HTML

HTML Introduction

What is HTML?

- HTML stands for Hyper Text Markup Language
- HTML is the standard markup language for creating Web pages
- HTML describes the structure of a Web page
- HTML consists of a series of elements
- HTML elements tell the browser how to display the content
- HTML elements label pieces of content such as "this is a heading", "this is a paragraph", "this is a link", etc.

HTML Editors

Learn HTML Using Notepad or TextEdit

Web pages can be created and modified by using professional HTML editors.
However, for learning HTML we recommend a simple text editor like Notepad (PC) or TextEdit (Mac).
We believe in that using a simple text editor is a good way to learn HTML.
Follow the steps below to create your first web page with Notepad or TextEdit.

Step 1: Open Notepad (PC)

Step 2: Write Some HTML

Write or copy the following HTML code into Notepad:

```
<!DOCTYPE html>
<html>
<body>
<p>My first paragraph.</p>
</body>
</html>
```

Step 3: Save the HTML Page

Save the file on your computer. Select **File > Save as** in the Notepad menu.
Name the file **"index.htm"** and set the encoding to **UTF-8** (which is the preferred encoding for HTML files).

Tip: You can use either **.htm or .html** as file extension. There is no difference, it is up to you.

Step 4: View the HTML Page in Your Browser

Open the saved HTML file in your favorite browser (double click on the file, or right-click - and choose "Open with").
The result will look much like this:

The <!DOCTYPE> Declaration

The <!DOCTYPE> declaration represents the document type, and helps browsers to display web pages correctly. It must only appear once, at the top of the page (before any HTML tags).
The <!DOCTYPE> declaration is not case sensitive.
The <!DOCTYPE> declaration for HTML5 is:

```
<!DOCTYPE html>
```

HTML Elements

An HTML element is defined by a start tag, some content, and an end tag.

HTML Elements

The HTML **element** is everything from the start tag to the end tag:

```
<tagname>Content goes here...</tagname>
```

Examples of some HTML elements:

```
<h1>My First Heading</h1>
<p>My first paragraph.</p>
```

Start tag	Element content	End tag
<h1>	My First Heading	</h1>
	My first paragraph.	
 	none	*none*

Note: Some HTML elements have no content (like the <br> element). These elements are called empty elements. Empty elements do not have an end tag!

Nested HTML Elements

HTML elements can be nested (this means that elements can contain other elements).
All HTML documents consist of nested HTML elements.
The following example contains four HTML elements (<html>, <body>, <h1> and):

```
<!DOCTYPE html>
<html>
<body>
<h1>My First Heading</h1>
```

```
<p>My first paragraph.</p>
</body>
</html>
```

Try it Yourself »

HTML Headings

HTML headings are defined with the <h1> to <h6> tags.
<h1> defines the most important heading. <h6> defines the least important heading:

```
<h1>This is heading 1</h1>
<h2>This is heading 2</h2>
<h3>This is heading 3</h3>
```

Try it Yourself »

HTML Paragraphs

HTML paragraphs are defined with the <p> tag:

Example

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

Try it Yourself »

HTML Links

HTML links are defined with the <a> tag:

Example

```
<a href="https://www.cometservices.org">This is a comet services link</a>
```

Try it Yourself »

The link's destination is specified in the href attribute.
Attributes are used to provide additional information about HTML elements.
You will learn more about attributes in a later chapter.

HTML Images

HTML images are defined with the <img> tag.
The source file (src), alternative text (alt), width, and height are provided as attributes:

```
<img src="w3schools.jpg" alt="W3Schools.com" width="104" height="142">
```

Try it Yourself »

The src Attribute

The required src attribute specifies the path (URL) to the image.

```
<img src="img_chania.jpg" alt="Flowers in Chania">
```

Try it Yourself »

The alt Attribute

The required 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:

```
<img src="img_chania.jpg" alt="Flowers in Chania">
```

Try it Yourself »

If a browser cannot find an image, it will display the value of the alt attribute:

```
<img src="wrongname.gif" alt="Flowers in Chania">
```

Try it Yourself »

Tip: A screen reader is a software program that reads the HTML code, and allows the user to "listen" to the content. Screen readers are useful for people who are visually impaired or learning disabled.

Image Size - Width and Height

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

```
<img src="img_girl.jpg" alt="Girl in a jacket" style="width:500px;height:600px;">
```

Try it Yourself »

Alternatively, you can use the width and height attributes:

```
<img src="img_girl.jpg" alt="Girl in a jacket" width="500" height="600">
```

Try it Yourself »

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

Note: Always specify the width and height of an image. If width and height are not specified, the web page might flicker while the image loads.

HTML Layout

In this tutorial you will learn about the different methods of creating a web page layout.

Creating Website Layouts

Creating a website layout is the activity of positioning the various elements that make a web page in a well-structured manner and give appealing look to the website.
You have seen most websites on the internet usually display their content in multiple rows and columns, formatted like a magazine or newspaper to provide the users a better reading and writing environment. This can be easily achieved by using the HTML tags, such as <table>, , <header>, <footer>, <section>, etc. and adding some CSS styles to them.

HTML meta Element

The <meta> element provides metadata about the HTML document. Metadata is a set of data that describes and gives information about other data. Here's an example:

Try this code »

```
<head>
  <title>Specifying Metadata</title>
  <meta charset="utf-8">
  <meta name="author" content="John Smith">
</head>
```

```
<head> <title>Defining Keywords and Description</title> <meta name="keywords" content="HTML, CSS, javaScript">
 <meta name="description" content="Easy to understand tutorials and references on HTML, CSS, javaScript and more..."> </head>
```

HTML script Element

The <script> element is used to define client-side script, such as JavaScript in HTML documents. The following example will display a greeting message in the browser:

Try this code »

```
<head>
  <title>Adding JavaScript</title>
  <script>
    document.write("<h1>Hello World!</h1>")
  </script>
</head>
```

HTML Attributes:

HTML attributes provide additional information about HTML elements.

- All HTML elements can have **attributes**
- Attributes provide **additional information** about elements
- Attributes are always specified in **the start tag**
- Attributes usually come in name/value pairs like: **name="value"**

The href Attribute

The <a> tag defines a hyperlink. The href attribute specifies the URL of the page the link goes to:

```
<a href="https://www.w3schools.com">Visit W3Schools</a>
```

Try it Yourself »

You will learn more about links in our HTML Links chapter.

The src Attribute

The <img> tag is used to embed an image in an HTML page. The src attribute specifies the path to the image to be displayed:

```
<img src="img_girl.jpg">
```

Try it Yourself »

The width and height Attributes

The <img> tag should also contain the width and height attributes, which specifies the width and height of the image (in pixels):

```
<img src="img_girl.jpg" width="500" height="600">
```

Try it Yourself »

The alt Attribute

The required alt attribute for the <img> tag specifies an alternate text for an image, if the image for some reason cannot be displayed. This can be due to slow connection, or an error in the src attribute, or if the user uses a screen reader.

```
<img src="img_girl.jpg" alt="Girl with a jacket">
```

Try it Yourself »

See what happens if we try to display an image that does not exist:

```
<img src="img_typo.jpg" alt="Girl with a jacket">
```

Try it Yourself »

You will learn more about images in our HTML Images chapter.

