HTML Headings

Headings are defined with the <h1> to <h6> tags.

<h1> defines the most important heading. <h6> defines the least important heading.

Example

```
<h1>This is a heading</h1>
<h2>This is a heading</h2>
<h3>This is a heading</
h3>
```

Note: Browsers automatically add some empty space (a margin) before and after each heading.

Headings Are Important

Use HTML headings for headings only. Don't use headings to make text **BIG** or **bold**.

Search engines use your headings to index the structure and content of your web pages.

Users skim your pages by its headings. It is important to use headings to show the document structure.

h1 headings should be main headings, followed by h2 headings, then the less important h3, and so on.

HTML Horizontal Rules

The **<hr>** tag creates a horizontal line in an HTML page.

The hr element can be used to separate content:

Example

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

The HTML <head> Element

The HTML **<head>** element has nothing to do with HTML headings.

The HTML <head> element contains **meta data**. Meta data are not displayed.

The HTML <head> element is placed between the <html> tag and the <body> tag:

Example

```
<!DOCTYPE html>
<html>

<head>
  <title>My First HTML</title>
  <meta charset="UTF-8">
</head>

<body>
.
.
.
```

Meta data means data **about** data. HTML meta data is data **about** the HTML document.

The HTML <title> Element

The HTML **<title>** element is meta data. It defines the HTML document's title.

The title will not be displayed in the document, but might be displayed in the browser tab.

The HTML <meta> Element

The HTML **<meta>** element is also meta data.

It can be used to define the character set, and other information about the HTML document.

More Meta Elements

In the chapter about HTML styles you discover more meta elements:

The HTML **<style>** element is used to define internal CSS style sheets.

The HTML **<link>** element is used to define external CSS style sheets.

HTML Tag Reference

Tag	Description
<html>	Defines an HTML document

<body>	Defines the document's body
<head>	Defines the document's head element
<h1> to <h6>	Defines HTML headings
<hr>	Defines a horizontal line

HTML Paragraphs

HTML documents are divided into paragraphs.

The HTML **<p>** element defines a **paragraph**.

Example

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


Browsers automatically add an empty line before and after a paragraph.

HTML Display

- You cannot be sure how HTML will be displayed.
- Large or small screens, and resized windows will create different results.
- With HTML, you cannot change the output by adding extra spaces or extra lines in your HTML code.
- The browser will remove extra spaces and extra lines when the page is displayed.
- Any number of spaces, and any number of new lines, count as **only one space**.

Example

```
<p>
This paragraph
contains a lot of lines
in the source code,
but the browser
ignores it.
</p>

<p>
This paragraph
contains        a lot of spaces
in the source        code,
but the        browser
ignores it.
</p>
```

Don't Forget the End Tag

Most browsers will display HTML correctly even if you forget the end tag:

Example

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

The example above will work in most browsers, but don't rely on it.

Forgetting the end tag can produce unexpected results or errors.

 Stricter versions of HTML, like XHTML, do not allow you to skip the end tag.

HTML Line Breaks

The HTML **
** element defines a **line break**.

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

Example

```
<p>This is<br>a para<br>graph with line breaks</p>
```

The
 element is an empty HTML element. It has no end tag.

The Poem Problem

Example

```
<p>This poem will display as one line:</p>

<p>

  My Bonnie lies over the ocean.

  My Bonnie lies over the sea.

  My Bonnie lies over the ocean.

  Oh, bring back my Bonnie to me.

</p>
```

The HTML <pre> Element

The HTML **<pre>** element defines a block of **pre-formatted** text, with structured spaces and lines.

To display anything, with right spacing and line-breaks, you must wrap the text in a **<pre>** element:

Example

```
<p>This will display as a poem:</p>

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

  My Bonnie lies over the sea.

  My Bonnie lies over the ocean.

  Oh, bring back my Bonnie to me.

</pre>
```

HTML Tag Reference

Tag	Description
<p>	Defines a paragraph
 	Inserts a single line break
<pre>	Defines pre-formatted text

HTML Styles

I am Red

<h2 style="color:red">I am Red</h2>

I am Blue

<h2 style="color:blue">I am Blue</h2>

HTML Styling

Every HTML element has a **default style** (background color is white, text color is black, text-size is 12px ...)

Changing the default style of an HTML element, can be done with the **style attribute**.

This example changes the default background color from white to lightgrey:

Example

<body style="background-color:lightgrey">

<h1>This is a heading</h1>

This is a paragraph.

</body>

The HTML Style Attribute

The HTML style attribute has the following **syntax**:

style="*property*:*value*"

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

HTML Text Color

The **color** property defines the text color to be used for an HTML element:

Example

<!DOCTYPE html>
<html>

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

</html>
```

HTML Text Fonts

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

Example

```
<!DOCTYPE html>
<html>

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

</html>
```

HTML Text Size

The **font-size** property defines the text size to be used for an HTML element:

Example

```
<!DOCTYPE html>
<html>

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

</html>
```

HTML Text Alignment

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

Example

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

```
<body>
  <h1 style="text-align:center">Centered Heading</h1>
  <p>This is a paragraph.</p>
</body>

</html>
```

Chapter Summary

- Use the **style** attribute for styling HTML elements
- Use **background-color** for background color
- Use **color** for text colors
- Use **font-family** for text fonts
- Use **font-size** for text sizes
- Use **text-align** for text alignment

HTML Text Formatting Elements

Text Formatting

This text is bold

This text is italic

This is superscript

HTML Formatting Elements

In the previous chapter, you learned about HTML **styling**, using the HTML **style attribute**.

HTML also defines special **elements**, for defining text with a special **meaning**.

HTML uses elements like <b> and <i> for formatting output, like **bold** or *italic* text.

Formatting elements were designed to display special **types of text**:

- Bold text
- Important text
- Italic text
- Emphasized text
- Marked text
- Small text
- Deleted text
- Inserted text
- Subscripts
- Superscripts

HTML Bold and Strong Formatting

The HTML **<b>** element defines **bold** text, without any extra importance.

Example

```
<p>This text is normal.</p>

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

The HTML **** element defines **strong** text, with added semantic "strong" importance.

Example

```
<p>This text is normal.</p>

<p><strong>This text is strong</strong>.</p>
```

HTML *Italic* and *Emphasized* Formatting

The HTML **<i>** element defines *italic* text, without any extra importance.

