

## HTML

HTML stands for HyperText Markup Language. HTML is the basic building block of World Wide Web.

Hypertext is text displayed on a computer or other electronic device with references to other text that the user can immediately access, usually by a mouse click or key press.

Apart from text, hypertext may contain tables, lists, forms, images, and other presentational elements. It is an easy-to-use and flexible format to share information over the Internet.

Markup languages use sets of markup tags to characterize text elements within a document, which gives instructions to the web browsers on how the document should appear.

HTML was originally developed by Tim Berners-Lee in 1990. He is also known as the father of the web. In 1996, the World Wide Web Consortium (W3C) became the authority to maintain the HTML specifications. HTML also became an international standard (ISO) in 2000. HTML5 is the latest version of HTML. HTML5 provides a faster and more robust approach to web development.

### What You Can Do with HTML

There are lot more things you can do with HTML.

- You can publish documents online with text, images, lists, tables, etc.
- You can access web resources such as images, videos or other HTML document via hyperlinks.
- You can create forms to collect user inputs like name, e-mail address, comments, etc.
- You can include images, videos, sound clips, flash movies, applications and other HTML documents directly inside an HTML document.
- You can create offline version of your website that work without internet.
- You can store data in the user's web browser and access later on.
- You can find the current location of your website's visitor.

### Creating Your First HTML Document

Let's walk through the following steps. At the end of this tutorial, you will have made an HTML file that displays "Hello world" message in your web browser.

#### Step 1: Creating the HTML file

Open up your computer's plain text editor and create a new file.

#### Step 2: Type some HTML code

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>A simple HTML document</title>
</head>
<body>
  <p>Hello World!</p>
</body>
</html>
```

#### Step 3: Saving the file

Now save the file on your desktop as "myfirstpage.html".

To open the file in a browser. Navigate to your file then double click on it. It will open in your default Web browser. If it does not, open your browser and drag the file to it.

### Explanation of code

You might wonder what that code was all about. Well, let's find out.

- The first line `<!DOCTYPE html>` is the [document type declaration](#). It instructs the web browser that this document is an HTML5 document. It is case-insensitive.
- The `<head>` element is a container for the tags that provides information about the document, for example, `<title>` tag defines the title of the document.
- The `<body>` element contains the document's actual content (paragraphs, links, images, tables, and so on) that is rendered in the web browser and displayed to the user.

### HTML Tags and Elements

HTML is written in the form of HTML elements consisting of markup tags. These markup tags are the fundamental characteristic of HTML. Every markup tag is composed of a keyword, surrounded by angle brackets, such as `<html>`, `<head>`, `<body>`, `<title>`, `<p>`, and so on.

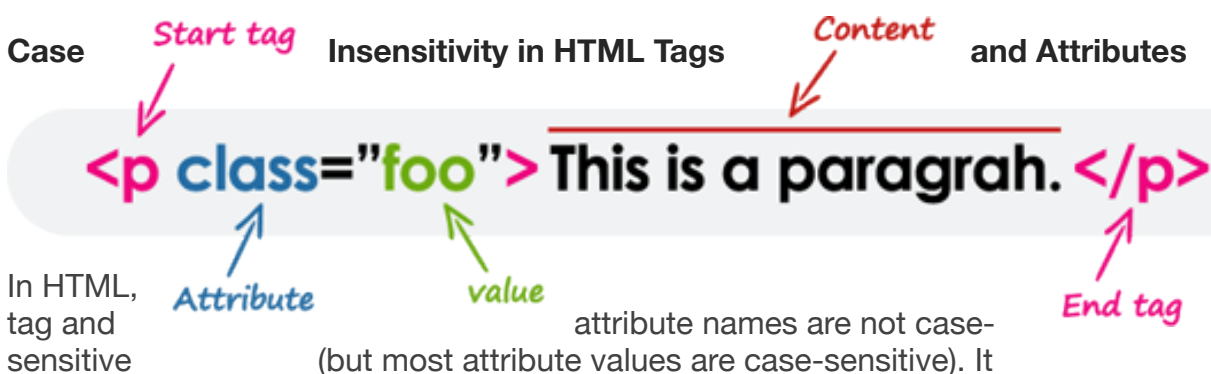
HTML tags normally come in pairs like `<html>` and `</html>`. The first tag in a pair is often called the opening tag (or start tag), and the second tag is called the closing tag (or end tag).

An opening tag and a closing tag are identical, except for a slash (/) after the opening angle bracket of the closing tag, to tell the browser that the command has been completed.

```
<p>This is a paragraph.</p>
<!-- Paragraph with nested element -->
<p>
    This is <b>another</b> paragraph.
</p>
```

### HTML Element Syntax

An HTML element is an individual component of an HTML document. It represents semantics, or meaning. For example, the `title` element represents the title of the document.



means the tag <P>, and the tag <p> defines the same thing in HTML which is a paragraph.

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

<P>This is also a valid paragraph.</P>

### Empty HTML Elements

Empty elements (also called self-closing or void elements) are not container tags — that means, you can not write <hr>some content</hr> or <br>some content</br>.

<p>This paragraph contains <br> a line break.</p>



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

### Address Tag

**Address tag** is used to add postal address in a webpage. Address is function element, and italic in style. Default font-size is 1em.

<address>

Tech Altum <br>

3rd Floor, OM Complex, <br>

Sec 15, Noida UP, 201301<br>

<abbr title="Phone No">Ph</abbr> (0120) 4280181

</address>

### Blockquote

**Blockquote tag** is used to write a **Block Level quotation** with extended quotation in HTML. This is mainly a text with Extended Quotation. By-default there is a margin of 40px from left and right in blockquote.

<blockquote> blockquote Tag </blockquote>

### PRE Tag

All HTML Textual Tags ignore white space and line break. Means text is not formatted.

PRE Tag is the only tag which follow the exact format in which you are writing.

<pre> PRE Tag </pre>

<pre> PRE Tag                      with white spacing more than one </pre>

<pre> PRE

Tag with line break </pre>                      x

### <b> tag

**html b tag** is used to bold text. The actual definition is offset text conventionally styled in bold as per Html5 documentation. b tag is inline level element.

<p>This text is <b>bold</b>.</p>

### <i> tag

**html i tag** is used to italic text. The actual definition is offset text conventionally styled in italic as per Html5 documentation. i tag is also inline level element.

<p>This text is <i>italic</i>.</p>

<s> tag

**html s tag** is used to struck text which is incorrect or removed. s tag is also inline level element. For semantic purpose, use <del> tag.

<p>This text is <s>struck</s>.</p>

<u> tag

**html u tag** is used to underline text. u tag is also inline level element.

<p>This text is <u>underlined</u>.</p>

<small> tag

**html small tag** is used to print small text. The default font-size of small tag is smaller (relative to parent element). small tag is also inline level element.

<h1>This text is <small>small</small>.</h1>

<p>This text is <small>small</small>.</p>

## HTML Headings

HTML includes Six headings elements. These headings are <h1>, <h2>, <h3>, <h4>, <h5> and <h6>.

**Heading elements** are used to write **headings** in a webpage, whereas p tag is only for plain text. All Headings are **bold** and **block level elements**.

Font size of all heading gradually decreases. For example h1 is twice of p, h2 is 1.5% of p tag, h3 is 1.17% of p tag and h4 and p tags are of same font size. h5 and h6 are smaller than p tag.

<h1> Heading 1 </h1>

<h2> Heading 2 </h2>

<h3> Heading 3 </h3>

<h4> Heading 4 </h4>

<h5> Heading 5 </h5>

<h6> Heading 6 </h6>

Example:

<!doctype html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>Web Designing</title>

</head>

<body>

<h1>Web Designing</h1>

<p>Description for web designing</p>

<h2>HTML</h2>

<p>Description for HTML</p>

<h3>HTML 4</h3>

<p>Description for HTML4</p>

<h3>HTML 5</h3>

<p>Description for HTML5</p>

<h2>CSS</h2>

```

<p>Description for CSS</p>
  <h3>CSS 2</h3>
  <p>Description for css2</p>
  <h3>CSS 3</h3>
  <p>Description for css3</p>
<h2>Javascript</h2>
<p>Description for JavaScript</p>
  <h3>Javascript</h3>
  <p>Description for Javascript</p>
  <h3>Jquery</h3>
  <p>Description for JQuery</p>
</body>
</html>

```

## HTML Block and Inline Elements

**HTML Elements** are classified as **Block Level** and **Inline Level** elements on the basics of their **display**. Some elements display as blocks and some inline.

To check an HTML Element is block or inline, open browser inspect and check display property in user agent stylesheet. If its display: block, it is block level, else it is inline level.

### Block Vs Inline level Elements

<p> <div> <h1> <h2> <address> etc are **block level elements**, whereas <b>, <strong>, <i>, <span>, <u> and <s> are **inline level elements**.

A **block level element** can have both inline and block elements as children or descendants, but **inline level elements** can have only inline elements as children or descendant. (*Except anchor tag or hyperlink*).

**HTML Block level elements** are elements who behave like blocks, like <p>, <h1>, <div>, <ul>, <ol>, <pre> and <address>. These elements always starts from a new line and occupy full width of parent element. **Block elements** can contain both **inline elements** and **block elements**. Here are some **block elements**.

### HTML Block Level Elements List

Element Name	Code	Use
<b>Html Tag</b>	<html> </html>	Root tag to build html page.
<b>Body Tag</b>	<body> </body>	To group visible content of a webpage.
<b>Para Tag</b>	<p> </p>	Create new <b>paragraph</b>
<b>Pre Tag</b>	<pre> </pre>	Create <b>pre formatted text</b> .
<b>hr</b>	<hr>	<b>Thematic Break</b> or formally known as <b>Horizontal rule</b> , used to break with 2px gradient shadow.
<b>Blockquote</b>	<blockquote> </blockquote>	Create a <b>blockquote</b> from new line.
<b>Div Tag</b>	<div> </div>	Create new New <b>Division</b>

<b>ul Tag</b>	<code>&lt;ul&gt; &lt;/ul&gt;</code>	Create new <b>Unordered List</b> . <a href="#">Unordered List</a>
<b>ol Tag</b>	<code>&lt;ol&gt; &lt;/ol&gt;</code>	Create new <b>Ordered List</b> . <a href="#">Ordered List</a>
<b>Address</b>	<code>&lt;address&gt; &lt;/address&gt;</code>	Create <b>Postal Address</b>
<b>Headings</b>	<code>&lt;h1&gt; &lt;/h1&gt;</code> , <code>&lt;h2&gt; &lt;/h2&gt;</code> till <code>&lt;h6&gt; &lt;/h6&gt;</code>	Create <b>Headings</b> and sub-headings. <a href="#">HTML Headings</a>
<b>Form Tag</b>	<code>&lt;form&gt; &lt;/form&gt;</code>	Used to group <b>Form controls and send form data..</b> <a href="#">HTML Form</a>
<b>Fieldset</b>	<code>&lt;fieldset&gt; &lt;/fieldset&gt;</code>	This is a fieldset, used to group form element.

HTML **inline elements** always start in the same line. Their width is equal to their content. Maximum **inline elements** are presentational, for exp, `<b>`, `<i>`, `<s>`, `<u>`. Some functional **inline elements** are `<strong>`, `<em>`, `<del>`, `<time>` etc.

HTML Inline Level Elements List

Element Name	Code	Use	Example
span Tag	<code>&lt;span&gt; &lt;/span&gt;</code>	Used to group <b>inline elements</b> .	This is a <b>span</b> .
anchor Tag	<code>&lt;a href=""&gt;Link&lt;/a&gt;</code>	Used to create <b>hyperlinks</b> .	I am <a href="#">hyperlink</a> .
b Tag	<code>&lt;b&gt; &lt;/b&gt;</code>	Used to give <b>bold appearance</b> .	I am <b>bold</b> .
i Tag	<code>&lt;i&gt; &lt;/i&gt;</code>	Presentational Element used to <b>italicize</b> text.	I am <i>italic</i> .
Strong Tag	<code>&lt;strong&gt; &lt;/strong&gt;</code>	Gives <b>bold appearance</b> and highlight content in <b>searching</b> .	I am <b>strong</b> .
em tag	<code>&lt;em&gt; &lt;/em&gt;</code>	<b>Italicize</b> text and highlight content in <b>searching</b> .	I am <i>em</i> .
small tag	I am <code>&lt;small&gt;small&lt;/small&gt;</code>	<b>small</b> print.	I am small.
u tag	<code>&lt;u&gt; &lt;/u&gt;</code>	<b>underline</b> text.	I am <u>underlined</u> .
s tag	<code>&lt;s&gt; &lt;/s&gt;</code>	Shows <b>struck</b> text.	I am <del>struck</del> text.
del tag	<code>&lt;del&gt; &lt;/del&gt;</code>	Shows <b>deleted</b> text.	Product Price is ₹100 95.
sup tag	<code>&lt;sup&gt; &lt;/sup&gt;</code>	Shows <b>superscript</b> text.	2000 = 2 * 10 <sup>3</sup> .
sub tag	<code>&lt;sub&gt; &lt;/sub&gt;</code>	Shows <b>subscript</b> text.	Chemical formula of water is H <sub>2</sub> O.
abbr tag	<code>&lt;abbr title="Prime Minister"&gt;PM&lt;/abbr&gt;</code>	Shows full version of <b>abbreviation</b> in title tag.	He is our PM.
kbd tag	<code>&lt;kbd&gt; &lt;/kbd&gt;</code>	Shows <b>keyboard command</b> .	To print this page, press Ctrl + p.
code tag	<code>&lt;code&gt; &lt;/code&gt;</code>	To show <b>computer code</b> .	Here is the equation var x = "string";.
q tag	<code>&lt;q&gt; &lt;/q&gt;</code>	To show <b>quotes</b> .	"This is a quote".
cite tag	I resides in <code>&lt;cite&gt;India &lt;/cite&gt;</code>	To show <b>cited title</b> of work.	I resides in India.

samp tag	<code>&lt;samp&gt; &lt;/samp&gt;</code>	To show <b>sample</b> .	This is a sample password.
ins tag	<code>&lt;ins&gt; &lt;/ins&gt;</code>	To indicate addition to document	<u>This is inserted text for above column</u>
var tag	<code>&lt;var&gt; &lt;/var&gt;</code>	to show <b>variables</b> in code .	$x + y = z$

### Div tag

Html **div tag** is used to create divisions. Div is **block level**, thus starts from new line and occupy full width of parent. Div can group all **block level elements**.