The title Attribute

The title attribute defines some extra information about an element.

The value of the title attribute will be displayed as a tooltip when you mouse over the element:

```
<p title="I'm a tooltip">This is a paragraph.</p>
```

Try it Yourself »

HTML Horizontal Rules

The <hr> tag defines a thematic break in an HTML page, and is most often displayed as a horizontal rule.

The <hr> element is used to separate content (or define a change) in an HTML page:

```
<h1>This is heading 1</h1>
<p>This is some text.</p>
<hr>
<h2>This is heading 2</h2>
<p>This is some other text.</p>
<hr>
```

Try it Yourself »

The <hr> tag is an empty tag, which means that it has no end tag.

HTML Line Breaks

The HTML <br> element defines a line break.

Use <br> if you want a line break (a new line) without starting a new paragraph:

```
<p>This is<br>a paragraph<br>with line breaks.</p>
```

Try it Yourself »

The <br> tag is an empty tag, which means that it has no end tag.

HTML <pre> Element

The HTML <pre> element defines preformatted text.

The text inside a <pre> element is displayed in a fixed-width font (usually Courier), and it preserves both spaces and line breaks:

```
<pre>
  My Bonnie lies over the ocean.


  My Bonnie lies over the sea.


  My Bonnie lies over the ocean.


  Oh, bring back my Bonnie to me.
</pre>
```

HTML Styles

The HTML style attribute is used to add styles to an element, such as color, font, size, and more.

```
I am Red
I am Blue
I am Big
```

Try it Yourself »

The HTML style attribute has the following syntax:

```
<tagname style="property:value;">
```

The ***property*** is a CSS property. The ***value*** is a CSS value.

You will learn more about CSS later in this tutorial.

Background Color

The CSS background-color property defines the background color for an HTML element.

Set the background color for a page to powderblue:

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

Try it Yourself »

Set background color for two different elements:

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

Try it Yourself »

Fonts

The CSS font-family property defines the font to be used for an HTML element:

```
<h1 style="font-family:verdana;">This is a heading</h1>
<p style="font-family:courier;">This is a paragraph.</p>
```

Try it Yourself »

Text Color

The CSS color property defines the text color for an HTML element:

```
<h1 style="color:blue;">This is a heading</h1>
<p style="color:red;">This is a paragraph.</p>
```

Try it Yourself »

Text Size

The CSS font-size property defines the text size for an HTML element:

```
<h1 style="font-size:300%;">This is a heading</h1>
<p style="font-size:160%;">This is a paragraph.</p>
```

Try it Yourself »

Text Alignment

The CSS text-align property defines the horizontal text alignment for an HTML element:

```
<h1 style="text-align:center;">Centered Heading</h1>
<p style="text-align:center;">Centered paragraph.</p>
```

Try it Yourself »

HTML Text Formatting

```
<b>This text is bold</b>
<strong>This text is important!</strong>
<i>This text is italic</i>
```

Try it Yourself »

HTML Colors

HTML colors are specified with predefined color names, or with RGB, HEX, HSL, RGBA, or HSLA values.

HTML Animated Images

HTML allows animated GIFs:

```
<img src="programming.gif" alt="Computer Man" style="width:48px;height:48px;">
```

Try it Yourself »

HTML Comments

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

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

```
<!-- This is a comment -->
<p>This is a paragraph.</p>
<!-- Remember to add more information here -->
```

Try it Yourself »

HTML Links

Links are found in nearly all web pages. Links allow users to click their way from page to page.

This example shows how to create a link to W3Schools.com:

```
<a href="https://www.w3schools.com/">Visit W3Schools.com!</a>
```

Try it Yourself »

Tip: Links can of course be styled with CSS, to get another look!

HTML Links - The target Attribute

By default, the linked page will be displayed in the current browser window. To change this, you must specify another target for the link.

The target attribute specifies where to open the linked document.

The target attribute can have one of the following values:

- _self - Default. Opens the document in the same window/tab as it was clicked
- _blank - Opens the document in a new window or tab
- _parent - Opens the document in the parent frame
- _top - Opens the document in the full body of the window

Use target="_blank" to open the linked document in a new browser window or tab:

```
<a href="https://www.w3schools.com/" target="_self">Visit W3Schools!</a>
<a href="https://www.w3schools.com/" target="_blank">Visit W3Schools!</a>
<a href="https://www.w3schools.com/" target="_parent">Visit W3Schools!</a>
<a href="https://www.w3schools.com/" target="_top">Visit W3Schools!</a>
```

Try it Yourself »

Absolute URLs vs. Relative URLs (Uniform Resource Locator)

There are two ways to specify the URL in the src attribute:

1. Absolute URL - Links to an external image that is hosted on another website.
Example: src="https://www.w3schools.com/images/img_girl.jpg".
Notes: External images might be under copyright. If you do not get permission to use it, you may be in violation of copyright laws. In addition, you cannot control external images; it can suddenly be removed or changed.
2. Relative URL - Links to an image that is hosted within the website. Here, the URL does not include the domain name. If the URL begins without a slash, it will be relative to the current page. Example: src="img_girl.jpg". If the URL begins with a slash, it will be relative to the domain. Example: src="/images/img_girl.jpg".
Tip: It is almost always best to use relative URLs. They will not break if you change domain.

```
<h2>Absolute URLs</h2>
<p><a href="https://www.w3.org/">W3C</a></p>
<p><a href="https://www.google.com/">Google</a></p>

<h2>Relative URLs</h2>
<p><a href="html_images.asp">HTML Images</a></p>
<p><a href="/css/default.asp">CSS Tutorial</a></p>
```

Try it Yourself »

HTML Links - Use an Image as a Link

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

```
<a href="comet.html">
<img src="smiley.gif" alt="HTML tutorial" style="width:42px;height:42px;">
</a>
```

Try it Yourself »

Link to an Email Address

Use mailto: inside the href attribute to create a link that opens the user's email program (to let them send a new email):

```
<a href="mailto:someone@example.com">Send email</a>
```

Try it Yourself »

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:

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

Try it Yourself »

Tip: Learn more about JavaScript in our JavaScript Tutorial.

Image Floating

Use the CSS float property to let the image float to the right or to the left of a text:

```
<p><img src="smiley.gif" alt="Smiley face" style="float:right;width:42px;height:42px;">
The image will float to the right of the text.</p>
```

```
<p><img src="smiley.gif" alt="Smiley face" style="float:left;width:42px;height:42px;">
The image will float to the left of the text.</p>
```

Try it Yourself »

Tip: To learn more about CSS Float, read our CSS Float Tutorial.

HTML Tables

HTML tables allow web developers to arrange data into rows and columns.

Company	Contact	Country
Alfreds Futterkiste	Maria Anders	Germany
Centro comercial Moctezuma	Francisco Chang	Mexico

Try it Yourself »

Define an HTML Table

A table in HTML consists of table cells inside rows and columns **table cells**.

