Initialize the loop counter value. The initial value of the for loop is done only once. This parameter is optional.

MAESTRO

Condition - Evaluate each iteration value. The loop continuously executes until the condition is false. If TRUE, the loop execution continues, otherwise the execution of the loop ends.

Increment/decrement - It increments or decrements the value of the variable.

PHP foreach loop

The foreach loop is used to traverse the array elements. It works only on array and object. It will issue an error if you try to use it with the variables of different datatype.

The foreach loop works on elements basis rather than index. It provides an easiest way to iterate the elements of an array.

In foreach loop, we don't need to increment the value.

```
foreach ($array as $value)

 {

   //code to be executed

}
```

There is one more syntax of foreach loop.

Syntax

```
foreach ($array as $key => $element) {

   //code to be executed

}
```

PHP Functions

PHP function is a piece of code that can be reused many times. It can take input as argument list and return value. There are thousands of built-in functions in PHP.

In PHP, we can define Conditional function, Function within Function and Recursive function also.

Advantage of PHP Functions

Code Reusability: PHP functions are defined only once and can be invoked many times, like in other programming languages.

MAESTRO

Less Code: It saves a lot of code because you don't need to write the logic many times. By the use of function, you can write the logic only once and reuse it.

Easy to understand: PHP functions separate the programming logic. So it is easier to understand the flow of the application because every logic is divided in the form of functions.

PHP User-defined Functions

We can declare and call user-defined functions easily. Let's see the syntax to declare user-defined functions.

Syntax

```
function functionname(){

//code to be executed

}
```

Note: Function name must be start with letter and underscore only like other labels in PHP. It can't be start with numbers or special symbols.

PHP Function Arguments

We can pass the information in PHP function through arguments which is separated by comma.

PHP supports **Call by Value** (default), **Call by Reference**, **Default argument values** and **Variable-length argument list**.

PHP Call By Reference

Value passed to the function doesn't modify the actual value by default (call by value). But we can do so by passing value as a reference.

By default, value passed to the function is call by value. To pass value as a reference, you need to use ampersand (&) symbol before the argument name.

PHP Parameterized Function

PHP Parameterized functions are the functions with parameters. You can pass any number of parameters inside a function. These passed parameters act as variables inside your function.

They are specified inside the parentheses, after the function name.

MAESTRO

PHP Call By Value

PHP allows you to call function by value and reference both. In case of PHP call by value, actual value is not modified if it is modified inside the function.

PHP Call By Reference

In case of PHP call by reference, actual value is modified if it is modified inside the function. In such case, you need to use & (ampersand) symbol with formal arguments. The & represents reference of the variable.

PHP Recursive Function

PHP also supports recursive function call like C/C++. In such case, we call current function within function. It is also known as recursion.

PHP Arrays

PHP array is an ordered map (contains value on the basis of key). It is used to hold multiple values of similar type in a single variable.

Advantage of PHP Array

Less Code: We don't need to define multiple variables.

Easy to traverse: By the help of single loop, we can traverse all the elements of an array.

Sorting: We can sort the elements of array.

PHP Array Types

There are 3 types of array in PHP.

1. Indexed Array
2. Associative Array
3. Multidimensional Array

PHP Indexed Array

PHP index is represented by number which starts from 0. We can store number, string and object in the PHP array. All PHP array elements are assigned to an index number by default.

There are two ways to define indexed array:

1st way:

```
1. $season=array("summer","winter","spring","autumn");
```

MAESTRO

2nd way:

```
1. $season[0]="summer";
2. $season[1]="winter";
3. $season[2]="spring";
4. $season[3]="autumn";
```

PHP Associative Array

We can associate name with each array elements in PHP using => symbol.

There are two ways to define associative array:

1st way:

```
1. $salary=array("Sonoo"=>"350000","John"=>"450000","Kartik"=>"200000");
```

2nd way:

```
1. $salary["Sonoo"]="350000";
2. $salary["John"]="450000";
3. $salary["Kartik"]="200000";
```

PHP Indexed Array

PHP indexed array is an array which is represented by an index number by default. All elements of array are represented by an index number which starts from 0.

PHP indexed array can store numbers, strings or any object. PHP indexed array is also known as numeric array.

Definition

There are two ways to define indexed array:

1st way:

```
$size=array("Big","Medium");
```

2nd way:

```
$size[0]="Big";
$size[1]="Medium";
```

MAESTRO

PHP Associative Array

PHP allows you to associate name/label with each array elements in PHP using => symbol. Such way, you can easily remember the element because each element is represented by label than an incremented number.

Definition

There are two ways to define associative array:

1st way:

```
$salary=array("Sonoo"=>"550000","Vimal"=>"250000","Ratan"=>"200000");
```

2nd way:

```
$salary["Sonoo"]="550000";
$salary["Vimal"]="250000";
$salary["Ratan"]="200000";
```

PHP Multidimensional Array

PHP multidimensional array is also known as array of arrays. It allows you to store tabular data in an array. PHP multidimensional array can be represented in the form of matrix which is represented by row * column.

Definition

```
$emp = array
 (
 array(1,"sonoo",400000),
 array(2,"john",500000),
 array(3,"rahul",300000)
 );
```

PHP Array Functions

PHP provides various array functions to access and manipulate the elements of array. The important PHP array functions are given below.

MAESTRO

1) PHP array() function

PHP array() function creates and returns an array. It allows you to create indexed, associative and multidimensional arrays.

Syntax

array array ([mixed $...])

2) PHP array_change_key_case() function

PHP array_change_key_case() function changes the case of all key of an array.

Note: It changes case of key only.

Syntax

array array_change_key_case (**array** $array [, int $case = CASE_LOWER])

3) PHP array_chunk() function

PHP array_chunk() function splits array into chunks. By using array_chunk() method, you can divide array into many parts.

Syntax

array array_chunk (**array** $array , int $size [, bool $preserve_keys = false])

4) PHP count() function

PHP count() function counts all elements in an array.

