

# SYLLABUS-SEMESTER 6<sup>TH</sup> (CBCS) - B.A/B.SC COMPUTER APPLICATIONS

## (GENERIC ELECTIVE COURSE- THEORY)

**COURSE CODE: 0CAPGENE04**

**COURSE TITLE: NETWORKING & INTERNET**

### *GENERIC ELECTIVE COURSE*

*(Syllabus wise Topic Description)*

#### **UNIT-4: Introduction to Web Design**

##### **What is Web Design?**

Web design is a process of conceptualizing, planning, and building a collection of electronic files that determine the layout, colors, text styles, structure, graphics, images, and use of interactive features that deliver pages to your site visitors. Web design is the process of creating websites. It encompasses several different aspects, including webpage layout, content production, and graphic design. While the terms web design and web development is often used interchangeably, web design is technically a subset of the broader category of web development. Websites are created using a markup language called HTML. Web designers build WebPages using HTML tags that define the content and metadata of each page. The layout and appearance of the elements within a webpage are typically defined using CSS, or cascading style sheets.

##### **Introduction to Hypertext Markup Language (HTML)**

In 1980, physicist Tim Berners-Lee, a contractor at CERN, proposed and prototyped ENQUIRE, a system for CERN researchers to use and share documents. In 1989, Berners-Lee wrote a memo proposing an Internet-based hypertext system. HTML is a markup language that web browsers use to interpret and compose text, images, and other material into visual or audible web pages.

Hypertext Markup Language (HTML) is the standard markup language for documents designed to be displayed in a web browser. It can be assisted by technologies such as Cascading Style Sheets (CSS) and scripting languages such as JavaScript. Web browsers receive HTML documents from a web server or from local storage and render the documents into multimedia web pages. HTML describes the structure of a web page semantically and originally included cues for the appearance of the document. HTML elements are the building blocks of HTML pages. With HTML constructs, images and other objects such as interactive forms may be embedded into the rendered page. HTML elements are delineated by tags, written using angle brackets. Tags such as `<img />` and `<input />` directly introduce content into the page. HTML can embed programs written in a scripting language such as JavaScript, which affects the behavior and content of web pages.

##### **Document type definition**

---

A document type definition (DTD) is a set of markup declarations that define a document type for an SGML-family markup language (GML, SGML, XML, HTML). A DTD defines the valid building blocks of an XML document. It defines the document structure with a list of validated elements and attributes. A DTD can be declared inline inside an XML document, or as an external reference. A DTD defines the structure and the legal elements and attributes of an XML document.

With a DTD, independent groups of people can agree on a standard DTD for interchanging data. An application can use a DTD to verify that XML data is valid.

### **An Internal DTD Declaration**

If the DTD is declared inside the XML file, it must be wrapped inside the `<!DOCTYPE>` definition

### **An External DTD Declaration**

If the DTD is declared in an external file, the `<!DOCTYPE>` definition must contain a reference to the DTD file

## **The Building Blocks of XML Documents**

Seen from a DTD point of view, all XML documents are made up by the following building blocks:

- Elements
- Attributes
- Entities
- PCDATA
- CDATA

### **Elements**

- Elements are the main building blocks of both XML and HTML documents.
- Examples of HTML elements are "body" and "table".

### **Attributes**

- Attributes provide extra information about elements.
- Attributes are always placed inside the opening tag of an element. Attributes always come in name/value pairs. The following "img" element has additional information about a source file:
- ``

### **Entities**

- Some characters have a special meaning in XML, like the less than sign (`<`) that defines the start of an XML tag.
- Most of you know the HTML entity: `"&nbsp;";`. This "no-breaking-space" entity is used in HTML to insert an extra space in a document. Entities are expanded when a document is parsed by an XML parser.

### **PCDATA**

- PCDATA means parsed character data.
- Think of character data as the text found between the start tag and the end tag of an XML element.

parsed character data should not contain any &, <, or > characters; these need to be represented by the &amp; &lt; and &gt; entities, respectively.

## CDATA

CDATA means character data. CDATA is text that will NOT be parsed by a parser. Tags inside the text will NOT be treated as markup and entities will not be expanded.

## Creating Web Page

**Web page:** A web page is stored in a special kind of file, called a "html file". html files can be created in almost any program that can be used to edit text files.

The file that contains your home page must have a special name, index.html

### What is a tag?

A tag is a kind of control code that defines the look of the page. An example of a tag is <br>, which inserts a line break. Many tags can/shall also have arguments, for example the img tag, which is used to include pictures on a page. An example of how it can look is 

Tags must be surrounded by the characters < and >

### The start of the page

At the start of the file, we must have the tag <html>, to indicate that the file is a html file. Then we want a title on the page. The title is placed in the head of the page. Thus we need a head: <head>. The title is written within the <title> tag: <title>My homepage</title>.

### The text on the page

Now we write the actual page. It is placed in the so called "body". Thus we enter the <body> tag, with a few arguments to indicate what colors we want on our page: <body bgcolor="#ffffff" text="#000000" link="#00ffff" alink="#000000">. Now we can enter the text that we want on the page. If we want a line break on a certain place in the text, we enter the <br> tag.

### Pictures

If you want pictures on your web page you insert them using a special tag: <img>. The <img> tag can have many different arguments, to specify the size of the image, the thickness of the frame around the picture, etc.

### Links

To link to other pages or documents, you write something like this: <a href="http://www.cnn.com/">cnn</a>. This will create a link to www.cnn.com, called "cnn".

## The end of the page

At the bottom of the html file one must terminate the tags that were "opened" at the beginning of the file. This is done by entering `</body>` and `</html>`.

### *An example*

```
<html>
<head>
<title>My homepage</title>
</head>

<body bgcolor="#ffffff" text="#000000" link="#00ffff" alink="#000000">
Hello and welcome to my homepage. Right now it doesn't contain any useful info, but just you
wait until I've learnt more html!<br>
<br>
Here is a link to <a href="http://www.cnn.com/">cnn</a> where you can read the latest
news..<br>
<br>
A picture of ACC's logo:
<br>
Bye, and welcome back!
</body>
</html>
```

## HTML Lists

1. An Unordered List
2. An Ordered List

- **Unordered HTML List:**

An unordered list starts with the `<ul>` tag. Each list item starts with the `<li>` tag.

```
<ul>
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ul>
```

Unordered HTML List - Choose List Item Marker

The CSS `list-style-type` property is used to define the style of the list item marker:

Value	Description
disc	Sets the list item marker to a bullet (default)
circle	Sets the list item marker to a circle
square	Sets the list item marker to a square
none	The list items will not be marked

Example - Disc

```
<ul style="list-style-type:disc;">  
  <li>Coffee</li>  
  <li>Tea</li>  
  <li>Milk</li>  
</ul>
```

- **Ordered HTML List**

An ordered list starts with the `<ol>` tag. Each list item starts with the `<li>` tag.

The list items will be marked with numbers by default:

***Example***

```
<ol>  
  <li>Coffee</li>  
  <li>Tea</li>  
  <li>Milk</li>  
</ol>
```

**Ordered HTML List: - The Type Attribute**

The type attribute of the <ol> tag, defines the type of the list item marker:

Type	Description
type="1"	The list items will be numbered with numbers (default)
type="A"	The list items will be numbered with uppercase letters
type="a"	The list items will be numbered with lowercase letters
type="I"	The list items will be numbered with uppercase roman numbers
type="i"	The list items will be numbered with lowercase roman numbers

***Example Numbers:***

```
<ol type="1">  
  <li>Coffee</li>  
  <li>Tea</li>  
  <li>Milk</li>  
</ol>
```

## HTML Links

Links allow users to click their way from page to page.