A simple HTML table:

```
<table>
  <tr>
    <th>Company</th>    <th>Contact</th>    <th>Country</th>
  </tr>
  <tr>
    <td>Alfreds Futterkiste</td>    <td>Maria Anders</td>    <td>Germany</td>
  </tr>
  <tr>
    <td>Centro comercial Moctezuma</td>    <td>Francisco Chang</td>    <td>Mexico</td>
  </tr>
</table>
```

Try it Yourself »

Table Cells

Each table cell is defined by a <td> and a </td> tag.

Table Headers

Sometimes you want your cells to be headers, in those cases use the <th> tag instead of the <td> tag:

Let the first row be table headers:

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

```
  <th>Person 1</th>   <th>Person 2</th>   <th>Person 3</th>
 </tr>
 <tr>
  <td>Emil</td>   <td>Tobias</td>   <td>Linus</td>
 </tr>
 <tr>
  <td>16</td>   <td>14</td>   <td>10</td>  </tr>
</table>
```

Try it Yourself »

By default, the text in <th> elements are bold and centered, but you can change that with CSS.

HTML Lists

Unordered HTML List

An unordered list starts with the <ul> tag. Each list item starts with the <li> tag.
The list items will be marked with bullets (small black circles) by default:

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

Try it Yourself »

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:

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

Try it Yourself »

HTML Description Lists

HTML also supports description lists.
A description list is a list of terms, with a description of each term.
The <dl> tag defines the description list, the <dt> tag defines the term (name), and the <dd> tag describes each term:

```
<dl>
 <dt>Coffee</dt>
```

```
 <dd>- black hot drink</dd>
 <dt>Milk</dt>
 <dd>- white cold drink</dd>
</dl>
```

Try it Yourself »

HTML - Forms

HTML Form Controls

1) Text Input Controls
2) Checkboxes Controls
3) Radio Box Controls
4) Select Box Controls
5) File Select boxes
6) Hidden Controls
7) Clickable Buttons
8) Submit and Reset Button

Text Input Controls

There are three types of text input used on forms –

- **Single-line text input controls** – This control is used for items that require only one line of user input, such as search boxes or names. They are created using HTML **<input>** tag.
- **Multi-line text input controls** – This is used when the user is required to give details that may be longer than a single sentence. Multi-line input controls are created using HTML **<textarea>** tag.
- **Password input controls** – This is also a single-line text input but it masks the character as soon as a user enters it. They are also created using HTMl <input> tag.

```
<!DOCTYPE html>
<html>
 <head>    <title> Input Controls</title>  </head>
  <body>
    <form >

      First name: <input type = "text" name = "first_name" />
      <br>
      Last name: <input type = "text" name = "last_name" />
    </form>
  </body>

</html>
```

Password input controls

```
<!DOCTYPE html>
<html>
  <head>     <title>Password Input Control</title></head>
  <body>
    <form >
      User ID : <input type = "text" name = "user_id" />
      <br>
      Password: <input type = "password" name = "password" />
    </form>
  </body>

</html>
```

Multiple-Line Text Input Controls

```
<!DOCTYPE html>
<html>

  <head>
    <title>Multiple-Line Input Control</title>
  </head>

  <body>
    <form>
      Description : <br/>
      <textarea rows = "5" cols = "50" name = "description">
        Enter description here...
      </textarea>
    </form>
  </body>

</html>
```

Checkbox Control

```
<!DOCTYPE html>
<html>

  <head>     <title>Checkbox Control</title>   </head>
  <body>
    <form>
      <input type = "checkbox" name = "maths" value = "on"> Maths
      <input type = "checkbox" name = "physics" value = "on"> Physics
```

```
    </form>
  </body>

</html>
```

Radio Button Control

```
<!DOCTYPE html>
<html>

  <head>
    <title>Radio Box Control</title>
  </head>

  <body>
    <form>
      <input type = "radio" name = "subject" value = "maths"> Maths
      <input type = "radio" name = "subject" value = "physics"> Physics
    </form>
  </body>

</html>
```

Select Box Control

```
<!DOCTYPE html>
<html>
  <head>     <title>Select Box Control</title>   </head>
  <body>
    <form>
      <select name = "dropdown">
        <option value = "Maths" selected>Maths</option>
        <option value = "Physics">Physics</option>
      </select>
    </form>
  </body>

</html>
```

File Upload Box

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

```
<head>    <title>File Upload Box</title></head>
<body>
  <form>
    <input type = "file" name = "fileupload" accept = "image/*" />
  </form>
</body>

</html>
```

Button Controls

There are various ways in HTML to create clickable buttons. You can also create a clickable button using <input>tag by setting its type attribute to **button**. The type attribute can take the following values −

Sr	Type & Description
1	**Submit** This creates a button that automatically submits a form.
2	**Reset** This creates a button that automatically resets form controls to their initial values.
3	**Button** This creates a button that is used to trigger a client-side script when the user clicks that button.
4	**Image** This creates a clickable button but we can use an image as background of the button.

```
<!DOCTYPE html>
<html>
  <head    <title>Button types</title>  </head>
  <body>
    <form>
      <input type = "submit" name = "submit" value = "Submit" />
      <input type = "reset" name = "reset"  value = "Reset" />
      <input type = "button" name = "ok" value = "OK" />
      <input type = "image" name = "imagebutton" src = "/html/images/logo.png" />
    </form>
  </body>
</html>
```

Hidden Form Controls

Hidden form controls are used to hide data inside the page which later on can be pushed to the server. This control hides inside the code and does not appear on the actual page. For example, following hidden form is being used to keep current page number. When a user will click next

page then the value of hidden control will be sent to the web server and there it will decide which page will be displayed next based on the passed current page.

```
<!DOCTYPE html>
<html>
  <head>     <title>Hidden form</title></head>
  <body>
    <form>
      <p>This is page 10</p>
      <input type = "hidden" name = "pagename" value = "10" />
      <input type = "submit" name = "submit" value = "Submit" />
      <input type = "reset" name = "reset"  value = "Reset" />
    </form>
  </body>

</html>
```

Multiple Select Box

This is like a radio button and checkbox in one package. It is embedded in the page with two elements – a select element and an option, which is always nested inside select.
By default, the user can only pick one of the options. But with multiple attributes, you can let the user select more than one of the options.

```
<select>
    <option value="HTML">Select a Language</option>
    <option value="HTML">HTML</option>
    <option value="CSS">CSS</option>
    <option value="JavaScript">JavaScript</option>
    <option value="React">React</option>
</select>
```

How to Label HTML Inputs

Assigning labels to form controls is important. When they're properly connected to the input field through their for attribute and the input's id attribute, it's easier for the user to use as they can just click the label itself to access the input.

```
<label for="name">Name</label>
<input type="text" id="name" /> <br />
<label for="check">Agree with terms</label>
<input type="checkbox" id="check" />
```

How HTML Forms Work

When a user fills in a form and submits it with the submit button, the data in the form controls are sent to the server through GET or POST HTTP request methods.
So how is the server indicated? The form element takes an action attribute, which must have its value specified to the URL of the server. It also takes a method attribute, where the HTTP method it uses to convey the values to the server is specified.
This method could be GET or POST. With the GET method, the values entered by the user are visible in the URL when the data is submitted. But with POST, the values are sent in HTTP headers, so those values are not visible in the URL.
If a method attribute is not used in the form, it is automatically assumed that the user wants to use the GET method, because it's the default.
So when should you use the GET or POST method? Use the GET method for submitting non-sensitive data or retrieving data from a server (for example, during searches). Use the POST request when submitting files or sensitive data.