```
<div>
  // content inside div
</div>
```

### Span tag

Html **span tag** is used to create division inside line. Span is **inline level**, thus starts in the same line and occupy width of content. Span is used to group **inline level elements**.

```
<span>
  // content inside div
</span>
```

### HTML Lists

**HTML Lists** includes three type of **lists** to show single or multiple **list item**. These three lists are [Unordered List](#), [Ordered List](#) and [Description List](#), formerly known as [Definition List](#).

For example, name of fruits, name of cities, top engineering colleges in sequence etc. Lists are styled by bullets or numbers based on the **type of list**, i.e [Unordered List](#) or [Ordered List](#). We can also create **Nested Lists** in HTML.

#### HTML List Example

#### HTML List examples

##### Unordered List

- List Item
- List Item
- List Item
- List Item

```
<ul>
  <li>List Item</li>
  <li>List Item</li>
  <li>List Item</li>
  <li>List Item</li>
</ul>
```



## Ordered List

1. List Item
2. List Item
3. List Item
4. List Item

```
<ol>
  <li>List Item</li>
  <li>List Item</li>
  <li>List Item</li>
  <li>List Item</li>
</ol>
```

## Description List

Term

data

Term

data

```
<dl>
  <dt>Term</dt>
  <dd>data</dd>
  <dt>Term</dt>
  <dd>data</dd>
</dl>
```

## Type of Lists

There are **three types of list in HTML**, i.e, **unordered lists** , **ordered lists** & **description list**.

**Unordered lists** and **ordered lists** works the same way, except that the **Unordered** is used for **non-sequential lists** with list items usually preceded by bullets and the numbers is for **Ordered List**, which are normally represented by **incremental numbers**.

The **ul tag** is used to define **unordered list** and the **ol tag** is used to define **ordered list**.

Inside the list, the **li tag** is used to define each **list item**.

### Unordered List

**Unordered List** are list with bullets and without sequence. By default, **unordered list** are styled with *disk* (•). Till html4/xhtml, type attribute was used to change list style. But in **HTML5**, type attribute of **unordered list** is deprecated. But we can remove or change list style using **CSS list-style** property..

Use **Unordered List** where sequence is not required. For Example, list of fruits, vegetables etc.

```
<ul>
  <li>List 1</li>
  <li>List 2</li>
  <li>List 3</li>
  <li>List 4</li>
</ul>
```

### Unordered List with type none

```
<ul type="none">
  <li>List 1</li>
  <li>List 2</li>
  <li>List 3</li>
  <li>List 4</li>
</ul>
```

### Unordered List with type square

```
<ul type="square">
  <li>List 1</li>
  <li>List 2</li>
  <li>List 3</li>
  <li>List 4</li>
</ul>
```

### Unordered List with type circle

```
<ul type="circle">
  <li>List 1</li>
  <li>List 2</li>
  <li>List 3</li>
  <li>List 4</li>
</ul>
```

### Nested Unordered List

An **unordered list** can have another descendant unordered list, but only inside list item. A `<ul>` cannot have another `<ul>` element. Only `<li>` is allowed as child element of `ul` or `ol`.

```
<ul>
  <li>List 1</li>
  <li>List 2</li>
  <li>List 3
    <ul>
      <li>List 31</li>
      <li>List 32</li>
    </ul>
  </li>
  <li>List 4</li>
</ul>
```

### Ordered List

**Ordered list** are Sequential List. **ol** use numbers, alphabets and Roman characters as list style. **Ordered list** are countable. By default, ordered list are styled with numbers. We can change list style using *type attribute* or [List Style](#).

## Attributes of OL

Attribute	Type	Use
type	optional	to change type of list
start	optional	to start list from particular no
reversed	optional	to reversed numbering

### OL type number

Default type of ol is 1, which means number.

```
<ol>
  <li>List 1</li>
  <li>List 2</li>
  <li>List 3</li>
  <li>List 4</li>
</ol>
```

### OL type Roman

to start an **ordered list** with roman characters, use type I or i.

```
<ol type="I">
  <li>List 1</li>
  <li>List 2</li>
  <li>List 3</li>
  <li>List 4</li>
</ol>
```

### Ordered list with lowercase roman

```
<ol type="i">
  <li>List 5</li>
  <li>List 6</li>
  <li>List 7</li>
  <li>List 8</li>
</ol>
```

### OL type Alphabet

**Ordered List** can have alphabets, both uppercase and lowercase.

```
<ol type="A">
  <li>List 1</li>
  <li>List 2</li>
```

```
        <li>List 3</li>
        <li>List 4</li>
</ol>
```

Ordered list with lowercase

```
<ol type="a">
    <li>List 1</li>
    <li>List 2</li>
    <li>List 3</li>
    <li>List 4</li>
</ol>
```

Start Attribute

**start attribute** in ordered list can start list from n number, instead of 1, A, a, I or i. value of **start attribute** is always a number.

```
<ol type="1" start="10">
    <li>List 1</li>
    <li>List 2</li>
    <li>List 3</li>
    <li>List 4</li>
</ol>
```

Ordered List starting from X

```
<ol type="A" start="24">
    <li>List 5</li>
    <li>List 6</li>
    <li>List 7</li>
</ol>
```

Reversed Attribute

**Ordered List** can also have optional **reversed attribute**. **Reversed attribute** can reverse sequence or order of ordered list.

```
<ol start="10" reversed>
    <li>List 1</li>
    <li>List 2</li>
    <li>List 3</li>
    <li>List 4</li>
</ol>
```

Description List

```
<dl>
    <dt>WEB DESIGNING</dt>
```

```

        <dd>To Design front end of a website.</dd>
</dl>
<dl>
    <dt>DATABASE</dt>
    <dd>ORACLE, My SQL, and SQL Server</dd>
</dl>

```

## HTML Attributes

### What are Attributes

Attributes define additional characteristics or properties of the element such as width and height of an image. Attributes are always specified in the start tag (or opening tag) and usually consist of name/value pairs like `name="value"`. Attribute values should always be enclosed in quotation marks.

```

```

```


<a href="https://www.google.com/" title="Search Engine">Google</a>
<abbr title="Hyper Text Markup Language">HTML</abbr>
<input type="text" value="John Doe">

```

There are several attributes in HTML5 that do not consist of name/value pairs but consist of just a name. Such attributes are called Boolean attributes. Examples of some commonly used Boolean attributes are checked, disabled, readonly, required, etc.

```

<input type="email" required id="firstName" class="email">
<input type="submit" value="Submit" disabled id="firstName">
<input type="checkbox" checked id="firstName">
<input type="text" value="Read only text" readonly>

```

### General Purpose Attributes

There are some attributes, such as `id`, `title`, `class`, `style`, etc. that you can use on the majority of HTML elements. The following section describes their usage.

#### The id Attribute

The `id` attribute is used to give a unique name or identifier to an element within a document. This makes it easier to select the element using CSS or JavaScript.

```

<input type="text" id="firstName">
<div id="container">Some content</div>
<p id="infoText">This is a paragraph.</p>

```

#### The class Attribute

Like `id` attribute, the `class` attribute is also used to identify elements. But unlike `id`, the `class` attribute does not have to be unique in the document. This means you can apply the same class to multiple elements in a document, as shown in the following example:

```
<input type="text" class="highlight">
<input type="password" class="pass">
<div class="box highlight">Some content</div>
<p class="highlight">This is a paragraph.</p>
```

### The title Attribute

The `title` attribute is used to provide advisory text about an element or its content. Try out the following example to understand how this actually works.

```
<abbr title="World Wide Web">WWW</abbr>
<a href="images/kites.jpg" title="Click to view a larger image">
  
</a>
```

### The style Attribute

The `style` attribute allows you to specify CSS styling rules such as color, font, border, etc. directly within the element. Let's check out an example to see how it works:

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

<div style="border: 1px solid red;">Some content</div>
```

### Hyperlink

**HTML Hyperlinks** are built using `<a>` tag. Formally known as anchor tag, **hyperlink** defines a **link** `<a href="">` in html document. A **hyperlink** connects a webpage with other webpages or external pages. Hyperlink can be used as **Internal Link**, **External Link**, **Email Link** or **telephone link**.

In HTML5, an **hyperlink** can contain both **block level** and **inline level** elements. That means, we can write **text**, image or a `div` inside **hyperlink**.

#### Anchor Tag

**Anchor Tag** was the previous name of **hyperlinks** till html4/xhtml. But HTML5 renamed anchor tag to **hyperlink**.

```
<a>Anchor Tag</a>
```

#### Hyperlink

**Hyperlink** means an **a tag** with **href attribute**. **Hyperlink** is used to link webpages.

```
<a href="">Hyperlink Tag</a>
```

## Creating Links in HTML

A link or hyperlink is a connection from one web resource to another. Links allow users to move seamlessly from one page to another, on any server anywhere in the world.

A link has two ends, called anchors. The link starts at the source anchor and points to the destination anchor, which may be any web resource, for example, an image, an audio or video clip, a PDF file, an HTML document or an element within the document itself, and so on.

By default, links will appear as follows in most of the browsers:

- An [unvisited link](#) is underlined and blue.
- A [visited link](#) is underlined and purple.
- An [active link](#) is underlined and red.

## HTML Link Syntax

Links are specified in HTML using the `<a>` tag.

A link or hyperlink could be a word, group of words, or image.

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

Anything between the opening `<a>` tag and the closing `</a>` tag becomes the part of the link that the user sees and clicks in a browser. Here are some examples of the links:

```
<a href="https://www.google.com/">Google Search</a>
<a href="https://www.tutorialrepublic.com/">Tutorial Republic</a>
<a href="images/kites.jpg">
    
</a>
```

The `href` attribute specifies the target of the link. Its value can be an absolute or relative URL.

An absolute URL is the URL that includes every part of the URL format, such as protocol, host name, and path of the document, e.g., `https://www.google.com/`, `https://www.example.com/form.php`, etc. While, relative URLs are page-relative paths, e.g., `contact.html`, `images/smiley.png`, and so on. A relative URL never includes the `http://` or `https://` prefix.

## Setting the Targets for Links

The `target` attribute tells the browser where to open the linked document. There are four defined targets, and each target name starts with an underscore(`_`) character:

- `_blank` — Opens the linked document in a new window or tab.
- `_parent` — Opens the linked document in the parent window.
- `_self` — Opens the linked document in the same window or tab as the source document. This is the default, hence it is not necessary to explicitly specify this value.
- `_top` — Opens the linked document in the full browser window.

Try out the following example to understand how the link's target basically works:

```
<a href="/about-us.php" target="_top">About Us</a>
<a href="https://www.google.com/" target="_blank">Google</a>
<a href="images/sky.jpg" target="_parent">
  
</a>
```

## HTML Styles

### Styling HTML Elements

HTML is quite limited when it comes to the presentation of a web page. It was originally designed as a simple way of presenting information. [CSS \(Cascading Style Sheets\)](#) was introduced in December 1996 by the [World Wide Web Consortium \(W3C\)](#) to provide a better way to style HTML elements.

With CSS, it becomes very easy to specify the things like, size and typeface for the fonts, colors for the text and backgrounds, alignment of the text and images, amount of space between the elements, border and outlines for the elements, and lots of other styling properties.

### Adding Styles to HTML Elements

Style information can either be attached as a separate document or embedded in the HTML document itself. These are the three methods of implementing styling information to an HTML document.

- **Inline styles** — Using the `style` attribute in the HTML start tag.
- **Embedded style** — Using the `<style>` element in the head section of the document.
- **External style sheet** — Using the `<link>` element, pointing to an external CSS files.

### Inline Styles

Inline styles are used to apply the unique style rules to an element, by putting the CSS rules directly into the start tag. It can be attached to an element using the `style` attribute.

The style attribute includes a series of CSS property and value pairs. Each property: value pair is separated by a semicolon (;), just as you would write into an embedded or external style sheet. But it needs to be all in one line i.e. no line break after the semicolon.



The following example demonstrates how to set the **color** and **font-size** of the text:

```
<h1 style="color:red; font-size:30px;">This is a heading</h1>
<p style="color:green; font-size:18px;">This is a paragraph.</p>
<div style="color:green; font-size:18px;">This is some text.</div>
```

## Embedded Style Sheets

Embedded or internal style sheets only affect the document they are embedded in.

Embedded style sheets are defined in the **<head>** section of an HTML document using the **<style>** tag. You can define any number of **<style>** elements inside the **<head>** section.

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

## HTML Images

HTML Images are defined within **<img>** tag. Image tag is an void tag. **src** and **alt** attributes are compulsory to add path of image and alternative text.

```





```

Each image must carry at least two attributes: the **src** attribute, and an **alt** attribute.