### HTML Links - Hyperlinks

You can click on a link and jump to another document.

When you move the mouse over a link, the mouse arrow will turn into a little hand.

**Note:** A link does not have to be text. It can be an image or any other HTML element.

## HTML Links - Syntax

Hyperlinks are defined with the HTML `<a>` tag:

```
<a href="url">link text</a>
```

### *Example*

```
<a href="https://www.w3schools.com/html/">Visit our HTML tutorial</a>
```

## HTML Links - Image as a Link

It is common to use images as links:

### *Example*

```
<a href="default.asp">  
    
</a>
```

## Button as a Link

To use an HTML button as a link, you have to add some JavaScript code.

JavaScript allows you to specify what happens at certain events, such as a click of a button:

### *Example*

```
<button onclick="document.location = 'default.asp'">HTML Tutorial</button>
```

- **External Paths**

External pages can be referenced with a full URL or with a path relative to the current web page.

This example uses a full URL to link to a web page:

### *Example*

```
<a href="https://www.w3schools.com/html/default.asp">HTML tutorial</a>
```

**HTML Images Syntax:** In HTML, images are defined with the `<img>` tag.

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

The `src` attribute specifies the URL (web address) of the image:

```

```

## The alt Attribute

The alt attribute provides an alternate text for an image

### *Example*

```

```

Image Size - Width and Height

You can use the style attribute to specify the width and height of an image.

### *Example*

```

```

### **Image as a Link**

To use an image as a link, put the <img> tag inside the <a> tag:

### *Example*

```
<a href="default.asp">  
    
</a>
```

## **HTML Table**

An HTML table is defined with the <table> tag. Each table row is defined with the <tr> tag. A table header is defined with the <th> tag. By default, table headings are bold and centered. A table data/cell is defined with the <td> tag.

### **Example**

```
<table style="width:100%">  
  <tr>  
    <th>Firstname</th>  
    <th>Lastname</th>  
    <th>Age</th>  
  </tr>  
  <tr>  
    <td>Jill</td>  
    <td>Smith</td>  
    <td>50</td>  
  </tr>  
  <tr>  
    <td>Eve</td>  
    <td>Jackson</td>  
    <td>94</td>
```



```
</tr>
</table>
```

## Output:

### Basic HTML Table

Firstname	Lastname	Age
Jill	Smith	50
Eve	Jackson	94
John	Doe	80

**Note:** The `<td>` elements are the data containers of the table. They can contain all sorts of HTML elements; text, images, lists, other tables, etc

### HTML Table - Adding a Border

If you do not specify a border for the table, it will be displayed without borders.

A border is set using the CSS border property:

#### *Example*

```
table, th, td {
  border: 1px solid black;
}
```

to define borders for both the table and the table cells.

### HTML Table - Adding Cell Padding

Cell padding specifies the space between the cell content and its borders.

If you do not specify a padding, the table cells will be displayed without padding.

To set the padding, use the CSS padding property:

#### *Example*

```
th, td {
  padding: 15px;
}
```

- **HTML Table - Cells that Span Many Columns**

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

```
<html>
<body>
<h2>Cell that spans two columns</h2>
<p>To make a cell span more than one column, use the colspan attribute.</p>
<table border=5 style="width:100%">
  <tr>
    <th>Name</th>
    <th colspan="2">Telephone</th>
  </tr>
  <tr>
    <td>Bill Gates</td>
    <td>55577854</td>
    <td>55577855</td>
  </tr>
</table>
</body>
</html>
```

**Output:**

**Cell that spans two columns**

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

Name	Telephone	
Bill Gates	55577854	55577855

- **HTML Table - Cells that Span Many Rows**

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

**Program:**

```
<html>
<head>
</head>
<body>
<h2>Cell that spans two rows</h2>
<p>To make a cell span more than one row, use the rowspan attribute.</p>
<table border=7 style="width:100%">
  <tr>
    <th>Name:</th>
    <td>Bill Gates</td>
  </tr>
  <tr>
    <th rowspan="2">Telephone:</th>
    <td>55577854</td>
  </tr>
  <tr>
    <td>55577855</td>
  </tr>
</table>
</body>
</html>
```

## Output:

### Cell that spans two rows

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

<b>Name:</b>	Bill Gates
<b>Telephone:</b>	55577854
	55577855

## HTML Table - Adding a Caption

To add a caption to a table, use the `<caption>` tag:

**Note:** The `<caption>` tag must be inserted immediately after the `<table>` tag.

```
<html>
<head>
</head>
<body>
<h2>Table Caption</h2>
<p>To add a caption to a table, use the caption tag.</p>
<table border=8 style="width:100%">
  <caption>Monthly savings</caption>
  <tr>
    <th>Month</th>
    <th>Savings</th>
  </tr>
  <tr>
    <td>January</td>
    <td>$100</td>
```

```
</tr>
<tr>
  <td>February</td>
  <td>$50</td>
</tr>
</table>
</body>
</html>
```

**Output:**

**Table Caption**

To add a caption to a table, use the caption tag.

Monthly savings	
Month	Savings
January	\$100
February	\$50

**WEB FORMS: HTML - Forms**

HTML Forms are required, when you want to collect some data from the site visitor. For example, during user registration you would like to collect information such as name, email address, credit card, etc.

A form will take input from the site visitor and then will post it to a back-end application such as CGI, ASP Script or PHP script etc. The back-end application will perform required processing on the passed data based on defined business logic inside the application.

There are various form elements available like text fields, textarea fields, drop-down menus, radio buttons, checkboxes, etc.

*The HTML `<form>` element defines a form that is used to collect user input:*

```
<form>
.
form elements
```

</form>

## HTML Form Controls

There are different types of form controls that you can use to collect data using HTML form –

- Text Input Controls
- Checkboxes Controls
- Radio Box Controls
- Select Box Controls
- File Select boxes
- Hidden Controls
- Clickable Buttons
- Submit and Reset Button

The <input> Element

The <input> element is the most important form element.