The Name Attribute for <input>

Notice that each input field must have a name attribute to be submitted.
If the name attribute is omitted, the value of the input field will not be sent at all.
This example will not submit the value of the "First name" input field:

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

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 file on the server when the user clicks on the submit button.
In the example below, the form data is sent to a file called "action_page.php". This file contains a server-side script that handles the form data:
On submit, send form data to "action_page.php":

```
<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>
```

The Target Attribute

The target attribute specifies where to display the response that is received after submitting the form.

The target attribute can have one of the following values:

Value	Description
_blank	The response is displayed in a new window or tab
_self	The response is displayed in the current window
_parent	The response is displayed in the parent frame
_top	The response is displayed in the full body of the window
Framename	The response is displayed in a named iframe

The default value is _self which means that the response will open in the current window.

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

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

The Method Attribute

The method attribute specifies the HTTP method to be used when submitting the form data. The form-data can be sent as URL variables (with method="get") or as HTTP post transaction (with method="post").

The default HTTP method when submitting form data is GET.

This example uses the GET method when submitting the form data:

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

Notes on GET:

- Appends the form data to the URL, in name/value pairs
- NEVER use GET to send sensitive data! (the submitted form data is visible in the URL!)
- The length of a URL is limited (2048 characters)
- Useful for form submissions where a user wants to bookmark the result
- GET is good for non-secure data, like query strings in Google

Notes on POST:

- Appends the form data inside the body of the HTTP request (the submitted form data is not shown in the URL)
- POST has no size limitations, and can be used to send large amounts of data.
- Form submissions with POST cannot be bookmarked

Tip: Always use POST if the form data contains sensitive or personal information!

The Autocomplete Attribute

The autocomplete attribute specifies whether a form should have autocomplete on or off. When autocomplete is on, the browser automatically complete values based on values that the user has entered before.

A form with autocomplete on:

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

Try it Yourself »

The Novalidate Attribute

The novalidate attribute is a boolean attribute.
When present, it specifies that the form-data (input) should not be validated when submitted.
A form with a novalidate attribute:

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

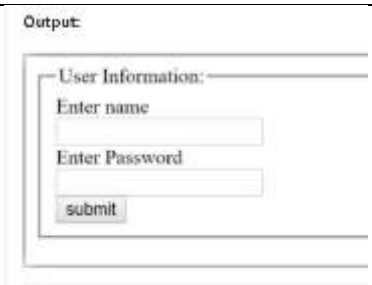
HTML <fieldset> element:

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>
</form>
```

Output:



HTML iFrame

An iframe or inline frame is used to display external objects including other web pages within a web page. An iframe pretty much acts like a mini web browser within a web browser. Also, the content inside an iframe exists entirely independent from the surrounding elements.
The basic syntax for adding an iframe to a web page can be given with:

```
<iframe src="URL"></iframe>
```

The URL specified in the src attribute points to the location of an external object or a web page.
The following example display "hello.html" file inside an iframe in current document.

Try this code »

```
<iframe src="hello.html"></iframe>
```

Setting Width and Height of an iFrame

The height and width attributes are used to specify the height and width of the iframe.

Try this code »

```
<iframe src="hello.html" width="400" height="200"></iframe>
```

You can also use CSS to set the width and height of an iframe, as shown here:

Try this code »

```
<iframe src="hello.html" style="width: 400px; height: 200px;"></iframe>
```

Removing Default Frameborder

The iframe has a border around it by default. However, if you want to modify or remove the iframe borders, the best way is to use the CSS border property.

The following example will simply render the iframe without any borders.

Try this code »

```
<iframe src="hello.html" style="border: none;"></iframe>
```

Similarly, you can use the border property to add the borders of your choice to an iframe. The following example will render the iframe with 2 pixels blue border.

Try this code »