The **src** attribute tells the browser where to find the image. Its value is the URL of the image file.

Whereas, the **alt** attribute provides an alternative text for the image, if it is unavailable or cannot be displayed for some reason. Its value should be a meaningful substitute for the image.

## Setting the Width and Height of an Image

The **width** and **height** attributes are used to specify the width and height of an image.

The values of these attributes are interpreted in pixels by default.

```



```

```



```

## HTML Tables

### Creating Tables in HTML

HTML table allows you to arrange data into rows and columns. They are commonly used to display tabular data like product listings, customer's details, financial reports, and so on.

You can create a table using the `<table>` element. Inside the `<table>` element, you can use the `<tr>` elements to create rows, and to create columns inside a row you can use the `<td>` elements. You can also define a cell as a header for a group of table cells using the `<th>` element.

```
<table>
  <tr>
    <th>No.</th>
    <th>Name</th>
    <th>Age</th>
  </tr>
  <tr>
    <td>1</td>
    <td>Peter Parker</td>
    <td>16</td>
  </tr>
  <tr>
    <td>2</td>
    <td>Clark Kent</td>
    <td>34</td>
  </tr>
</table>
```

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

```
th, td {  
    padding: 10px;  
}
```

## Spanning Multiple Rows and Columns

Spanning allow you to extend table rows and columns across multiple other rows and columns.

Normally, a table cell cannot pass over into the space below or above another table cell. But, you can use the `rowspan` or `colspan` attributes to span multiple rows or columns in a table.

```
<table>  
  <tr>  
    <th>Name</th>  
    <th colspan="2">Phone</th>  
  </tr>  
  <tr>  
    <td>John Carter</td>  
    <td>5550192</td>  
    <td>5550152</td>  
  </tr>  
</table>
```

Similarly, you can use the `rowspan` attribute to create a cell that spans more than one row. Let's try out an example to understand how row spanning basically works:

```
<table>  
  <tr>  
    <th>Name:</th>  
    <td>John Carter</td>  
  </tr>  
  <tr>  
    <th rowspan="2">Phone:</th>  
    <td>55577854</td>  
  </tr>  
  <tr>  
    <td>55577855</td>  
  </tr>  
</table>
```

## Adding Captions to Tables

You can specify a caption (or title) for your tables using the `<caption>` element.

The `<caption>` element must be placed directly after the opening `<table>` tag. By default, caption appears at the top of the table, but you can change its position using the CSS `caption-side` property.

```
<table>
  <caption>Users Info</caption>
  <tr>
    <th>No.</th>
    <th>Name</th>
    <th>Age</th>
  </tr>
  <tr>
    <td>1</td>
    <td>Peter Parker</td>
    <td>16</td>
  </tr>
  <tr>
    <td>2</td>
    <td>Clark Kent</td>
    <td>34</td>
  </tr>
</table>
```

### Defining a Table Header, Body, and Footer

HTML provides a series of tags `<thead>`, `<tbody>`, and `<tfoot>` that helps you to create more structured table, by defining header, body and footer regions, respectively.

```
<table>
  <thead>
    <tr>
      <th>Items</th>
      <th>Expenditure</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td>Stationary</td>
      <td>2,000</td>
    </tr>
    <tr>
      <td>Furniture</td>
      <td>10,000</td>
    </tr>
  </tbody>
</table>
```

```

        </tr>
    </tbody>
    <tfoot>
        <tr>
            <th>Total</th>
            <td>12,000</td>
        </tr>
    </tfoot>
</table>

```

## HTML iFrame

### What is 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.

```
<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.

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

```
<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.

```
<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.

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

## What is HTML Form

HTML Forms are required to collect different kinds of user inputs, such as contact details like name, email address, phone numbers, or details like credit card information, etc.

Forms contain special elements called controls like inputbox, checkboxes, radio-buttons, submit buttons, etc. Users generally complete a form by modifying its controls e.g. entering text, selecting items, etc. and submitting this form to a web server for further processing.

```
<form>
  <label>Username: <input type="text"></label>
  <label>Password: <input type="password"></label>
  <input type="submit" value="Submit">
</form>
```

## Input Element

This is the most commonly used element within HTML forms.

It allows you to specify various types of user input fields, depending on the `type` attribute. An input element can be of type *text field*, *password field*, *checkbox*, *radio button*, *submit button*, *reset button*, *file select box*, as well as several [new input types](#) introduced in HTML5.

## Text Fields

Text fields are one line areas that allow the user to input text.

Single-line text input controls are created using an `<input>` element, whose `type` attribute has a value of `text`. Here's an example of a single-line text input used to take username:

```
<form>
  <label for="username">Username:</label>
  <input type="text" name="username" id="username">
</form>
```

Username:

## Password Field

Password fields are similar to text fields. The only difference is; characters in a password field are masked, i.e. they are shown as asterisks or dots. This is to prevent someone else from reading the password on the screen. This is also a single-line text input controls created using an `<input>` element whose `type` attribute has a value of `password`.

Password:

## Radio Buttons

Radio buttons are used to let the user select exactly one option from a pre-defined set of options. It is created using an `<input>` element whose `type` attribute has a value of `radio`.

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

☐ Male ☐ Female

## Checkboxes

Checkboxes allows the user to select one or more option from a pre-defined set of options. It is created using an `<input>` element whose `type` attribute has a value of `checkbox`.

```
<form>
  <input type="checkbox" name="sports" id="soccer">
  <label for="soccer">Soccer</label>
  <input type="checkbox" name="sports" id="cricket">
  <label for="cricket">Cricket</label>
  <input type="checkbox" name="sports" id="baseball">
  <label for="baseball">Baseball</label>
</form>
```

☐ Soccer ☐ Cricket ☐ Baseball

## File Select box

The file fields allow a user to browse for a local file and send it as an attachment with the form data. Web browsers such as Google Chrome and Firefox render a file select input field with a Browse button that enables the user to navigate the local hard drive and select a file.

```
<form>
  <label for="file-select">Upload:</label>
  <input type="file" name="upload" id="file-select">
```

</form>

Upload:  No file chosen

## Textarea

Textarea is a multiple-line text input control that allows a user to enter more than one line of text. Multi-line text input controls are created using an <textarea> element.

```
<form>
  <label for="address">Address:</label>
  <textarea rows="3" cols="30" name="address" id="address"></
textarea>
</form>
```

Address:

## Select Boxes

A select box is a dropdown list of options that allows user to select one or more option from a pull-down list of options. Select box is created using the <select> element and <option> element.

```
<form>
  <label for="city">City:</label>
  <select name="city" id="city">
    <option value="sydney">Sydney</option>
    <option value="melbourne">Melbourne</option>
    <option value="cromwell">Cromwell</option>
  </select>
</form>
```

City:

## Submit and Reset Buttons

A submit button is used to send the form data to a web server. When submit button is clicked the form data is sent to the file specified in the form's action attribute to process the submitted data.



```

<form action="action.php" method="post">
  <label for="first-name">First Name:</label>
  <input type="text" name="first-name" id="first-name">
  <input type="submit" value="Submit">
  <input type="reset" value="Reset">
</form>

```

First Name:

## Grouping Form Controls

You also group logically related controls and labels within a web form using the `<legend>` element. Grouping form controls into categories makes it easier for users to locate a control which makes the form more user-friendly.

```

<form>
  <fieldset>
    <legend>Contact Details</legend>
    <label>Email Address: <input type="email" name="email"></label>
    <label>Phone Number: <input type="text" name="phone"></label>
  </fieldset>
</form>

```

## Frequently Used Form Attributes

The following table lists the most frequently used form element's attributes:

Attribute	Description
name	Specifies the name of the form.
action	Specifies the URL of the program or script on the web server that will be used for processing the information submitted via form.
method	Specifies the HTTP method used for sending the data to the web server by the browser. The value can be either <code>get</code> (the default) and <code>post</code> .
target	Specifies where to display the response that is received after submitting the form. Possible values are <code>_blank</code> , <code>_self</code> , <code>_parent</code> and <code>_top</code> .
enctype	Specifies how the form data should be encoded when submitting the form to the server. Applicable only when the value of the <code>method</code> attribute is <code>post</code> .

## Understanding the HTML5 Doctype

A Document Type Declaration, or DOCTYPE for short, is an instruction to the web browser about the version of markup language in which a web page is written.

A DOCTYPE declaration appears at the top of a web page before all other elements. According to the HTML specification or standards, every HTML document requires a valid document type declaration to insure that your web pages are displayed the way they are intended to be displayed.

The doctype declaration is usually the very first thing defined in an HTML document (even before the opening `<html>` tag); however the doctype declaration itself is not an HTML tag.

The DOCTYPE for HTML5 is very short, concise, and case-insensitive.

```
<!DOCTYPE html>
```

With HTML5 this is no longer the case and the doctype declaration is only needed to enable the standard mode for documents written using the HTML syntax.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <title><!-- Insert your title here --></title>
</head>
<body>
  <!-- Insert your content here -->
</body>
</html>
```

**Note:** The doctype declaration refers to a Document Type Definition (DTD). It is an instruction to the web browser about what version of the markup language the page is written in. The World Wide Web Consortium (W3C) provides DTDs for all HTML versions.

## 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>`, `<div>`, `<header>`, `<footer>`, `<section>`, etc. and adding some [CSS styles](#) to them.

## HTML Table Based Layout

Table provides the simplest way for creating layouts in HTML. Generally, this involves the process of putting the contents such as text, images, and so on into rows and columns.

The following layout is created using an HTML table with 3 rows and 2 columns — the first and last row spans both columns using the table's `colspan` attribute:

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<title>HTML Table Layout</title>
</head>
<body style="margin:0px;">
  <table style="width:100%; border-collapse:collapse; font:14px
  Arial,sans-serif;">
    <tr>
      <td colspan="2" style="padding:10px 20px; background-
      color:#acb3b9;">
        <h1 style="font-size:24px;">Tutorial Republic</h1>
      </td>
    </tr>
    <tr style="height:170px;">
      <td style="width:20%; padding:20px; background-
      color:#d4d7dc; vertical-align: top;">
        <ul style="list-style:none; padding:0px; line-
        height:24px;">
          <li><a href="#" style="color:#333;">Home</a></
          li>
          <li><a href="#" style="color:#333;">About</
          a></li>
          <li><a href="#" style="color:#333;">Contact</
          a></li>
        </ul>
      </td>
      <td style="padding:20px; background-color:#f2f2f2;
      vertical-align:top;">
        <h2>Welcome to our site</h2>
        <p>Here you will learn how to create websites...</
        p>
      </td>
    </tr>
    <tr>
      <td colspan="2" style="padding:5px; background-
      color:#acb3b9; text-align:center;">
```

```

        <p>copyright &copy; tutorialrepublic.com</p>
    </td>
</tr>
</table>
</body>
</html>

```

Tutorial Republic	
<ul style="list-style-type: none"> <li>• <a href="#">Home</a></li> <li>• <a href="#">About</a></li> <li>• <a href="#">Contac</a></li> </ul>	<h2>Welcome to our site</h2> <p>Here you will learn how to create websites...</p>
copyright © tutorialrepublic.com	

## The HTML head Element

The `<head>` element primarily is the container for all the head elements, which provide extra information about the document (metadata), or reference to other resources that are required for the document to display or behave correctly in a web browser.

The head elements collectively describes the properties of the document such as title, provide meta information like character set, instruct the browser where to find the style sheets or scripts that allows you to extend the HTML document in a highly active and interactive ways.

The HTML elements that can be used inside the `<head>` element are: `<title>`, `<base>`, `<link>`, `<style>`, `<meta>`, `<script>` and the `<noscript>` element.

## The HTML title Element

The `<title>` element defines the title of the document.

The title element is required in all HTML/XHTML documents to produce a valid document. Only one title element is permitted in a document and it must be placed within the `<head>` element. The title element contains plain text and entities; it may not contain other markup tags.

The title of the document may be used for different purposes. For example:

- To display a title in the browser title bar and in the task bar.
- To provide a title for the page when it is added to favorites or bookmarked.

- To displays a title for the page in search-engine results.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>A simple HTML document</title>
</head>
<body>
  <p>Hello World!</p>
</body>
</html>
```

## The HTML base Element

The HTML <base> element is used to define a base URL for all relative links contained in the document, e.g. you can set the base URL once at the top of your page, and then all subsequent relative links will use that URL as a starting point.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>Defining a base URL</title>
  <base href="https://www.tutorialrepublic.com/">
</head>
<body>
  <p><a href="html-tutorial/html-head.php">HTML Head</a>.</p>
</body>
</html>
```

## The HTML link Element

The <link> element defines the relationship between the current document and an external documents or resources. A common use of link element is to link to external style sheets.

```
<head>
  <title>Linking Style Sheets</title>
  <link rel="stylesheet" href="style.css">
</head>
```

## The HTML style Element

The <style> element is used to define embedded style information for an HTML document. The style rules inside the <style> element specify how HTML elements render in a browser.

```
<head>
  <title>Embedding Style Sheets</title>
  <style>
```

```

    body { background-color: YellowGreen; }
    h1 { color: red; }
    p { color: green; }
</style>
</head>

```

## The 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.

```

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

```

## The HTML script Element

The <script> element is used to define client-side script, such as JavaScript in HTML documents.

```

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

```

## Defining Metadata

The <meta> tags are typically used to provide structured metadata such as a document's *keywords*, *description*, *author name*, *character encoding*, and other metadata. Any number of meta tags can be placed inside the [head section](#) of an HTML or XHTML document.

## Declaring Character Encoding in HTML

Meta tag typically used to declare character encoding inside HTML document.

