

**HTML**

奇  
妙  
世  
界





## Basic HTML Document

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

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

## The <!DOCTYPE> Declaration

The <!DOCTYPE> declaration tag is used by the web browser to understand the version of the HTML used in the document. Current version of HTML is 5 and it makes use of the following declaration:

```
<!DOCTYPE html>
```

# HTML – OVERVIEW

- As told earlier, HTML is a markup language and makes use of various tags to format the content. These tags are enclosed within angle braces **<Tag Name>**. Except few tags, most of the tags have their corresponding closing tags.

Above example of HTML document uses the following tags:

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



## HTML Document Structure

A typical HTML document will have the following structure:

Document declaration tag

```
<html>
```

```
  <head>
```

Document header related tags

```
  </head>
```

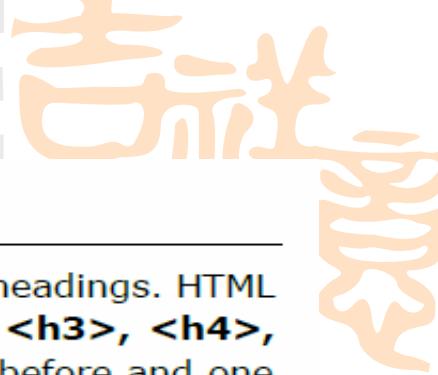
```
<body>
```

Document body related tags

```
  </body>
```

```
</html>
```





## Heading Tags

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

### Example

```
<!DOCTYPE html>
<html>
<head>
<title>Heading Example</title>
</head>
<body>
<h1>This is heading 1</h1>
<h2>This is heading 2</h2>
<h3>This is heading 3</h3>
<h4>This is heading 4</h4>
<h5>This is heading 5</h5>
<h6>This is heading 6</h6>
</body>
</html>
```

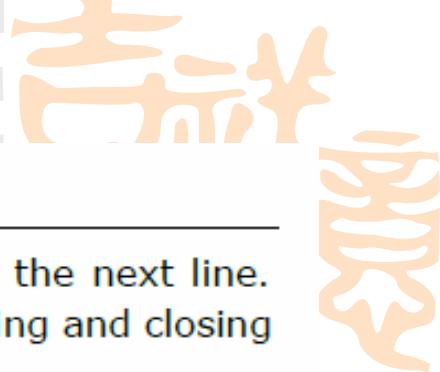


## Paragraph Tag

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

### Example

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



## Line Break Tag

Whenever you use the `<br />` element, anything following it starts from the next line. This tag is an example of an **empty** element, where you do not need opening and closing tags, as there is nothing to go in between them.

### Example

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



## Horizontal Lines

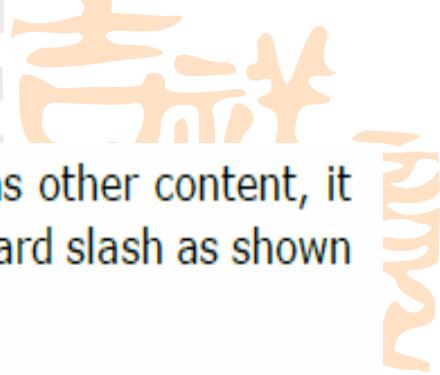
Horizontal lines are used to visually break-up sections of a document. The **<hr>** tag creates a line from the current position in the document to the right margin and breaks the line accordingly.

### Example

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



# HTML – ELEMENTS



An **HTML element** is defined by a starting tag. If the element contains other content, it ends with a closing tag, where the element name is preceded by a forward slash as shown below with few tags:

Start Tag	Content	End Tag
<p>	This is paragraph content.	</p>
<h1>	This is heading content.	</h1>
<div>	This is division content.	</div>





## Bold Text

Anything that appears within **<b>...</b>** element, is displayed in bold as shown below:

## Example

```
<!DOCTYPE html>

<html>
  <head>
    <title>Bold Text Example</title>
  </head>
  <body>
    <p>The following word uses a <b>bold</b> typeface.</p>
  </body>
</html>
```





## Italic Text

Anything that appears within `<i>...</i>` element is displayed in italicized as shown below:

### Example

```
<!DOCTYPE html>

<html>
  <head>
    <title>Italic Text Example</title>
  </head>
  <body>
    <p>The following word uses a <i>italicized</i> typeface.</p>
  </body>
</html>
```





## Underlined Text

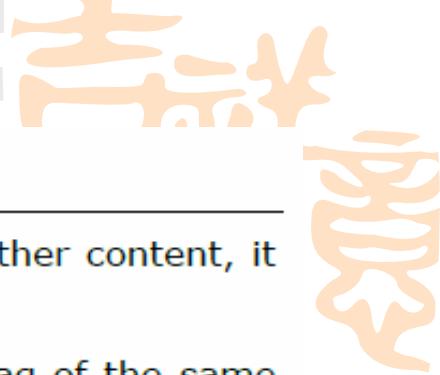
Anything that appears within `<u>...</u>` element, is displayed with underline as shown below:

### Example

```
<!DOCTYPE html>

<html>
<head>
<title>Underlined Text Example</title>
</head>
<body>
<p>The following word uses a <u>underlined</u> typeface.</p>
</body>
</html>
```





## HTML Tag vs. Element

---

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

For example, **<p>** is starting tag of a paragraph and **</p>** is closing tag of the same paragraph but **<p>This is paragraph</p>** is a paragraph element.

## Nested HTML Elements

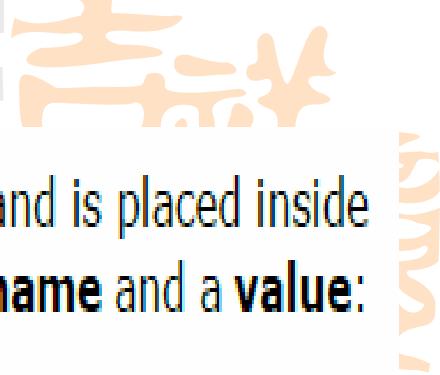
---

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

### Example

```
<!DOCTYPE html>  
  
<html>  
  <head>  
    <title>Nested Elements Example</title>  
  </head>  
  <body>  
    <h1>This is <i>italic</i> heading</h1>  
    <p>This is <u>underlined</u> paragraph</p>  
  </body>  
</html>
```

# HTML – ATTRIBUTES



An attribute is used to define the characteristics of an HTML element and is placed inside the element's opening tag. All attributes are made up of two parts: a **name** and a **value**:

- The **name** is the property you want to set. For example, the paragraph `<p>` element in the example carries an attribute whose name is **align**, which you can use to indicate the alignment of paragraph on the page.
- The **value** is what you want the value of the property to be set and always put within quotations. The below example shows three possible values of align attribute: **left**, **center** and **right**.

Attribute names and attribute values are case-insensitive. However, the World Wide Web Consortium (W3C) recommends lowercase attributes/attribute values in their HTML 4 recommendation.