Syntax

Int count (mixed $array_or_countable [, int $mode = COUNT_NORMAL])

5) PHP sort() function

PHP sort() function sorts all the elements in an array.

Syntax

bool sort (**array** &$array [, int $sort_flags = SORT_REGULAR])

MAESTRO

PHP String

PHP string is a sequence of characters i.e., used to store and manipulate text. PHP supports only 256-character set and so that it does not offer native Unicode support. There are 4 ways to specify a string literal in PHP.

1. single quoted
2. double quoted
3. heredoc syntax
4. newdoc syntax (since PHP 5.3)

PHP Form Handling

We can create and use forms in PHP. To get form data, we need to use PHP superglobals $_GET and $_POST.

The form request may be get or post. To retrieve data from get request, we need to use $_GET, for post request $_POST.

PHP Get Form

Get request is the default form request. The data passed through get request is visible on the URL browser so it is not secured. You can send limited amount of data through get request.

PHP Post Form

Post request is widely used to submit form that have large amount of data such as file upload, image upload, login form, registration form etc.

The data passed through post request is not visible on the URL browser so it is secured. You can send large amount of data through post request.

MAESTRO

PHP Include and Require

PHP allows us to create various elements and functions, which are used several times in many pages. It takes much time to script these functions in multiple pages. Therefore, use the concept of **file inclusion** that helps to include files in various programs and saves the effort of writing code multiple times.

"PHP allows you to include file so that a page content can be reused many times. It is very helpful to include files when you want to apply the same HTML or PHP code to multiple pages of a website." There are two ways to include file in PHP.

1. include
2. require

Both include and require are identical to each other, except failure.

- **include** only generates a warning, i.e., E_WARNING, and continue the execution of the script.
- **require** generates a fatal error, i.e., E_COMPILE_ERROR, and stop the execution of the script.

Advantage

Code Reusability: By the help of include and require construct, we can reuse HTML code or PHP script in many PHP scripts.

Easy editable: If we want to change anything in webpages, edit the source file included in all webpage rather than editing in all the files separately.

PHP include

PHP include is used to include a file on the basis of given path. You may use a relative or absolute path of the file.

Syntax

There are two syntaxes available for include:

include 'filename ';

Or

include ('filename');

MAESTRO

PHP require

PHP require is similar to include, which is also used to include files. The only difference is that it stops the execution of script if the file is not found whereas include doesn't.

Syntax

There are two syntaxes available for require:

require 'filename';

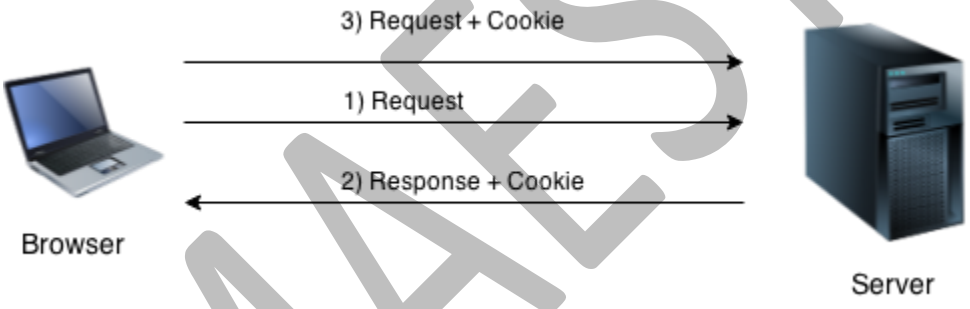
Or

require ('filename');

PHP Cookie

PHP cookie is a small piece of information which is stored at client browser. It is used to recognize the user.

Cookie is created at server side and saved to client browser. Each time when client sends request to the server, cookie is embedded with request. Such way, cookie can be received at the server side.



In short, cookie can be created, sent and received at server end.

Note: PHP Cookie must be used before <html> tag.

PHP setcookie() function

PHP setcookie() function is used to set cookie with HTTP response. Once cookie is set, you can access it by $_COOKIE superglobal variable.

Syntax

bool setcookie (string $name [, string $value [, int $expire = 0 [, string $path

[, string $domain [, bool $secure = false [, bool $httponly = false]]]]])

MAESTRO

PHP $_COOKIE

PHP $_COOKIE superglobal variable is used to get cookie.

Example

```
$value=$_COOKIE["CookieName"];//returns cookie value
```

PHP Delete Cookie

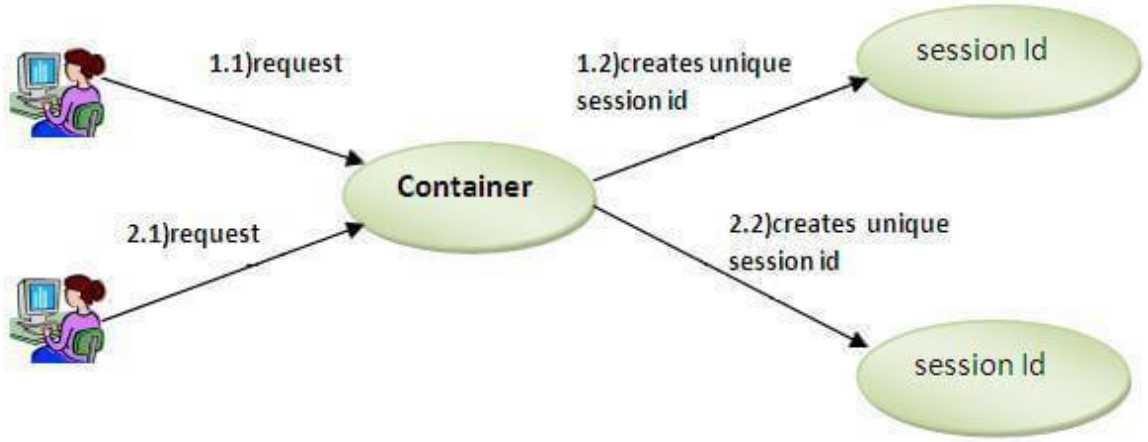
If you set the expiration date in past, cookie will be deleted.