Example

```
<p>This text is normal.</p>

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

The HTML **** element defines *emphasized* text, with added semantic importance.

Example

```
<p>This text is normal.</p>

<p><em>This text is emphasized</em>.</p>
```

Browsers display as <b>, and as <i>.

However, there is a difference in the meaning of these tags: <b> and <i> defines bold and italic text,
but and means that the text is "important".

HTML Small Formatting

The HTML **<small>** element defines **small** text:

Example

```
<h2>HTML <small>Small</small> Formatting</h2>
```

HTML Marked Formatting

The HTML **<mark>** element defines **marked** or highlighted text:

Example

```
<h2>HTML <mark>Marked</mark> Formatting</h2>
```

HTML ~~Deleted~~ Formatting

The HTML **<del>** element defines **deleted** (removed) of text.

Example

```
<p>My favorite color is <del>blue</del> red.</p>
```

HTML Inserted Formatting

The HTML **<ins>** element defines **inserted** (added) text.

Example

```
<p>My favorite <ins>color</ins> is red.</p>
```

HTML Subscript Formatting

The HTML **<sub>** element defines **subscripted** text.

Example

```
<p>This is <sub>subscripted</sub> text.</p>
```

HTML Superscript Formatting

The HTML **<sup>** element defines **superscripted** text.

Example

```
<p>This is <sup>superscripted</sup> text.</p>
```

HTML Text Formatting Elements

Tag	Description
<b>	Defines bold text
	Defines emphasized text
<i>	Defines italic text
<small>	Defines smaller text
	Defines important text

<sub>	Defines subscripted text
<sup>	Defines superscripted text
<ins>	Defines inserted text
<del>	Defines deleted text
<mark>	Defines marked/highlighted text

HTML Computer Code Elements

Computer Code

```
var person = {
   firstName:"John",
   lastName:"Doe",
   age:50,
   eyeColor:"blue"
}
```

HTML `Computer Code` Formatting

Normally, HTML uses **variable** letter size, and variable letter spacing.

This is not wanted when displaying examples of **computer code**.

The **<kbd>**, **<samp>**, and **<code>** elements all support **fixed** letter size and spacing.

HTML `Keyboard` Formatting

The HTML **<kbd>** element defines **keyboard input**:

Example

```
<!DOCTYPE html>
<html>
<body style="font-size:16px">

<p>The kbd element represents keyboard input:</p>

<p><kbd>File | Open...</kbd></p>

<p>Hold down <kbd>CTRL</kbd>, <kbd>ALT</kbd>, and <kbd>DELETE</kbd>,
   then select Task Manager</p>

<p>To get a list of directories, type <kbd>dir</kbd> in the command line and press
Enter.</p>

<p>To undo your last action, press <kbd><kbd>Ctrl</kbd> + <kbd>Z</kbd></kbd>.</p>

</body>
</html>
```

HTML `Sample` Formatting

The HTML **<samp>** element defines a **computer output sample**:

Example

```
<!DOCTYPE html>
<html>
<body style="font-size:16px">

<p>The samp element represents a computer output sample:</p>

<p>It wasn't the most helpful of error messages as it simply said: <samp>An error has occurred</samp>.</p>

<p>A message will appear on the screen informing you what to do next: <samp>Press <kbd>Enter</kbd> to continue</samp>.</p>

</body>
</html>
```

Both the `kbd` and `samp` elements can also be used in conjunction with the `pre` element.

HTML Code Formatting

The HTML **<code>** element defines **programming code sample**:

Example

```
<!DOCTYPE html>
<html>
<body style="font-size:16px">

<p>Programming code examples:</p>

<code>
int a[] = {10, 20, 30, 40, 50};
</code>
<br><br>

<code>
var person = {firstName:"John", lastName:"Doe", age:50}
</code>

</body>
</html>
```

- The <code> element does **not** preserve extra **whitespace** and **line-breaks**:

Example

```
<!DOCTYPE html>
```

```
<html>
<body style="font-size:16px">

<p>The code element does not preserve whitespace and line-breaks:</p>

<code>
var person = {
   firstName:"John",
   lastName:"Doe",
   age:50,
   eyeColor:"blue"
}
</code>

</body>
</html>
```

- To fix this, you must wrap the code in a <pre> element:

Example

```
<!DOCTYPE html>
<html>
<body style="font-size:16px">

<p>The code element does not preserve whitespace and line-breaks.</p>

<p>To fix this, you must wrap the code in a pre element:</p>

<code>
<pre>
var person = {
   firstName:"John",
   lastName:"Doe",
   age:50,
   eyeColor:"blue"
}
</pre>
</code>

</body>
</html>
```

HTML *Variable* Formatting

The HTML **<var>** element defines a **mathematical variable**:

Example

```
<!DOCTYPE html>
<html>
<body style="font-size:16px">

<p>Einstein wrote:</p>

<p><var>E</var> = <var>m</var> <var>c</var><sup>2</sup></p>

</body>
</html>
```

HTML Computer Code Elements

Tag	Description
<code>	Defines computer code text
<kbd>	Defines keyboard text
<samp>	Defines sample computer code
<var>	Defines a variable
<pre>	Defines preformatted text

HTML Comments

Comment tags <!-- and --> are used to insert comments in HTML.

HTML Comment Tags

Comments can be added to HTML source by using the following syntax:

<!-- Write your comments here -->

Comments are not displayed by the browser, but they can help document your HTML.

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

Example

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

<!-- This is a comment -->
<p>This is a paragraph.</p>
<!-- Comments are not displayed in the browser -->

</body>
</html>
```

Comments are also great for debugging HTML, because you can comment out HTML lines of code, one at a time, to search for errors:

Example

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

<!-- Do not display this at the moment
<img border="0" src="pic_mountain.jpg" alt="Mountain">
-->

</body>
</html>
```

Conditional Comments

You might stumble upon conditional comments in HTML:

```
<!--[if IE 8]>
   .... some HTML here ....
<![endif]-->
```

Conditional comments defines HTML tags to be executed by Internet Explorer only.

HTML Fonts

Fonts play 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 **** 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 tag. The text that follows will remain changed until you close with the tag. You can change one or all of the font attributes within one <font> tag.

Note: *The font and basefont tags are deprecated and it is supposed to be removed in a future version of HTML. So they should not be used rather, it's suggested to use CSS styles to manipulate your fonts. But still for learning purpose, this chapter will explain font and basefont tags in detail.*

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>
```

The <basefont> Element:

The <basefont> element is supposed to set a default font size, color, and typeface for any parts of the document that are not otherwise contained within a <font> tag. You can use the <font> elements to override the <basefont> settings.

The <basefont> tag also takes color, size and face attributes and it will support relative font setting by giving size a value of +1 for a size larger or -2 for two sizes smaller.