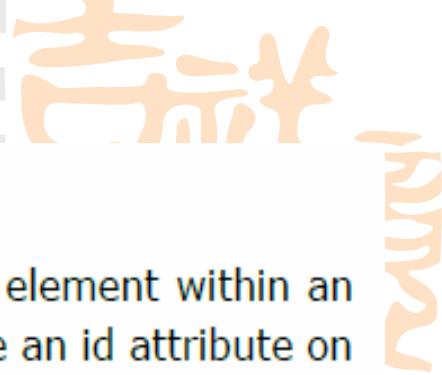


## Example

```
<!DOCTYPE html>

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





## The Id Attribute

The **id** attribute of an HTML tag can be used to uniquely identify any element within an HTML page. There are two primary reasons that you might want to use an id attribute on an element:

- If an element carries an id attribute as a unique identifier, it is possible to identify just that element and its content.
- If you have two elements of the same name within a Web page (or style sheet), you can use the id attribute to distinguish between elements that have the same name.

## Example

```
<p id="html">This para explains what is HTML</p>  
<p id="css">This para explains what is Cascading Style Sheet</p>
```





## The style Attribute

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

```
<!DOCTYPE html>

<html>
<head>
<title>The style Attribute</title>
</head>
<body>
<p style="font-family:arial; color:#FF0000;">Some text...</p>
</body>
</html>
```



# HTML – COLORS



Colors are very important to give a good look and feel to your website. You can specify colors on page level using `<body>` tag or you can set colors for individual tags using `bgcolor` attribute.

The `<body>` tag has following attributes which can be used to set different colors:

- `bgcolor` - sets a color for the background of the page.
- `text` - sets a color for the body text.
- `alink` - sets a color for active links or selected links.
- `link` - sets a color for linked text.
- `vlink` - sets a color for *visited links* - that is, for linked text that you have already clicked on.





## HTML Color Coding Methods

---

There are following three different methods to set colors in your web page:

- **Color names** - You can specify color names directly like green, blue or red.
- **Hex codes** - A six-digit code representing the amount of red, green, and blue that makes up the color.
- **Color decimal or percentage values** - This value is specified using the `rgb()` property.





## HTML Colors - Color Names

You can specify direct a color name to set text or background color. W3C has listed 16 basic color names that will validate with an HTML validator but there are over 200 different color names supported by major browsers.

### W3C Standard 16 Colors

Here is the list of W3C Standard 16 Colors names and it is recommended to use them.

	Black		Gray		Silver		White
	Yellow		Lime		Aqua		Fuchsia
	Red		Green		Blue		Purple
	Maroon		Olive		Navy		Teal





## Example

Here are the examples to set background of an HTML tag by color name:

```
<!DOCTYPE html>

<html>
<head>
<title>HTML Colors by Name</title>
</head>
<body text="blue" bgcolor="green">
<p>Use different color names for body and table and see the result.</p>
<table bgcolor="black">
<tr>
<td>
<font color="white">This text will appear white on black background.</font>
</td>
</tr>
</table>
</body>
</html>
```



## HTML Colors - Hex Codes

---

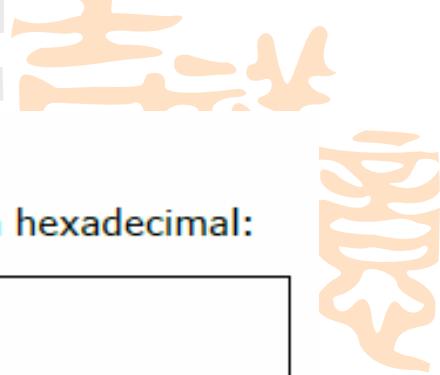
A hexadecimal is a 6 digit representation of a color. The first two digits(RR) represent a red value, the next two are a green value(GG), and the last are the blue value(BB).

A hexadecimal value can be taken from any graphics software like Adobe Photoshop, Paintshop Pro or MS Paint.

Each hexadecimal code will be preceded by a pound or hash sign #. Following is a list of few colors using hexadecimal notation.

Color	Color HEX
Black	#000000
Red	#FF0000
Green	#00FF00
Blue	#0000FF
Yellow	#FFFF00
Cyan	#00FFFF
Magenta	#FF00FF
Grey	#C0C0C0
White	#FFFFFF

# HTML – COLORS

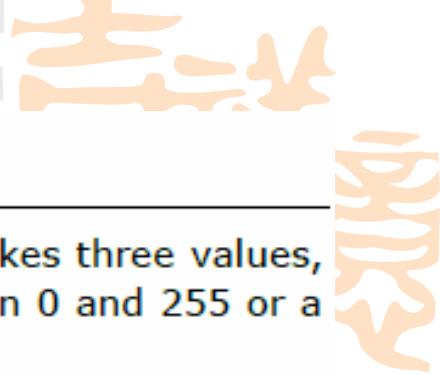


## Example

Here are the examples to set background of an HTML tag by color code in hexadecimal:

```
<!DOCTYPE html>

<html>
<head>
<title>HTML Colors by Hex</title>
</head>
<body text="#0000FF" bgcolor="#00FF00">
<p>Use different color hexa for for body and table and see the result.</p>
<table bgcolor="#000000">
<tr>
<td>
<font color="#FFFFFF">This text will appear white on black background.</font>
</td>
</tr>
</table>
</body>
</html>
```



## HTML Colors - RGB Values

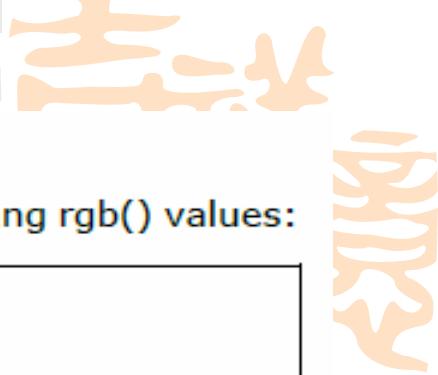
This color value is specified using the `rgb( )` property. This property takes three values, one each for red, green, and blue. The value can be an integer between 0 and 255 or a percentage.

**Note:** All the browsers does not support `rgb()` property of color so it is recommended not to use it.

Following is a list to show few colors using RGB values.

Color	Color RGB
Black	rgb(0,0,0)
Red	rgb(255,0,0)
Green	rgb(0,255,0)
Blue	rgb(0,0,255)
Yellow	rgb(255,255,0)
Cyan	rgb(0,255,255)
Magenta	rgb(255,0,255)
Grey	rgb(192,192,192)
White	rgb(255,255,255)

# HTML – COLORS



## Example

Here are the examples to set background of an HTML tag by color code using `rgb()` values:

```
<!DOCTYPE html>
<html>
<head>
<title>HTML Colors by RGB code</title>
</head>
<body text="rgb(0,0,255)" bgcolor="rgb(0,255,0)">
<p>Use different color code for body and table and see the result.</p>
<table bgcolor="rgb(0,0,0)">
<tr>
<td>
<font color="rgb(255,255,255)">This text will appear white on black
background.</font>
</td>
</tr>
</table>
</body>
</html>
```

# HTML – COMMENTS



HTML comments are placed in between `<!-- ... -->` tags. So, any content placed within `<!-- ... -->` tags will be treated as comment and will be completely ignored by the browser.

## Example

```
<!DOCTYPE html>  
  
<html>  
  
<head> <!-- Document Header Starts -->  
  
<title>This is document title</title>  
  
</head> <!-- Document Header Ends -->  
  
<body>  
  
<p>Document content goes here.....</p>  
  
</body>  
  
</html>
```





## Valid vs Invalid Comments

---

Comments do not nest which means a comment cannot be put inside another comment. Second the double-dash sequence "--" may not appear inside a comment except as part of the closing --> tag. You must also make sure that there are no spaces in the start-of-comment string.

### Example

But, following line is not a valid comment and will be displayed by the browser. This is because there is a space between the left angle bracket and the exclamation mark.