```

<!DOCTYPE html>
<html lang="en">
<head>
  <title>Declaring Character Encoding</title>
  <meta charset="utf-8">
</head>
<body>
  <h1>Hello World!</h1>
</body>

```

```
</html>
```

## Defining the Author of a Document

You can also use the meta tag to clearly define who is the author or creator of the web page.

```
<head>
  <title>Defining Document's Author</title>
  <meta name="author" content="Alexander Howick">
</head>
```

## Keywords and Description for Search Engines

Some search engines use metadata especially keywords and descriptions to index web pages; however this may not necessarily be true. Keywords giving extra weight to a document's keywords and description provide a short synopsis of the page.

```
<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>
```

## Configuring the Viewport for Mobile Devices

You can use the viewport meta tag to display the web pages correctly on mobile devices.

Without a viewport meta tag, mobile browsers render the web pages at typical desktop screen widths, and then scale it down to fit the mobile screen. As a result, it requires pinch-and-zoom to view the web page properly in mobile devices, which is very inconvenient.

```
<head>
  <title>Configuring the Viewport</title>
  <meta name="viewport" content="width=device-width, initial-
scale=1">
</head>
```

The width=device-width key-value pair inside the content attribute sets the width of the viewport to same as the screen width of the device, whereas the initial-scale=1 sets the initial scale or zoom level to 100% when the page is first loaded by the browser.

## New Input Types in HTML5

HTML5 introduces several new `<input>` types like email, date, time, color, range, and so on. to improve the user experience and to make the forms more interactive.

### Input Type Color

The `color` input type allows the user to select a color from a color picker and returns the color value in hexadecimal format (`#rrggbb`). If you don't specify a value, the default is `#000000`, which is black.

```
<form>
  <label for="mycolor">Select Color:</label>
  <input type="color" value="#00ff00" id="mycolor">
</form>
```

### Input Type Date

The `date` input type allows the user to select a date from a drop-down calendar.

The date value includes the year, month, and day, but not the time.

```
<form>
  <label for="mydate">Select Date:</label>
  <input type="date" value="2019-04-15" id="mydate">
</form>
```

### Input Type Datetime-local

The `datetime-local` input type allows the user to select both local date and time, including the year, month, and day as well as the time in hours and minutes.

```
<form>
  <label for="mydatetime">Choose Date and Time:</label>
  <input type="datetime-local" id="mydatetime">
</form>
```

### Input Type Email

The `email` input type allows the user to enter e-mail address. It is very similar to a standard text input type, but if it is used in combination with the `required` attribute, the browser may look for the patterns to ensure a properly-formatted e-mail address should be entered.

```
<form>
  <label for="myemail">Enter Email Address:</label>
  <input type="email" id="myemail" required>
</form>
```

### Input Type Month

The `month` input type allows the user to select a month and year from a drop-down calendar.

The value is a string in the format "YYYY-MM", where YYYY is the four-digit year and MM is the month number.



```
<form>
  <label for="mymonth">Select Month:</label>
  <input type="month" id="mymonth">
</form>
```

## Input Type Number

The number input type can be used for entering a numerical value. You can also restrict the user to enter only acceptable values using the additional attributes min, max, and step.

```
<form>
  <label for="mynumber">Enter a Number:</label>
  <input type="number" min="1" max="10" step="0.5"
id="mynumber">
</form>
```

## Input Type Range

The range input type can be used for entering a numerical value within a specified range. It works very similar to number input, but it offers a simpler control for entering a number.

```
<form>
  <label for="mynumber">Select a Number:</label>
  <input type="range" min="1" max="10" step="0.5" id="mynumber">
</form>
```

## Input Type Search

The search input type can be used for creating search input fields.

A search field typically behaves like a regular text field, but in some browsers like Chrome and Safari as soon as you start typing in the search box a small cross appears on the right side of the field that lets you quickly clear the search field

```
<form>
  <label for="mysearch">Search Website:</label>
  <input type="search" id="mysearch">
</form>
```

## Input Type Tel

The tel input type can be used for entering a telephone number.

Browsers don't support tel input validation natively. However, you can use the placeholder attribute to help users in entering the correct format for a phone number, or specify a [regular expression](#) to validate the user input using the pattern attribute.

```
<form>
  <label for="myphone">Telephone Number:</label>
  <input type="tel" id="myphone" placeholder="xx-xxxx-xxxx"
required>
</form>
```

## Input Type Time

The time input type can be used for entering a time (hours and minutes).

Browser may use 12- or 24-hour format for inputting times, based on local system's time setting.

```
<form>
  <label for="mytime">Select Time:</label>
  <input type="time" id="mytime">
</form>
```

## Input Type URL

The url input type can be used for entering URL's or web addresses.

You can use the multiple attribute to enter more than one URL. Also, if required attribute is specified browser will automatically carry out validation to ensure that only text that matches the standard format for URLs is entered into the input box.

```
<form>
  <label for="myurl">Enter Website URL:</label>
  <input type="url" id="myurl" required>
</form>
```

## Input Type Week

The week input type allows the user to select a week and year from a drop-down calendar.

```
<form>
  <label for="myweek">Select Week:</label>
  <input type="week" id="myweek">
</form>
```

What are Semantic Elements? A semantic element clearly describes its meaning to both the browser and the developer. Examples of non-semantic elements: <div> and <span> - Tells nothing about its content. Examples of semantic elements: <form> , <table> , and <article> - Clearly defines its content.

## Semantic Elements in HTML5

Index	Semantic Tag	Description
1	<article>	Defines an article
2	<aside>	Defines content aside from the page content
3	<details>	Defines additional details that the user can view or hide
4	<figcaption>	Defines a caption for a <figure> element
5	<figure>	Specifies self-contained content, like illustrations, diagrams, photos, code listings, etc.
6	<footer>	Defines a footer for a document or section
7	<header>	Specifies a header for a document or section
8	<main>	Specifies the main content of a document
9	<mark>	Defines marked/highlighted text
10	<nav>	Defines navigation links
11	<section>	Defines a section in a document
12	<summary>	Defines a visible heading for a <details> element
13	<time>	Defines a date/time

## Examples

Below are some examples of semantic tags:

- The **<article>** element specifies the independent, self-contained content.

```
<!DOCTYPE html>
<html>
<head>
  <title>Home Page</title>
</head>
<body>
<article>
<h1>Imarticus</h1>
<h2>Leading provider of online learning platforms</h2>
```

```
<p>Imarticus is the leading provider of online learning platforms for today's developers.  
Text-based courses with interactive in-browser coding helps you learn only what you  
need.</p>  
</article>  
</body>  
</html>
```

The **<nav>** element defines a set of navigation links.

```
<!DOCTYPE html>  
<html>  
<head>  
  <title>Home Page</title>  
</head>  
<body>  
<nav>  
  <a href="file/js/">JavaScript</a> |  
  <a href="file/jquery/">jQuery</a>  
  <a href="file/html/">HTML</a> |  
  <a href="file/css/">CSS</a> |  
</nav>  
</body>  
</html>
```

The **<aside>** element represents sidebars or call-out boxes.

```
<!DOCTYPE html>  
<html>  
<head>  
  <title>Home Page</title>  
</head>  
<body>  
<aside>  
<h1>Imarticus</h1>  
<h2>Leading provider of online learning platforms</h2>  
<p>Imarticus is the leading provider of online learning platforms for today's developers.  
Text-based courses with interactive in-browser coding helps you learn only what you  
need.</p>  
</aside>  
</body>  
</html>
```

The **<section>** element represents a section of the document.

```
<!DOCTYPE html>  
<html>  
<head>  
  <title>Home Page</title>  
</head>  
<body>  
<section>  
<h1>Imarticus</h1>  
<h2>Leading provider of online learning platforms</h2>
```

```
<p>Imarticus is the leading provider of online learning platforms for today's developers.  
Text-based courses with interactive in-browser coding helps you learn only what you  
need.</p>  
</section>  
</body>  
</html>
```

The **<details>** tag specifies the tag for additional details.

```
<!DOCTYPE html>  
<html>  
<head>  
  <title>Home Page</title>  
</head>  
<body>  
<details>  
<summary>Imarticus</summary>  
<h2>Leading provider of online learning platforms</h2>  
<p>Imarticus is the leading provider of online learning platforms for today's developers.  
Text-based courses with interactive in-browser coding helps you learn only what you  
need.</p>  
</details>  
</body>  
</html>
```

The **<header>** tag represents a container for introductory content or a set of navigational links.

```
<!DOCTYPE html>  
<html>  
<head>  
  <title>Home Page</title>  
</head>  
<body>  
<header>  
  <h1>Imarticus</h1>  
<h2>Leading provider of online learning platforms</h2>  
</header>  
</body>  
</html>
```

The **<footer>** tag defines a footer for a document or section.

```
<!DOCTYPE html>  
<html>  
<head>  
  <title>Home Page</title>  
</head>  
<body>  
<footer>  
  <p>Leading provider of online learning platforms</p>  
  <p><a href="mailto:Example@gmail.com">Mail For Details</a></p>
```

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

- The **<main>** tag specifies the main page.
- Output**

```
<!DOCTYPE html>
<html>
<head>
  <title>Home Page</title>
</head>
<body>
<main>
  <h1>Imarticus</h1>
  <p>Leading provider of online learning platforms</p>

  <article>
    <h1>Imarticus</h1>
    <p>Leading provider of online learning platforms</p>
  </article>
</main>
</body>
</html>
```

## HTML Form Tag

**Form Tag** defines the **form** and within this tag, there is **action attribute** which tells the form where its contents will be sent when it is submitted.

An **HTML form** can have **input elements, checkbox, radio buttons, submit button** and more. A form can also contain **select dropdown, textarea, fieldset, legend, and label** elements.

```
<form action="" method="get" name="enquiry">
```

```
  /* Content */
```

```
</form>
```

### Form tag Attributes

HTML Form Attributes

Attribute	Values	Use
method	get or post	http get method submit form data but is visible in url. post includes data in body. more secure as data is not visible to user in url

action	path	the backend file collecting form data
name	any name	name of form control

## Input Element

The most important form element is the **input element**. The **input element** is used to get user information. An input element can be of type **text**, **password**, **checkbox**, **radio button**, **submit button** and more.

The default input type is text.

Attributes in Input element

Attribute Name	values	Use
type	text, password, file, radio, checkbox, button, submit, and reset	type defines <b>type of input</b> control.
size	default value is 20	change size of input control
tabindex	any numeric value	used to define a sequence followed by user when he navigate using Tab key
value	any possible value	set a default value of input control
maxlength	n digits	set maximum characters limit
disabled		disabled input control, or fieldset tag
checked		Check checkbox or radio button
multiple		Used in input type file for multiple files upload

## Input Type Text

**Input type text** `<input type="text">` is the common **input element** for name, surname, country, numbers and symbols in single line. Default input type is always text. But still it is recommended to define type attribute in input element.

Inputs are always used inside label.

```
<label>First Name: <input type="text"></label>
```

```
<label>Last Name: <input type="text"></label>
```

## Label

**Label tag** is used to write the content just before text field. To Specify particular label, place input inside label or the value of for attribute inside label should match the id of input control.

## Input in Label

### First Name:

```
<label>First Name: <input type="text"></label>
```

Input Outside Label

### Last Name:

```
<label for="lname">Last Name:</label>  
<input type="text" id="lname">
```

## Label Usage

Its always recommended to use inputs with labels for web accessibility. If label is not required, use aria-label attribute with placeholder.