Here are some examples:

Type	Description
<input type="text">	Defines a single-line text input field
<input type="radio">	Defines a radio button (for selecting one of many choices)
<input type="submit">	Defines a submit button (for submitting the form)

## Text Fields

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

**Note:** The form itself is not visible. Also note that the default width of an input field is 20 characters.

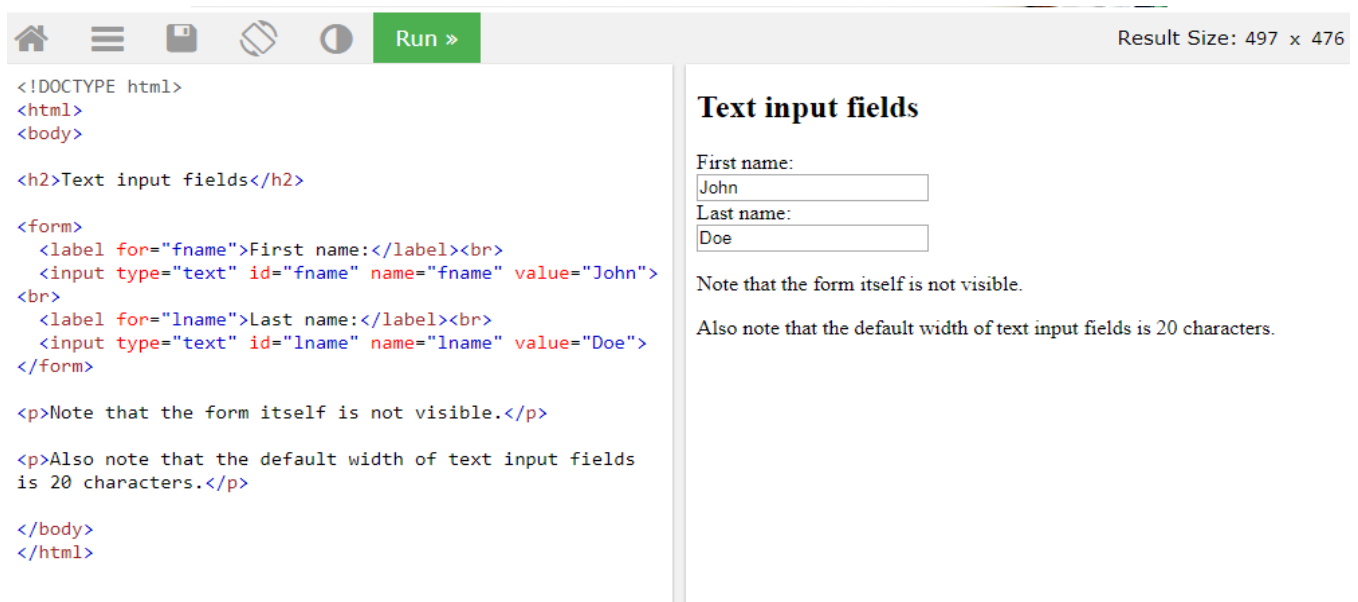
The <label> Element

Notice the use of the <label> element in the example above.

The <label> tag defines a label for many form elements.

The `<label>` element is useful for screen-reader users, because the screen-reader will read out loud the label when the user is focused on the input element.

## Example



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

<h2>Text input fields</h2>

<form>
  <label for="fname">First name:</label><br>
  <input type="text" id="fname" name="fname" value="John">
<br>
  <label for="lname">Last name:</label><br>
  <input type="text" id="lname" name="lname" value="Doe">
</form>

<p>Note that the form itself is not visible.</p>

<p>Also note that the default width of text input fields
is 20 characters.</p>

</body>
</html>
```

**Text input fields**  
First name:  
  
Last name:  
  
Note that the form itself is not visible.  
Also note that the default width of text input fields is 20 characters.

## Radio Buttons

`<input type="radio">` defines a **radio button**.

Radio buttons let a user select ONE of a limited number of choices.

## Example

A form with radio buttons:



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

<h2>Radio Buttons</h2>

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

</body>
</html>
```

**Radio Buttons**  
 Male  
 Female  
 Other

## The Submit Button

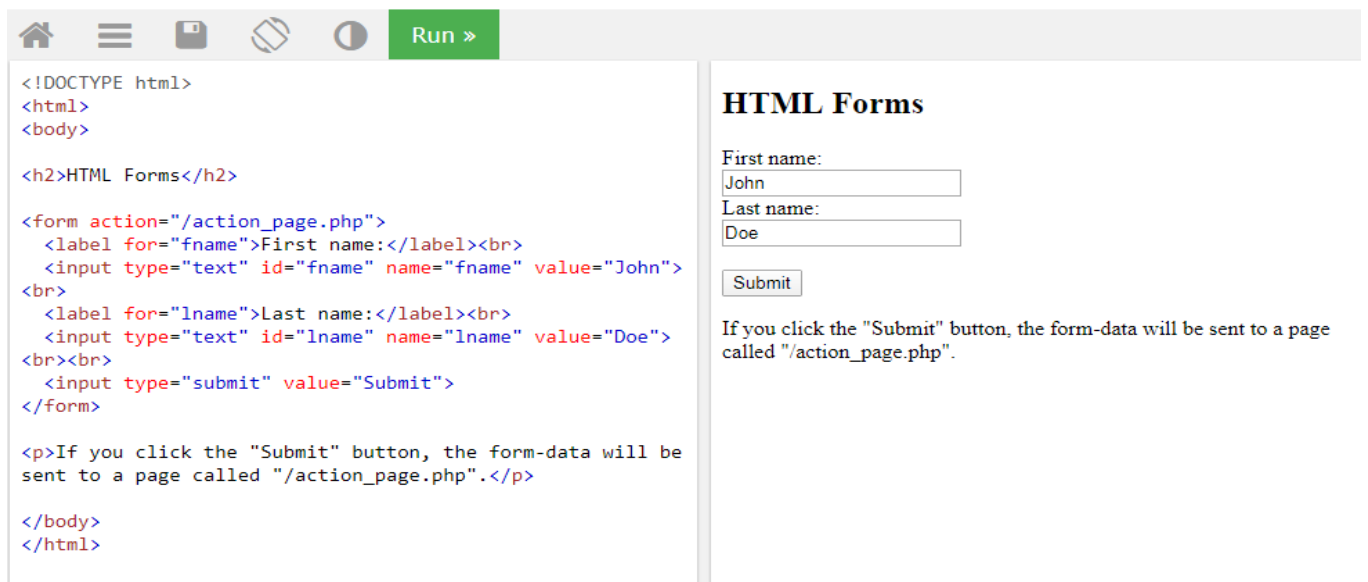
`<input type="submit">` defines a button for **submitting** the form data to a **form-handler**.

The form-handler is typically a page on the server with a script for processing input data.

The form-handler is specified in the form's **action** attribute.

## Example

A form with a submit button:



The screenshot shows a web browser interface. On the left, there is a code editor with the following HTML code:

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

<h2>HTML Forms</h2>

<form action="/action_page.php">
  <label for="fname">First name:</label><br>
  <input type="text" id="fname" name="fname" value="John">
<br>
  <label for="lname">Last name:</label><br>
  <input type="text" id="lname" name="lname" value="Doe">
<br><br>
  <input type="submit" value="Submit">
</form>

<p>If you click the "Submit" button, the form-data will be
sent to a page called "/action_page.php".</p>

</body>
</html>
```

On the right, the rendered form is displayed under the heading "HTML Forms". It contains two text input fields: "First name:" with the value "John" and "Last name:" with the value "Doe". Below the fields is a "Submit" button. A paragraph of text below the button reads: "If you click the 'Submit' button, the form-data will be sent to a page called '/action\_page.php'."

## The Action Attribute

The `action` attribute defines the action to be performed when the form is submitted.

Usually, the form data is sent to a page on the server when the user clicks on the submit button.

In the example above, the form data is sent to a page on the server called `"/action_page.php"`. This page contains a server-side script that handles the form data:

```
<form action="/action_page.php">
```

*If the `action` attribute is omitted, the action is set to the current page.*



## The Target Attribute

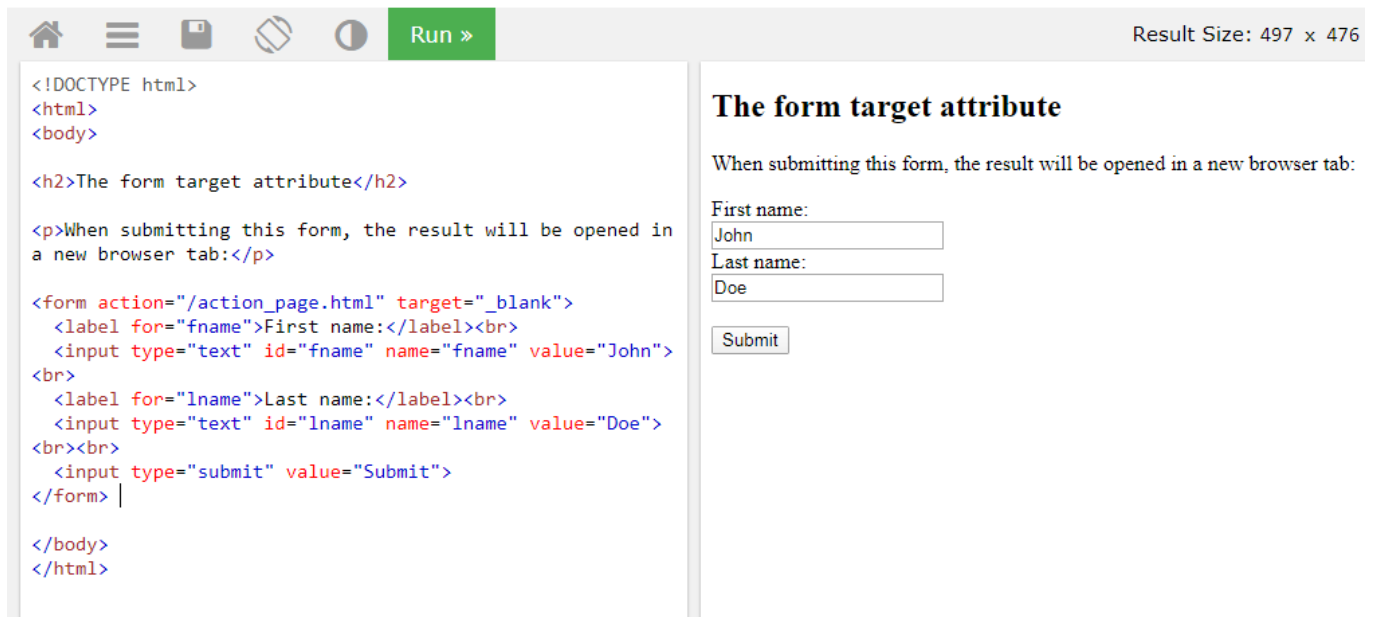
The `target` attribute specifies if the submitted result will open in a new browser tab, a frame, or in the current window.

The default value is "`_self`" which means the form will be submitted in the current window.

To make the form result open in a new browser tab, use the value "`_blank`".

## Example

*Here, the submitted result will open in a new browser tab:*



The screenshot shows a web browser window with a form titled "The form target attribute". The form contains two text input fields: "First name" with the value "John" and "Last name" with the value "Doe". A "Submit" button is located below the input fields. The browser's developer tools are open, showing the HTML code for the form. The code is as follows:

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

<h2>The form target attribute</h2>

<p>When submitting this form, the result will be opened in
a new browser tab:</p>

<form action="/action_page.html" target="_blank">
  <label for="fname">First name:</label><br>
  <input type="text" id="fname" name="fname" value="John">
<br>
  <label for="lname">Last name:</label><br>
  <input type="text" id="lname" name="lname" value="Doe">
<br><br>
  <input type="submit" value="Submit">
</form> |

</body>
</html>
```

## The Method Attribute

The `method` attribute specifies the HTTP method (**GET** or **POST**) to be used when submitting the form data.

- **GET METHOD:**

The default HTTP method when submitting form data is GET.

However, when GET is used, the form data will be **visible in the page's address field**:

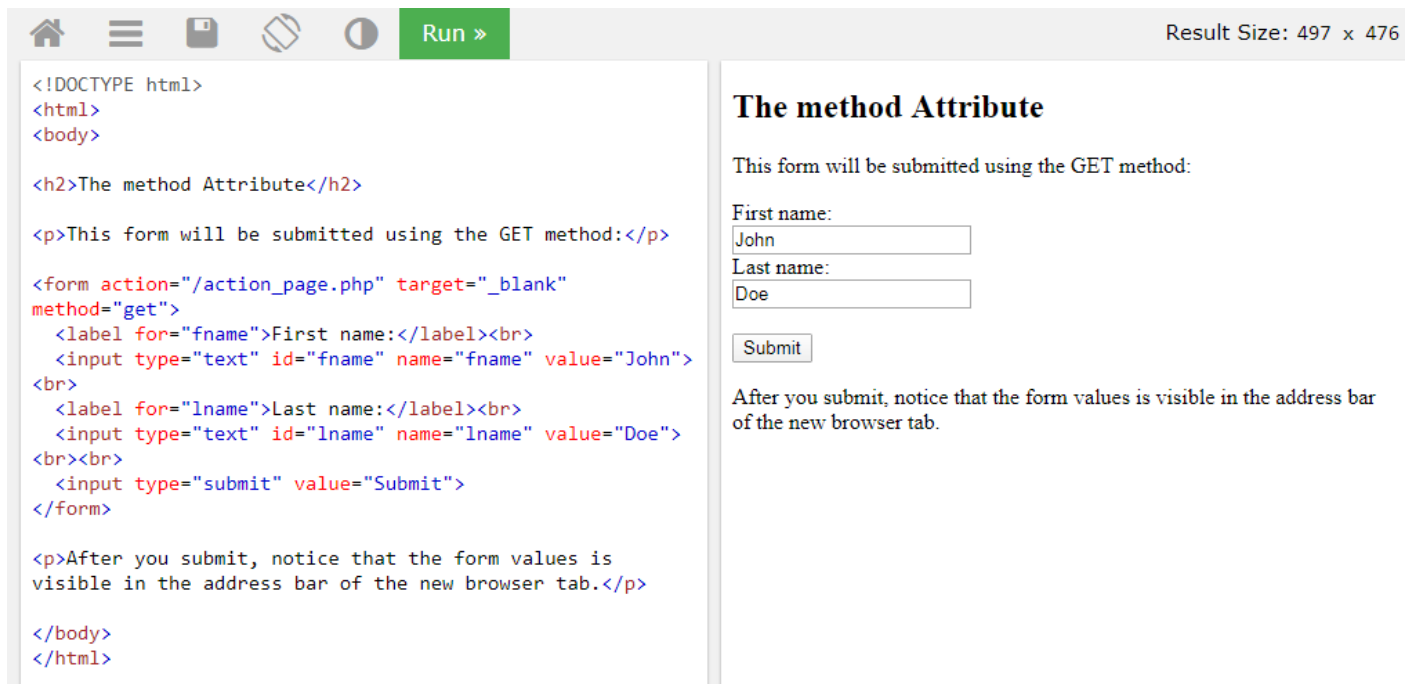
### Notes on GET:

- Appends form-data into the URL in name/value pairs
- The length of a URL is limited (2048 characters)
- Never use GET to send sensitive data! (will be visible in the URL)
- Useful for form submissions where a user wants to bookmark the result

- GET is better for non-secure data, like query strings in Google.