```
<iframe src="hello.html" style="border: 2px solid blue;"></iframe>
```

HTML Entity

Some characters are reserved in HTML, e.g. you cannot use the less than (<) or greater than (>) signs or angle brackets within your text, because the browser could mistake them for markup, while some characters are not present on the keyboard like copyright symbol ©.

To display these special characters, they must be replaced with the character entities.

Character entity references, or entities for short, enable you to use the characters that cannot be expressed in the document's character encoding or that cannot be entered by a keyboard.

Frequently Used HTML Character Entities

Result	Description	Entity Name	Numerical reference
	non-breaking space		
<	less than	&lt;	&#60;
>	greater than	&gt;	&#62;
&	Ampersand	&amp;	&#38;
"	quotation mark	&quot;	&#34;
'	Apostrophe	&apos;	&#39;
¢	Cent	&cent;	&#162;
£	Pound	&pound;	&#163;
¥	Yen	&yen;	&#165;
€	Euro	&euro;	&#8364;

©	Copyright	&copy;	&#169;
®	registered trademark	&reg;	&#174;
™	Trademark	&trade;	&#8482;

You can use numeric character references, instead of entity names. A key benefit of using numeric character references is that, they have stronger browser support, and can be used to specify any Unicode character, whereas entities are limited to a subset of this.

Note: HTML entities names are case-sensitive! Please check out the HTML character entities reference for a complete list of character entities of special characters and symbols.

Tip: Nonbreaking space ( ) is used to create a blank space between two items that cannot be separated by a line break. They are also used to display multiple spaces since web browsers display only one space if multiple spaces are created using the spacebar key.

DHTML

DHTML stands for Dynamic Hypertext Markup language i.e., Dynamic HTML.
Dynamic HTML is not a markup or programming language but it is a term that combines the features of various web development technologies for creating the web pages dynamic and interactive.
The DHTML application was introduced by Microsoft with the release of the 4th version of IE (Internet Explorer) in 1997.

Components of Dynamic HTML

DHTML consists of the following four components or languages:

- HTML 4.0
- CSS
- JavaScript
- DOM.

HTML 4.0

HTML is a client-side markup language, which is a core component of the DHTML. It defines the structure of a web page with various defined basic elements or tags.

CSS

CSS stands for Cascading Style Sheet, which allows the web users or developers for controlling the style and layout of the HTML elements on the web pages.

JavaScript

JavaScript is a scripting language, which is done on a client-side. The various browser supports JavaScript technology. DHTML uses the JavaScript technology for accessing, controlling, and manipulating the HTML elements. The statements in JavaScript are the commands which tell the browser for performing an action.

DOM

DOM is the document object model. It is a w3c standard, which is a standard interface of programming for HTML. It is mainly used for defining the objects and properties of all elements in HTML.

Uses of DHTML

Following are the uses of DHTML (Dynamic HTML):

- It is used for designing the animated and interactive web pages that are developed in real-time.
- DHTML helps users by animating the text and images in their documents.
- It allows the authors for adding the effects on their pages.
- It also allows the page authors for including the drop-down menus or rollover buttons.
- This term is also used to create various browser-based action games.
- It is also used to add the ticker on various websites, which needs to refresh their content automatically.

Features of DHTML

Following are the various characteristics or features of DHTML (Dynamic HTML):

- Its simplest and main feature is that we can create the web page dynamically.
- **Dynamic Style** is a feature that allows the users to alter the font, size, color, and content of a web page.
- It provides the facility for using the events, methods, and properties. And, also provides the feature of code reusability.
- It also provides the feature in browsers for data binding.
- Using DHTML, users can easily create dynamic fonts for their web sites or web pages.
- With the help of DHTML, users can easily change the tags and their properties.
- The web page functionality is enhanced because the DHTML uses low-bandwidth effect.

Advantages of DHTML

Following are the various benefits or the advantages of DHTML (Dynamic HTML):

- Those web sites and web pages which are created using this concept are fast.
- There is no plug-in required for creating the web page dynamically.

- Due to the low-bandwidth effect by the dynamic HTML, the web page functionality is enhanced.
- This concept provides advanced functionalities than the static HTML.
- It is highly flexible, and the user can make changes easily in the web pages.

Disadvantages of DHTML

Following are the various disadvantages or limitations of DHTML (Dynamic HTML):

- The scripts of DHTML does not run properly in various web browsers. Or in simple words, we can say that various web browsers do not support the DHTML. It is only supported by the latest browsers.
- The coding of those websites that are created using DHTML is long and complex.
- For understanding the DHTML, users must know about HTML, CSS, and JavaScript. If any user does not know these languages, then it is a time-consuming and long process in itself.

Difference between HTML and DHTML

Following table describes the differences between HTML and DHTML:

HTML (Hypertext Markup language)	DHTML (Dynamic Hypertext Markup language)
1. HTML is simply a markup language.	1. DHTML is not a language, but it is a set of technologies of web development.
2. It is used for developing and creating web pages.	2. It is used for creating and designing the animated and interactive web sites or pages.
3. This markup language creates static web pages.	3. This concept creates dynamic web pages.
4. It does not contain any server-side scripting code.	4. It may contain the code of server-side scripting.
5. The files of HTML are stored with the .html or .htm extension in a system.	5. The files of DHTML are stored with the .dhtm extension in a system.
6. A simple page which is created by a user without using the scripts or styles called as an HTML page.	6. A page which is created by a user using the HTML, CSS, DOM, and JavaScript technologies called a DHTML page.
7. This markup language does not need database connectivity.	7. This concept needs database connectivity because it interacts with users.

DHTML JavaScript

JavaScript can be included in HTML pages, which creates the content of the page as dynamic. We can easily type the JavaScript code within the <head> or <body> tag of a HTML page. If we want to add the external source file of JavaScript, we can easily add using the <src> attribute.

Following are the various examples, which describes how to use the JavaScript technology with the DHTML:

Document.write() Method

The document.write() method of JavaScript, writes the output to a web page.

Example 1: Display "Hello World" in webpage, using JS

```
1)  <HTML>
2)  <head>
3)  <title>  Method of a JavaScript  </title>
4)  </head>
5)  <body>
6)  <script type="text/javascript">
7)  document.write("Hello world");
8)  </script>
9)  </body>
10) </html>
```

Example 2: display date and time using JS

```
1)  <html>
2)  <head>
3)  <title>  DHTML with JavaScript  </title>
4)  </head>
5)  <body>
6)  <p> displaying Date and time
7)  <script type="text/javascript">
8)  document.write (Date());
9)  </script>
10) </body>
11) </html>
```

Example 3: Function call in JS.

```
12) <html>
13) <head>
```

```
14) <title>
15) DHTML with JavaScript
16) </title>
17) <script type="text/javascript">
18) function dateandtime()
19) {
20) alert(Date());
21) }
22) </script>
23) </head>
24) <body>
25) dateandtime()
26) </body>
27) </html>
```

JavaScript and HTML event

A JavaScript code can also be executed when some event occurs. Suppose, a user clicks an HTML element on a webpage, and after clicking, the JavaScript function associated with that HTML element is automatically invoked. And, then the statements in the function are performed.

Example 4: shows the current date and time with the JavaScript and **HTML event (Onclick)**.

```
1)  <html>
2)  <head>
3)  <title>
4)  DHTML with JavaScript
5)  </title>
6)  <script type="text/javascript">
7)  function dateandtime()
8)  {
9)  alert(Date());
10) }
11) </script>
12) </head>
13) <body>
14)  <p> Click here # <a href="#" onClick="dateandtime();"> Date and Time </a> </p>
15) </body>
16) </html>
```

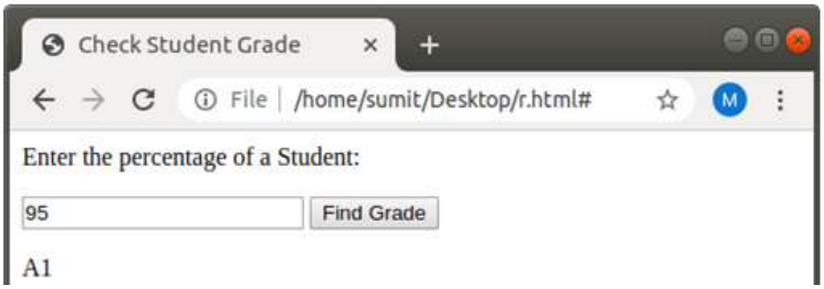
JavaScript and HTML **DOM**

Example 5: decision control statement (if-else statement) in JS and HTML DOM.

```
<html>
<head>
<title> Check Student Grade
</title>
</head>
<body>
<p>Enter the percentage of a Student:</p>
<input type="text" id="percentage">
<button type="button" onclick="checkGrade()">  Find Grade  </button>
<p id="demo"></p>

<script type="text/javascript">
function checkGrade()
{
var x,p, text;
p = document.getElementById("percentage").value;
x=parseInt(p);
if (x>90 && x <= 100)
{
document.getElementById("demo").innerHTML = "A1";
}
else if (x>80 && x <= 90)
{
document.getElementById("demo").innerHTML = "A2";
}
else if (x>70 && x <= 80)
{
document.getElementById("demo").innerHTML = "A3";
}
else if (x <= 70)
{
document.getElementById("demo").innerHTML = "B";
}
 }
</script>
</body>
</html>
```

Output:



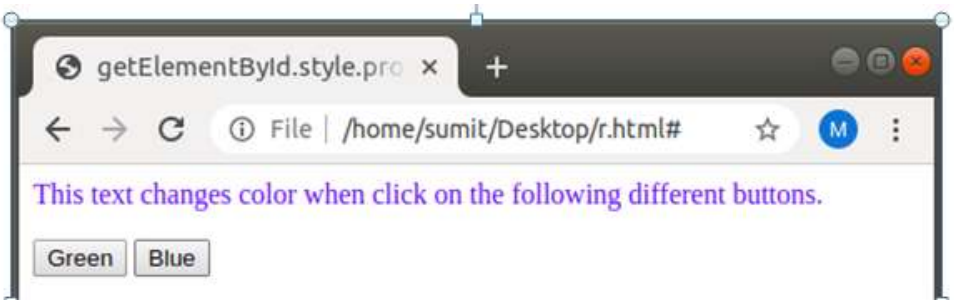
CSS with JavaScript in DHTML

With version 4 of HTML, JavaScript code can also change the style such as size, color, and face of an HTML element.

Example 6: The following example changes the color of a text.