PHP Session

PHP session is used to store and pass information from one page to another temporarily (until user close the website).

PHP session technique is widely used in shopping websites where we need to store and pass cart information e.g. username, product code, product name, product price etc from one page to another.

PHP session creates unique user id for each browser to recognize the user and avoid conflict between multiple browsers.



PHP session_start() function

PHP session_start() function is used to start the session. It starts a new or resumes existing session. It returns existing session if session is created already. If session is not available, it creates and returns new session.

Syntax

```
bool session_start ( void )
```

Example

```
session_start();
```

PHP $_SESSION

PHP $_SESSION is an associative array that contains all session variables. It is used to set and get session variable values.

PHP File Handling

PHP File System allows us to create file, read file line by line, read file character by character, write file, append file, delete file and close file.

PHP Open File - fopen()

The PHP fopen() function is used to open a file.

Syntax

Example

```
<?php
$handle = fopen("c:\\folder\\file.txt", "r");
?>
```

PHP Close File - fclose()

The PHP fclose() function is used to close an open file pointer.

Syntax

Example

```
<?php
fclose($handle);
?>
```

PHP Read File - fread()

The PHP fread() function is used to read the content of the file. It accepts two arguments: resource and file size.

Syntax

```
string fread ( resource $handle , int $length )
```

MAESTRO

PHP Write File - fwrite()

The PHP fwrite() function is used to write content of the string into file.

Syntax

int fwrite (resource $handle , string $string [, int $length])

JSP (JAVA SERVER PAGE)



JSP technology is used to create web application just like Servlet technology. It can be thought of as an extension to Servlet because it provides more functionality than servlet such as expression language, JSTL, etc.

A JSP page consists of HTML tags and JSP tags. The JSP pages are easier to maintain than Servlet because we can separate designing and development. It provides some additional features such as Expression Language, Custom Tags, etc.

Advantages of JSP over Servlet

There are many advantages of JSP over the Servlet. They are as follows:

1) Extension to Servlet

JSP technology is the extension to Servlet technology. We can use all the features of the Servlet in JSP. In addition to, we can use implicit objects, predefined tags, expression language and Custom tags in JSP, that makes JSP development easy.

2) Easy to maintain

JSP can be easily managed because we can easily separate our business logic with presentation logic. In Servlet technology, we mix our business logic with the presentation logic.

3) Fast Development: No need to recompile and redeploy

If JSP page is modified, we don't need to recompile and redeploy the project. The Servlet code needs to be updated and recompiled if we have to change the look and feel of the application.

MAESTRO

4) Less code than Servlet

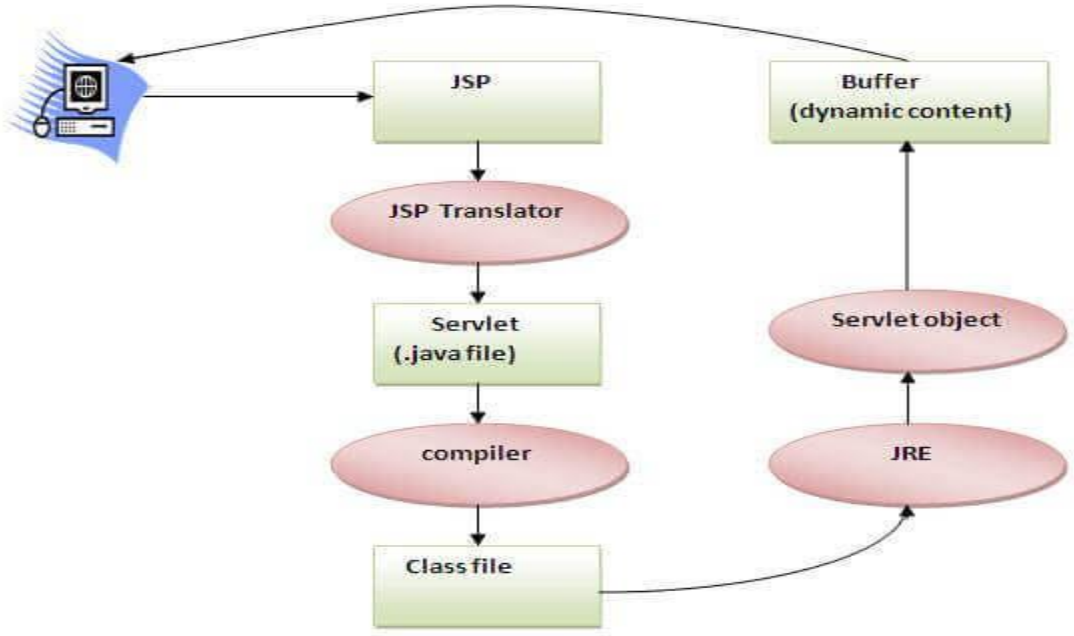
In JSP, we can use many tags such as action tags, JSTL, custom tags, etc. that reduces the code. Moreover, we can use EL, implicit objects, etc.

The Lifecycle of a JSP Page

The JSP pages follow these phases:

- Translation of JSP Page
- Compilation of JSP Page
- Classloading (the classloader loads class file)
- Instantiation (Object of the Generated Servlet is created).
- Initialization (the container invokes jspInit() method).
- Request processing (the container invokes _jspService() method).
- Destroy (the container invokes jspDestroy() method).

Note: jspInit(), _jspService() and jspDestroy() are the life cycle methods of JSP.