## Example

Use the GET method when submitting the form:



The screenshot shows a web browser window with a green 'Run' button in the top right corner. The browser's address bar displays 'Result Size: 497 x 476'. The main content area is titled 'The method Attribute' and contains the following text: 'This form will be submitted using the GET method:'. Below this text is a form with two text input fields: 'First name:' with the value 'John' and 'Last name:' with the value 'Doe'. A 'Submit' button is located below the input fields. Below the form, the text reads: 'After you submit, notice that the form values is visible in the address bar of the new browser tab.' The left sidebar of the browser shows the HTML source code for the page, which includes the following code:

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

<h2>The method Attribute</h2>

<p>This form will be submitted using the GET method:</p>

<form action="/action_page.php" target="_blank"
method="get">
  <label for="fname">First name:</label><br>
  <input type="text" id="fname" name="fname" value="John">
<br>
  <label for="lname">Last name:</label><br>
  <input type="text" id="lname" name="lname" value="Doe">
<br><br>
  <input type="submit" value="Submit">
</form>

<p>After you submit, notice that the form values is
visible in the address bar of the new browser tab.</p>

</body>
</html>
```

## • POST METHOD

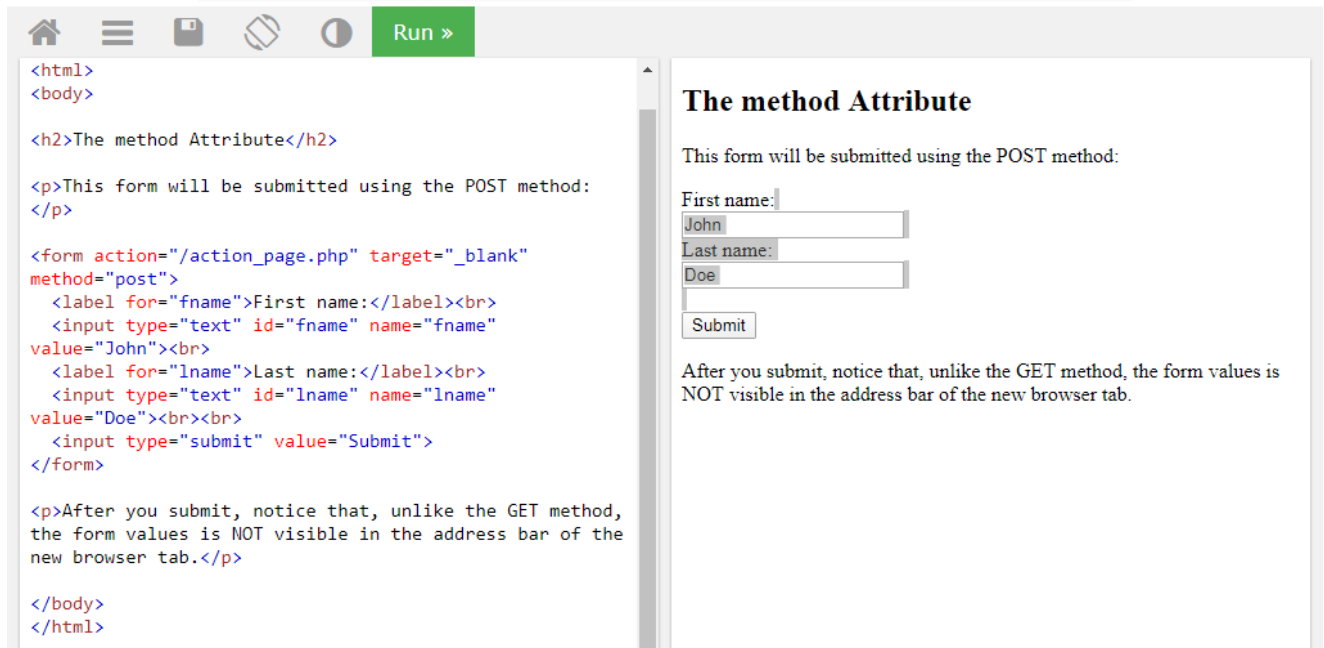
Always use POST if the form data contains sensitive or personal information. The POST method does not display the form data in the page address field.

### Notes on POST:

- POST has no size limitations, and can be used to send large amounts of data.
- Form submissions with POST cannot be bookmarked

## Example

Use the POST method when submitting the form:



The screenshot shows a web browser window with a code editor on the left and a rendered form on the right. The code editor contains the following HTML code:

```
<html>
<body>

<h2>The method Attribute</h2>

<p>This form will be submitted using the POST method:
</p>

<form action="/action_page.php" target="_blank"
method="post">
  <label for="fname">First name:</label><br>
  <input type="text" id="fname" name="fname"
value="John"><br>
  <label for="lname">Last name:</label><br>
  <input type="text" id="lname" name="lname"
value="Doe"><br><br>
  <input type="submit" value="Submit">
</form>

<p>After you submit, notice that, unlike the GET method,
the form values is NOT visible in the address bar of the
new browser tab.</p>

</body>
</html>
```

The rendered form on the right is titled "The method Attribute" and contains the following text and form elements:

This form will be submitted using the POST method:

First name:

Last name:

After you submit, notice that, unlike the GET method, the form values is NOT visible in the address bar of the new browser tab.

## HTML Input Types

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

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

- `<input type="week">`
- **Input Type Text**

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

- **Input Type Password**

`<input type="password">` defines a **password field**.

- **Input Type Submit**

`<input type="submit">` defines a button for **submitting** form data to a **form-handler**.

The form-handler is typically a server page with a script for processing input data.

- **Input Type Reset**

`<input type="reset">` defines a **reset button** that will reset all form values to their default values.

- **Input Type Radio**

`<input type="radio">` defines a **radio button**.

Radio buttons let a user select **ONLY ONE** of a limited number of choices.

- **Input Type Checkbox**

`<input type="checkbox">` defines a **checkbox**.

Checkboxes let a user select **ZERO** or **MORE** options of a limited number of choices.

## Example