```
1.  <html>
2.  <head>
3.  <title>  Example to change the font color </title>
4.  </head>
5.  <body>
6.  <p id="demo"> This text will changes color when buttons are clicked. </p>
7.  <button onclick="change_Color('green');"> Green </button>
8.  <button onclick="change_Color('blue');"> Blue </button>
9.  <script type="text/javascript">
10. function change_Color(newColor)
11. {
12.     var element = document.getElementById('demo').style.color = newColor;
13. }
14. </script>
15. </body>
16. </html>
```

Output:



DHTML CSS

We can easily use the CSS with the DHTML page with the help of JavaScript and HTML DOM. With the help of **this.style.property=new style** statement, we can change the style of the currently used HTML element. Or, we can also update the style of any particular HTML element by **document.getElementById(id).style.property = new_style** statement.

Example 7: The following example uses the DHTML CSS for changing the style of current element:

```
1. <html>
2. <head>
3. <title>  Changes current HTML element  </title>
4. </head>
5. <body>
6. <center>
7. <h1 onclick="this.style.color='blue'"> This is a COMET Site </h1>
8. <center>
9. </body>
10. </html>
```

DHTML Events

An event is defined as changing the occurrence of an object.

It is compulsory to add the events in the DHTML page. Without events, there will be no dynamic content on the HTML page. The event is a term in the HTML, which triggers the actions in the web browsers.

Suppose, any user clicks an HTML element, then the JavaScript code associated with that element is executed. Actually, the event handlers catch the events performed by the user and then execute the code.

Example of events:

1. Click a button.
2. Submitting a form.
3. An image loading or a web page loading, etc.

Following table describes the Event Handlers used in the DHTML:
Following are the different examples using the different event handlers, which helps us to understand the concept of DHTML events:

Example 8: This example uses the onclick event handler, which is used to change the text after clicking.

```
1. <html>
2. <head>
3. <title>
4. Example of onclick event
5. </title>
6. <script type="text/javascript">
7. function ChangeText(ctext)
8. {
9. ctext.innerHTML=" Hi COMET! ";
10. }
11. </script>
12. </head>
13. <body>
14. <font color="red"> Click on the Given text for changing it: <br>
15. </font>
16. <font color="blue">
17. <h1 onclick="ChangeText(this)"> Hello World! </h1>
18. </font>
19. </body>
20. </html>
```

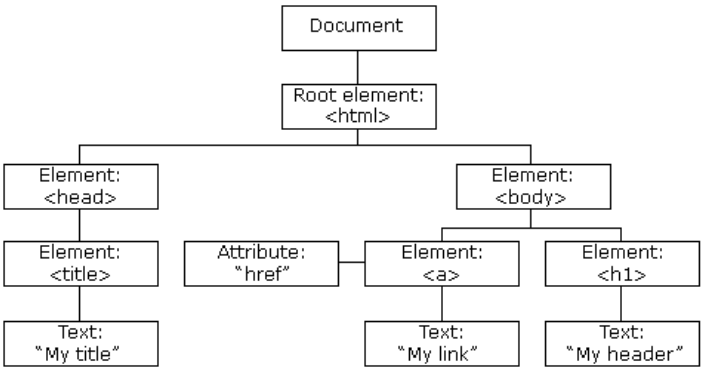
DHTML DOM

DHTML DOM stands for Dynamic HTML Document Object Model.

The HTML DOM is:

- A standard object model for HTML
- A standard programming interface for HTML
- Platform- and language-independent
- A W3C standard

The DOM presents HTML as a tree-structure (a node tree), with elements, attributes, and text:



The HTML DOM defines the **objects and properties** of all HTML elements, and the **methods** (interface) to access them.
In other words:

The HTML DOM is a standard for how to get, change, add, or delete HTML elements.
It is a w3c standard, which is a standard interface of programming for HTML. It is mainly used for defining the objects and properties of all elements in HTML. It also defines the methods for accessing the HTML elements.

Example 9: The following program helps in understanding the concept of DHTML DOM. This example changes the color of text and displays the text in red colour automatically.

```
1. <html>
2. <head>
3. <title>
4. Example of DHTML DOM
5. </title>
6. </head>
7. <body>
8. <font color = "blue">
9. <p id="demo"> This text changes color when the page loaded. </p>
10. </font>
11. <script type="text/javascript">
12. document.getElementById('demo').style.color = "red";
13. </script>
14. </body>
15. </html>
```

Example 10:
Using the HTML DOM to Change an HTML Element
The HTML DOM can be used to change the content of an HTML element:

```
<html>
<body>
<h1 id="header">Old Header</h1>
<script type="text/javascript">
document.getElementById("header").innerHTML="New Header";
</script>
</body>
</html>
```

HTML output:

New Header

Example explained:

- The HTML document contains a header with id="header"
- The DOM is used to get the element with id="header"
- A JavaScript is used to change the HTML content (innerHTML)

Example 11:

Using the HTML DOM to Change an HTML Attribute

- The HTML DOM can be used to change the attribute of an HTML element:

```
<html>
<body>
<img id="image" src="smiley.gif">
<script type="text/javascript">
document.getElementById("image").src="landscape.jpg";
</script>
</body>
</html>
```

- HTML output:

Event handlers

An event handler allows you to execute code when an event occurs.
Events are generated by the browser when the user clicks an element, when the page loads, when a form is submitted, etc.

Event	Occurs when...
Onabort	a user aborts page loading
Onblur	a user leaves an object
Onchange	a user changes the value of an object
Onclick	a user clicks on an object
Ondblclick	a user double-clicks on an object
Onfocus	a user makes an object active
Onkeydown	a keyboard key is on its way down
Onkeypress	a keyboard key is pressed
Onkeyup	a keyboard key is released
Onload	a page is finished loading
Onmousedown	a user presses a mouse-button

Onmousemove	a cursor moves on an object
Onmouseover	a cursor moves over an object
Onmouseout	a cursor moves off an object
Onmouseup	a user releases a mouse-button
Onreset	a user resets a form
Onselect	a user selects content on a page
Onsubmit	a user submits a form
Onunload	a user closes a page

A header changes when the user clicks it:

```
<h1 onclick="this.innerHTML='Ooops!'">Click on this text</h1>
```

Try it yourself

Example 12:

You can also add a script in the head section of the page and then call the function from the event handler:

```
<html>
<head>
<script type="text/javascript">
function changetext(id)
{
id.innerHTML="Ooops!";
}
</script>
</head>
<body>
<h1 onclick="changetext(this)">Click on this text</h1>
</body>
</html>
```

HTML Styles - CSS

Cascading Style Sheets (CSS) is used to format the layout of a webpage.
With CSS, you can control the color, font, the size of text, the spacing between elements, how elements are positioned and laid out, what background images or background colors are to be used, different displays for different devices and screen sizes, and much more!

A CSS comprises of style rules that are interpreted by the browser and then applied to the corresponding elements in your document. A style rule is made of three parts −

- **Selector** − A selector is an HTML tag at which a style will be applied. This could be any tag like <h1> or <table> etc.
- **Property** − A property is a type of attribute of HTML tag. Put simply, all the HTML attributes are converted into CSS properties. They could be *color*, *border* etc.
- **Value** − Values are assigned to properties. For example, *color* property can have value either *red* or *#F1F1F1* etc.

Example − You can define a table border as follows −

table{ border :1px solid #C00; }

Here table is a selector and border is a property and given value *1px solid #C00* is the value of that property. You can define selectors in various simple ways based on your comfort. Let me put these selectors one by one.

Application / Advantages of CSS

- **CSS saves time** − You can write CSS once and then reuse same sheet in multiple HTML pages. You can define a style for each HTML element and apply it to as many Web pages as you want.
- **Pages load faster** − If you are using CSS, you do not need to write HTML tag attributes every time. Just write one CSS rule of a tag and apply it to all the occurrences of that tag. So less code means faster download times.
- **Easy maintenance** − To make a global change, simply change the style, and all elements in all the web pages will be updated automatically.
- **Superior styles to HTML** − CSS has a much wider array of attributes than HTML, so you can give a far better look to your HTML page in comparison to HTML attributes.
- **Multiple Device Compatibility** − Style sheets allow content to be optimized for more than one type of device. By using the same HTML document, different versions of a website can be presented for handheld devices such as PDAs and cell phones or for printing.
- **Global web standards** − Now HTML attributes are being deprecated and it is being recommended to use CSS. So its a good idea to start using CSS in all the HTML pages to make them compatible to future browsers.

Using CSS

CSS can be added to HTML documents in 3 ways:

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

The most common way to add CSS, is to keep the styles in external CSS files. However, in this tutorial we will use inline and internal styles, because this is easier to demonstrate, and easier for you to try it yourself.

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.
The following example sets the text color of the <h1> element to blue, and the text color of the <p> element to red:

```
<h1 style="color:blue;">A Blue Heading</h1>
<p style="color:red;">A red paragraph.</p>
```

Try it Yourself »

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.
The following example sets the text color of ALL the <h1> elements (on that page) to blue, and the text color of ALL the <p> elements to red. In addition, the page will be displayed with a "powderblue" background color:

```
1) <html>
2) <head>
        <style>
            body {background-color: powderblue;}
            h1 {color: blue;}
            p {color: red;}
        </style>
3) </head>
4) <body>
5) <h1>This is a heading</h1>
6) <p>This is a paragraph.</p>
7) </body>
8) </html>
```

Try it Yourself »

External CSS

An external style sheet (.CSS file) is used to define the style for many HTML pages.
To use an external style sheet, add a link to it in the <head> section of each HTML page:

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

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

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

Try it Yourself »

The 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 what the "styles.css" file looks like:

"styles.css":

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

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

CSS Colors, Fonts and Sizes

```
<!DOCTYPE html>
<html>
<head>
<style>
h1 {
  color: blue;
  font-family: verdana;
  font-size: 300%;
}
p {
  color: red;
  font-family: courier;
  font-size: 160%;
}
</style>
</head>
<body>
<h1>This is a heading</h1>
```

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

</body>
</html>
```

Try it Yourself »

CSS color

You can specify your color values in various formats. Following table lists all the possible formats –

Format	Syntax	Example
Hex Code	#RRGGBB	p{color:#FF0000;}
Short Hex Code	#RGB	p{color:#6A7;}
RGB %	rgb(rrr%,ggg%,bbb%)	p{color:rgb(50%,50%,50%);}
RGB Absolute	rgb(rrr,ggg,bbb)	p{color:rgb(0,0,255);}
keyword	aqua, black, etc.	p{color:teal;}

CSS background

This chapter teaches you how to set backgrounds of various HTML elements. You can set the following background properties of an element –

- The **background-color** property is used to set the background color of an element.

```
<p style = "background-color:yellow;">
      This text has a yellow background color.
   </p>
```

- The **background-image** property is used to set the background image of an element.

```
<style>
      body {
        background-image: url("/css/images/css.jpg");
        background-color: #cccccc;
      }
   </style>
```

- The **background-repeat** property is used to control the repetition of an image in the background.

```
<style>
    body {
      background-image: url("/css/images/css.jpg");
      background-repeat: repeat;
    }
  </style>
```

- The **background-position** property is used to control the position of an image in the background.

```
<style>
    body {
      background-image: url("/css/images/css.jpg");
      background-position:100px;
    }
  </style>
```

Font

- The **font-family** property is used to change the face of a font.

```
<p style = "font-family:georgia,garamond,serif;">
    This text is rendered in either georgia, garamond, or the
    default serif font depending on which font  you have at your system.
  </p>
```

- The **font-style** property is used to make a font italic or oblique.

```
<p style = "font-style:italic;">
    This text will be rendered in italic style
  </p>
```

- The **font-weight** property is used to increase or decrease how bold or light a font appears.

```
<p style = "font-weight:bold;">
    This font is bold.
  </p>

  <p style = "font-weight:bolder;">
    This font is bolder.
  </p>

  <p style = "font-weight:500;">
    This font is 500 weight.
```

```
    </p>
```

- The **font-size** property is used to increase or decrease the size of a font.

```
<p style = "font-size:20px;">
    This font size is 20 pixels
</p>

<p style = "font-size:small;">
    This font size is small
</p>
```

CSS Text

- The **color** property is used to set the color of a text.

```
<p style = "color:red;">
    This text will be written in red.
</p>
```

- The **direction** property is used to set the text direction.
  The following example demonstrates how to set the direction of a text. Possible values are *ltr or rtl*.

```
<p style = "direction:rtl;">
    This text will be rendered from right to left
</p>
```

- The **letter-spacing** property is used to add or subtract space between the letters that make up a word.

```
<p style = "letter-spacing:5px;">
    This text is having space between letters.
</p>
```

- The **word-spacing** property is used to add or subtract space between the words of a sentence.

```
<p style = "word-spacing:5px;">
    This text is having space between words.
</p>
```

- The **text-indent** property is used to indent the text of a paragraph.

```
<p style = "text-indent:1cm;">
    This text will have first line indented by 1cm and this line will remain at
    its actual position this is done by CSS text-indent property.
```

```
    </p>
```