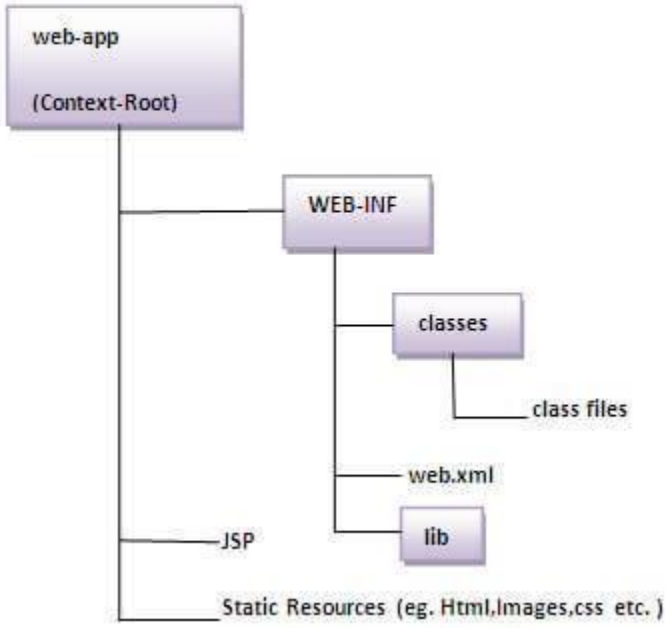


Creating a simple JSP Page

To create the first JSP page, write some HTML code as given below, and save it by .jsp extension. We have saved this file as index.jsp. Put it in a folder and paste the folder in the web-apps directory in apache tomcat to run the JSP page.

The Directory structure of JSP

The directory structure of JSP page is same as Servlet. We contain the JSP page outside the WEB-INF folder or in any directory.



JSP API

The JSP API consists of two packages:

1. javax.servlet.jsp
2. javax.servlet.jsp.tagext

javax.servlet.jsp package

The javax.servlet.jsp package has two interfaces and classes.The two interfaces are as follows:

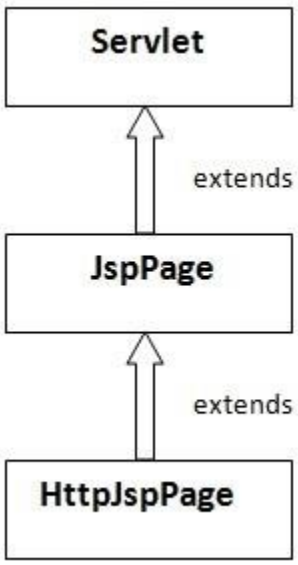
1. JspPage
2. HttpJspPage

The classes are as follows:

- JspWriter
- PageContext
- JspFactory
- JspEngineInfo
- JspException
- JspError

The JspPage interface

According to the JSP specification, all the generated servlet classes must implement the JspPage interface. It extends the Servlet interface. It provides two life cycle methods.



Methods of JspPage interface

1. **public void jspInit():** It is invoked only once during the life cycle of the JSP when JSP page is requested firstly. It is used to perform initialization. It is same as the init() method of Servlet interface.

2. **public void jspDestroy():** It is invoked only once during the life cycle of the JSP before the JSP page is destroyed. It can be used to perform some clean up operation.

W.AESTRO

The HttpJspPage interface

The HttpJspPage interface provides the one life cycle method of JSP. It extends the JspPage interface.

Method of HttpJspPage interface:

public void _jspService(): It is invoked each time when request for the JSP page comes to the container. It is used to process the request. The underscore _ signifies that you cannot override this method.

JSP Scriptlet tag (Scripting elements)

In JSP, java code can be written inside the jsp page using the scriptlet tag. Let's see what are the scripting elements first.

JSP Scripting elements

The scripting elements provides the ability to insert java code inside the jsp. There are three types of scripting elements:

- scriptlet tag
- expression tag
- declaration tag

JSP scriptlet tag

A scriptlet tag is used to execute java source code in JSP. Syntax is as follows:

<% java source code %>

JSP expression tag

The code placed within **JSP expression tag** is written to the output stream of the response. So you need not write out.print() to write data. It is mainly used to print the values of variable or method.

Syntax of JSP expression tag

<%= statement **%>**

JSP Declaration Tag

The **JSP declaration tag** is used *to declare fields and methods*.

The code written inside the jsp declaration tag is placed outside the service() method of auto generated servlet.

So it doesn't get memory at each request.

MAESTRO

Syntax of JSP declaration tag

The syntax of the declaration tag is as follows:

<%! field or method declaration %**>**

Difference between JSP Scriptlet tag and Declaration tag

Jsp Scriptlet Tag	**Jsp Declaration Tag**
The jsp scriptlet tag can only declare variables not methods.	The jsp declaration tag can declare variables as well as methods.
The declaration of scriptlet tag is placed inside the _jspService() method.	The declaration of jsp declaration tag is placed outside the _jspService() method.

JSP Implicit Objects

There are **9 jsp implicit objects**. These objects are *created by the web container* that are available to all the jsp pages.

The available implicit objects are out, request, config, session, application etc.

A list of the 9 implicit objects is given below

Object	**Type**
out	JspWriter
request	HttpServletRequest
response	HttpServletResponse
config	ServletConfig
application	ServletContext

session	HttpSession
pageContext	PageContext
page	Object
exception	Throwable

JSP directives

The **jsp directives** are messages that tells the web container how to translate a JSP page into the corresponding servlet.

There are three types of directives:

- page directive
- include directive
- taglib directive

Syntax of JSP Directive

<%@ directive attribute="value" %>

JSP page directive

The page directive defines attributes that apply to an entire JSP page.

Syntax of JSP page directive

<%@ page attribute="value" %>

Attributes of JSP page directive

- import
- contentType
- extends
- info
- buffer

MAESTRO

- language
- isELIgnored
- isThreadSafe
- autoFlush
- session
- pageEncoding
- errorPage
- isErrorPage

The include directive is used to include the contents of any resource it may be jsp file, html file or text file. The include directive includes the original content of the included resource at page translation time (the jsp page is translated only once so it will be better to include static resource).