```
<!DOCTYPE html>
<html>
<head>
<title>Invalid Comment Example</title>
</head>
<body>
<!-- This is not a valid comment -->
<p>Document content goes here.....</p>
</body>
</html>
```



## Multiline Comments

You can comment multiple lines by the special beginning tag `<!--` and ending tag `-->` placed before the first line and end of the last line as shown in the given example below.

### Example

```
<!DOCTYPE html><html>
<head>
<title>Multiline Comments</title>
</head>
<body>
<!--
This is a multiline comment and it can
span through as many as lines you like.
-->
<p>Document content goes here.....</p>
</body>
</html>
```





## Commenting Script Code

Though you will learn JavaScript with HTML, in a separate tutorial, but here you must make a note that if you are using Java Script or VB Script in your HTML code then it is recommended to put that script code inside proper HTML comments so that old browsers can work properly.

### Example

```
<!DOCTYPE html><html>
<head>
<title>Commenting Script Code</title>
<script>
<!--
    document.write("Hello World!")
//-->
</script>
</head>
<body>
<p>Hello , World!</p>
</body>
</html>
```



## Commenting Style Sheets

Though you will learn using style sheets with HTML in a separate tutorial, but here you must make a note that if you are using Cascading Style Sheet (CSS) in your HTML code then it is recommended to put that style sheet code inside proper HTML comments so that old browsers can work properly.

### Example

```
<!DOCTYPE html><html>
<head>
<title>Commenting Style Sheets</title>
<style>
<!--
.example {
    border:1px solid #4a7d49;
}
//-->
</style>
</head>
<body>
<div class="example">Hello , World!</div>
</body>
</html>
```



HTML lets you specify metadata - additional important information about a document in a variety of ways. The META elements can be used to include name/value pairs describing properties of the HTML document, such as author, expiry date, a list of keywords, document author etc.

The **<meta>** tag is used to provide such additional information. This tag is an empty element and so does not have a closing tag but it carries information within its attributes.

- | You can include one or more meta tags in your document based on what information you want to keep in your document but in general, meta tags do not impact physical appearance of the document so from appearance point of view, it does not matter if you include them or not.





## Adding Meta Tags to Your Documents

You can add metadata to your web pages by placing `<meta>` tags inside the header of the document which is represented by `<head>` and `</head>` tags. A meta tag can have following attributes in addition to core attributes:

Attribute	Description
Name	Name for the property. Can be anything. Examples include, keywords, description, author, revised, generator etc.
content	Specifies the property's value.
scheme	Specifies a scheme to interpret the property's value (as declared in the content attribute).
http-equiv	Used for http response message headers. For example, http-equiv can be used to refresh the page or to set a cookie. Values include content-type, expires, refresh and set-cookie.





## Specifying Keywords

---

You can use <meta> tag to specify important keywords related to the document and later these keywords are used by the search engines while indexing your webpage for searching purpose.

## Document Description

---

You can use <meta> tag to give a short description about the document. This again can be used by various search engines while indexing your webpage for searching purpose.

## Document Revision Date

---

You can use <meta> tag to give information about when last time the document was updated. This information can be used by various web browsers while refreshing your webpage.

## Document Refreshing

---

A <meta> tag can be used to specify a duration after which your web page will keep refreshing automatically.





## Example

If you want your page keep refreshing after every 5 seconds then use the following syntax.

```
<!DOCTYPE html>
<html>
<head>
<title>Meta Tags Example</title>
<meta name="keywords" content="HTML, Meta Tags, Metadata" />
<meta name="description" content="Learning about Meta Tags." />
<meta name="revised" content="Tutorialspoint, 3/7/2014" />
<meta http-equiv="refresh" content="5" />
</head>
<body>
<p>Hello HTML5!</p>
</body>
</html>
```



## Html Background with Colors

The **bgcolor** attribute is used to control the background of an HTML element, specifically page body and table backgrounds. Following is the syntax to use bgcolor attribute with any HTML tag.

```
<tagname bgcolor="color_value"....>
```

This color\_value can be given in any of the following formats:

```
<!-- Format 1 - Use color name -->
```

```
<table bgcolor="lime" >
```

```
<!-- Format 2 - Use hex value -->
```

```
<table bgcolor="#f1f1f1" >
```

```
<!-- Format 3 - Use color value in RGB terms -->
```

```
<table bgcolor="rgb(0,0,120)" >
```



# HTML – BACKGROUNDS



## Example

Here are the examples to set background of an HTML tag:

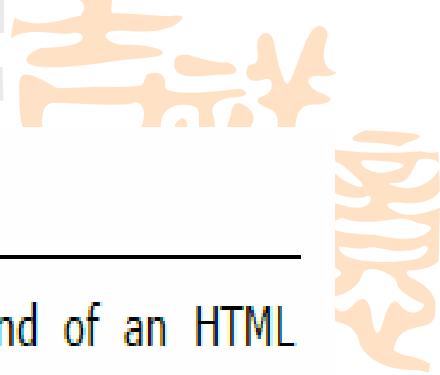
```
<!DOCTYPE html>
<html>
<head>
<title>HTML Background Colors</title>
</head>
<body>

<!-- Format 1 - Use color name --&gt;
&lt;table bgcolor="yellow" width="100%"&gt;
&lt;tr&gt;&lt;td&gt;
This background is yellow
&lt;/td&gt;&lt;/tr&gt;
&lt;/table&gt;

<!-- Format 2 - Use hex value --&gt;
&lt;table bgcolor="#6666FF" width="100%"&gt;
&lt;tr&gt;&lt;td&gt;
This background is sky blue
&lt;/td&gt;&lt;/tr&gt;
&lt;/table&gt;

<!-- Format 3 - Use color value in RGB terms --&gt;
&lt;table bgcolor="rgb(255,0,255)" width="100%"&gt;
&lt;tr&gt;&lt;td&gt;
This background is green
&lt;/td&gt;&lt;/tr&gt;
&lt;/table&gt;

&lt;/body&gt;
&lt;/html&gt;</pre>
```



## Html Background with Images

The **background** attribute can also be used to control the background of an HTML element, specifically page body and table backgrounds. You can specify an image to set background of your HTML page or table. Following is the syntax to use background attribute with any HTML tag.

**Note:** The *background* attribute is deprecated and it is recommended to use Style Sheet for background setting.