- The **text-align** property is used to align the text of a document.

```
<p style = "text-align:right;">
   This will be right aligned.
</p>

<p style = "text-align:center;">
   This will be center aligned.
</p>

<p style = "text-align:left;">
   This will be left aligned.
</p>
```

- The **text-decoration** property is used to underline, overline, and strikethrough text.

```
<p style = "text-decoration:underline;">
   This will be underlined
</p>

<p style = "text-decoration:line-through;">
   This will be striked through.
</p>

<p style = "text-decoration:overline;">
   This will have a over line.
</p>

<p style = "text-decoration:blink;">
   This text will have blinking effect
</p>
```

- The **text-transform** property is used to capitalize text or convert text to uppercase or lowercase letters.

```
<p style = "text-transform:capitalize;">
   This will be capitalized
</p>

<p style = "text-transform:uppercase;">
   This will be in uppercase
</p>

<p style = "text-transform:lowercase;">
```

```
    This will be in lowercase
  </p>
```

CSS Image

CSS plays a good role to control image display. You can set the following image properties using CSS.

- The **border** property is used to set the width of an image border.

```
<body>
   <img style = "border:0px;" src = "/css/images/logo.png" />
   <br />
   <img style = "border:3px dashed red;" src = "/css/images/logo.png" />
 </body>
```

- The **height** property is used to set the height of an image.

```
<body>
   <img style = "border:1px solid red; height:100px;" src = "/css/images/logo.png" />
   <br />
   <img style = "border:1px solid red; height:50%;" src = "/css/images/logo.png" />
 </body>
```

- The **width** property is used to set the width of an image.

```
<body>
   <img style = "border:1px solid red; width:150px;" src = "/css/images/logo.png" />
   <br />
   <img style = "border:1px solid red; width:100%;" src = "/css/images/logo.png" />
 </body>
```

Links

- **Set the Color of Links**

```
<html>
  <head>
    <style type = "text/css">
      a:link {color:#000000}
    </style>
  </head>

  <body>
    <a href = "">Link</a>
  </body>
```

```
</html>
```

- **Set the Color of Visited Links**

```
<html>
  <head>
    <style type = "text/css">
      a:visited {color: #006600}
    </style>
  </head>

  <body>
    <a href = ""> link</a>
  </body>
</html>
```

- **Change the Color of Links when Mouse is Over**

```
<html>
  <head>
    <style type = "text/css">
      a:hover {color: #FFCC00}
    </style>
  </head>

  <body>
    <a href = "">Link</a>
  </body>
</html>
```

- **Change the Color of Active Links**

```
<html>
  <head>
    <style type = "text/css">
      a:active {color: #FF00CC}
    </style>
  </head>

  <body>
    <a href = "">Link</a>
  </body>
</html>
```

CSS Box Model

All HTML elements can be considered as boxes. In CSS, the term "box model" is used when talking about design and layout. The CSS box model is essentially a box that wraps around every HTML element. It consists of: margins, borders, padding, and the actual content. The image below illustrates the box model:

Explanation of the different parts:

- **Content** - The content of the box, where text and images appear
- **Padding** - Clears an area around the content. The padding is transparent
- **Border** - A border that goes around the padding and content
- **Margin** - Clears an area outside the border. The margin is transparent

The box model allows us to add a border around elements, and to define space between elements.

Demonstration of the box model:

```
div {
  width: 300px;
  border: 15px solid green;
  padding: 50px;
  margin: 20px;
}
```

CSS Border

The CSS border property defines a border around an HTML element.

Tip: You can define a border for nearly all HTML elements.

Use of CSS border property:

```
p {
  border: 2px solid powderblue;
}
```

CSS Padding

The CSS padding property defines a padding (space) between the text and the border.

Use of CSS border and padding properties:

```
p {
  border: 2px solid powderblue;
  padding: 30px;
}
```

CSS Margin

The CSS margin property defines a margin (space) outside the border.

Use of CSS border and margin properties:

```
p {
  border: 2px solid powderblue;
  margin: 50px;
}
```

Styling Forms

```
input[type=text]

{
  width: 100%;
  padding: 12px 20px;
  margin: 8px 0;
  box-sizing: border-box;
}
```

Styling Select Menus

```
select {
  width: 100%;
  padding: 16px 20px;
  border: none;
  border-radius: 4px;
  background-color: #f1f1f1;
}
```

Styling Input Buttons

```
input[type=button], input[type=submit], input[type=reset] {
  background-color: #4CAF50;
  border: none;
  color: white;
  padding: 16px 32px;
  text-decoration: none;
  margin: 4px 2px;
  cursor: pointer;
}
```

Assignments

Create a web page COMET.html to –

1) Display hello world
2) Display 6 different heading types
3) Display a paragraph in web page
4) Link webpage to website https://cometservices.org/
5) Display your picture
6) Display your picture in different heights and width
7) Show the use of alt attribute
8) Show the use of style attribute
9) Show the use of title attribute with paragraph tag
10) Show the use of line breaks.
11) Show the use of pre-element
12) Write 5 lines about India with different color
13) Write 5 lines about India with different font sizes
14) Show web page with different background colors
15) Display text in different fonts.
16) Show the use of comments in html
17) Show 4 links each link should have different target attributes.
18) Display image with absolute url
19) Display image with relative url
20) Use Image as a hyper link
21) Use button as a hyper link

22) Use links to open e-mail program
23) Create a Table with 5 columns (name, age, address, gender, class,) and 4 rows (details about 4 students)
24) Show table with specific column width and heights.
25) Link to an email address
26) Use button as a link
27) Show Floating Image
28) Show animated image
29) Display different types of lists.
30) Display HTML form with all form elements.
31) Display different types of Input text boxes.
32) Display different types of buttons.
33) Display the use of HTML entities (display <, > and ?
34) Create a webpage (COMET.html) and display 4 frames in this web page (comet1.html, comet2.html, comet3.html, comet4.html)
35) Show the use of field set element
36) WAP to demonstrate the use of JS in html page
37) WAP to display your name in html page using JS
38) WAP to display DateTime in html page using JS
39) Show the use of iframe element
40) Use html character entities
41) Apply CSS to different tags, change Color, Font, Size
42) Apply CSS to Text, image and link tag
43) Create a HTML page with Inline CSS.
44) Create a HTML page with Internal CSS.
45) Demonstrate External CSS.
46) Execute and analyze example 1 in DHTML section
47) Execute and analyze example 2 in DHTML section
48) Execute and analyze example 3 in DHTML section
49) Execute and analyze example 4 in DHTML section
50) Execute and analyze example 5 in DHTML section
51) Execute and analyze example 6 in DHTML section
52) Execute and analyze example 7 in DHTML section
53) Execute and analyze example 8 in DHTML section
54) Execute and analyze example 9 in DHTML section
55) Execute and analyze example 10 in DHTML section
56) Execute and analyze example 11 in DHTML section
57) Execute and analyze example 12 in DHTML section