Advantage of Include directive

Code Reusability

Syntax of include directive

```
<%@ include file="resourceName" %>
```

JSP Taglib directive

The JSP taglib directive is used to define a tag library that defines many tags. We use the TLD (Tag Library Descriptor) file to define the tags. In the custom tag section we will use this tag so it will be better to learn it in custom tag.

Syntax JSP Taglib directive

```
<%@ taglib uri="uriofthetaglibrary" prefix="prefixoftaglibrary" %>
```

Servlets

Servlet technology is used to create a web application (resides at server side and generates a dynamic web page).

Servlet technology is robust and scalable because of java language. Before Servlet, CGI (Common Gateway Interface) scripting language was common as a server-side programming language. However, there were many disadvantages to this technology. We have discussed these disadvantages below.

There are many interfaces and classes in the Servlet API such as Servlet, GenericServlet, HttpServlet, ServletRequest, ServletResponse, etc.

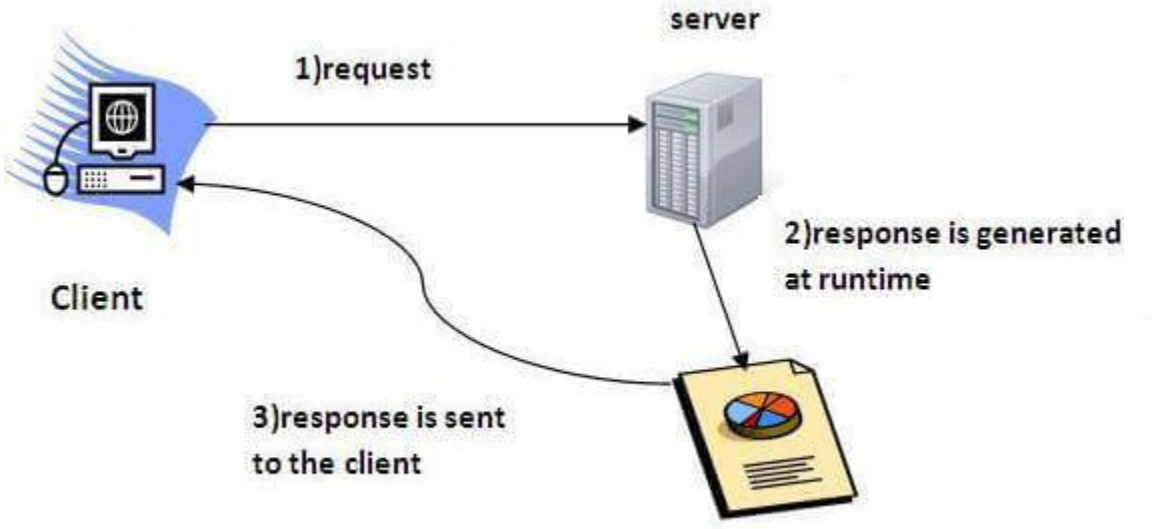
MAESTRO



What is a Servlet?

Servlet can be described in many ways, depending on the context.

- Servlet is a technology which is used to create a web application.
- Servlet is an API that provides many interfaces and classes including documentation.
- Servlet is an interface that must be implemented for creating any Servlet.
- Servlet is a class that extends the capabilities of the servers and responds to the incoming requests. It can respond to any requests.
- Servlet is a web component that is deployed on the server to create a dynamic web page



MAESTRO

What is a web application?

A web application is an application accessible from the web. A web application is composed of web components like Servlet, JSP, Filter, etc. and other elements such as HTML, CSS, and JavaScript. The web components typically execute in Web Server and respond to the HTTP request.

Servlet API

The javax.servlet and javax.servlet.http packages represent interfaces and classes for servlet api.

The **javax.servlet** package contains many interfaces and classes that are used by the servlet or web container. These are not specific to any protocol.

The **javax.servlet.http** package contains interfaces and classes that are responsible for http requests only.

Interfaces in javax.servlet package

There are many interfaces in javax.servlet package. They are as follows:

1. Servlet
2. ServletRequest
3. ServletResponse
4. RequestDispatcher
5. ServletConfig
6. ServletContext
7. SingleThreadModel
8. Filter
9. FilterConfig
10. FilterChain
11. ServletRequestListener
12. ServletRequestAttributeListener
13. ServletContextListener
14. ServletContextAttributeListener

MAESTRO

Servlet Interface

Servlet interface provides commonbehaviorto all the servlets.Servlet interface defines methods that all servlets must implement.

Servlet interface needs to be implemented for creating any servlet (either directly or indirectly). It provides 3 life cycle methods that are used to initialize the servlet, to service the requests, and to destroy the servlet and 2 non-life cycle methods.

Methods of Servlet interface

There are 5 methods in Servlet interface. The init, service and destroy are the life cycle methods of servlet. These are invoked by the web container.

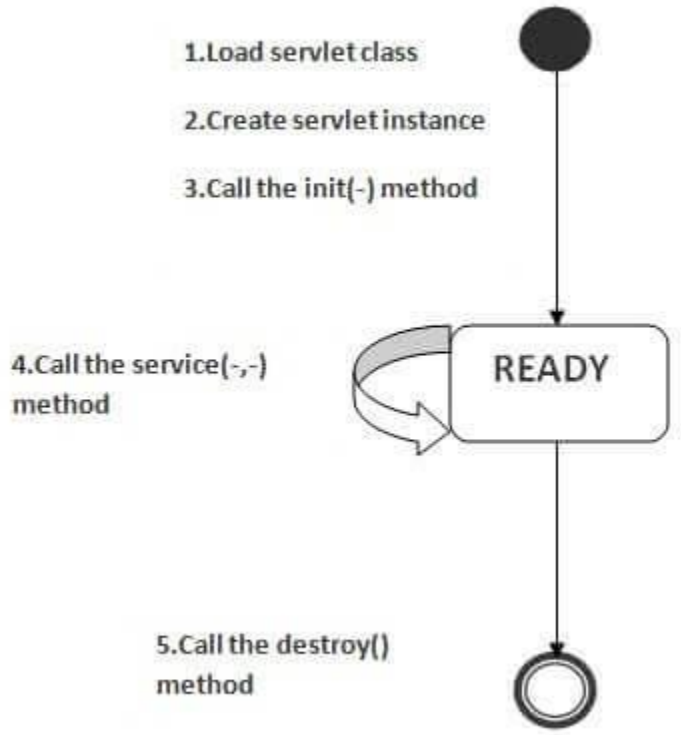
Method	Description
public void init(ServletConfig config)	initializes the servlet. It is the life cycle method of servlet and invoked by the web container only once.
public void service(ServletRequest request,ServletResponse response)	provides response for the incoming request. It is invoked at each request by the web container.
public void destroy()	is invoked only once and indicates that servlet is being destroyed.
public ServletConfig getServletConfig()	returns the object of ServletConfig.
public String getServletInfo()	returns information about servlet such as writer, copyright, version etc.