```
<tagname background="Image URL" ...>
```

The most frequently used image formats are JPEG, GIF and PNG images.



# HTML – BACKGROUNDS



## Example

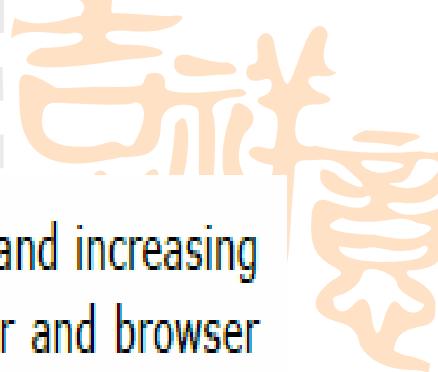
Here are the examples to set background images of a table.

```
<!DOCTYPE html>
<html>
<head>
<title>HTML Background Images</title>
</head>
<body>

<!-- Set table background -->
<table background="/images/html.gif" width="100%" height="100">
<tr><td>
This background is filled up with HTML image.
</td></tr>
</table>

</body>
</html>
```

## HTML – FONTS



Fonts play a very important role in making a website more user friendly and increasing content readability. Font face and color depends entirely on the computer and browser that is being used to view your page but you can use HTML `<font>` tag to add style, size, and color to the text on your website. You can use a `<basefont>` tag to set all of your text to the same size, face, and color.

The font tag is having three attributes called `size`, `color`, and `face` to customize your fonts. To change any of the font attributes at any time within your webpage, simply use the `<font>` tag. The text that follows will remain changed until you close with the `</font>` tag. You can change one or all of the font attributes within one `<font>` tag.





## Set Font Size

You can set content font size using **size** attribute. The range of accepted values is from 1(smallest) to 7(largest). The default size of a font is 3.

### Example

```
<!DOCTYPE html>
<html>
<head>
<title>Setting Font Size</title>
</head>
<body>
<font size="1">Font size="1"</font><br />
<font size="2">Font size="2"</font><br />
<font size="3">Font size="3"</font><br />
<font size="4">Font size="4"</font><br />
<font size="5">Font size="5"</font><br />
<font size="6">Font size="6"</font><br />
<font size="7">Font size="7"</font>
</body>
</html>
```



## Relative Font Size

You can specify how many sizes larger or how many sizes smaller than the preset font size should be. You can specify it like **<font size="+n">** or **<font size="-n">**

### Example

```
<!DOCTYPE html>

<html>
<head>
<title>Relative Font Size</title>
</head>
<body>
<font size="-1">Font size="-1"</font><br />
<font size="+1">Font size="+1"</font><br />
<font size="+2">Font size="+2"</font><br />
<font size="+3">Font size="+3"</font><br />
<font size="+4">Font size="+4"</font>
</body>
</html>
```



## Setting Font Face

You can set font face using *face* attribute but be aware that if the user viewing the page doesn't have the font installed, they will not be able to see it. Instead user will see the default font face applicable to the user's computer.

### Example

```
<!DOCTYPE html>

<html>
  <head>
    <title>Font Face</title>
  </head>
  <body>
    <font face="Times New Roman" size="5">Times New Roman</font><br />
    <font face="Verdana" size="5">Verdana</font><br />
    <font face="Comic sans MS" size="5">Comic Sans MS</font><br />
    <font face="WildWest" size="5">WildWest</font><br />
    <font face="Bedrock" size="5">Bedrock</font><br />
  </body>
</html>
```

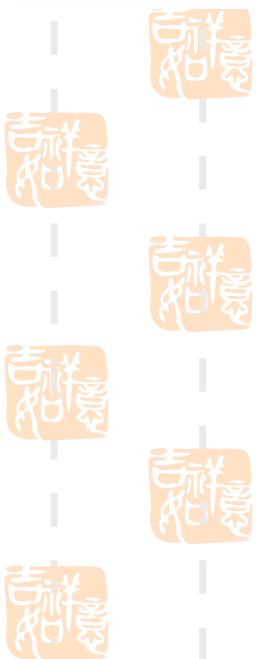


## Specify alternate font faces

A visitor will only be able to see your font if they have that font installed on their computer. So, it is possible to specify two or more font face alternatives by listing the font face names, separated by a comma.

```
<font face="arial,helvetica">  
<font face="Lucida Calligraphy,Comic Sans MS,Lucida Console">
```

When your page is loaded, their browser will display the first font face available. If none of the given fonts are installed, then it will display the default font face *Times New Roman*.





## Setting Font Color

You can set any font color you like using *color* attribute. You can specify the color that you want by either the color name or hexadecimal code for that color.

### Example

```
<!DOCTYPE html>

<html>
  <head>
    <title>Setting Font Color</title>
  </head>
  <body>
    <font color="#FF00FF">This text is in pink</font><br />
    <font color="red">This text is red</font>
  </body>
</html>
```





## Insert Image

You can insert any image in your web page by using **<img>** tag. Following is the simple syntax to use this tag.

```

```

The **<img>** tag is an empty tag, which means that, it can contain only list of attributes and it has no closing tag.

## Example

To try following example, let's keep our HTML file test.htm and image file test.png in the same directory:

```
<!DOCTYPE html>
<html>
<head>
<title>Using Image in Webpage</title>
</head>
<body>
<p>Simple Image Insert</p>

</body>
</html>
```

### Note:

You can use PNG, JPEG or GIF image file based on your comfort but make sure you specify correct image file name in **src** attribute. Image name is always case sensitive.

The **alt** attribute is a mandatory attribute which specifies an alternate text for an image, if the image cannot be displayed.



## Set Image Location

Usually we keep all the images in a separate directory. So let's keep HTML file test.htm in our home directory and create a subdirectory **images** inside the home directory where we will keep our image test.png.

### Example

Assuming our image location is "image/test.png", try the following example:

```
<!DOCTYPE html>
<html>
<head>
<title>Using Image in Webpage</title>
</head>
<body>
<p>Simple Image Insert</p>

</body>
</html>
```





## Set Image Width/Height

You can set image width and height based on your requirement using **width** and **height** attributes. You can specify width and height of the image in terms of either pixels or percentage of its actual size.

### Example

```
<!DOCTYPE html>
<html>
<head>
<title>Set Image Width and Height</title>
</head>
<body>
<p>Setting image width and height</p>

</body>
</html>
```





## Set Image Border

By default, image will have a border around it, you can specify border thickness in terms of pixels using border attribute. A thickness of 0 means, no border around the picture.

### Example

```
<!DOCTYPE html>
<html>
<head>
<title>Set Image Border</title>
</head>
<body>
<p>Setting image Border</p>