Example

```
<!DOCTYPE html>
<html>
<head>
<title>Setting Basefont Color</title>
</head>
<body>
<basefont face="arial, verdana, sans-serif" size="2" color="#ff0000">
<p>This is the page's default font.</p>

<h2>Example of the &lt;basefont&gt; Element</h2>
<p><font size="+2" color="darkgray">
This is darkgray text with two sizes larger
</font></p>

<p><font face="courier" size="-1" color="#000000">
It is a courier font, a size smaller and black in color.
</font></p>

</body>
</html>
```

CSS = Styles and Colors

Manipulate Text
Colors, Boxes

HTML Styles - CSS

Example

```
<!DOCTYPE html>
<html>

<head>
<style>
  body {background-color:lightgray}
  h1   {color:blue}
  p    {color:green}
</style>
</head>

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

</html>
```

Styling HTML with CSS

CSS stands for **C**ascading **S**tyle **S**heets

Styling can be added to HTML elements in 3 ways:

- Inline - using a **style attribute** in HTML elements
- Internal - using a **<style> element** in the HTML <head> section
- External - using one or more **external CSS files**

The most common way to add styling, is to keep CSS syntax in separate CSS files. But, in this tutorial, internal styling is used in order to demonstrate easily.

CSS Syntax

CSS styling has the following **syntax**:

element { *property*:*value ; property:value }*

The **element** is an HTML element name. The ***property*** is a CSS property. The ***value*** is a CSS value.

Multiple styles are separated with semicolon.

Inline Styling (Inline CSS)

- **Inline styling** is useful for applying a unique style to a single HTML element:
- Inline styling uses the **style attribute**.
- This inline styling changes the text color of a single heading:

Example

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

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

</body>
</html>
```

Internal Styling (Internal CSS)

An internal style sheet can be used to define a common style for all HTML elements on a page.

Internal styling is defined in the **<head>** section of an HTML page, using a **<style>** element:

Example

```
<!DOCTYPE html>
<html>

<head>
<style>
  body {background-color:lightgrey}
  h1   {color:blue}
  p    {color:green}
</style>
</head>

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

```
</html>
```

External Styling (External CSS)

External style sheet are ideal when the style is applied to many pages.

With external style sheets, you can change the look of an entire site by changing one file.

External styles are defined in the **<head>** section of an HTML page, in the **<link>** element:

Example

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

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

</html>
```

CSS Fonts

The CSS property **color** defines the text color to be used for an HTML element.

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

The CSS property **font-size** defines the text size to be used for an HTML element.

Example

```
<!DOCTYPE html>
<html>

<head>
<style>
h1 {
    color:blue;
    font-family:verdana;
    font-size:300%;
```

```
}
p  {
   color:red;
   font-family:courier;
   font-size:160%;
}
</style>
</head>

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

</html>
```

The CSS Box Model

- Every visible HTML element has a box around it, even if you cannot see it.
- The CSS **border** property defines a visible border around an HTML element

Example

```
<!DOCTYPE html>
<html>

<head>
<style>
p {
   border:1px solid blue;
}
</style>
</head>
<body>

<h1>This is a heading</h1>

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

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

<p>This is a paragraph</p>

</body>
</html>
```

The CSS **padding** property defines a padding (space) inside the border:

Example

```
p {
   border:1px solid black;
   padding:10px;
}
```

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

Example

```
p {
   border:1px solid black;
   padding:10px;
   margin:30px;
}
```

The CSS examples above use px to define sizes in pixels (screen pixels).

The id Attribute

- All the examples above use CSS to style HTML elements in a general way.
- The CSS styles define an equal style for all equal elements.

To define a special style for a special element, first add an id attribute to the element:

Example

```
<!DOCTYPE html>
<html>

<head>
<style>
p#p01 {
   color: blue;
}
</style>
</head>
<body>

<p>This is a paragraph.</p>
<p>This is a paragraph.</p>
<p>This is a paragraph.</p>
<p id="p01">I am different.</p>

</body>
</html>
```

The class Attribute

- To define a style for a special type (class) of elements, add a class attribute to the element:

Example

```
<!DOCTYPE html>
<html>

<head>
<style>
p.error {
   color:red;
}
</style>
</head>
<body>

<p>This is a paragraph.</p>
<p>This is a paragraph.</p>
<p class="error">I am different.</p>
<p>This is a paragraph.</p>
<p class="error">I am different too.</p>

</body>
</html>
```

 Use **id** to address **single** elements. Use **class** to address **groups** of elements.

Chapter Summary

- Use the HTML **style** attribute for inline styling
- Use the HTML **<style>** element to define internal CSS
- Use the HTML **<link>** element to define external CSS
- Use the HTML **<head>** element to store <style> and <link> elements
- Use the CSS **color** property for text colors
- Use the CSS **font-family** property for text fonts
- Use the CSS **font-size** property for text sizes
- Use the CSS **border** property for visible element borders
- Use the CSS **padding** property for space inside the border
- Use the CSS **margin** property for space outside the border

---

HTML Style Tags

Tag	Description
<style>	Defines style information for a document

<link> Defines a link between a document and an external resource

HTML Links

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

HTML Links - Hyperlinks

HTML links are hyperlinks. A hyperlink is an element, a text, or an image that you can click on, and jump to another document.

HTML Links - Syntax

In HTML, links are defined with the **<a>** tag:

Link Syntax:

link text

Example:

```
<!DOCTYPE html>
<html>
<body>
<p>
<a href="http://www.vit.ac.in/">Visit our University website</a>
</p>
</body>
</html>
```

- The **href** attribute specifies the destination address
- The **link text** is the visible part

Clicking on the link text, will send you to the specified address.

> The link text does not have to be text. It can be an HTML image or any other HTML element.

Local Links

The example above used an absolute URL (A full web address).

A local link (link to the same web site) is specified with a relative URL (without http://www....).

Example:

```
<a href="html_images.asp">HTML Images</a>
```

HTML Links - The target Attribute

The **target** attribute specifies where to open the linked document.

This example will open the linked document in a new browser window or in a new tab:

Example

```
<a href="http://www.vit.ac.in/" target="_blank">Visit VIT University website! </a>
```

Target Value	Description
_blank	Opens the linked document in a new window or tab
_self	Opens the linked document in the same frame as it was clicked (this is default)
_parent	Opens the linked document in the parent frame
_top	Opens the linked document in the full body of the window
framename	Opens the linked document in a named frame

If your webpage is locked in a frame, you can use target="_top" to break out of the frame:

Example

```
<a href="http://www.w3schools.com/html/" target="_top">HTML5 tutorial!</a>
```

HTML Links - Image as Link

It is common to use images as links:

Example