Life Cycle of a Servlet (Servlet Life Cycle)

The web container maintains the life cycle of a servlet instance. Let's see the life cycle of the servlet:

1. Servlet class is loaded.
2. Servlet instance is created.
3. init method is invoked.

4. service method is invoked.
5. destroy method is invoked.



As displayed in the above diagram, there are three states of a servlet: new, ready and end. The servlet is in new state if servlet instance is created. After invoking the init() method, Servlet comes in the ready state. In the ready state, servlet performs all the tasks. When the web container invokes the destroy() method, it shifts to the end state.

ServletRequest Interface

An object of ServletRequest is used to provide the client request information to a servlet such as content type, content length, parameter names and values, header informations, attributes etc.

Methods of ServletRequest interface

There are many methods defined in the ServletRequest interface. Some of them are as follows:

Method	Description
public String getParameter(String name)	is used to obtain the value of a parameter by name.
public String[] getParameterValues(String name)	returns an array of String containing all values of given parameter name. It is mainly used to obtain values of a Multi select list box.
java.util.Enumeration getParameterNames()	returns an enumeration of all of the request parameter names.
public int getContentLength()	Returns the size of the request entity data, or -1 if not known.
public String getCharacterEncoding()	Returns the character set encoding for the input of this request.
public String getContentType()	Returns the Internet Media Type of the request entity data, or null if not known.
public ServletInputStream getInputStream() throws IOException	Returns an input stream for reading binary data in the request body.
public abstract String getServerName()	Returns the host name of the server that received the request.
public int getServerPort()	Returns the port number on which this request was received.

How Servlet works?

It is important to learn how servlet works for understanding the servlet well. Here, we are going to get the internal detail about the first servlet program.

The server checks if the servlet is requested **for the first time**.

If yes, web container does the following tasks:

- loads the servlet class.
- instantiates the servlet class.
- calls the init method passing the ServletConfig object

else

- calls the service method passing request and response objects

The web container calls the destroy method when it needs to remove the servlet such as at time of stopping server or undelaying the project.

How web container handles the servlet request?

The web container is responsible to handle the request. Let's see how it handles the request.

- maps the request with the servlet in the web.xml file.
- creates request and response objects for this request
- calls the service method on the thread
- The public service method internally calls the protected service method
- The protected service method calls the doGet method depending on the type of request.
- The doGet method generates the response and it is passed to the client.
- After sending the response, the web container deletes the request and response objects. The thread is contained in the thread pool or deleted depends on the server implementation

Event and Listener in Servlet

Events are basically occurrence of something. Changing the state of an object is known as an event.

We can perform some important tasks at the occurrence of these exceptions, such as counting total and current logged-in users, creating tables of the database at time of deploying the project, creating database connection object etc.

There are many Event classes and Listener interfaces in the javax.servlet and javax.servlet.http packages.

Event classes

The event classes are as follows:

1. ServletRequestEvent
2. ServletContextEvent

MAESTRO

3. ServletRequestAttributeEvent
4. ServletContextAttributeEvent
5. HttpSessionEvent
6. HttpSessionBindingEvent

Event interfaces

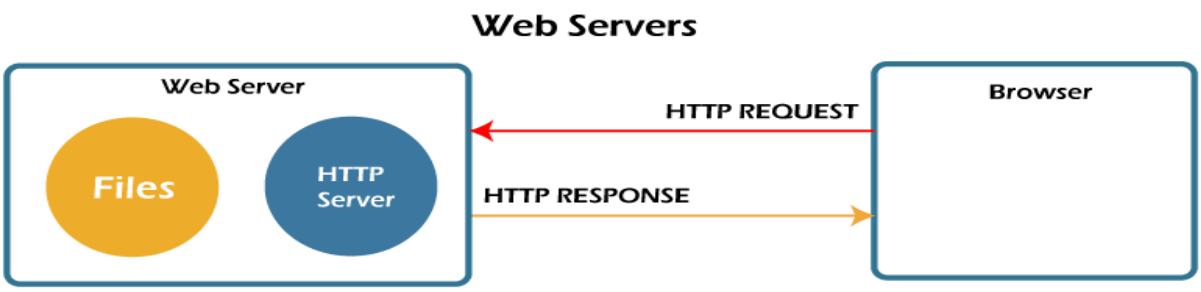
The event interfaces are as follows:

1. ServletRequestListener
2. ServletRequestAttributeListener
3. ServletContextListener
4. ServletContextAttributeListener
5. HttpSessionListener
6. HttpSessionAttributeListener
7. HttpSessionBindingListener
8. HttpSessionActivationListener