</body>
</html>
```





## Set Image Alignment

By default, image will align at the left side of the page, but you can use **align** attribute to set it in the center or right.

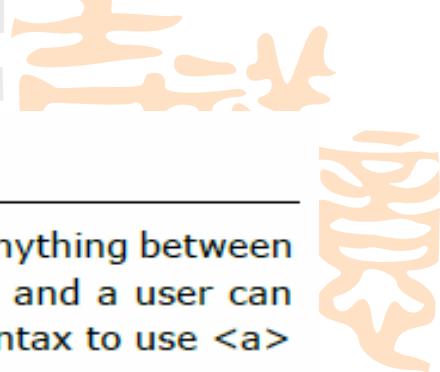
### Example

```
<!DOCTYPE html>

<html>
<head>
<title>Set Image Alignment</title>
</head>
<body>
<p>Setting image Alignment</p>

</body>
</html>
```





## Linking Documents

A link is specified using HTML tag `<a>`. This tag is called **anchor tag** and anything between the opening `<a>` tag and the closing `</a>` tag becomes part of the link and a user can click that part to reach to the linked document. Following is the simple syntax to use `<a>` tag.

```
<a href="Document URL" ... attributes-list>Link Text</a>
```

## Example

Let's try following example which links <http://www.tutorialspoint.com> at your page:

```
<!DOCTYPE html>
<html>
<head>
<title>Hyperlink Example</title>
</head>
<body>
<p>Click following link</p>
<a href="http://www.tutorialspoint.com" target="_self">Tutorials Point</a>
</body>
</html>
```



## Linking to a Page Section

You can create a link to a particular section of a given webpage by using **name** attribute. This is a two-step process.

First create a link to the place where you want to reach with-in a webpage and name it using `<a>` tag as follows:

```
<h1>HTML Text Links <a name="top"></a></h1>
```

Second step is to create a hyperlink to link the document and place where you want to reach:

```
<a href="/html/html_text_links.htm#top">Go to the Top</a>
```

This will produce following link, where you can click on the link generated **Go to the Top** to reach to the top of the HTML Text Link tutorial.





## Setting Link Colors

You can set colors of your links, active links and visited links using **link**, **alink** and **vlink** attributes of **<body>** tag.

### Example

Save the following in test.htm and open it in any web browser to see how **link**, **alink** and **vlink** attributes work.

```
<!DOCTYPE html>
<html>
<head>
<title>Hyperlink Example</title>
<base href="http://www.tutorialspoint.com/">
</head>
<body alink="#54A250" link="#040404" vlink="#F40633">
<p>Click following link</p>
<a href="/html/index.htm" target="_blank" >HTML Tutorial</a>
</body>
</html>
```





## Download Links

You can create text link to make your PDF, or DOC or ZIP files downloadable. This is very simple; you just need to give complete URL of the downloadable file as follows:

```
<!DOCTYPE html>
<html>
<head>
<title>Hyperlink Example</title>
</head>
<body>
<a href="http://www.tutorialspoint.com/page.pdf">Download PDF File</a>
</body>
</html>
```





## Example

It's simple to use an image as hyperlink. We just need to use an image inside hyperlink at the place of text as shown below:

```
<!DOCTYPE html>

<html>
  <head>
    <title>Image Hyperlink Example</title>
  </head>
  <body>
    <p>Click following link</p>
    <a href="http://www.tutorialspoint.com" target="_self">
      
    </a>
  </body>
</html>
```





## HTML Email Tag

HTML `<a>` tag provides you option to specify an email address to send an email. While using `<a>` tag as an email tag, you will use `mailto: email address` along with `href` attribute. Following is the syntax of using `mailto` instead of using `http`.

```
<a href= "mailto: abc@example.com">Send Email</a>
```

This code will generate the following link which you can use to send email.

```
Send Email
```

Now, if a user clicks this link, it launches one Email Client (like Lotus Notes, Outlook Express etc. ) installed on your user's computer. There is another risk to use this option to send email because if user do not have email client installed on their computer then it would not be possible to send email.

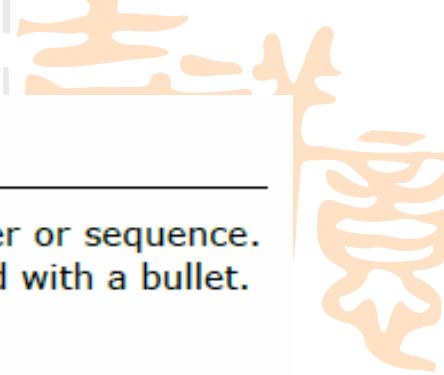




HTML offers web authors three ways for specifying lists of information. All lists must contain one or more list elements. Lists may contain:

- **<ul>** - An unordered list. This will list items using plain bullets.
- **<ol>** - An ordered list. This will use different schemes of numbers to list your items.
- **<dl>** - A definition list. This arranges your items in the same way as they are arranged in a dictionary.





## HTML Unordered Lists

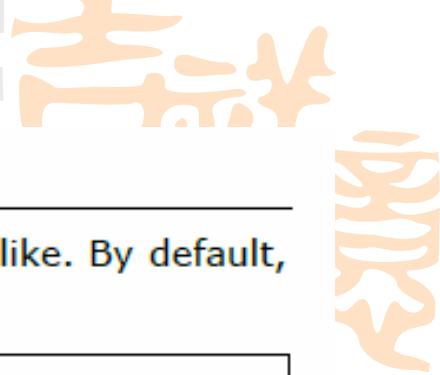
An unordered list is a collection of related items that have no special order or sequence. This list is created by using HTML **<ul>** tag. Each item in the list is marked with a bullet.

### Example

```
<!DOCTYPE html>

<html>
<head>
<title>HTML Unordered List</title>
</head>
<body>
<ul>
<li>Beetroot</li>
<li>Ginger</li>
<li>Potato</li>
<li>Radish</li>
</ul>
</body>
</html>
```





## The type Attribute

You can use **type** attribute for `<ul>` tag to specify the type of bullet you like. By default, it is a disc. Following are the possible options:

```
<ul type="square">  
<ul type="disc">  
<ul type="circle">
```

### Example

Following is an example where we used `<ul type="square">`

```
<!DOCTYPE html>  
<html>  
<head>  
<title>HTML Unordered List</title>  
</head>  
<body>  
    <ul type="square">  
        <li>Beetroot</li>  
        <li>Ginger</li>  
        <li>Potato</li>  
        <li>Radish</li>  
    </ul>  
</body>  
</html>
```





## HTML Ordered Lists

If you are required to put your items in a numbered list instead of bulleted, then HTML ordered list will be used. This list is created by using `<ol>` tag. The numbering starts at one and is incremented by one for each successive ordered list element tagged with `<li>`.

### Example

```
<!DOCTYPE html>
<html>
<head>
<title>HTML Ordered List</title>
</head>
<body>
<ol>
<li>Beetroot</li>
<li>Ginger</li>
<li>Potato</li>
<li>Radish</li>
</ol>
</body>
</html>
```



## The type Attribute

You can use **type** attribute for `<ol>` tag to specify the type of numbering you like. By default, it is a number. Following are the possible options:

```
<ol type="1"> - Default-Case Numerals.  
<ol type="I"> - Upper-Case Numerals.  
<ol type="i"> - Lower-Case Numerals.  
<ol type="a"> - Lower-Case Letters.  
<ol type="A"> - Upper-Case Letters.
```

### Example

Following is an example where we used `<ol type="1">`

```
<!DOCTYPE html>  
<html>  
<head>  
<title>HTML Ordered List</title>  
</head>  
<body>  
    <ol type="1">  
        <li>Beetroot</li>  
        <li>Ginger</li>  
        <li>Potato</li>  
        <li>Radish</li>  
    </ol>  
</body>  
</html>
```



## The start Attribute

You can use **start** attribute for `<ol>` tag to specify the starting point of numbering you need. Following are the possible options:

- |  |                            |
|--|----------------------------|
| <code>&lt;ol type="1" start="4"&gt;</code> | - Numerals starts with 4.  |
| <code>&lt;ol type="I" start="4"&gt;</code> | - Numerals starts with IV. |
| <code>&lt;ol type="i" start="4"&gt;</code> | - Numerals starts with iv. |
| <code>&lt;ol type="a" start="4"&gt;</code> | - Letters starts with d.   |
| <code>&lt;ol type="A" start="4"&gt;</code> | - Letters starts with D.   |

## Example

Following is an example where we used `<ol type="i" start="4" >`

```
<!DOCTYPE html>
<html>
<head>
<title>HTML Ordered List</title>
</head>
<body>
  <ol type="i" start="4">
    <li>Beetroot</li>
    <li>Ginger</li>
    <li>Potato</li>
    <li>Radish</li>
  </ol>
</body>
</html>
```



## HTML Definition Lists

HTML and XHTML supports a list style which is called **definition lists** where entries are listed like in a dictionary or encyclopedia. The definition list is the ideal way to present a glossary, list of terms, or other name/value list.

Definition List makes use of following three tags.

- `<dl>` - Defines the start of the list
- `<dt>` - A term
- `<dd>` - Term definition
- `</dl>` - Defines the end of the list

### Example

```
<!DOCTYPE html>
<html>
<head>
<title>HTML Definition List</title>
</head>
<body>
<dl>
<dt><b>HTML</b></dt>
<dd>This stands for Hyper Text Markup Language</dd>
<dt><b>HTTP</b></dt>
<dd>This stands for Hyper Text Transfer Protocol</dd>
</dl>
</body>
</html>
```



# HTML – TABLES



The HTML tables are created using the **<table>** tag in which the **<tr>** tag is used to create table rows and **<td>** tag is used to create data cells.

## Example

```
<!DOCTYPE html>
<html>
<head>
<title>HTML Tables</title>
</head>
<body>
<table border="1">
<tr>
<td>Row 1, Column 1</td>
<td>Row 1, Column 2</td>
</tr>
<tr>
<td>Row 2, Column 1</td>
<td>Row 2, Column 2</td>
</tr>
</table>
</body>
</html>
```

### Note:

Here, the **border** is an attribute of **<table>** tag and it is used to put a border across all the cells. If you do not need a border, then you can use **border="0"**.



## Table Heading

Table heading can be defined using **<th>** tag. This tag will be put to replace **<td>** tag, which is used to represent actual data cell. Normally you will put your top row as table heading as shown below, otherwise you can use **<th>** element in any row.

### Example

```
<!DOCTYPE html>
<html>
<head>
<title>HTML Table Header</title>
</head>
<body>
<table border="1">
<tr>
<th>Name</th>
<th>Salary</th>
</tr>
<tr>
<td>Ramesh Raman</td>
<td>5000</td>
</tr>
<tr>
<td>Shabbir Hussein</td>
<td>7000</td>
</tr>
</table>
</body>
</html>
```



## Cellpadding and Cellspacing Attributes

There are two attributes called *cellpadding* and *cellspacing* which you will use to adjust the white space in your table cells. The cellspacing attribute defines the width of the border, while cellpadding represents the distance between cell borders and the content within a cell.

### Example

```
<!DOCTYPE html>
<html>
<head>
<title>HTML Table Cellpadding</title>
</head>
<body>
<table border="1" cellpadding="5" cellspacing="5">
<tr>
<th>Name</th>
<th>Salary</th>
</tr>
<tr>
<td>Ramesh Raman</td>
<td>5000</td>
</tr>
<tr>
<td>Shabbir Hussein</td>
<td>7000</td>
</tr>
</table>
</body>
</html>
```



## Colspan and Rowspan Attributes

You will use **colspan** attribute if you want to merge two or more columns into a single column. Similar way you will use **rowspan** if you want to merge two or more rows.

### Example

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



## Tables Backgrounds

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

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

You can also set border color also using **bordercolor** attribute.

### Example

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

# HTML – TABLES



Here is an example of using **background** attribute. Here we will use an image available in /images directory.

```
<!DOCTYPE html>
<html>
<head>
<title>HTML Table Background</title>
</head>
<body>
<table border="1" bordercolor="green" background="/images/test.png">
<tr>
<th>Column 1</th>
<th>Column 2</th>
<th>Column 3</th>
</tr>
<tr><td rowspan="2">Row 1 Cell 1</td><td>Row 1 Cell 2</td><td>Row 1 Cell 3</td></tr>
<tr><td>Row 2 Cell 2</td><td>Row 2 Cell 3</td></tr>
<tr><td colspan="3">Row 3 Cell 1</td></tr>
</table>
</body>
</html>
```



## Table Height and Width

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

### Example

```
<!DOCTYPE html>
<html>
<head>
<title>HTML Table Width/Height</title>
</head>
<body>
<table border="1" width="400" height="150">
<tr>
<td>Row 1, Column 1</td>
<td>Row 1, Column 2</td>
</tr>
<tr>
<td>Row 2, Column 1</td>
<td>Row 2, Column 2</td>
</tr>
</table>
</body>
</html>
```



## Nested Tables

You can use one table inside another table. Not only tables you can use almost all the tags inside table data tag <td>.

### Example

Following is the example of using another table and other tags inside a table cell.