```
<label for="name">Name: <input type="text"></label>
```

```
<input type="text" id="name" placeholder="name">
```

```
<span>Name: <input type="text"></span>
```

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

## value attribute

**value attribute** can also be used inside input or textarea. Usually we ask user to fill values, but if value is fixed, use value attribute.

**First Name:** First Name

**Last Name:** Last Name

```
<form>  
  <label for="fname">First Name:</label>  
  <input type="text" value="First Name" id="fname" >
```



```
<label for="lname">Last Name:</label>
<input type="text" value="Last Name" id="lname" >
</form>
Maxlength
```

**maxlength** attribute is used to restrict no of characters in a text field. **maxlength** value is a number. Maxlength attribute is useful for form validations.

Input with maxlength

**First Name:**

**Age:**

```
<form>
<label for="fname">First Name:</label>
<input type="text" id="fname" maxlength="10">
```

```
<label for="age">Age:</label>
<input type="text" id="age" maxlength="3">
</form>
```

**Size**

**Size attribute** is used to set the size of input or textarea. The default size is 20. To increase or decrease input size, change value of size.

size attribute example

**Age:**

```
<label>Age: <input type="text" size="3"> </label>
```

**Pincode:**

```
<label>Pincode: <input type="text" size="6" maxlength="6"></label>
```

**Input type Password**

The **input type password** is used to write passwords. The password value is written in **encrypt form**. i.e. a user cannot see, copy or cut password data from input type password.

Input type password example

**Password:**

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

## Input type File

**Input type file** let user to choose file from his system. This can be used to upload a picture, upload resume, upload a video or audio etc.

Default value of input type file is "No file chosen". Once the file is uploaded, the file name replace this text followed by extension.

### Resume:

```
<form>
  <label for="resume">Resume:</label>
  <input type="file" id="resume">
</form>
```

### Input type file with multiple

### Upload pictures :

```
<form>
  <label for="pics">Upload pictures : </label>
  <input type="file" id="pics" multiple>
</form>
```

### accept

**accept attribute** is used in input type file to validate type of file or extension. accept can choose one or multiple values.

accept attribute example

**Choose Resume:**      [PDF Online](#)

```
<label>Age: <input type="file" accept=".pdf"> </label>
```

accept attribute with multiple extensions

**Upload Picture:**      [JPG or PNG](#)

```
<label>Age: <input type="file" accept=".jpg, .png"> </label>
```

## Radio Buttons

**Radio Buttons** are used to choose a **single element** among a group.

To allow window to choose a single radio, use **name attribute** with **same value** on both radio inputs.

### Radio Button For Gender Selection

#### Male Female

```
<input type="radio" name="gender" id="male">
<label for="male">Male</label>
```

```
<input type="radio" name="gender" id="female">
<label for="female">Female</label>
```

## Radio Button For Multiple Questions

Question 1

**Answer 1 Answer 2 Answer 3 Answer 4**

Question 2

**Answer 1 Answer 2 Answer 3 Answer 4**

## Checkbox

**Checkbox** are used to select multiple selections unlike radio button. They can be checked and unchecked. We can use checkbox for hobbies, interests, terms & conditions, etc.

## Form Checkbox

**Bike**

**Car**

```
<label> <input type="checkbox"> :Bike</label>
<label> <input type="checkbox"> :Car</label>
```

Checkbox with disabled

Checkbox can also have **disabled** attribute. A **disabled** checkbox can't be checked, means checked will remain checked, and unchecked will remain unchecked. See example

## HTML Checkbox with disable

**: I Agree**

```
<input type="checkbox" disabled>>
```

## Checkbox with checked

Default checkbox state is unchecked. But we can change default state to **checked** by using **checked attribute** in input type checkbox. See example

## HTML Checkbox with checked

**: I Agree**

```
<label><input type="checkbox" checked>: I Agree</label>
```

## Select Dropdown

**select** or **select dropdown** is used to fetch single or multiple options in dropdown list. **Select** options are fixed, thus used can choose only given option or options. To select city, country, date, month, year etc, **Select Dropdown** is used.

Select dropdown example

**Current City:**

--Select City--  
New York  
Chicago  
Los Angeles  
Washington DC

```
<select>
  <option selected disabled>--Select City--</option>
  <option>New York</option>
  <option>Chicago</option>
  <option>Los Angeles</option>
  <option>Washington DC</option>
</select>
```

## Select with Optgroup

**Optgroup** element is used to group multiple options in select dropdown. The name of optgroup is set using *label attribute* in optgroup.

**Current City:**

--Select City--

New Delhi  
Kolkata  
Mumbai  
Chennai

Noida  
Gurgram  
Faridabad  
Gaziabad

```
<select>
  <option selected disabled>--Select City--</option>
  <optgroup label="Metros">
    <option>New Delhi</option>
    <option>Kolkata</option>
    <option>Mumbai</option>
    <option>Chennai</option>
```

```
</optgroup>
<optgroup label="Others">
  <option>Noida</option>
  <option>Gurgram</option>
  <option>Faridabad</option>
  <option>Ghaziabad</option>
</optgroup>
</select>
```

## Multiple Attribute in Select

**Multiple** attribute is used in select dropdown to select more than one options by using Ctrl or Cmd key.

**Choose Multiple Cities:** --Select City-- New Delhi Kolkata Mumbai Chennai Noida Gurgram Faridabad Gaziabad **Press Ctrl or Cmd for multi selections.**

```
<select multiple>
  <option selected disabled>--Select City--</option>
  <optgroup label="Metros">
    <option>New Delhi</option>
    <option>Kolkata</option>
    <option>Mumbai</option>
    <option>Chennai</option>
  </optgroup>
  <optgroup label="Others">
    <option>Noida</option>
    <option>Gurgram</option>
    <option>Faridabad</option>
    <option>Ghaziabad</option>
  </optgroup>
</select>
```

## Textarea

**Textarea** is used to write multiple line. Like post, query, address, comments, reviews etc. Textarea can have **row** and **col** attributes. Default rows are 2, and default columnns are 20.

### Textarea Example

```
<textarea></textarea>

<textarea rows="4"></textarea>

<textarea rows="4" cols="30"></textarea>
```

## Submit Button

**Submit Button** or **input type submit** is used to send information from user to web server. Submit button can be used only once in a form tag.

Submit Button example

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

## Reset Button

**Reset Button** or **input type reset** is used to reload form data, without refreshing webpage. Reset is also used once in a form tag.

Create reset Button

**Name:**

```
<input type="reset">
```

Button Tag

**Button Tag** or **Button** can create buttons, like input type button. Button Tag is pair element. We can use image, icon or child element inside button tag.

Type of buttons

- Button, `<button>Button</button>`.
- Reset button, `<button type="reset">Button</button>`.
- Submit button, `<button type="submit">Button</button>`.

Button Tag Example

Button 1 Button 2 Reset Submit

```
<button>Button</button>
```

## Fieldset

Form or form controls can also be placed inside **fieldset tag**. **Fieldset tag** is used to group form or multiple input controls. **Fieldset** group form controls in bordered area. We can also use **legend tag** inside fieldset.

The main functionality of **Fieldset** is to disable multiple form controls.

Fieldset with Legend

Enquiry Form

**Name:**

```
<fieldset>
  <legend>Enquiry Form</legend>
```

```

    <form onsubmit="return false">
      <label>Name:<input type="text"></label>
      <input type="submit">
      <input type="reset"> </form>
</fieldset>

```

### Fieldset with disabled

Fieldset also supports *disabled* attribute. By adding disabled attribute, all form controls including submit button are disabled. Thus user cannot fill and submit form data.

Enquiry Form

Name: **Name:**

```

<fieldset disabled>
  <legend>Enquiry Form</legend>
  <form onsubmit="return false">
    <label>Name:<input type="text"></label>
    <input type="submit">
    <input type="reset"> </form>
</fieldset>

```

### Contact Form Example

A complete **HTML Form** with all inputs, select dropdown, radio buttons, checkbox, textarea, submit and reset buttons.

```

<form>
  <table>
    <tr>
      <td>
        <label for="uname">Name</label>
      </td>
      <td>
        <input type="text" id="uname" name="username">
      </td>
    </tr>
    <tr>
      <td>
        <label for="uemail">Email</label>
      </td>
      <td>
        <input type="text" id="uemail" name="usermail">
        <button type="button">Check</button>
      </td>
    </tr>
    <tr>
      <td>
        <label for="age">Age</label>
      </td>
    </tr>
  </table>

```

```

        <td>
            <input type="text" name="userage" id="age" size="2" maxlength="2">
        </td>
    </tr>
    <tr>
        <td>
            <label>Country</label>
        </td>
        <td>
            <input type="text" value="India" name="country" disabled>
        </td>
    </tr>
    <tr>
        <td>
            <label for="pass">Password</label>
        </td>
        <td>
            <input type="password" id="pass">
        </td>
    </tr>
    <tr>
        <td>
            <label for="res">Resume</label>
        </td>
        <td>
            <input type="file" id="res">
        </td>
    </tr>
    <tr>
        <td>
            <label>Hobbies</label>
        </td>
        <td>
            <label>
                <input type="checkbox" checked> Cricket
            </label>
            <label>
                <input type="checkbox"> Football
            </label>
        </td>
    </tr>
    <tr>
        <td>
            <label>Gender</label>
        </td>
        <td>
            <label>
                <input type="radio" value="f" name="gender"> Female</label>
                <label>
                    <input value="m" type="radio" name="gender"> Male</label>
            </td>
    </tr>

```



```

<tr>
  <td>
    <label for="city">City</label>
  </td>
  <td>
    <select id="city" name="city">
      <option disabled selected>--Choose City--</option>
      <optgroup label="Metros">
        <option>New Delhi</option>
        <option>Mumbai</option>
        <option>Chennai</option>
        <option>Kolkata</option>
      </optgroup>
      <optgroup label="Others">
        <option>Noida</option>
        <option>Gurgaon</option>
        <option>Faridabad</option>
        <option>Ghaziabad</option>
      </optgroup>
    </select>
  </td>
</tr>
<tr>
  <td>
    <label>Address</label>
  </td>
  <td>
    <textarea rows="4" cols="40"></textarea>
  </td>
</tr>
<tr>
  <td></td>
  <td>
    <input type="submit" value="Submit">
    <input type="reset">
  </td>
</tr>
</table>
</form>

```

## Marquee Tag

**HTML Marquee Tag** is a **non standard** element used to **scroll text**, image or any content from left to right, right to left, bottom to top and top to bottom directions. We can also control **marquee speed**, **direction**, **scrolldelay** and **Stop or start marquee on mouseover** and **mouseout** using marquee attributes.

```
<marquee> This is a marquee </marquee>
```

Change Marquee Direction

**Direction attribute** can change **direction of marquee**. We can change **direction of marquee** from left to right, right to left, top to bottom or bottom to top. Default direction of marquee is left.

### How to Change Marquee direction?

Direction attribute in marquee can have four different values. Default value is left. Other values are right, top and bottom.

**<marquee direction="left">marquee with direction left </marquee>**

```
<marquee direction="right"> This marquee will scroll right </marquee>
```

```
<marquee direction="up" height="200"> This marquee will scroll upwards </marquee>
```

```
<marquee direction="down" height="200"> This marquee will scroll downwards </marquee>
```

```
<marquee behavior="scroll"> This is scroll behavior of a marquee </marquee>
```

```
<marquee behavior="alternate">This is alternate behavior of a marquee </marquee>
```

### Scrollamount, Change Marquee speed

```
<marquee scrollamount="10"> This is a marquee with scrollamount 10 </marquee>
```

```
<marquee scrollamount="25"> This is a marquee with scrollamount 25 </marquee>
```

```
<marquee scrollamount="50"> This is a marquee with scrollamount 50 </marquee>
```

```
<marquee scrollamount="100"> This is a marquee with scrollamount 100 </marquee>
```

### Scrolldelay, Delay Marquee

```
<marquee scrolldelay="10">This is a marquee with scrolldelay 10</marquee>
```

```
<marquee scrolldelay="25">This is a marquee with scrolldelay 25</marquee>
```

```
<marquee scrolldelay="50">This is a marquee with scrolldelay 50</marquee>
```

```
<marquee scrolldelay="100">This is a marquee with scrolldelay 100</marquee>
```

```
<marquee scrolldelay="200">This is a marquee with scrolldelay 200</marquee>
```

```

<marquee scrolldelay="300">This is a marquee with scrolldelay
300</marquee>
<marquee scrolldelay="500">This is a marquee with scrolldelay
500</marquee>
<marquee scrolldelay="1000">This is a marquee with scrolldelay
1000</marquee>

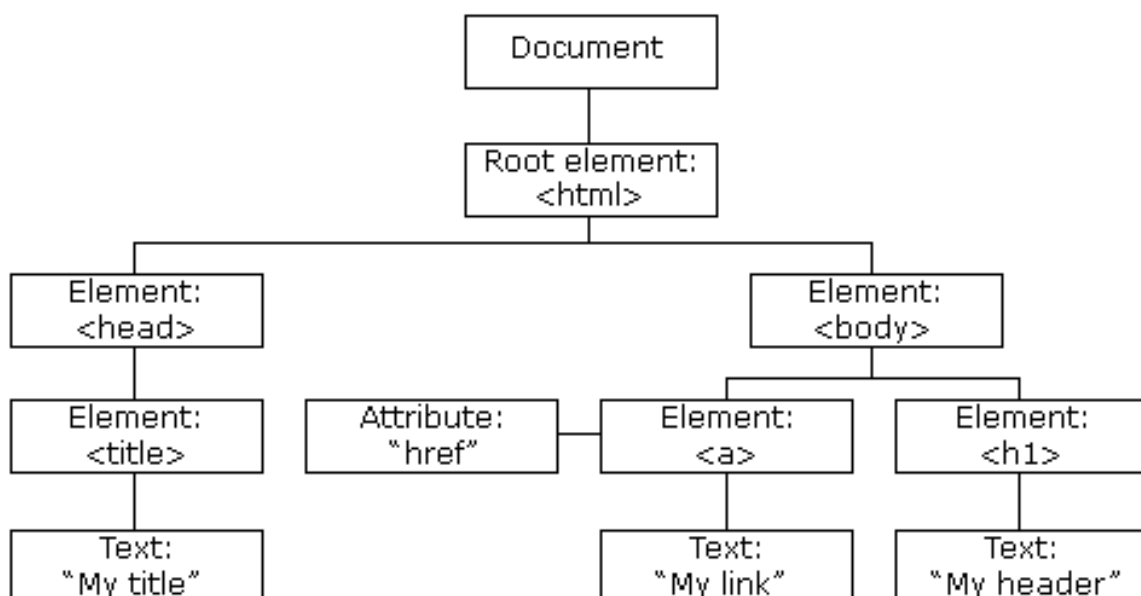
```

HTML4	XHTML	HTML5
<DIV></DIV>	<div></div>	Both allowed, but lowercase preferred
<img src="" alt="" >	<img src="" alt="" />	Both allowed, but <img src="" alt=""> is preferred
 	 	  is preferred
<img src="" alt="" width=200 >	<img src="" alt="" width="200" />	<img src="" alt="" width="200">
<input type="text" disabled >	<input type="text" disabled="disabled" />	<input type="text" disabled>

## HTML DOM Node Tree

The HTML DOM views a HTML document as a tree-structure. The tree structure is called a **node-tree**.

All nodes can be accessed through the tree. Their contents can be modified or deleted, and new elements can be created.



## Node Parents, Children, and Siblings

The nodes in the node tree have a hierarchical relationship to each other.

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

- In a node tree, the top node is called the root
- Every node, except the root, has exactly one parent node
- A node can have any number of children
- A leaf is a node with no children
- Siblings are nodes with the same parent

In the HTML above, every node except for the document node has a parent node:

- The `<html>` node has no parent node; the root node
- The parent node of the `<head>` and `<body>` nodes is the `<html>` node
- The parent node of the "Hello world!" text node is the `<p>` node

Most element nodes have child nodes:

- The `<html>` node has two child nodes; `<head>` and `<body>`
- The `<head>` node has one child node; the `<title>` node
- The `<title>` node also has one child node; the text node "DOM Tutorial"
- The `<h1>` and `<p>` nodes are siblings, and both child nodes of `<body>`

## HTML DOM Properties and Methods

### Programming Interface

The DOM models HTML as a set of node objects. The nodes can be accessed with JavaScript or other programming languages. In this tutorial we use JavaScript.

The programming interface to the DOM is defined by a set standard properties and methods.

**Properties** are often referred to as something that is (i.e. `nodename` is "p").

**Methods** are often referred to as something that is done (i.e. `delete` "p").

## HTML DOM Properties

These are some typical DOM properties:

- x.innerHTML - the inner text value of x (a HTML element)
- x.nodeName - the name of x
- x.nodeValue - the value of x
- x.parentNode - the parent node of x
- x.childNodes - the child nodes of x
- x.attributes - the attributes nodes of x

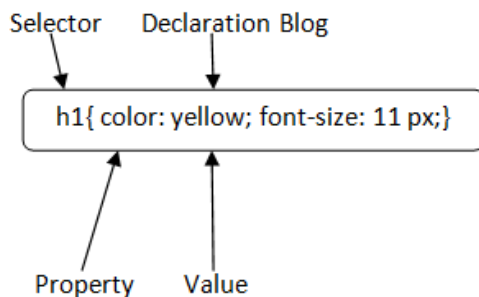
## CSS

The major points of CSS are given below:

- CSS stands for Cascading Style Sheet.
- CSS is used to design HTML tags.
- CSS is a widely used language on the web.
- HTML, CSS and JavaScript are used for web designing. It helps the web designers to apply style on HTML tags.

## CSS Syntax

A CSS rule set contains a selector and a declaration block.



### Without CSS