Web Servers

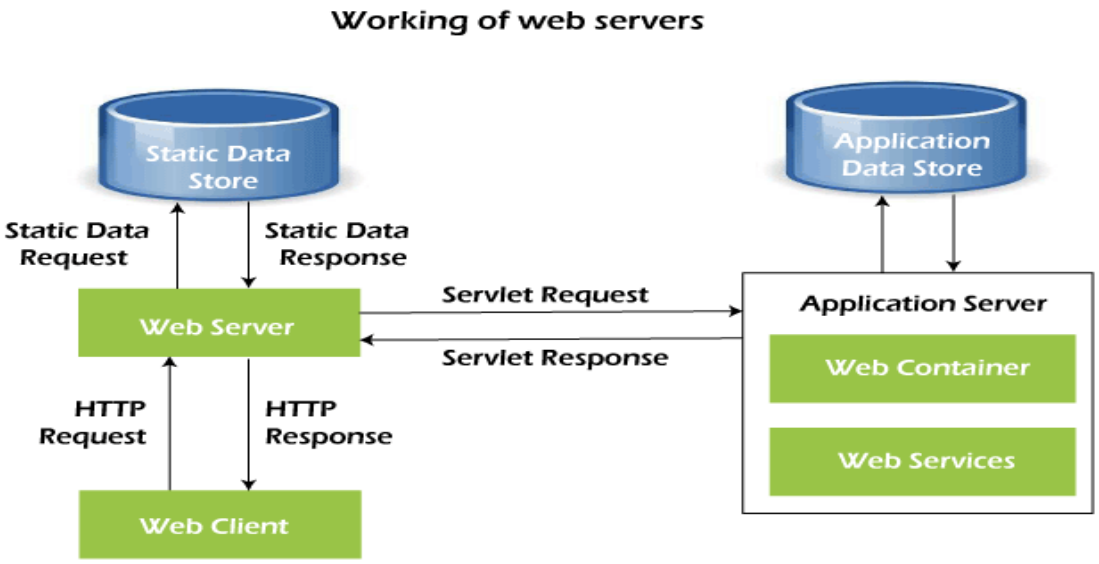
Web pages are a collection of data, including images, text files, hyperlinks, database files etc., all located on some computer (also known as server space) on the Internet. A web server is dedicated software that runs on the server-side. When any user requests their web browser to run any web page, the webserver places all the data materials together into an organized web page and forwards them back to the web browser with the help of the Internet. Therefore, we can conclude that: -

"A web server is a dedicated computer responsible for running websites sitting out on those computers somewhere on the Internet. They are specialized programs that circulate web pages as summoned by the user. The primary objective of any web server is to collect, process and provide web pages to the users."



How do web servers work?

The term web server can denote server hardware or server software, or in most cases, both hardware and software might be working together.



1. ***On the hardware side***, a web server is defined as a computer that stores software and another website raw data, such as HTML files, images, text documents, and JavaScript files. The hardware of the web servers are connected to the web and supports the data exchange with different devices connected to the Internet.
2. ***On the software side***, a web server includes server software accessed through website domain names. It controls how web users access the web files and ensures the supply of website content to the end-user. The web server contains several components, including an HTTP server.

Web server security Methods

Though there are various security techniques available in the market that a user can implement to have a safe web server experience, below given are some examples of security practices that can include processes:

1. **A reverse proxy** is a proxy server that is accessible to the clients, therefore hiding the internal server. It acts as an intermediary as wherever any user makes requests to the web server for any data or file, the proxy server seizes those requests and then communicates with the webserver.
2. **Access restriction** is a technique that limits the web host's access to infrastructure machines or using Secure Socket Shell (SSH);
3. **Keeping web servers mended and updated**, as it benefits to ensure the web server isn't vulnerable to exposures;

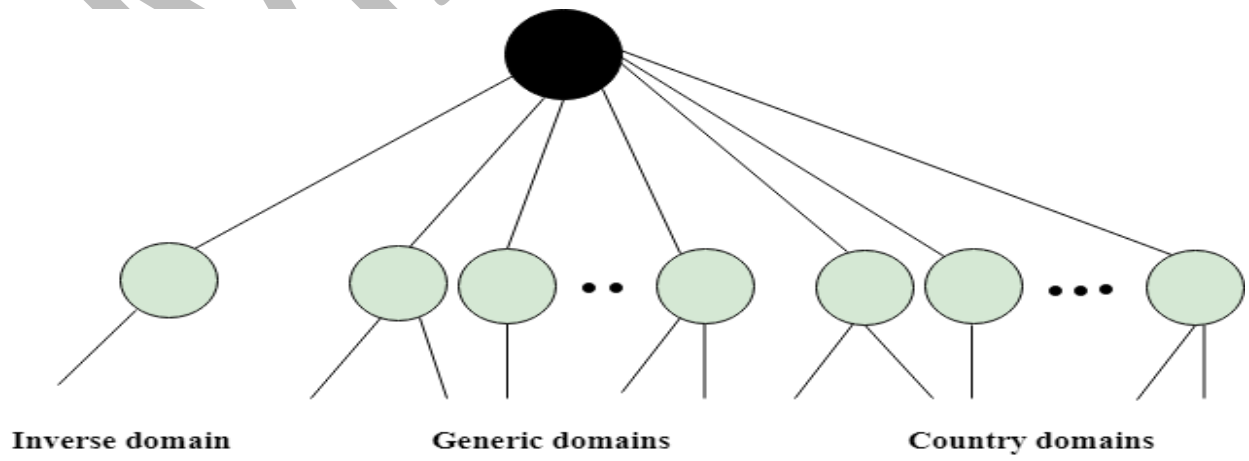
4. **Network monitoring** is a security practice that ensures that no unauthorized activity is going on the web server; and
5. **Using a firewall and SSL safeguards** the web server as firewalls can supervise HTTP request traffic while a Secure Sockets Layer (SSL) supports securing the data.

DNS

An application layer protocol defines how the application processes running on different systems, pass the messages to each other.