```
<a href="default.asp">
  <img src="smiley.gif" alt="HTML tutorial" style="width:42px;height:42px;border:0">
</a>
```

border:0 is added to prevent IE9 (and earlier) from displaying a border around the image.

HTML Links - The id Attribute

The **id** attribute can be used to create bookmarks inside HTML documents.

Bookmarks are not displayed in any special way. They are invisible to the reader.

Example

Add an id attribute to any <a> element:

Useful Tips Section

Then create a link to the <a> element (Useful Tips Section):

Visit the Useful Tips Section

Or, create a link to the <a> element (Useful Tips Section) from another page:

Visit the Useful Tips Section

Without a trailing slash on subfolder addresses, you might generate two requests to the server.
Many servers will automatically add a slash to the address, and then create a new request.

Chapter Summary

- Use the HTML **<a>** element to define a link
- Use the HTML **href** attribute to define the link address
- Use the HTML **target** attribute to define where to open the linked document
- Use the HTML **<img>** element (inside <a>) to use an image as a link
- Use the HTML **id** attribute (id=*value*) to define bookmarks in a page
- Use the HTML **href** attribute (href="*#value"*) to address the bookmark

---

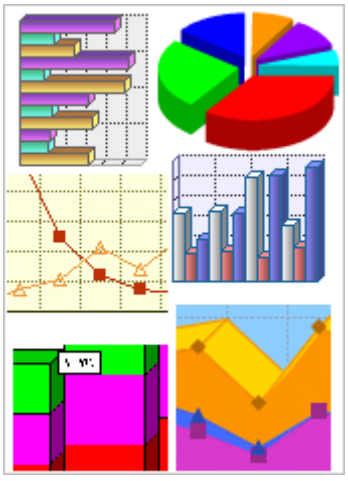
HTML Images

Example

GIF Images

JPG Images

PNG Images



```
<!DOCTYPE html>
<html>

<body>
 <h2>Spectacular Mountains</h2>
 <img src="pic_mountain.jpg" alt="Mountain View" style="width:304px;height:228px">
</body>

</html>
```

Always specify image size. If the size is unknown, the page will flicker while the image loads.

HTML Images Syntax

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

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

The **src** attribute defines the url (web address) of the image:

<img src="*url*" alt="*some_text*">

The alt Attribute

The **alt** attribute specifies an alternate text for the image, if it cannot be displayed.

The value of the alt attribute should describe the image in words:

Example

```
<img src="html5.gif" alt="The official HTML5 Icon">
```

The alt attribute is **required**. A web page will not validate correctly without it.

HTML Screen Readers

- Screen readers are software programs that can read what is displayed on a screen.
- Used on the web, screen readers can "reproduce" HTML as text-to-speech, sound icons, or braille output.
- Screen readers are used by people who are blind, visually impaired, or learning disabled.

Screen readers can read the **alt** attribute.

Image Size - Width and Height

- You can use the **style** attribute to specify the **width** and **height** of an image.
- The values are specified in pixels (use px after the value):

Example

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

<img src="VIT.gif" alt="VIT Icon" style="width:128px;height:128px">

</body>
</html>
```

- Alternatively, you can use width and height **attributes**.
- The values are specified in pixels (without px after the value):

Example

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

<img src="VIT.gif" alt="VIT Icon" width="128" height="128">

</body>
</html>
```

Width and Height or Style?

- Both the width, the height, and the style attributes, are valid in the latest HTML5 standard.
- We suggest you use the style attribute. It prevents styles sheets from changing the default size of images:

Example

```
<!DOCTYPE html>
<html>
<head>
<style>
  img { width:100%; }
</style>
</head>
```

```
<body>

<img src="VIT.gif" alt="VIT Icon" style="width:128px;height:128px">
<img src="VIT.gif" alt="VIT Icon" width="128" height="128">

</body>

</html>
```

Images in Another Folder

If not specified, the browser expects to find the image in the same folder as the web page. However, it is common on the web, to store images in a sub-folder, and refer to the folder in the image name:

Example

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

<img src="/images/VIT.gif" alt="VIT Icon" style="width:128px;height:128px">

</body>
</html>
```

- ***If a browser cannot find an image, it will display a broken link icon:***

Example

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

<p>
If a browser cannot find an image, it will display a broken link icon.
</p>

<img src="wrongname.gif" alt="HTML5 Icon" style="width:128px;height:128px">

</body>
</html>
```

Images on Another Server

Some web sites store their images on image servers. Actually, you can access images from any web address in the world:

Example

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

<img src="http://www.vit.ac.in/images/VIT_green.jpg" alt="VIT University"
style="width:104px;height:142px">

</body>
</html>
```

Animated Images

- The GIF standard allows animated images:

Example

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

<p>
The GIF standard allows moving images.
</p>

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

</body>
</html>
```

Note: that the syntax of inserting animated images is no different from non-animated images.

Using an Image as a Link

- It is common to use images as links:

Example

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

```
<p>The image is a link. You can click on it.</p>

<a href="default.asp">
<img src="smiley.gif" alt="HTML tutorial" style="width:42px;height:42px;border:0">
</a>

<p>
We have added "border:0" to prevent IE9 (and earlier) from displaying a border around the image.
</p>

</body>
</html>
```

 We have added border:0 to prevent IE9 (and earlier) from displaying a border around the image.

Image Floating

- You can let an image float to the left or right of a paragraph:

Example

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

<p>
  <img src="smiley.gif" alt="Smiley face"
  style="float:left;width:42px;height:42px">
  A paragraph with an image. A paragraph with an image.
  A paragraph with an image. A paragraph with an image.
  A paragraph with an image. A paragraph with an image.
</p>

<p>The image floats to the left of the text.</p>

<p>Please use the CSS float property. The align attribute is deprecated in HTML 4, and not supported in HTML5.</p>

</body>
</html>
```

Chapter Summary

- Use the HTML **<img>** element to define images
- Use the HTML **src** attribute to define the image file name
- Use the HTML **alt** attribute to define an alternative text
- Use the HTML **width** and **height** attributes to define the image size
- Use the CSS **width** and **height** properties to define the image size (alternatively)
- Use the CSS **float** property to define image floating
- Use the HTML **usemap** attribute to point to an image map
- Use the HTML **<map>** element to define an image map
- Use the HTML **<area>** element to define image map areas

 Loading images takes time. Large images can slow down your page. Use images carefully.