```
<form>
  <input type="checkbox" id="vehicle1" name="vehicle1" value="Bike">
  <label for="vehicle1"> I have a bike</label><br>
  <input type="checkbox" id="vehicle2" name="vehicle2" value="Car">
  <label for="vehicle2"> I have a car</label><br>
  <input type="checkbox" id="vehicle3" name="vehicle3" value="Boat">
  <label for="vehicle3"> I have a boat</label>
</form>
```

## OUTPUT:

HTML code above will be displayed in a browser:

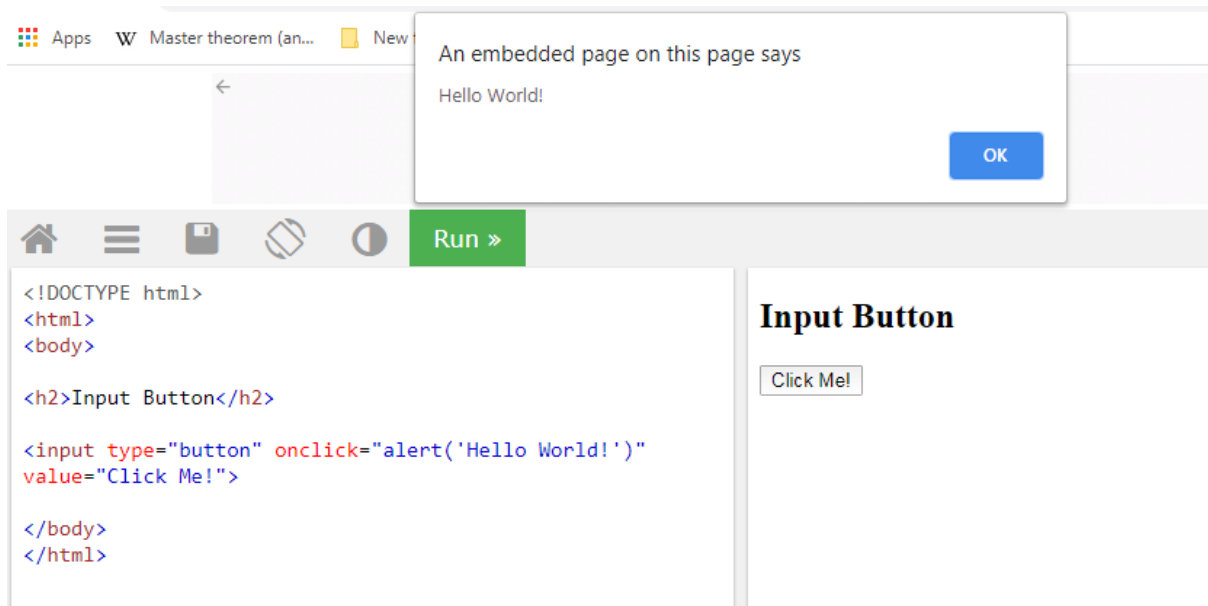
- I have a bike
- I have a car
- I have a boat

- **Input Type Button**

`<input type="button">` defines a **button**:

### Example

```
<input type="button" onclick="alert('Hello World!')" value="Click Me!">
```



- **Input Type Date**

The `<input type="date">` is used for input fields that should contain a date.

Depending on browser support, a date picker can show up in the input field.

### Example

```
<form>  
  <label for="birthday">Birthday:</label>  
  <input type="date" id="birthday" name="birthday">  
</form>
```

- **Input Type Time**

The `<input type="time">` allows the user to select a time (no time zone).

Depending on browser support, a time picker can show up in the input field.

## Example

```
<form>
  <label for="appt">Select a time:</label>
  <input type="time" id="appt" name="appt">
</form>
```

## HTML Images

Images can improve the design and the appearance of a web page.

### HTML Images Syntax

In HTML, images are defined with the `<img>` tag.

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

The `src` attribute specifies the URL (web address) of the image:

```

```

### The alt Attribute

The `alt` attribute provides an alternate text for an image, if the user for some reason cannot view it (because of slow connection, an error in the `src` attribute, or if the user uses a screen reader).

The value of the `alt` attribute should describe the image:

### *Example*

```

```

### *Example*

```

```

## Image Size - Width and Height

To specify the width and height of an image. The width and height attributes always define the width and height of the image in pixels.

```
<html>
```

```
<body>
```

```
<h2>Image Size</h2>
```

```
<p>Use the style attribute to specify the width and height of an image:</p>

</body>
</html>
```

## Animated Images

HTML allows animated GIFs:

### *Example*

```

```

## Background Image on a Page

If you want the entire page to have a background image, then you must specify the background image on the <body> element:

### *Program.*

```
<html>
<body background="img_girl.jpg" width=200 height=200>
<h2>Background Image</h2>
<p> background image shown on web page</p>
</body>
</html>
```

### *Output:*



**Background Repeat:** If the background image is smaller than the element, the image will repeat itself, horizontally and vertically, until it has reached the end of the element.



## HTML frames

HTML frames are used to divide your browser window into multiple sections where each section can load a separate HTML document. A collection of frames in the browser window is known as a frameset. The window is divided into frames in a similar way the tables are organized: into rows and columns.

### Creating Frames

To use frames on a page we use `<frameset>` tag instead of `<body>` tag. The `<frameset>` tag defines how to divide the window into frames. The **rows** attribute of `<frameset>` tag defines horizontal frames and **cols** attribute defines vertical frames. Each frame is indicated by `<frame>` tag and it defines which HTML document shall open into the frame.

### Example

Following is the example to create three horizontal frames –

```
<html>
  <head>
    <title>Example of HTML Frames using row attribute</title>
  </head>
```



```

<frameset rows = "20%, 60%, 20%">
  <frame name = "top" src =
  "C:/Users/dharam/Desktop/attr1.png" />
  <frame name = "main" src =
  "C:/Users/dharam/Desktop/gradient3.png" />
  <frame name = "bottom" src =
  "C:/Users/dharam/Desktop/col_last.png" />
  <noframes>
    <body>The browser you are working does
    not support frames.</body>
  </noframes>
</frameset>
</html>

```

**Output:** The above example basically used to create three horizontal frames: top, middle and bottom using row attribute of frameset tag and the noframe tag is used for those browser who doesn't support noframe.

**Example:** This example illustrates the col attribute of frameset tag.

```

<html>
  <head>
    <title>Example of HTML Frames Using col Attribute</title>
  </head>

  <frameset cols = "30%, 40%, 30%">
    <frame name = "top" src =
    "C:/Users/dharam/Desktop/attr1.png" />
    <frame name = "main" src =
    "C:/Users/dharam/Desktop/gradient3.png" />
    <frame name = "bottom" src =
    "C:/Users/dharam/Desktop/col_last.png" />
    <noframes>
      <body>The browser you are working does
      not support frames.</body>
    </noframes>
  </frameset>
</html>

```

**Output:** The above example basically used to create three vertical frames: left, center and right using col attribute of frameset tag.

#### Attributes of Frameset tag:

- **cols:** The cols attribute is used to create vertical frames in web browser. This attribute is basically used to define the no of columns and its size inside the frameset tag. The size or width of the column is set in the frameset in the following ways:

Use absolute value in pixel

**Example:**

```
<frameset cols = "300, 400, 300">
```

Use percentage value

**Example:**

```
<frameset cols = "30%, 40%, 30%">
```

- **rows:** The rows attribute is used to create horizontal frames in web browser. This attribute is used to define no of rows and its size inside the frameset tag.

The size of rows or height of each row use the following ways:

Use absolute value in pixel

**Example:**

```
<frameset rows = "300, 400, 300">
```

Use percentage value

**Example:**

```
<frameset rows = "30%, 40%, 30%">
```

- **border:** This attribute of frameset tag defines the width of border of each frames in pixels. Zero value is used for no border.

**Example:**

```
<frameset border="4" frameset>
```

- **frameborder:** This attribute of frameset tag is used to specify whether the three-dimensional border should be displayed between the frames or not for this use two values 0 and 1, where 0 defines for no border and value 1 signifies for yes there will be border.
- **framespacing:** This attribute of frameset tag is used to specify the amount of spacing between the frames in a frameset. This can take any integer value as an parameter which basically denotes the value in pixel.

### Attributes of Frame Tag:

- **name:** This attribute is used to give names to the frame. It differentiate one frame from another. It is also used to indicate which frame a document should loaded into.

**Example:**