```
<!DOCTYPE html>
<html>
<head>
</head>
<body>
<h3> Here</h3>
<p>Hello, this is without CSS</p>
</body>
</html>
```

### Output

Here

Hello, this is without CSS

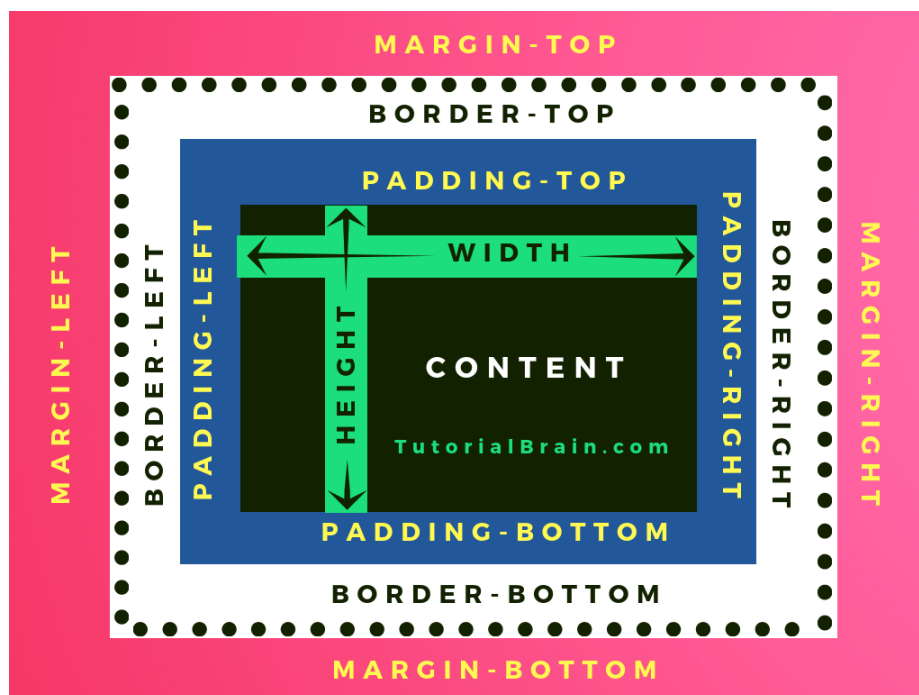
### With CSS

```
<!DOCTYPE html>
<html>
<head>
<style>
h3 {
color:blue;
background-color:yellow;
padding:3px;
}
p {
color:black;
}
</style>
</head>
<body>
<h3> Here</h3>
<p>This is with CSS</p>
</body>
</html>
```

### Output

[Here](#)

This is with CSS



**Selector:** Selector indicates the HTML element you want to style. It could be any tag like `<h1>`, `<title>` etc.

**Declaration Block:** The declaration block can contain one or more declarations separated by a semicolon. For the above example, there are two declarations:

1. color: yellow;
2. font-size: 11 px;

Each declaration contains a property name and value, separated by a colon.

**Property:** A Property is a type of attribute of HTML element. It could be color, border etc.

**Value:** Values are assigned to CSS properties. In the above example, value "yellow" is assigned to color property.

## CSS Selector

**CSS selectors** are used *to select the content you want to style*. Selectors are the part of CSS rule set. CSS selectors select HTML elements according to its id, class, type, attribute etc.

There are several different types of selectors in CSS.

1. CSS Element Selector
2. CSS Id Selector
3. CSS Class Selector
4. CSS Universal Selector
5. CSS Group Selector

### 1) CSS Element Selector

The element selector selects the HTML element by name.

```
<!DOCTYPE html>
<html>
<head>
<style>
p{
  text-align: center;
  color: blue;
}
</style>
</head>
<body>
<p>This style will be applied on every paragraph.</p>
<p id="para1">Me too!</p>
<p>And me!</p>
</body>
</html>
```

### 2) CSS Id Selector

The id selector selects the id attribute of an HTML element to select a specific element. An id is always unique within the page so it is chosen to select a single, unique element.

It is written with the hash character (#), followed by the id of the element

```
<!DOCTYPE html>
<html>
<head>
<style>
#para1 {
    text-align: center;
    color: blue;
}
</style>
</head>
<body>
<p id="para1">Hello Imarticus</p>
<p>This paragraph will not be affected.</p>
</body>
</html>
```

### 3) CSS Class Selector

The class selector selects HTML elements with a specific class attribute. It is used with a period character . (full stop symbol) followed by the class name.

```
<!DOCTYPE html>
<html>
<head>
<style>
.center {
    text-align: center;
    color: blue;
}
</style>
</head>
<body>
<h1 class="center">This heading is blue and center-aligned.</h1>
<p class="center">This paragraph is blue and center-aligned.</p>
</body>
</html>
```

### CSS Class Selector for specific element

If you want to specify that only one specific HTML element should be affected then you should use the element name with class selector.

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



```

<style>
p.center {
  text-align: center;
  color: blue;
}
</style>
</head>
<body>
<h1 class="center">This heading is not affected</h1>
<p class="center">This paragraph is blue and center-aligned.</p>
</body>
</html>

```

#### 4) CSS Universal Selector

```

<!DOCTYPE html>
<html>
<head>
<style>
* {
  color: green;
  font-size: 20px;
}
</style>
</head>
<body>
<h2>This is heading</h2>
<p>This style will be applied on every paragraph.</p>
<p id="para1">Me too!</p>
<p>And me!</p>
</body>
</html>

```

#### 5) CSS Group Selector

The grouping selector is used to select all the elements with the same style definitions.

Grouping selector is used to minimize the code. Commas are used to separate each selector in grouping.

```

h1 {
  text-align: center;
  color: blue;
}
h2 {
  text-align: center;
  color: blue;
}
p {
  text-align: center;
}

```

```
    color: blue;
}

<!DOCTYPE html>
<html>
<head>
<style>
h1, h2, p {
    text-align: center;
    color: blue;
}
</style>
</head>
<body>
<h1>Hello google.com</h1>
<h2>Hello google.com (In smaller font)</h2>
<p>This is a paragraph.</p>
</body>
</html>
```

## How to add CSS

CSS is added to HTML pages to format the document according to information in the style sheet. There are three ways to insert CSS in HTML documents.

1. Inline CSS
2. Internal CSS
3. External CSS

### 1) Inline CSS

Inline CSS is used to apply CSS on a single line or element.

For example:

```
<p style="color:blue">Hello CSS</p>
```

### 2) Internal CSS

Internal CSS is used to apply CSS on a single document or page. It can affect all the elements of the page. It is written inside the style tag within head section of html.

```
<style>
p{color:blue}
</style>
```

### 3) External CSS

External CSS is used to apply CSS on multiple pages or all pages. Here, we write all the CSS code in a css file. Its extension must be .css for example style.css.

```
p{color:blue}
```

```
<link rel="stylesheet" type="text/css" href="style.css">
```

### 1. What is the CSS syntax?

The Styling of the HTML elements can be defined using CSS syntax.

For example, if we want to give h1 in red color then the syntax for that is:

```
h1{  
color:red;}
```

So in this way, we can define CSS syntax.

### 2. What is CSS Selector Syntax?

Selecting HTML elements to apply style properties is defined as CSS selectors.

#### Syntax:

```
elementName {  
property1: value1;  
property2: value2;  
}
```

### 3. The basic syntax for CSS

There is no particular syntax for CSS. But, we can define a syntax for HTML elements in CSS.

For example,

#### Syntax for <p> tag:

```
p{  
color:green;  
font-size:18px;}
```

```
<!DOCTYPE html>  
<html>  
<head>  
<style>  
  .h4class {  
    background-color:#ffa500;  
  }  
  .square {  
    list-style-type: square;  
  }  
</style>  
</head>
```

```

<body>
  <h4 class="h4class">Internal CSS is included in the <mark>head</
mark> tag.
  All the CSS style is put inside <mark>style</mark></h4>
  <p>Example of hot skills:</p>
  <ul class="square">
    <li>Mainframe</li>
    <li>Java</li>
    <li>PHP</li>
  </ul>
</body>
</html>

```

## 1) CSS background-color

The background-color property is used to specify the background color of the element.

```

<!DOCTYPE html>
<html>
<head>
<style>
h2,p{
  background-color: #b0d4de;
}
</style>
</head>
<body>
<h2>My first CSS page.</h2>
<p>Hello Javatpoint. This is an example of CSS background-color.</p>
</body>
</html>

```

## 2) CSS background-image

The background-image property is used to set an image as a background of an element. By default the image covers the entire element. You can set the background image for a page like this.

```

<!DOCTYPE html>
<html>
<head>
<style>
body {
background-image: url("paper1.gif");
margin-left:100px;
}
</style>
</head>
<body>

```

```
<h1>Hello Imarticus</h1>
</body>
</html>
```

### 3) CSS background-repeat

By default, the background-image property repeats the background image horizontally and vertically. Some images are repeated only horizontally or vertically.

```
<!DOCTYPE html>
<html>
<head>
<style>
body {
    background-image: url("gradient_bg.png");
    background-repeat: repeat-x;
}
</style>
</head>
<body>
<h1>Hello Imarticus</h1>
</body>
</html>
```

background-repeat: repeat-y;

```
<!DOCTYPE html>
<html>
<head>
<style>
body {
    background-image: url("gradient_bg.png");
    background-repeat: repeat-y;
}
</style>
</head>
<body>
<h1>Hello Imarticus</h1>
</body>
</html>
```

### 4) CSS background-attachment

The background-attachment property is used to specify if the background image is fixed or scroll with the rest of the page in browser window. If you set fixed the background image then the image will not move during scrolling in the browser. Let's take an example with fixed background image.

```
<!DOCTYPE html>
```

```
<html>
```

&lt;head&gt;

<style>

```
body {
```

```
background: white url('bbb.gif');
```

```
background-repeat: no-repeat;
```

```
background-attachment: fixed;
```

```
margin-left:200px;
```

}

&lt;/style&gt;

&lt;/head&gt;

<body>

This is a fixed background-image. Scroll down the page.

This is a fixed background-image. Scroll down the page.

This is a fixed background-image. Scroll down the page.

This is a fixed background-image. Scroll down the page.

This is a fixed background-image. Scroll down the page.

This is a fixed background-image. Scroll down the page.

This is a fixed background-image. Scroll down the page.

This is a fixed background-image. Scroll down the page.

This is a fixed background-image. Scroll down the page.

This is a fixed background-image. Scroll down the page.

This is a fixed background-image. Scroll down the page.

This is a fixed background-image. Scroll down the page.