HTML Image Tags

Tag	Description
<img>	Defines an image
<map>	Defines an image-map
<area>	Defines a clickable area inside an image-map

HTML Tables

HTML Table Example

Number	First Name	Last Name	Points
1	Eve	Jackson	94
2	John	Doe	80
3	Adam	Johnson	67
4	Jill	Smith	50

Defining HTML Tables

Example

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

<table style="width:100%">
  <tr>
    <td>Jill</td>
    <td>Smith</td>
    <td>50</td>
  </tr>
  <tr>
    <td>Eve</td>
    <td>Jackson</td>
    <td>94</td>
  </tr>
  <tr>
    <td>John</td>
    <td>Doe</td>
    <td>80</td>
  </tr>
</table>

</body>
</html>
```

Example explained:

- Tables are defined with the **<table>** tag.
- Tables are divided into **table rows** with the **<tr>** tag.
- Table rows are divided into **table data** with the **<td>** tag.

- A table row can also be divided into **table headings** with the **<th>** tag.


Table data <td> are the data containers of the table.
They can contain all sorts of HTML elements like text, images, lists, other tables, etc.

An HTML Table with a Border Attribute

If you do not specify a border for the table, it will be displayed without borders. A border can be added using the border attribute:

Example

```
<table border="1" style="width:100%">
  <tr>
    <td>Jill</td>
    <td>Smith</td>
    <td>50</td>
  </tr>
  <tr>
    <td>Eve</td>
    <td>Jackson</td>
    <td>94</td>
  </tr>
</table>
```


The border attribute is on its way out of the HTML standard! It is better to use CSS.

To add borders, use the **CSS border** property:

Example

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

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

An HTML Table with Collapsed Borders

If you want the borders to collapse into one border, add **CSS border-collapse**:

Example

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

```
  border-collapse: collapse;
}
```

An HTML Table with Cell Padding

- Cell padding specifies the space between the cell content and its borders.
- If you do not specify a padding, the table cells will be displayed without padding.
- To set the padding, use the **CSS padding** property:

Example

```
table, th, td {
  border: 1px solid black;
  border-collapse: collapse;
}
th,td {
  padding: 15px;
}
```

HTML Table Headings

- Table headings are defined with the **<th>** tag.
- By default, all major browsers display table headings as bold and centered:

Example

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

To left-align the table headings, use the **CSS text-align** property:

Example

```
th {
  text-align: left;
}
```

An HTML Table with Border Spacing

Border spacing specifies the space between the cells.

To set the border spacing for a table, use the **CSS border-spacing** property:

Example

```
table {
   border-spacing: 5px;
}
```

 If the table has collapsed borders, border-spacing has no effect.

Table Cells that Span Many Columns

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

Example

```
<table style="width:100%">
  <tr>
    <th>Name</th>
    <th colspan="2">Telephone</th>
  </tr>
  <tr>
    <td>Bill Gates</td>
    <td>555 77 854</td>
    <td>555 77 855</td>
  </tr>
</table>
```

Table Cells that Span Many Rows

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

Example

```
<table style="width:100%">
  <tr>
    <th>Name:</th>
    <td>Bill Gates</td>
  </tr>
  <tr>
    <th rowspan="2">Telephone:</th>
    <td>555 77 854</td>
  </tr>
```

```
  <tr>
    <td>555 77 855</td>
  </tr>
</table>
```

An HTML Table With a Caption

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

Example

```
<table style="width:100%">
  <caption>Monthly savings</caption>
  <tr>
    <th>Month</th>
    <th>Savings</th>
  </tr>
  <tr>
    <td>January</td>
    <td>$100</td>
  </tr>
  <tr>
    <td>February</td>
    <td>$50</td>
  </tr>
</table>
```


The <caption> tag must be inserted immediately after the <table> tag.

Different Styles for Different Tables

- Most of the examples above use a style attribute (width="100%") to define the width of each table.
- This makes it easy to define different widths for different tables.
- The styles in the <head> section, however, define a style for all tables in a page.
- To define a special style for a special table, add an **id attribute** to the table:

Example

```
<table id="t01">
  <tr>
    <th>Firstname</th>
    <th>Lastname</th>
    <th>Points</th>
  </tr>
  <tr>
    <td>Eve</td>
    <td>Jackson</td>
```

```
    <td>94</td>
  </tr>
</table>
```

Now you can define a different style for this table:

```
table#t01 {
    width: 100%;
    background-color: #f1f1c1;
}
```

And add more styles:

```
table#t01 tr:nth-child(even) {
    background-color: #eee;
}
table#t01 tr:nth-child(odd) {
    background-color: #fff;
}
table#t01 th {
    color: white;
    background-color: black;
}
```

Chapter Summary

- Use the HTML **<table>** element to define a table
- Use the HTML **<tr>** element to define a table row
- Use the HTML **<td>** element to define a table data
- Use the HTML **<th>** element to define a table heading
- Use the HTML **<caption>** element to define a table caption
- Use the CSS **border** property to define a border
- Use the CSS **border-collapse** property to collapse cell borders
- Use the CSS **padding** property to add padding to cells
- Use the CSS **text-align** property to align cell text
- Use the CSS **border-spacing** property to set the spacing between cells
- Use the **colspan** attribute to make a cell span many columns
- Use the **rowspan** attribute to make a cell span many rows
- Use the **id** attribute to uniquely define one table

HTML Table Tags

Tag	Description
<table>	Defines a table
<th>	Defines a header cell in a table
<tr>	Defines a row in a table
<td>	Defines a cell in a table
<caption>	Defines a table caption
<colgroup>	Specifies a group of one or more columns in a table for formatting
<col>	Specifies column properties for each column within a <colgroup> element
<thead>	Groups the header content in a table
<tbody>	Groups the body content in a table
<tfoot>	Groups the footer content in a table

HTML Lists

HTML can have Unordered Lists, Ordered Lists, or Description Lists:

Unordered HTML List

- The first item
- The second item
- The third item
- The fourth item

Ordered HTML List

1. The first item
2. The second item
3. The third item
4. The fourth item

HTML Description List

The first item
Description of item
The second item
Description of item

Unordered HTML Lists

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

Unordered List:

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

Unordered HTML Lists - The Style Attribute

- A **style** attribute can be added to an **unordered list**, to define the style of the marker:

Style	Description
list-style-type:disc	The list items will be marked with bullets (default)
list-style-type:circle	The list items will be marked with circles
list-style-type:square	The list items will be marked with squares
list-style-type:none	The list items will not be marked

Disc:

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

Circle:

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

Square:

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

None:

```
<ul style="list-style-type:none">
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ul>
```

Using a type attribute <ul type="disc">, instead of <ul style="list-style-type:disc">, also works.
But in HTML5, the type attribute is not valid in unordered lists, only in ordered list.

Ordered HTML Lists

- An ordered list starts with the **<ol>** tag. Each list item starts with the **<li>** tag.
- The list items will be marked with numbers.

Ordered List:

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

Ordered HTML Lists - The Type Attribute

A **type** attribute can be added to an **ordered list**, to define the type of the marker:

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

type="I" The list items will be numbered with uppercase roman numbers
type="i" The list items will be numbered with lowercase roman numbers

Numbers:

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

Upper Case:

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

Lower Case:

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

Roman Upper Case:

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

Roman Lower Case:

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

HTML Description Lists

- A description list, is a list of terms, with a description of each term.
- The **<dl>** tag defines a description list.
- The **<dt>** tag defines the term (name), and the **<dd>** tag defines the data (description).

Description List:

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

Nested HTML Lists

- List can be nested (lists inside lists).

Nested Lists:

```
<ul>
  <li>Coffee</li>
  <li>Tea
    <ul>
      <li>Black tea</li>
      <li>Green tea</li>
    </ul>
  </li>
  <li>Milk</li>
</ul>
```


List items can contain new list, and other HTML elements, like images and links, etc.

Horizontal Lists

- HTML lists can be styled in many different ways with CSS.
- One popular way, is to style a list to display horizontally:

Horizontal List:

```
<!DOCTYPE html>
<html>

<head>
<style>
ul#menu li {
   display:inline;
}
</style>
</head>

<body>
```

```
<h2>Horizontal List</h2>

<ul id="menu">
  <li>Apples</li>
  <li>Bananas</li>
  <li>Lemons</li>
  <li>Oranges</li>
</ul>

</body>
</html>
```

With a little extra style, you can make it look like a menu:

Tables Lists Blocks Classes

New Style:

```
<!DOCTYPE html>
<html>

<head>
<style>
ul#menu {
    padding: 0;
}

ul#menu li {
    display: inline;
}

ul#menu li a {
    background-color: black;
    color: white;
    padding: 10px 20px;
    text-decoration: none;
    border-radius: 4px 4px 0 0;
}

ul#menu li a:hover {
    background-color: orange;
}
</style>
</head>

<body>

<h2>Horizontal List</h2>

<ul id="menu">
```

```
 <li><a href="html_tables.asp">Tables</a></li>
 <li><a href="html_lists.asp">Lists</a></li>
 <li><a href="html_blocks.asp">Blocks</a></li>
 <li><a href="html_classes.asp">Classes</a></li>
</ul>

</body>
</html>
```

Chapter Summary

- Use the HTML **<ul>** element to define an unordered list
- Use the HTML **style** attribute to define the bullet style
- Use the HTML **<ol>** element to define an ordered list
- Use the HTML **type** attribute to define the numbering type
- Use the HTML **<li>** element to define a list item
- Use the HTML **<dl>** element to define a description list
- Use the HTML **<dt>** element to define the description term
- Use the HTML **<dd>** element to define the description data
- Lists can be nested inside lists
- List items can contain other HTML elements
- Use the CSS property **display:inline** to display a list horizontally

---

HTML List Tags

Tag	Description
	Defines an unordered list
	Defines an ordered list
	Defines a list item
<dl>	Defines a description list
<dt>	Defines the term in a description list
<dd>	Defines the description in a description list

HTML Frames

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

Disadvantages of Frames

There are few drawbacks with using frames, so it's never recommended to use frames in your webpages:

- Some smaller devices cannot cope with frames often because their screen is not big enough to be divided up.
- Sometimes your page will be displayed differently on different computers due to different screen resolution.
- The browser's *back button* might not work as the user hopes.
- There are still few browsers that do not support frame technology.

Creating Frames

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

Example

Following is the example to create three horizontal frames:

```
<!DOCTYPE html>
<html>
<head>
<title>HTML Frames</title>
</head>
<frameset rows="10%,80%,10%">
   <frame name="top" src="/html/top_frame.htm" />
   <frame name="main" src="/html/main_frame.htm" />
   <frame name="bottom" src="/html/bottom_frame.htm" />
   <noframes>
   <body>
      Your browser does not support frames.
   </body>
   </noframes>
</frameset>
</html>
```

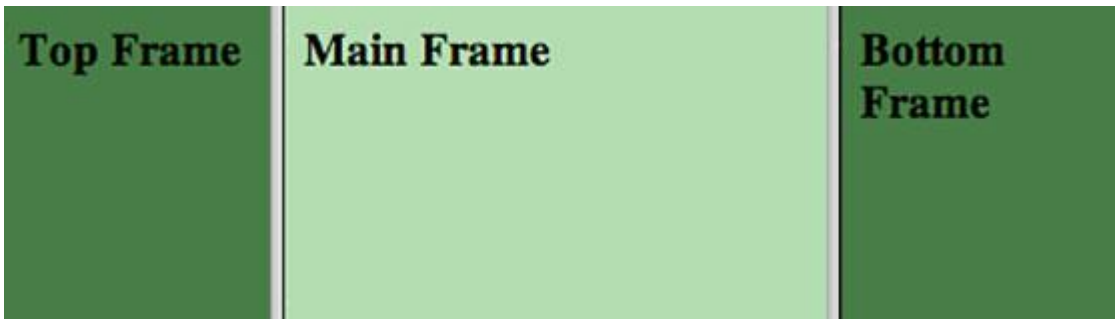
This will produce following result:

Example

Let's put above example as follows, here we replaced rows attribute by cols and changed their width. This will create all the three frames vertically:

```
<!DOCTYPE html>
<html>
<head>
<title>HTML Frames</title>
</head>
<frameset cols="25%,50%,25%">
   <frame name="left" src="/html/top_frame.htm" />
   <frame name="center" src="/html/main_frame.htm" />
   <frame name="right" src="/html/bottom_frame.htm" />
   <noframes>
   <body>
      Your browser does not support frames.
   </body>
   </noframes>
</frameset>
</html>
```

This will produce following result:



The <frameset> Tag Attributes

Following are important attributes of the <frameset> tag:

Attribute	Description
cols	specifies how many columns are contained in the frameset and the size of each

	column. You can specify the width of each column in one of four ways: • Absolute values in pixels. For example to create three vertical frames, use *cols="100, 500,100"*. • A percentage of the browser window. For example to create three vertical frames, use *cols="10%, 80%,10%"*. • Using a wildcard symbol. For example to create three vertical frames, use *cols="10%, *,10%"*. In this case wildcard takes remainder of the window. • As relative widths of the browser window. For example to create three vertical frames, use *cols="3*,2*,1*"*. This is an alternative to percentages. You can use relative widths of the browser window. Here the window is divided into sixths: the first column takes up half of the window, the second takes one third, and the third takes one sixth.
rows	This attribute works just like the cols attribute and takes the same values, but it is used to specify the rows in the frameset. For example to create two horizontal frames, use *rows="10%, 90%"*. You can specify the height of each row in the same way as explained above for columns.
border	This attribute specifies the width of the border of each frame in pixels. For example border="5". A value of zero means no border.
frameborder	This attribute specifies whether a three-dimensional border should be displayed between frames. This attrubute takes value either 1 (yes) or 0 (no). For example frameborder="0" specifies no border.
framespacing	This attribute specifies the amount of space between frames in a frameset. This can take any integer value. For example framespacing="10" means there should be 10 pixels spacing between each frames.

The <frame> Tag Attributes

Following are important attributes of <frame> tag:

Attribute	Description
src	This attribute is used to give the file name that should be loaded in the frame. Its value can be any URL. For example, src="/html/top_frame.htm" will load an HTML file available in html directory.
name	This attribute allows you to give a name to a frame. It is used to indicate which frame a document should be loaded into. This is especially important when you want to create links in one frame that load pages into an another frame, in which case the second frame needs a name to identify itself as the target of the link.
frameborder	This attribute specifies whether or not the borders of that frame are shown; it overrides the value given in the frameborder attribute on the <frameset> tag if one is given, and this can take values either 1 (yes) or 0 (no).
marginwidth	This attribute allows you to specify the width of the space between the left and right of the frame's borders and the frame's content. The value is given in pixels. For example marginwidth="10".

marginheight	This attribute allows you to specify the height of the space between the top and bottom of the frame's borders and its contents. The value is given in pixels. For example marginheight="10".
noresize	By default you can resize any frame by clicking and dragging on the borders of a frame. The noresize attribute prevents a user from being able to resize the frame. For example noresize="noresize".
scrolling	This attribute controls the appearance of the scrollbars that appear on the frame. This takes values either "yes", "no" or "auto". For example scrolling="no" means it should not have scroll bars.
longdesc	This attribute allows you to provide a link to another page containing a long description of the contents of the frame. For example longdesc="framedescription.htm"

Browser Support for Frames

If a user is using any old browser or any browser which does not support frames then <noframes> element should be displayed to the user.

So you must place a <body> element inside the <noframes> element because the <frameset> element is supposed to replace the <body> element, but if a browser does not understand <frameset> element then it should understand what is inside the <body> element which is contained in a <noframes> element.

You can put some nice message for your user having old browsers. For example *Sorry!! your browser does not support frames.* as shown in the above example.

Frame's name and target attributes

One of the most popular uses of frames is to place navigation bars in one frame and then load main pages into a separate frame.

Let's see following example where a test.htm file has following code:

```
<!DOCTYPE html>
<html>
<head>
<title>HTML Target Frames</title>
</head>
<frameset cols="200, *">
   <frame src="/html/menu.htm" name="menu_page" />
   <frame src="/html/main.htm" name="main_page" />
   <noframes>
   <body>
      Your browser does not support frames.
   </body>
   </noframes>
</frameset>
</html>
```

Here we have created two columns to fill with two frames. The first frame is 200 pixels wide and will contain the navigation menubar implemented by **menu.htm** file. The second column

fills in remaining space and will contain the main part of the page and it is implemented by **main.htm** file. For all the three links available in menubar, we have mentioned target frame as **main_page**, so whenever you click any of the links in menubar, available link will open in main_page.

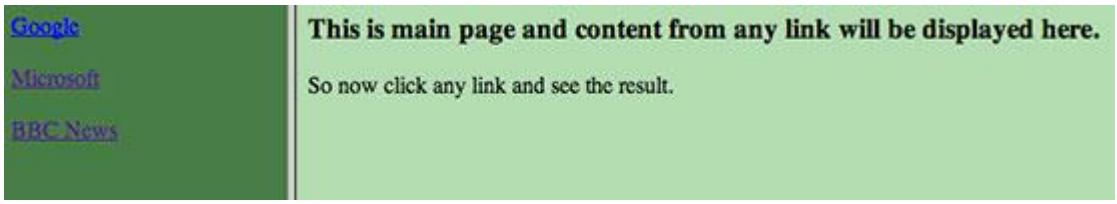
Following is the content of menu.htm file

```
<!DOCTYPE html>
<html>
<body bgcolor="#4a7d49">
<a href="http://www.google.com" target="main_page">Google</a>
<br /><br />
<a href="http://www.microsoft.com" target="main_page">Microsoft</a>
<br /><br />
<a href="http://news.bbc.co.uk" target="main_page">BBC News</a>
</body>
</html>
```

Following is the content of main.htm file:

```
<!DOCTYPE html>
<html>
<body bgcolor="#b5dcb3">
<h3>This is main page and content from any link will be displayed
here.</h3>
<p>So now click any link and see the result.</p>
</body>
</html>
```

When we load **test.htm** file, it produces following result:



Now you can try to click links available in the left panel and see the result. The *target* attribute can also take one of the following values:

Option	Description
_self	Loads the page into the current frame.
_blank	Loads a page into a new browser window.opening a new window.
_parent	Loads the page into the parent window, which in the case of a single frameset is the main browser window.
_top	Loads the page into the browser window, replacing any current frames.
targetframe	Loads the page into a named targetframe.

HTML Forms

The **<form>** element defines an HTML form:

Example

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

- HTML forms contain **form elements**.
- Form elements are different types of input elements, checkboxes, radio buttons, submit buttons, and more.

The <input> Element

- The **<input>** element is the most important **form element**.
- The <input> element has many variations, depending on the **type** attribute.

Here are the types used in this chapter:

Type	Description
text	Defines normal text input
radio	Defines radio button input (for selecting one of many choices)
submit	Defines a submit button (for submitting the form)

Text Input

<input type="text"> defines a one-line input field for **text input**:

Example

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

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

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

```
<p>Also note that the default width of a text field is 20 characters.</p>

</body>
</html>
```

This is how it will look like in a browser:

First name:

Last name:

Note: The form itself is not visible. Also note that the default width of a text field is 20 characters.

Radio Button Input

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

- Radio buttons let a user select ONE of a limited number of choices:

Example

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

<form>
<input type="radio" name="sex" value="male" checked>Male
<br>
<input type="radio" name="sex" value="female">Female
</form>

</body>
</html>
```

This is how the HTML code above will be displayed in a browser:

Male

Female

The Submit Button

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

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

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

Example

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

<form action="action_page.php">
First name:<br>
<input type="text" name="firstname" value="Mickey">
<br>
Last name:<br>
<input type="text" name="lastname" value="Mouse">
<br><br>
<input type="submit" value="Submit">
</form>

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

</body>
</html>
```

This is how the HTML code above will be displayed in a browser:

First name:

Mickey

Last name:

Mouse

The Action Attribute

The **action attribute** defines the action to be performed when the form is submitted. The common way to submit a form to a server, is by using a submit button. Normally, the form is submitted to a web page on a web server. In the example above, a server-side script is specified to handle the submitted form:

<form **action="action_page.php**">

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

The Method Attribute

The **method attribute** specifies the HTTP method (**GET** or **POST**) to be used when submitting the forms:

<form action="action_page.php" **method="GET"**>

or:

<form action="action_page.php" **method="POST"**>

When to Use GET?

You can use GET (the default method):

- If the form submission is
  - passive (like a search engine query), and
  - without sensitive information.
- When you use GET, the form data will be visible in the page address:
  Eg. *action_page.php?firstname=Mickey&lastname=Mouse*
- GET is best suited to short amounts of data. [Size limitations are set in your browser].

When to Use POST?

You should use POST:

- If the form is updating data, or
- Includes sensitive information (password).
- POST offers better security because the submitted data is not visible in the page address.

The Name Attribute

To be submitted correctly, each input field must have a name attribute.

This example will only submit the "Last name" input field:

Example

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

<form action="action_page.php">
First name:<br>
```

```
<input type="text" value="Mickey">
<br>
Last name:<br>
<input type="text" name="lastname" value="Mouse">
<br><br>
<input type="submit" value="Submit">
</form>

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

<p>The first name will not be submitted, because the input element does not have a name attribute.</p>

</body>
</html>
```

RESULT:

Your input was received as:

lastname=Mouse

The server has processed your input and returned this answer.