```
<frame name = "top" src = "C:/Users/dharam/Desktop/attr1.png" />
```

```
<frame name = "main" src = "C:/Users/dharam/Desktop/gradient3.png" />
```

```
<frame name = "bottom" src = "C:/Users/dharam/Desktop/col_last.png" />
```

Here we use three frames with names as left center and right.

- **src:** This attribute in frame tag is basically used to define the source file that should be loaded into the frame. The value of src can be any url.

**Example:**

```
<frame name = "left" src = "/html/left.htm" />
```

In the above example name of frame is left and source file will be loaded from “/html/left.htm” in frame.

- **marginwidth:** This attribute in frame tag is used to specify width of the spaces in pixels between the border and contents of left and right frame.

**Example:**

```
<frame marginwidth="20">
```

- **marginheight:** This attribute in frame tag is used to specify height of the spaces in pixels between the border and contents of top and bottom frame.

**Example:**

```
<frame marginheight="20">
```

- **scrollbar:** To control the appearance of scroll bar in frame use scrollbar attribute in frame tag. This is basically used to control the appearance of scrollbar. The value of this attribute can be yes, no, auto. Where the value no denotes there will be no appearance of scroll bar.

**Example:**

```
<frame scrollbar="no">
```

## Web hosting

Web hosting is a service that allows organizations and individuals to post a website or web page onto the Internet. A web host, or web hosting service provider, is a business that provides the technologies and services needed for the website or webpage to be viewed in the Internet. Websites are hosted, or stored, on special computers called servers. When Internet users want to view your website, all they need to do is type your website address or domain into their browser. Their computer will then connect to your server and your web pages will be delivered to them through the browser.

Every website that you visit consists of a domain name and a web host. The easiest way to picture it is to think of the domain name as the address and the web hosting as the physical building.

When you're browsing the web and you type in a URL (i.e. <http://www.hostinger.com>), your web browser basically queries that domain name (i.e. [hostinger.com](http://www.hostinger.com)) and asks it where the website is hosted. The browser then loads up the contents that are stored there.

Web hosting is effectively the process of using a server to host a website, and there are all sorts of different types of web hosting available out there on the market. Hostigner, just like most web hosts offer different packages so that their customers can pay for just the resources that they need. The more popular the website is, the more resources it's likely to need to function effectively.

**Web Hosting** is an account on a computer (aka server) that can store and serve **website** files via the **Internet**. **Domain** Registration is leasing a human-readable word (e.g., [amazon.com](http://amazon.com)) that directs people to specific **website** files via a browser. As an analogy, a **domain** is an “address” on the **Internet**.

Most hosting companies require that you own your domain in order to host with them. If you do not have a domain, the hosting companies will help you purchase one.



## **Here are some features you should be expecting from your hosting provider:**

- **Email Accounts** As mentioned earlier, most hosting providers require users to have their own domain name. With a domain name (e.g. www.yourwebsite.com) and email account features provided by your hosting company, you can create domain email accounts (e.g.yourname@yourwebsite.com).
- **FTP Access** The use of FTP lets you upload files from your local computer to your web server. If you build your website using your own HTML files, you can transfer the files from your computer to the web server through FTP, allowing your website to be accessed through the internet.
- **WordPress Support** Word Press is an online website creation tool. It is a powerful blogging and website content management system, which is a convenient way to create and manage website. Word Press powers over 25% of websites on the internet.

## **Types of Web Hosting Services**

There are various types of web hosting services available to host your website. Before signing up for web hosting services, it is important to understand what kind of service your website needs, the kind of server you or your business needs, your budget, and what type of services the web host offers.

### **Hosting options available are:**

- Website Builders
- Shared Hosting
- Dedicated Hosting
- Collocated Hosting

#### **Website Builders**

Website builder services are a type of hosting service that caters to beginners who need to host a website, but lack the technical skills and knowledge to build one. Website builder services typically provide you with an online browser-based interface to build your website, and also host the website for you without any additional setup.

#### **Shared Hosting**

In a shared hosting environment, your and other website owners shared one server. This includes sharing the physical server and the software applications within the server. Shared hosting services are affordable because the cost to operate the server is shared between you and these other owners. There are, however, a number of down sides, such as being slower.

## **Dedicated Hosting**

In a dedicated hosting environment, you have the entire web server to yourself. This allows for faster performance, as you have all the server's resources entirely, without sharing with other website owners. However, this also means that you will be responsible for the cost of server operation entirely. This is a good choice for websites that requires a lot of system resources, or need a higher level of security.

## **Collocated Hosting**

In this type of hosting, you will purchase your own server and have it housed at a web host's facilities. You will be responsible for the server itself. An advantage of this type of hosting service is you have full control of the web server. You can install any scripts or applications you need.

## **Domain name registration**

**Domain name registration** is the act of reserving a name on the Internet for a certain period, usually one year. It is important to know that this domain will remain yours for as long as you renew it and there is no way to purchase a domain name forever. Domain name registration is necessary for a website, an email or another web service. The most important thing about a domain name registration is that it gives you personality and recognized identity. The price for a domain name registration varies greatly on its extension. For example, the most common .com is the most affordable one, while some country-specific domain name extensions are 5 times more expensive.

Domain names are used to identify one or more IP addresses. For example, the domain name microsoft.com represents about a dozen IP addresses. Domain names are used in URLs to identify particular Web pages.

**For example**, in the URL <http://www.pcwebopedia.com/index.html>, the **domain** name is pcwebopedia.com.

## **Domain name registrar**

A domain name registrar is a company that manages the reservation of Internet domain names. A domain name registrar must be accredited by a generic top-level domain (gTLD) registry or a country code top-level domain (ccTLD) registry. A registrar operates in accordance with the guidelines of the designated domain name registries.

## **DNS hosting**

Registration of a domain name establishes a set of name server records in the DNS servers of the parent domain, indicating the IP addresses of DNS servers that are authoritative for the domain. This provides a reference for direct queries of domain data.

Registration of a domain does not automatically imply the provision of DNS services for the registered domain. Most registrars do offer DNS hosting as an optional free service for domains registered through them. If DNS services are not offered, or the end-user opts out, the end-user is responsible for procuring or self-hosting DNS services. Registrars require the specification of usually at least two name servers.

## Domain Name System

---

The Domain Name System (DNS) is a hierarchical and decentralized naming system for computers, services, or other resources connected to the Internet or a private network. It associates various information with domain names assigned to each of the participating entities. Most prominently, it translates more readily memorized domain names to the numerical IP addresses needed for locating and identifying computer services and devices with the underlying network protocols. By providing a worldwide, distributed directory service, the Domain Name System has been an essential component of the functionality of the Internet since 1985.

The Domain Name System delegates the responsibility of assigning domain names and mapping those names to Internet resources by designating authoritative name servers for each domain. Network administrators may delegate authority over sub-domains of their allocated name space to other name servers. This mechanism provides distributed and fault-tolerant service and was designed to avoid a single large central database.

## Cascading Style Sheet: Styling HTML with CSS

Cascading Style Sheets (CSS) is a simple mechanism for adding style (e.g., fonts, colors, and spacing) to Web documents. Cascading Style Sheets is a style sheet language used for describing the presentation of a document written in a markup language like HTML. CSS is a cornerstone technology of the World Wide Web, alongside HTML and JavaScript. The most common way to add CSS is to keep the styles in separate CSS files.

CSS describes how HTML elements are to be displayed on screen, paper, or in other media.

CSS **saves a lot of work**. It can control the layout of multiple web pages all at once.

CSS can be added to HTML elements in 3 ways:

- **Inline** - by using the style attribute in HTML elements
- **Internal** - by using a <style> element in the <head> section
- **External** - by using an external CSS file

### Inline CSS

An inline CSS is used to apply a unique style to a single HTML element.

An inline CSS uses the style attribute of an HTML element.

This example sets the text color of the <h1> element to blue:

#### Program:

```
<html>
```

```
<body>
```

```
<h1 style="color:blue;">This is a Blue Heading</h1>
```

```
</body>
```

```
</html>
```

**Output:** [Blue Heading](#)

## **Internal CSS**

An internal CSS is used to define a style for a single HTML page.

An internal CSS is defined in the `<head>` section of an HTML page, within a `<style>` element:

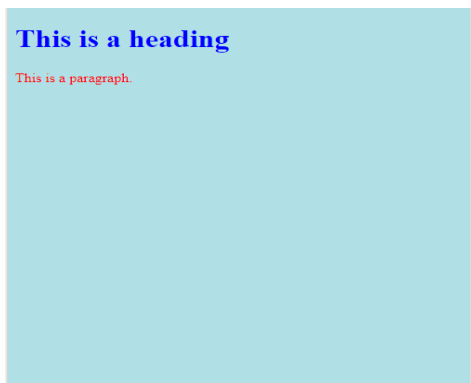
### **Program:**

```
<html>
<head>
<style>
body {background-color: blue;}
h1 {color: blue;}
p {color: red;}
</style>
</head>
<body>

<h1>This is a heading</h1>
<p>This is a paragraph.</p>

</body>
</html>
```

### **Output:**



## External CSS

An external style sheet is used to define the style for many HTML pages.

**With an external style sheet, you can change the look of an entire web site, by changing one file!**

To use an external style sheet, add a link to it in the `<head>` section of the HTML page:

### Program:

```
<html>
<head>
  <link rel="stylesheet" href="styles.css">
</head>
<body>

<h1>This is a heading</h1>
<p>This is a paragraph.</p>

</body>
</html>
```

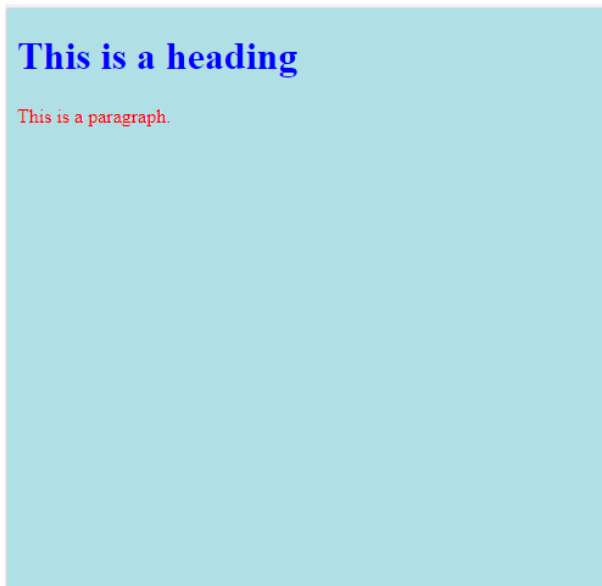
An external style sheet can be written in any text editor. The file must not contain any HTML code, and must be saved with a .css extension.

*Here is how the "styles.css" looks:*

```
body {
  background-color: powderblue;
}
h1 {
  color: blue;
}
p {
  color: red;
}
```



## Output:



## Text Formatting Manipulations

HTML provides us with the ability for formatting text just like we do it in MS Word or any text editing software. In this article, we would go through few such options.

1. **Making text Bold or Strong:** We can make the text **bold** using the `<b>` tag. The tag uses both opening and closing tag. The text that needs to be made bold must be within `<b>` and `</b>` tag.

We can also use the `<strong>` tag to make the text strong, with added semantic importance. It also opens with `<strong>` and ends with `</strong>` tag.

### Example:

```
<html>

<head>
  <title>Bold</title>
</head>
<body>
  <!--Normal text-->
  <p>Hello World</p>
  <!--Text in Bold-->
  <p><b>Hello Hello World </b></p>
  <!--Text in Strong-->
  <p><strong> Hello World </strong></p>
</body>
</html>
```

### Output:

Hello World

**Hello World**

**Hello World**

- 2. Making text *Italic* or *emphasize*:** The `<i>` tag is used to *italicise* the text. It opens with `<i>` and ends with `</i>` tag.  
The `<em>` tag is used to *emphasize* the text, with added semantic importance. It opens with `<em>` and ends with `</em>` tag.
- 3. Highlighting a text:** It is also possible to highlight a text in HTML using the `<mark>` tag. It has an opening tag `<mark>` and a closing tag `</mark>`.
- 4. Making a text Subscript or Superscript:** The `<sup>` element is used to superscript a text and `<sub>` element is used to subscript a text. They both have opening and a closing tag.
- 5. Making text smaller:** The `<small>` element is used to make the text smaller. The text that needs to be displayed smaller should be written inside `<small>` and `</small>` tag.
- 6. Striking through the text:** The `<del>` element is used to strike through the text marking the part as deleted. It also has an opening and a closing tag.

## TEXT FORMATTING

### ❖ Text Color

The `color` property is used to set the color of the text. The color is specified by:

- a color name - like "red"
- a HEX value - like "#ff0000"
- an RGB value - like "rgb(255,0,0)"

### ❖ Text Alignment

The `text-align` property is used to set the horizontal alignment of a text.

A text can be left or right aligned, centered, or justified.

### ❖ Text Direction

This property can be used to change the text direction of an element, by default direction is from left to right.

### ❖ Text Transformation

The `text-transform` property is used to specify uppercase and lowercase letters in a text.

It can be used to turn everything into uppercase or lowercase letters, or capitalize the first letter of each word:

### *Example*

```
p.uppercase {  
  text-transform: uppercase;  
}
```

```
p.lowercase {  
  text-transform: lowercase;  
}
```

```
p.capitalize {  
  text-transform: capitalize;  
}
```

### ❖ **Text Indentation**

The `text-indent` property is used to specify the indentation of the first line of a text:

### ❖ **Letter Spacing**

The `letter-spacing` property is used to specify the space between the characters in a text.

### ❖ **Line Height**

The `line-height` property is used to specify the space between lines

### ❖ **Word Spacing**

The `word-spacing` property is used to specify the space between the words in a text

### ❖ **Text Shadow**

The `text-shadow` property adds shadow to text.

## **Font Families**

In CSS, there are two types of font family names:

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

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

#### ❖ **Font Style**

The `font-style` property is mostly used to specify italic text.

This property has three values:

- normal - The text is shown normally
- italic - The text is shown in italics

#### ❖ **Font Weight**

The `font-weight` property specifies the weight of a font:

#### ❖ **Font Size**

The `font-size` property sets the size of the text.

Being able to manage the text size is important in web design. However, you should not use font size adjustments to make paragraphs look like headings, or headings look like paragraphs.

Always use the proper HTML tags, like `<h1>` - `<h6>` for headings and `<p>` for paragraphs.