```
<p>This is a fixed background-image. Scroll down the page.</p>

<p>This is a fixed background-image. Scroll down the page.</p>

<p>This is a fixed background-image. Scroll down the page.</p>

<p>This is a fixed background-image. Scroll down the page.</p>

<p>This is a fixed background-image. Scroll down the page.</p>

<p>This is a fixed background-image. Scroll down the page.</p>

<p>If you do not see any scrollbars, Resize the browser window.</p>

</body>

</html>
```

## 5) CSS background-position

The background-position property is used to define the initial position of the background image. By default, the background image is placed on the top-left of the webpage.

You can set the following positions:

center  
top  
bottom  
left  
right

[illegible]

```
<p>This is a fixed background-image. Scroll down the page.</p>
<p>This is a fixed background-image. Scroll down the page.</p>
<p>This is a fixed background-image. Scroll down the page.</p>
<p>This is a fixed background-image. Scroll down the page.</p>
<p>This is a fixed background-image. Scroll down the page.</p>
<p>This is a fixed background-image. Scroll down the page.</p>
<p>This is a fixed background-image. Scroll down the page.</p>
<p>This is a fixed background-image. Scroll down the page.</p>
<p>This is a fixed background-image. Scroll down the page.</p>
<p>This is a fixed background-image. Scroll down the page.</p>
<p>This is a fixed background-image. Scroll down the page.</p>
<p>This is a fixed background-image. Scroll down the page.</p>
<p>If you do not see any scrollbars, Resize the browser window.</p>
</body>
</html>
```

## CSS Border

The CSS border is a shorthand property used to set the border on an element.

The **CSS** border properties are use to specify the style, color and size of the border of an element. The CSS border properties are given below

border-style  
border-color  
border-width  
border-radius

### 1) CSS border-style

The Border style property is used to specify the border type which you want to display on the web page.

There are some border style values which are used with border-style property to define a border.

| Value  | Description  |
|--------|--|
| none   | It doesn't define any border.                            |
| dotted | It is used to define a dotted border.                    |
| dashed | It is used to define a dashed border.                    |
| solid  | It is used to define a solid border.                     |
| double | It defines two borders with the same border-width value. |



|        |  |
|--------|--|
| groove | It defines a 3d grooved border. effect is generated according to border-color value. |
| ridge  | It defines a 3d ridged border. effect is generated according to border-color value.  |
| inset  | It defines a 3d inset border. effect is generated according to border-color value.   |
| outset | It defines a 3d outset border. effect is generated according to border-color value.  |

```

<!DOCTYPE html>
<html>
<head>
<style>
p.none {border-style: none;}
p.dotted {border-style: dotted;}
p.dashed {border-style: dashed;}
p.solid {border-style: solid;}
p.double {border-style: double;}
p.groove {border-style: groove;}
p.ridge {border-style: ridge;}
p.inset {border-style: inset;}
p.outset {border-style: outset;}
p.hidden {border-style: hidden;}
</style>
</head>
<body>
<p class="none">No border.</p>
<p class="dotted">A dotted border.</p>
<p class="dashed">A dashed border.</p>
<p class="solid">A solid border.</p>
<p class="double">A double border.</p>
<p class="groove">A groove border.</p>
<p class="ridge">A ridge border.</p>
<p class="inset">An inset border.</p>
<p class="outset">An outset border.</p>
<p class="hidden">A hidden border.</p>
</body>
</html>

```

## 2) CSS border-width

The border-width property is used to set the border's width. It is set in pixels. You can also use the one of the three pre-defined values, thin, medium or thick to set the width of the border.

```

<!DOCTYPE html>
<html>
<head>
<style>
p.one {
  border-style: solid;
  border-width: 5px;
}
p.two {
  border-style: solid;
  border-width: medium;
}
p.three {
  border-style: solid;
  border-width: 1px;
}
</style>
</head>
<body>
<p class="one">Write your text here.</p>
<p class="two">Write your text here.</p>
<p class="three">Write your text here.</p>
</body>
</html>

```

### 3) CSS border-color

There are three methods to set the color of the border.

Name: It specifies the color name. For example: "red".

RGB: It specifies the RGB value of the color. For example: "rgb(255,0,0)".

Hex: It specifies the hex value of the color. For example: "#ff0000".

There is also a border color named "transparent". If the border color is not set it is inherited from the color property of the element.

```

<!DOCTYPE html>
<html>
<head>
<style>
p.one {
  border-style: solid;
  border-color: red;
}
p.two {
  border-style: solid;
  border-color: #98bf21;
}
</style>
</head>

```

```

<body>
<p class="one">This is a solid red border</p>
<p class="two">This is a solid green border</p>
</body>
</html>

```

## CSS - Fonts

- The font-family property is used to change the face of a font.
- The font-style property is used to make a font italic or oblique.
- The font-variant property is used to create a small-caps effect.
- The font-weight property is used to increase or decrease how bold or light a font appears.
- The font-size property is used to increase or decrease the size of a font.
- The font property is used as shorthand to specify a number of other font properties.

```

<html>
  <head>
  </head>

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

```

### Set the Font Style

Following is the example, which demonstrates how to set the font style of an element. Possible values are normal, italic and oblique.

```

<html>
  <head>
  </head>

  <body>
    <p style = "font-variant:small-caps;">
      This text will be rendered as small caps
    </p>
  </body>
</html>

```

### Set the Font Weight

The following example demonstrates how to set the font weight of an element. The font-weight property provides the functionality to specify how bold a font is. Possible values could be normal, bold, bolder, lighter, 100, 200, 300, 400, 500, 600, 700, 800, 900.

```

<html>
  <head>
  </head>

```

```

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

```

### Set the Font Size

The following example demonstrates how to set the font size of an element. The font-size property is used to control the size of fonts. Possible values could be xx-small, x-small, small, medium, large, x-large, xx-large, smaller, larger, size in pixels or in %.

```

<html>
  <head>
  </head>

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

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

    <p style = "font-size:large;">
      This font size is large
    </p>
  </body>
</html>

```

### Set the Font Size Adjust

The following example demonstrates how to set the font size adjust of an element. This property enables you to adjust the x-height to make fonts more legible. Possible value could be any number.

```

<html>
  <head>
  </head>

  <body>
    <p style = "font-size-adjust:0.61;">
      This text is using a font-size-adjust value.
    </p>
  </body>
</html>

```

### Shorthand Property

You can use the font property to set all the font properties at once. For example –

```

<html>
  <head>
  </head>

  <body>
    <p style = "font:italic small-caps bold 15px georgia;">
      Applying all the properties on the text at once.
    </p>
  </body>
</html>

```

## CSS - Text

This chapter teaches you how to manipulate text using CSS properties. You can set following text properties of an element –

- The color property is used to set the color of a text.
- The direction property is used to set the text direction.
- The letter-spacing property is used to add or subtract space between the letters that make up a word.
- The word-spacing property is used to add or subtract space between the words of a sentence.
- The text-indent property is used to indent the text of a paragraph.
- The text-align property is used to align the text of a document.
- The text-decoration property is used to underline, overline, and strikethrough text.
- The text-transform property is used to capitalize text or convert text to uppercase or lowercase letters.
- The white-space property is used to control the flow and formatting of text.
- The text-shadow property is used to set the text shadow around a text.

### Set the Text Color

The following example demonstrates how to set the text color. Possible value could be any color name in any valid format.

```

<html>
  <head>
  </head>

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

```

### Set the Text Direction

The following example demonstrates how to set the direction of a text. Possible values are ltr or rtl.

```

<html>
  <head>
  </head>

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

```

### Set the Space between Characters

The following example demonstrates how to set the space between characters. Possible values are normal or a number specifying space..

```
<html>
  <head>
  </head>

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

### Set the Space between Words

The following example demonstrates how to set the space between words. Possible values are normal or a number specifying space.

```
<html>
  <head>
  </head>

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

### Set the Text Indent

The following example demonstrates how to indent the first line of a paragraph. Possible values are % or a number specifying indent space.

```
<html>
  <head>
  </head>

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

### Set the Text Alignment

The following example demonstrates how to align a text. Possible values are left, right, center, justify.

```
<html>
  <head>
  </head>

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

### Decorating the Text

The following example demonstrates how to decorate a text. Possible values are none, underline, overline, line-through, blink.

```
<html>
  <head>
  </head>

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

### Set the Text Cases

The following example demonstrates how to set the cases for a text. Possible values are none, capitalize, uppercase, lowercase.

```
<html>
  <head>
  </head>

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

```

### Set the White Space between Text

The following example demonstrates how white space inside an element is handled. Possible values are normal, pre, nowrap.

```

<html>
  <head>
  </head>

  <body>
    <p style = "white-space:pre;">
      This text has a line break and the white-space pre setting
      tells the browser to honor it just like the HTML pre tag.
    </p>
  </body>
</html>

```

### Set the Text Shadow

The following example demonstrates how to set the shadow around a text. This may not be supported by all the browsers.

```

<html>
  <head>
  </head>

  <body>
    <p style = "text-shadow:4px 4px 8px blue;">
      If your browser supports the CSS text-shadow property,
      this text will have a blue shadow.
    </p>
  </body>
</html>

```

## CSS - Using Images

Images play an important role in any webpage. Though it is not recommended to include a lot of images, but it is still important to use good images wherever required.

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.
- The height property is used to set the height of an image.
- The width property is used to set the width of an image.
- The -moz-opacity property is used to set the opacity of an image.

### The Image Border Property

The border property of an image is used to set the width of an image border. This property can have a value in length or in %.

A width of zero pixels means no border.



```

<html>
  <head>
  </head>

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

```

### The Image Height Property

The height property of an image is used to set the height of an image. This property can have a value in length or in %. While giving value in %, it applies it in respect of the box in which an image is available.

```

<html>
  <head>
  </head>

  <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>
</html>

```

### The Image Width Property

The width property of an image is used to set the width of an image. This property can have a value in length or in %. While giving value in %, it applies it in respect of the box in which an image is available.

```

<html>
  <head>
  </head>

  <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>
</html>

```

### CSS - Links

- The :link signifies unvisited hyperlinks.
- The :visited signifies visited hyperlinks.

- The `:hover` signifies an element that currently has the user's mouse pointer hovering over it.
- The `:active` signifies an element on which the user is currently clicking.

```
<style type = "text/css">
  a:link {color: #000000}
  a:visited {color: #006600}
  a:hover {color: #FFCC00}
  a:active {color: #FF00CC}
</style>
```

### Set the Color of Links

The following example demonstrates how to set the link color. Possible values could be any color name in any valid format.

```
<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

The following example demonstrates how to set the color of visited links. Possible values could be any color name in any valid format.

```
<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

The following example demonstrates how to change the color of links when we bring a mouse pointer over that link. Possible values could be any color name in any valid format.

```
<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

The following example demonstrates how to change the color of active links. Possible values could be any color name in any valid format.

```

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

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

```

### CSS - Tables

- The border-collapse specifies whether the browser should control the appearance of the adjacent borders that touch each other or whether each cell should maintain its style.
- The border-spacing specifies the width that should appear between table cells.
- The caption-side captions are presented in the <caption> element. By default, these are rendered above the table in the document. You use the caption-side property to control the placement of the table caption.
- The empty-cells specifies whether the border should be shown if a cell is empty.
- The table-layout allows browsers to speed up layout of a table by using the first width properties it comes across for the rest of a column rather than having to load the whole table before rendering it.

The border-collapse Property

This property can have two values collapse and separate. The following example uses both the values

```

<html>
  <head>
    <style type = "text/css">
      table.one {border-collapse:collapse;}
      table.two {border-collapse:separate;}

      td.a {
        border-style:dotted;
        border-width:3px;
        border-color:#000000;
        padding: 10px;
      }
      td.b {
        border-style:solid;

```

```

        border-width:3px;
        border-color:#333333;
        padding:10px;
    }
</style>
</head>

<body>
    <table class = "one">
        <caption>Collapse Border Example</caption>
        <tr><td class = "a"> Cell A Collapse Example</td></tr>
        <tr><td class = "b"> Cell B Collapse Example</td></tr>
    </table>
    <br />

    <table class = "two">
        <caption>Separate Border Example</caption>
        <tr><td class = "a"> Cell A Separate Example</td></tr>
        <tr><td class = "b"> Cell B Separate Example</td></tr>
    </table>
</body>
</html>

```

### The border-spacing Property

The border-spacing property specifies the distance that separates adjacent cells' borders. It can take either one or two values; these should be units of length.

```

<html>
<head>
    <style type = "text/css">
        table.one {
            border-collapse:separate;
            width:400px;
            border-spacing:10px;
        }
        table.two {
            border-collapse:separate;
            width:400px;
            border-spacing:10px 50px;
        }
    </style>
</head>

<body>

    <table class = "one" border = "1">
        <caption>Separate Border Example with border-spacing</
caption>
        <tr><td> Cell A Collapse Example</td></tr>
        <tr><td> Cell B Collapse Example</td></tr>
    </table>
    <br />

```

```

        <table class = "two" border = "1">
            <caption>Separate Border Example with border-spacing</
caption>
            <tr><td> Cell A Separate Example</td></tr>
            <tr><td> Cell B Separate Example</td></tr>
        </table>

    </body>
</html>

```

### The caption-side Property

The caption-side property allows you to specify where the content of a <caption> element should be placed in relationship to the table. The table that follows lists the possible values.

This property can have one of the four values top, bottom, left or right. The following example uses each value.

```

<html>
    <head>
        <style type = "text/css">
            caption.top {caption-side:top}
            caption.bottom {caption-side:bottom}
            caption.left {caption-side:left}
            caption.right {caption-side:right}
        </style>
    </head>

    <body>

        <table style = "width:400px; border:1px solid black;">
            <caption class = "top">
                This caption will appear at the top
            </caption>
            <tr><td > Cell A</td></tr>
            <tr><td > Cell B</td></tr>
        </table>
        <br />

        <table style = "width:400px; border:1px solid black;">
            <caption class = "bottom">
                This caption will appear at the bottom
            </caption>
            <tr><td > Cell A</td></tr>
            <tr><td > Cell B</td></tr>
        </table>
        <br />

        <table style = "width:400px; border:1px solid black;">
            <caption class = "left">
                This caption will appear at the left
            </caption>

```

```

        <tr><td > Cell A</td></tr>
        <tr><td > Cell B</td></tr>
    </table>
    <br />

    <table style = "width:400px; border:1px solid black;">
        <caption class = "right">
            This caption will appear at the right
        </caption>
        <tr><td > Cell A</td></tr>
        <tr><td > Cell B</td></tr>
    </table>

</body>
</html>

```

### The empty-cells Property

The empty-cells property indicates whether a cell without any content should have a border displayed.

This property can have one of the three values - show, hide or inherit.