- DNS stands for Domain Name System.
- DNS is a directory service that provides a mapping between the name of a host on the network and its numerical address.
- DNS is required for the functioning of the internet.
- Each node in a tree has a domain name, and a full domain name is a sequence of symbols specified by dots.
- DNS is a service that translates the domain name into IP addresses. This allows the users of networks to utilize user-friendly names when looking for other hosts instead of remembering the IP addresses.
- For example, suppose the FTP site at EduSoft had an IP address of 132.147.165.50, most people would reach this site by specifying ftp.EduSoft.com. Therefore, the domain name is more reliable than IP address.

DNS is a TCP/IP protocol used on different platforms. The domain name space is divided into three different sections: generic domains, country domains, and inverse domain.



Country Domain

The format of country domain is same as a generic domain, but it uses two-character country abbreviations (e.g., us for the United States) in place of three character organizational abbreviations.

Inverse Domain

The inverse domain is used for mapping an address to a name. When the server has received a request from the client, and the server contains the files of only authorized clients. To determine whether the client is on the authorized list or not, it sends a query to the DNS server and ask for mapping an address to the name.

Working of DNS

- DNS is a client/server network communication protocol. DNS clients send requests to the. server while DNS servers send responses to the client.
- Client requests contain a name which is converted into an IP address known as a forward DNS lookups while requests containing an IP address which is converted into a name known as reverse DNS lookups.
- DNS implements a distributed database to store the name of all the hosts available on the internet.
- If a client like a web browser sends a request containing a hostname, then a piece of software such as **DNS resolver** sends a request to the DNS server to obtain the IP address of a hostname. If DNS server does not contain the IP address associated with a hostname, then it forwards the request to another DNS server. If IP address has arrived at the resolver, which in turn completes the request over the internet protocol.

Advantages

- The messages are delivered to users with zero downtime.
- Automatically corrects the typos.
- In case of maintenance or downtime the requests are answered by the closest node through Anycast technology.

Features

- DNS server gives a high performance and user interface. User interface has to handle a vast number of DNS.
- DNS has a distributed database.
- Additional records are also stored in DNS database.

MAESTRO

- It contains number of record types to facilitate other applications. For example, Mail Exchanger (MX) record.
- Any application which uses internet to connect from two or more host to share some information relies on DNS.

Message Log

The Message Log records information about each email message (inbound, outbound, and internal) processed by Email Security. Access the Message Log on the **Main > Status > Logs** page.

You can configure the number of entries per log page, between 25 and 200, in the **Per page** drop-down list in the log table banner. At the top and bottom of the page, scroll through Message Log pages by clicking the back and next arrows, or enter a specific page number in the **Page** field and click **Go**.

The length of time message records are saved in the database depends on your message volume and database partition capacity. To preserve message records, use the Export option to export the log on a regular basis. Exporting does not remove records from the Message Log. It transfers log data to a CSV or HTML file.

When the Message Log page appears, the most recent records are shown. Use the **View from/to** fields to specify the date/time range for the log entries you want to see. The calendar includes the following options:

- Change the month and year by using the back and next arrows around the month and year at the top of the calendar.
- Set the calendar to the current date by clicking the date in the lower left corner of the calendar.
- Click **Clean** to clear the current date/time calendar selection.
- Click **Today** to set the calendar date to today's date.

Set the time range in hours and minutes in the entry fields to the right of the calendar.

Message Log details

After you click a message in the Message Log ID column to view recipient details, a new **View Log Details** button is available at the bottom of the page. Message Log details appear in a table, with columns for the date and time of receipt, and the source of the message details. Detail sources can include message and connection control data, email policy data, and delivery data.

The log details appear in a third column, which can contain information about

- Message size, sender, and recipients

MAESTRO

- Connection type, sender IP address, and the Email Security appliance that received the connection request
- Email policies and actions applied, including policy and rule names (filter and action), email direction (inbound, outbound, or internal), name of the virus or spam encountered, and the action taken as a result of filtering
- Hybrid service scanning results, including a DKIM validation, if applicable
- Message delivery dispositions, including recipient email and IP address, encryption type, and delivery status

Message Log search options

The Message Log includes several search options, including date range or keyword searches. Determine the date/time range for a search by selecting dates in the **View from/to** field calendar controls. Default value for the **from** or **to** field is the date and time that you open the log.

You can perform a keyword search by selecting the log elements on which you want the search done from the **Keyword search** drop-down list and then entering a term in the field to the right of the list. Search for a keyword in all Message Log elements, or in 1 of the following Message Log components:

- Message Log ID
- Subject
- Sender Address
- Sender IP
- Recipient Address
- Scanning Result
- Message Status

Wildcard entries (*) in keywords are supported. However, for the sender IP address, an asterisk wildcard entry must be placed at the end of the address entry, after a period (e.g., 10.0.*).

Click **Set to Default** to return the keyword search options to the default settings (all Message Log components and keyword field blank).

MESSAGE LOG contents and format

The **MESSAGE LOG** file contains row and column information for each integrity constraint violation logged. For a given load, there are three types of messages logged: a timestamped header, row information, and a timestamped trailer. The header appears once per load. The tailer appears once if the statement executes successfully. The row information appears once for each integrity constraint violation logged.

MAESTRO

Web Service Components

There are three major web service components.

1. SOAP
2. WSDL
3. UDDI

SOAP Web Services

SOAP stands for Simple Object Access Protocol. It is a XML-based protocol for accessing web services.

SOAP is a W3C recommendation for communication between two applications.

SOAP is XML based protocol. It is platform independent and language independent. By using SOAP, you will be able to interact with other programming language applications.

Advantages of Soap Web Services

WS Security: SOAP defines its own security known as WS Security.

Language and Platform independent: SOAP web services can be written in any programming language and executed in any platform.

Disadvantages of Soap Web Services

Slow: SOAP uses XML format that must be parsed to be read. It defines many standards that must be followed while developing the SOAP applications. So it is slow and consumes more bandwidth and resource.

WSDL dependent: SOAP uses WSDL and doesn't have any other mechanism to discover the service.

MAESTRO

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!ALL THE BEST!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!