```
<!DOCTYPE html>
<html>
<head>
<title>HTML Table</title>
</head>
<body>
<table border="1" width="100%">
<tr>
<td>
<table border="1" width="100%">
<tr>
<th>Name</th>
<th>Salary</th>
</tr>
<tr>
<td>Ramesh Raman</td>
<td>5000</td>
</tr>
<tr>
<td>Shabbir Hussein</td>
<td>7000</td>
</tr>
</table>
</td>
</tr>
</table>
</body>
</html>
```

# HTML – FORMS



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

The HTML **<form>** tag is used to create an HTML form and it has following syntax:

```
<form action="Script URL" method="GET|POST">  
    form elements like input, textarea etc.  
</form>
```

## Form Attributes

Apart from common attributes, following is a list of the most frequently used form attributes:

Attribute	Description
action	Backend script ready to process your passed data.
method	Method to be used to upload data. The most frequently used are GET and POST methods.
target	Specify the target window or frame where the result of the script will be displayed. It takes values like _blank, _self, _parent etc.
enctype	You can use the enctype attribute to specify how the browser encodes the data before it sends it to the server. Possible values are:  application/x-www-form-urlencoded - This is the standard method most forms use in simple scenarios.  multipart/form-data - This is used when you want to upload binary data in the form of files like image, word file etc.



## HTML Form Controls

---

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

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

## Text Input Controls

---

There are three types of text input used on forms:

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





## Single-line text input controls

This control is used for items that require only one line of user input, such as search boxes or names. They are created using HTML <input> tag.

### Example

Here is a basic example of a single-line text input used to take first name and last name:

```
<!DOCTYPE html>
<html>
<head>
<title>Text Input Control</title>
</head>
<body>
<form >
First name: <input type="text" name="first_name" />
<br>
Last name: <input type="text" name="last_name" />
</form>
</body>
</html>
```

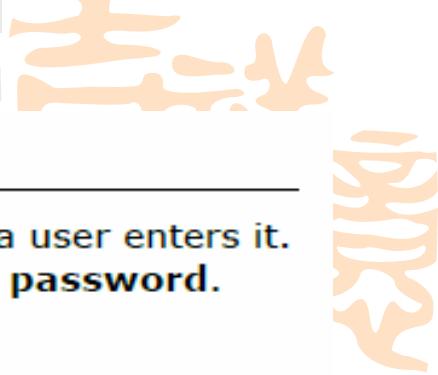


## Attributes

Following is the list of attributes for <input> tag for creating text field.

Attribute	Description
type	Indicates the type of input control and for text input control it will be set to text.
name	Used to give a name to the control which is sent to the server to be recognized and get the value.
value	This can be used to provide an initial value inside the control.
size	Allows to specify the width of the text-input control in terms of characters.
maxlength	Allows to specify the maximum number of characters a user can enter into the text box.





## Password Input controls

This is also a single-line text input but it masks the character as soon as a user enters it. They are also created using HTML `<input>` tag but type attribute is set to **password**.

### Example

Here is a basic example of a single-line password input used to take user password:

```
<!DOCTYPE html>

<html>
<head>
<title>Password Input Control</title>
</head>
<body>
<form >

User ID : <input type="text" name="user_id" />
<br>
Password: <input type="password" name="password" />
</form>
</body>
</html>
```



## Attributes

Following is the list of attributes for <input> tag for creating password field.

Attribute	Description
type	Indicates the type of input control and for password input control it will be set to password.
name	Used to give a name to the control which is sent to the server to be recognized and get the value.
value	This can be used to provide an initial value inside the control.
size	Allows to specify the width of the text-input control in terms of characters.
maxlength	Allows to specify the maximum number of characters a user can enter into the text box.





## Multiple-Line Text Input Controls

This is used when the user is required to give details that may be longer than a single sentence. Multi-line input controls are created using HTML <textarea> tag.

### Example

Here is a basic example of a multi-line text input used to take item description:

```
<!DOCTYPE html>
<html>
<head>
<title>Multiple-Line Input Control</title>
</head>
<body>
<form>
Description: <br />
<textarea rows="5" cols="50" name="description">
Enter description here...
</textarea>
</form>
</body>
</html>
```



## Attributes

---

Following is the list of attributes for <textarea> tag.

Attribute	Description
name	Used to give a name to the control which is sent to the server to be recognized and get the value.
rows	Indicates the number of rows of text area box.
cols	Indicates the number of columns of text area box





## Checkbox Control

Checkboxes are used when more than one option is required to be selected. They are also created using HTML `<input>` tag but type attribute is set to **checkbox**.

### Example

Here is an example HTML code for a form with two checkboxes:

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

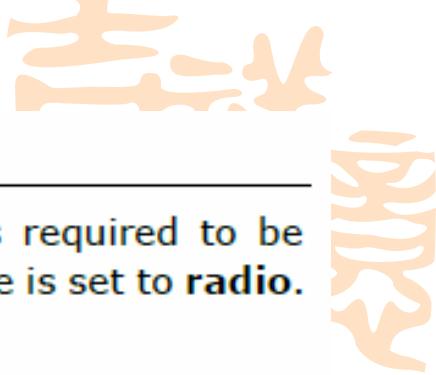


## Attributes

Following is the list of attributes for <checkbox> tag.

Attribute	Description
type	Indicates the type of input control and for checkbox input control it will be set to checkbox.
name	Used to give a name to the control which is sent to the server to be recognized and get the value.
value	The value that will be used if the checkbox is selected.
checked	Set to <i>checked</i> if you want to select it by default.





## Radio Button Control

Radio buttons are used when out of many options, just one option is required to be selected. They are also created using HTML <input> tag but type attribute is set to **radio**.

### Example

Here is example HTML code for a form with two radio buttons:

```
<!DOCTYPE html>
<html>
<head>
<title>Radio Box Control</title>
</head>
<body>
<form>
<input type="radio" name="subject" value="maths"> Maths
<input type="radio" name="subject" value="physics"> Physics
</form>
</body>
</html>
```



## Attributes

Following is the list of attributes for radio button.

Attribute	Description
type	Indicates the type of input control and for checkbox input control it will be set to radio.
name	Used to give a name to the control which is sent to the server to be recognized and get the value.
value	The value that will be used if the radio box is selected.
checked	Set to <i>checked</i> if you want to select it by default.





## Select Box Control

A select box, also called drop down box which provides option to list down various options in the form of drop down list, from where a user can select one or more options.

### Example

Here is example HTML code for a form with one drop down box

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



## Attributes

Following is the list of important attributes of <select> tag:

Attribute	Description
name	Used to give a name to the control which is sent to the server to be recognized and get the value.
size	This can be used to present a scrolling list box.
multiple	If set to "multiple" then allows a user to select multiple items from the menu.

Following is the list of important attributes of <option> tag:

Attribute	Description
value	The value that will be used if an option in the select box is selected.
selected	Specifies that this option should be the initially selected value when the page loads.
label	An alternative way of labeling options



## Button Controls

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

Type	Description
submit	This creates a button that automatically submits a form.
reset	This creates a button that automatically resets form controls to their initial values.
button	This creates a button that is used to trigger a client-side script when the user clicks that button.
image	This creates a clickable button but we can use an image as background of the button.





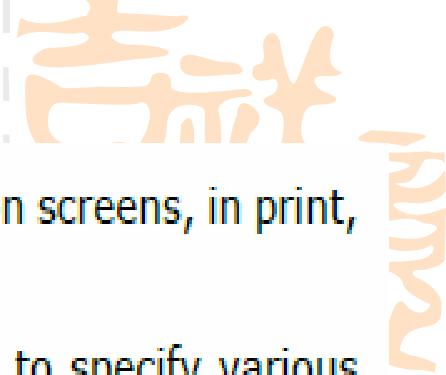
## Example

Here is example HTML code for a form with three types of buttons:

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



# HTML – STYLE SHEET



Cascading Style Sheets (CSS) describe how documents are presented on screens, in print, or perhaps how they are pronounced.

Cascading Style Sheets (CSS) provide easy and effective alternatives to specify various attributes for the HTML tags. Using CSS, you can specify a number of style properties for a given HTML element. Each property has a name and a value, separated by a colon (:). Each property declaration is separated by a semi-colon (;).

You can use CSS in three ways in your HTML document:

- **External Style Sheet:** Define style sheet rules in a separate .css file and then include that file in your HTML document using HTML <link> tag.
- **Internal Style Sheet:** Define style sheet rules in header section of the HTML document using <style> tag.
- **Inline Style Sheet:** Define style sheet rules directly along-with the HTML elements using **style** attribute.



# HTML – STYLE SHEET



## Example

First let's consider an example of HTML document which makes use of `<font>` tag and associated attributes to specify text color and font size:

```
<!DOCTYPE html>
<html>
<head>
<title>HTML CSS</title>
</head>
<body>
<p><font color="green" size="5">Hello, World!</font></p>
</body>
</html>
```

We can re-write above example with the help of Style Sheet as follows:

```
<!DOCTYPE html>
<html>
<head>
<title>HTML CSS</title>
</head>
<body>
<p style="color:green;font-size:24px;">Hello, World!</p>
</body>
</html>
```



## External Style Sheet

If you need to use your style sheet to various pages, then it's always recommended to define a common style sheet in a separate file. A cascading style sheet file will have extension as **.css** and it will be included in HTML files using **<link>** tag.

### Example

Consider we define a style sheet file **style.css** which has following rules:

```
.red{  
    color: red;  
}  
.thick{  
    font-size:20px;  
}  
.green{  
    color:green;  
}
```



# HTML – STYLE SHEET



Here we defined three CSS rules which will be applicable to three different classes defined for the HTML tags. Now let's make use of the above external CSS file in our following HTML document:

```
<!DOCTYPE html>

<html>
<head>
<title>HTML External CSS</title>
<link rel="stylesheet" type="text/css" href="/html/style.css">
</head>
<body>
<p class="red">This is red</p>
<p class="thick">This is thick</p>
<p class="green">This is green</p>
<p class="thick green">This is thick and green</p>
</body>
</html>
```





## Internal Style Sheet

If you want to apply Style Sheet rules to a single document only, then you can include those rules in header section of the HTML document using `<style>` tag.

Rules defined in internal style sheet overrides the rules defined in an external CSS file.

### Example

Let's re-write above example once again, but here we will write style sheet rules in the same HTML document using `<style>` tag:

```
<!DOCTYPE html>
<html>
<head>
<title>HTML Internal CSS</title>
<style type="text/css">
.red{
  color: red;
}
.thick{
  font-size:20px;
}
.green{
  color:green;
}
</style>
</head>
<body>
<p class="red">This is red</p>
<p class="thick">This is thick</p>
<p class="green">This is green</p>
<p class="thick green">This is thick and green</p>
</body>
</html>
```



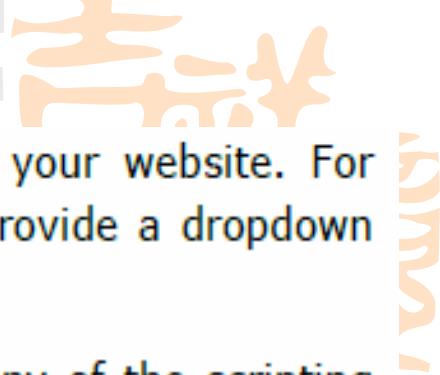
## Inline Style Sheet

You can apply style sheet rules directly to any HTML element using **style** attribute of the relevant tag. This should be done only when you are interested to make a particular change in any HTML element only.

Rules defined inline with the element overrides the rules defined in an external CSS file as well as the rules defined in <style> element.

### Example

```
<!DOCTYPE html>
<html>
<head>
<title>HTML Inline CSS</title>
</head>
<body>
<p style="color:red;">This is red</p>
<p style="font-size:20px;">This is thick</p>
<p style="color:green;">This is green</p>
<p style="color:green;font-size:20px;">This is thick and green</p>
</body>
</html>
```



A **script** is a small piece of program that can add interactivity to your website. For example, a script could generate a pop-up alert box message, or provide a dropdown menu. This script could be written using JavaScript or VBScript.

You can write various small functions, called event handlers using any of the scripting language and then you can trigger those functions using HTML attributes.

Now-a-days, only **JavaScript** and associated frameworks are being used by most of the web developers, VBScript is not even supported by various major browsers.

You can keep JavaScript code in a separate file and then include it wherever it's needed, or you can define functionality inside HTML document itself. Let's see both the cases one by one with suitable examples.

## External JavaScript

---

If you are going to define a functionality which will be used in various HTML documents then it's better to keep that functionality in a separate JavaScript file and then include that file in your HTML documents. A JavaScript file will have extension as **.js** and it will be included in HTML files using **<script>** tag.





## Example

Consider we define a small function using JavaScript in **script.js** which has following code:

```
function Hello()
{
    alert("Hello, World");
}
```

Now let's make use of the above external JavaScript file in our following HTML document:

```
<!DOCTYPE html>
<html>
<head>
<title>JavaScript External Script</title>
<script src="/html/script.js" type="text/JavaScript"/></script>
</head>
<body>
<input type="button" onclick="Hello();" name="ok" value="Click Me" />
</body>
</html>
```



## Internal Script

You can write your script code directly into your HTML document. Usually we keep script code in header of the document using `<script>` tag, otherwise there is no restriction and you can put your source code anywhere in the document but inside `<script>` tag.

### Example

```
<!DOCTYPE html>
<html>
<head>
<title>JavaScript Internal Script</title>
<base href="http://www.tutorialspoint.com/" />
<script type="text/JavaScript">
function Hello(){
    alert("Hello, World");
}
</script>
</head>
<body>
<input type="button" onclick="Hello();" name="ok" value="Click Me" />
</body>
</html>
```



## Event Handlers

Event handlers are nothing but simply defined functions which can be called against any mouse or keyboard event. You can define your business logic inside your event handler which can vary from a single to 1000s of line code.

Following example explains how to write an event handler. Let's write one simple function `EventHandler()` in the header of the document. We will call this function when any user brings mouse over a paragraph.

```
<!DOCTYPE html>
<html>
<head>
<title>Event Handlers Example</title>
<base href="http://www.tutorialspoint.com/" />
<script type="text/JavaScript">
function EventHandler(){
    alert("I'm event handler!!");
}
</script>
</head>
<body>
<p onmouseover="EventHandler();">Bring your mouse here to see an alert</p>
</body>
</html>
```