```

<html>
  <head>
    <style type = "text/css">
      table.empty {
        width:350px;
        border-collapse:separate;
        empty-cells:hide;
      }
      td.empty {
        padding:5px;
        border-style:solid;
        border-width:1px;
        border-color:#999999;
      }
    </style>
  </head>

  <body>

    <table class = "empty">
      <tr>
        <th></th>
        <th>Title one</th>
        <th>Title two</th>
      </tr>

      <tr>
        <th>Row Title</th>
        <td class = "empty">value</td>
        <td class = "empty">value</td>

```

```

        </tr>

        <tr>
            <th>Row Title</th>
            <td class = "empty">value</td>
            <td class = "empty"></td>
        </tr>
    </table>

</body>
</html>

```

### The table-layout Property

The table-layout property is supposed to help you control how a browser should render or lay out a table.

This property can have one of the three values: fixed, auto or inherit.

```

<html>
  <head>
    <style type = "text/css">
      table.auto {
        table-layout: auto
      }
      table.fixed {
        table-layout: fixed
      }
    </style>
  </head>

  <body>

    <table class = "auto" border = "1" width = "100%">
      <tr>
        <td width = "20%">10000000000000000000000000000000</td>
        <td width = "40%">100000000</td>
        <td width = "40%">100</td>
      </tr>
    </table>
    <br />

    <table class = "fixed" border = "1" width = "100%">
      <tr>
        <td width = "20%">10000000000000000000000000000000</td>
        <td width = "40%">100000000</td>
        <td width = "40%">100</td>
      </tr>
    </table>

  </body>
</html>

```

## CSS - Borders

The border properties allow you to specify how the border of the box representing an element should look. There are three properties of a border you can change –

- The border-color specifies the color of a border.
- The border-style specifies whether a border should be solid, dashed line, double line, or one of the other possible values.
- The border-width specifies the width of a border.

### The border-color Property

The border-color property allows you to change the color of the border surrounding an element. You can individually change the color of the bottom, left, top and right sides of an element's border using the properties –

- border-bottom-color changes the color of bottom border.
- border-top-color changes the color of top border.
- border-left-color changes the color of left border.
- border-right-color changes the color of right border.

```
<html>
  <head>
    <style type = "text/css">
      p.example1 {
        border:1px solid;
        border-bottom-color:#009900; /* Green */
        border-top-color:#FF0000;    /* Red */
        border-left-color:#330000;   /* Black */
        border-right-color:#0000CC;  /* Blue */
      }
      p.example2 {
        border:1px solid;
        border-color:#009900;        /* Green */
      }
    </style>
  </head>

  <body>
    <p class = "example1">
      This example is showing all borders in different colors.
    </p>

    <p class = "example2">
      This example is showing all borders in green color only.
    </p>
  </body>
</html>
```

### The border-style Property

The border-style property allows you to select one of the following styles of border –

- none – No border. (Equivalent of border-width:0;)
- solid – Border is a single solid line.
- dotted – Border is a series of dots.
- dashed – Border is a series of short lines.
- double – Border is two solid lines.



- groove – Border looks as though it is carved into the page.
- ridge – Border looks the opposite of groove.
- inset – Border makes the box look like it is embedded in the page.
- outset – Border makes the box look like it is coming out of the canvas.
- hidden – Same as none, except in terms of border-conflict resolution for table elements.

You can individually change the style of the bottom, left, top, and right borders of an element using the following properties –

- border-bottom-style changes the style of bottom border.
- border-top-style changes the style of top border.
- border-left-style changes the style of left border.
- border-right-style changes the style of right border.

```
<html>
  <head>
  </head>

  <body>
    <p style = "border-width:4px; border-style:none;">
      This is a border with none width.
    </p>

    <p style = "border-width:4px; border-style:solid;">
      This is a solid border.
    </p>

    <p style = "border-width:4px; border-style:dashed;">
      This is a dashed border.
    </p>

    <p style = "border-width:4px; border-style:double;">
      This is a double border.
    </p>

    <p style = "border-width:4px; border-style:groove;">
      This is a groove border.
    </p>

    <p style = "border-width:4px; border-style:ridge">
      This is a ridge border.
    </p>

    <p style = "border-width:4px; border-style:inset;">
      This is a inset border.
    </p>

    <p style = "border-width:4px; border-style:outset;">
      This is a outset border.
    </p>

    <p style = "border-width:4px; border-style:hidden;">
      This is a hidden border.
```

```

</p>

<p style = "border-width:4px;
border-top-style:solid;
border-bottom-style:dashed;
border-left-style:groove;
border-right-style:double;">
    This is a a border with four different styles.
</p>
</body>
</html>

```

### The border-width Property

The border-width property allows you to set the width of an element borders. The value of this property could be either a length in px, pt or cm or it should be set to thin, medium or thick.

You can individually change the width of the bottom, top, left, and right borders of an element using the following properties –

- border-bottom-width changes the width of bottom border.
- border-top-width changes the width of top border.
- border-left-width changes the width of left border.
- border-right-width changes the width of right border.

```

<html>
<head>
</head>

<body>
    <p style = "border-width:4px; border-style:solid;">
        This is a solid border whose width is 4px.
    </p>

    <p style = "border-width:4pt; border-style:solid;">
        This is a solid border whose width is 4pt.
    </p>

    <p style = "border-width:thin; border-style:solid;">
        This is a solid border whose width is thin.
    </p>

    <p style = "border-width:medium; border-style:solid;">
        This is a solid border whose width is medium;
    </p>

    <p style = "border-width:thick; border-style:solid;">
        This is a solid border whose width is thick.
    </p>

    <p style = "border-bottom-width:4px;border-top-width:10px;
border-left-width: 2px;border-right-width:15px;border-
style:solid;">

```

```

        This is a a border with four different width.
    </p>
</body>
</html>

```

### Border Properties Using Shorthand

The border property allows you to specify color, style, and width of lines in one property –

```

<html>
  <head>
  </head>

  <body>
    <p style = "border:4px solid red;">
      This example is showing shorthand property for border.
    </p>
  </body>
</html>

```

### CSS - Margins

The margin property defines the space around an HTML element. It is possible to use negative values to overlap content.

The values of the margin property are not inherited by the child elements. Remember that the adjacent vertical margins (top and bottom margins) will collapse into each other so that the distance between the blocks is not the sum of the margins, but only the greater of the two margins or the same size as one margin if both are equal.

We have the following properties to set an element margin.

- The margin specifies a shorthand property for setting the margin properties in one declaration.
- The margin-bottom specifies the bottom margin of an element.
- The margin-top specifies the top margin of an element.
- The margin-left specifies the left margin of an element.
- The margin-right specifies the right margin of an element.

#### The Margin Property

The margin property allows you set all of the properties for the four margins in one declaration. Here is the syntax to set margin around a paragraph –

```

<html>
  <head>
  </head>

  <body>
    <p style = "margin: 15px; border:1px solid black;">
      all four margins will be 15px
    </p>

    <p style = "margin:10px 2%; border:1px solid black;">
      top and bottom margin will be 10px, left and right margin
will be 2%
      of the total width of the document.
    </p>
  </body>
</html>

```

</p>

<p style = "margin: 10px 2% -10px; border:1px solid black;">  
top margin will be 10px, left and right margin will be 2%

of the

total width of the document, bottom margin will be -10px

</p>

<p style = "margin: 10px 2% -10px auto; border:1px solid  
black;">

top margin will be 10px, right margin will be 2% of the  
total

width of the document, bottom margin will be -10px, left  
margin

will be set by the browser

</p>

</body>

</html>

### The margin-bottom Property

The margin-bottom property allows you set bottom margin of an element. It can have a value in length, % or auto.

<html>

<head>

</head>

<body>

<p style = "margin-bottom: 15px; border:1px solid black;">

This is a paragraph with a specified bottom margin

</p>

<p style = "margin-bottom: 5%; border:1px solid black;">

This is another paragraph with a specified bottom margin  
in percent

</p>

</body>

</html>

### The margin-top Property

The margin-top property allows you set top margin of an element. It can have a value in length, % or auto.

<html>

<head>

</head>

<body>

<p style = "margin-top: 15px; border:1px solid black;">

This is a paragraph with a specified top margin

</p>

```

    <p style = "margin-top: 5%; border:1px solid black;">
      This is another paragraph with a specified top margin in
percent
    </p>
  </body>
</html>

```

### The margin-left Property

The margin-left property allows you set left margin of an element. It can have a value in length, % or auto.

```

<html>
  <head>
  </head>

  <body>
    <p style = "margin-left: 15px; border:1px solid black;">
      This is a paragraph with a specified left margin
    </p>

    <p style = "margin-left: 5%; border:1px solid black;">
      This is another paragraph with a specified top margin in
percent
    </p>
  </body>
</html>

```

### The margin-right Property

The margin-right property allows you set right margin of an element. It can have a value in length, % or auto.

```

<html>
  <head>
  </head>

  <body>
    <p style = "margin-right: 15px; border:1px solid black;">
      This is a paragraph with a specified right margin
    </p>
    <p style = "margin-right: 5%; border:1px solid black;">
      This is another paragraph with a specified right margin
in percent
    </p>
  </body>
</html>

```

## CSS - Lists

- The list-style-type allows you to control the shape or appearance of the marker.
- The list-style-position specifies whether a long point that wraps to a second line should align with the first line or start underneath the start of the marker.
- The list-style-image specifies an image for the marker rather than a bullet point or number.

- The list-style serves as shorthand for the preceding properties.
- The marker-offset specifies the distance between a marker and the text in the list.

### The list-style-type Property

The list-style-type property allows you to control the shape or style of bullet point (also known as a marker) in the case of unordered lists and the style of numbering characters in ordered lists.

```
<html>
  <head>
  </head>

  <body>
    <ul style = "list-style-type:circle;">
      <li>Maths</li>
      <li>Social Science</li>
      <li>Physics</li>
    </ul>

    <ul style = "list-style-type:square;">
      <li>Maths</li>
      <li>Social Science</li>
      <li>Physics</li>
    </ul>

    <ol style = "list-style-type:decimal;">
      <li>Maths</li>
      <li>Social Science</li>
      <li>Physics</li>
    </ol>

    <ol style = "list-style-type:lower-alpha;">
      <li>Maths</li>
      <li>Social Science</li>
      <li>Physics</li>
    </ol>

    <ol style = "list-style-type:lower-roman;">
      <li>Maths</li>
      <li>Social Science</li>
      <li>Physics</li>
    </ol>
  </body>
</html>
```

### The list-style-position Property

The list-style-position property indicates whether the marker should appear inside or outside of the box containing the bullet points. It can have one the two values –

```
<html>
  <head>
  </head>
```

```

<body>
  <ul style = "list-style-type:circle; list-style-
position:outside;">
    <li>Maths</li>
    <li>Social Science</li>
    <li>Physics</li>
  </ul>

  <ul style = "list-style-type:square;list-style-
position:inside;">
    <li>Maths</li>
    <li>Social Science</li>
    <li>Physics</li>
  </ul>

  <ol style = "list-style-type:decimal;list-style-
position:outside;">
    <li>Maths</li>
    <li>Social Science</li>
    <li>Physics</li>
  </ol>

  <ol style = "list-style-type:lower-alpha;list-style-
position:inside;">
    <li>Maths</li>
    <li>Social Science</li>
    <li>Physics</li>
  </ol>
</body>
</html>

```

### The list-style-image Property

The list-style-image allows you to specify an image so that you can use your own bullet style. The syntax is similar to the background-image property with the letters url starting the value of the property followed by the URL in brackets.

```

<html>
  <head>
  </head>

  <body>
    <ul>
      <li style = "list-style-image: url(/images/
bullet.gif);">Maths</li>
      <li>Social Science</li>
      <li>Physics</li>
    </ul>

    <ol>
      <li style = "list-style-image: url(/images/
bullet.gif);">Maths</li>
      <li>Social Science</li>
    </ol>
  </body>
</html>

```

```

        <li>Physics</li>
    </ol>
</body>
</html>

```

### The list-style Property

The list-style allows you to specify all the list properties into a single expression. These properties can appear in any order.

```

<html>
  <head>
  </head>

  <body>
    <ul style = "list-style: inside square;">
      <li>Maths</li>
      <li>Social Science</li>
      <li>Physics</li>
    </ul>

    <ol style = "list-style: outside upper-alpha;">
      <li>Maths</li>
      <li>Social Science</li>
      <li>Physics</li>
    </ol>
  </body>
</html>

```

### The marker-offset Property

The marker-offset property allows you to specify the distance between the marker and the text relating to that marker. Its value should be a length as shown in the following example

–

```

<html>
  <head>
  </head>

  <body>
    <ul style = "list-style: inside square; marker-offset:2em;">
      <li>Maths</li>
      <li>Social Science</li>
      <li>Physics</li>
    </ul>

    <ol style = "list-style: outside upper-alpha; marker-
offset:2cm;">
      <li>Maths</li>
      <li>Social Science</li>
      <li>Physics</li>
    </ol>
  </body>
</html>

```