Use the back button in the browser to return to the example.

The HTML tutorial will not teach you how servers are processing input. Server input is explained in our PHP and ASP tutorials.

Grouping Form Data with <fieldset>

- The **<fieldset>** element groups related data in a form.
- The **<legend>** element defines a caption for the <fieldset> element.

Example

```
<form action="action_page.php">
<fieldset>
<legend>Personal information:</legend>
First name:<br>
<input type="text" name="firstname" value="Mickey">
<br>
Last name:<br>
<input type="text" name="lastname" value="Mouse">
<br><br>
```

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

This is how the HTML code above will be displayed in a browser:
Personal information:
First name:
Mickey
Last name:
Mouse

HTML Form Attributes

An HTML <form> element, with all possible attributes set, will look like this:

Example

```
<form action="action_page.php" method="GET" target="_blank" accept-charset="UTF-8"
enctype="application/x-www-form-urlencoded" autocomplete="off" novalidate>
.
form elements
.
</form>
```

Here is the list of <form> attributes:

Attribute	Description
accept-charset	Specifies the charset used in the submitted form (default: the page charset).
Action	Specifies an address (url) where to submit the form (default: the submitting page).
autocomplete	Specifies if the browser should autocomplete the form (default: on).
Enctype	Specifies the encoding of the submitted data (default: is url-encoded).
Method	Specifies the HTTP method used when submitting the form (default: GET).
Name	Specifies a name used to identify the form (for DOM usage: document.forms.name).
Novalidate	Specifies that the browser should not validate the form.
Target	Specifies the target of the address in the action attribute (default: _self).


You will learn more about attributes in the next chapters.

HTML Form Elements

The <input> Element

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

The <input> element can vary in many ways, depending on the **type** attribute.

 All HTML input types are covered in the next chapter.

The <select> Element (Drop-Down List)

The **<select>** element defines a **drop-down** list:

Example

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

<form action="action_page.php">
<select name="cars">
<option value="volvo">Volvo</option>
<option value="saab">Saab</option>
<option value="fiat">Fiat</option>
<option value="audi">Audi</option>
</select>
<br><br>
<input type="submit">
</form>

</body>
</html>
```

- The **<option>** elements defines the options to select.
- The list will normally show the first item as selected.
- You can add a selected attribute to define a predefined option.

Example

```
<option value="fiat" selected>Fiat</option>
```

The <textarea> Element

The <**textarea**> element defines a multi-line input field (**a text area**):

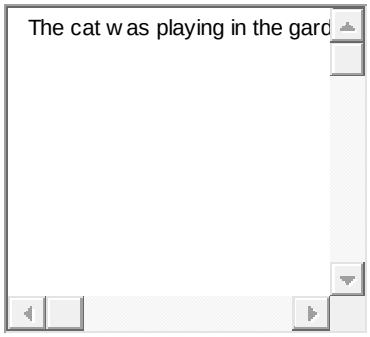
Example

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

<form action="action_page.php">
<textarea name="message" rows="10" cols="30">
The cat was playing in the garden.
</textarea>
<br><br>
<input type="submit">
</form>

</body>
</html>
```

This is how the HTML code above will be displayed in a browser:



The <button> Element

The <**button**> element defines a a clickable **button**:

Example

```
<!DOCTYPE html>
<html>
<body>
<button type="button" onclick="alert('Hello World!')">Click Me!</button>
</body>
</html>
```

HTML Form Elements

Tag	Description
<form>	Defines an HTML form for user input
<input>	Defines an input control
<textarea>	Defines a multiline input control (text area)
<label>	Defines a label for an <input> element
<fieldset>	Groups related elements in a form
<legend>	Defines a caption for a <fieldset> element
<select>	Defines a drop-down list
<button>	Defines a clickable button

HTML Input Types

Input Type: text

<input type="text"> defines a one-line input field for **text input**:

Example

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

This is how the HTML code above will be displayed in a browser:
First name:

Last Name:

Input Type: password

<input type="password"> defines a **password field**:

Example

```
<form>
User name:<br>
<input type="text" name="username">
<br>
User password:<br>
<input type="password" name="psw">
</form>
```

This is how the HTML code above will be displayed in a browser:
First name:

Password:


The characters in a password field are masked (shown as asterisks or circles).

Input Type: submit

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

- The form-handler is typically a server page with a script for processing input data.
- The form-handler is specified in the form's action attribute:

Example

```
<form action="action_page.php">
First name:<br>
<input type="text" name="firstname" value="Mickey">
<br>
Last name:<br>
<input type="text" name="lastname" value="Mouse">
<br><br>
<input type="submit" value="Submit">
</form>
```

This is how the HTML code above will be displayed in a browser:

First name:

Mickey

Last name:

Mouse

If you omit the submit button's value attribute, the button will get a default text:

Example

```
<form action="action_page.php">
First name:<br>
<input type="text" name="firstname" value="Mickey">
<br>
Last name:<br>
<input type="text" name="lastname" value="Mouse">
<br><br>
<input type="submit">
</form>
```

Input Type: radio

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

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

Example

```
<form>
<input type="radio" name="sex" value="male" checked>Male
<br>
<input type="radio" name="sex" value="female">Female
</form>
```

This is how the HTML code above will be displayed in a browser:

Male

Female

Input Type: checkbox

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

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

Example

```
<form>
<input type="checkbox" name="vehicle" value="Bike">I have a bike
<br>
<input type="checkbox" name="vehicle" value="Car">I have a car
</form>
```

This is how the HTML code above will be displayed in a browser:

I have a bike

I have a car

Input Type: button

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

Example

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

This is how the HTML code above will be displayed in a browser:

HTML5 Input Types

HTML5 added several new input types:

- color
- date
- datetime
- datetime-local
- email
- month
- number
- range
- search
- tel
- time
- url
- week

 Input types, not supported by old web browsers, will behave as input type text.

Input Type: number

The **<input type="number">** is used for input fields that should contain a numeric value.

You can set restrictions on the numbers.

Depending on browser support, the restrictions can apply to the input field.

Example

```
<form>
  Quantity (between 1 and 5):
  <input type="number" name="quantity" min="1" max="5">
</form>
```

Input Restrictions

Here is a list of some common input restrictions (some are new in HTML5):

Attribute	Description
disabled	Specifies that an input field should be disabled
max	Specifies the maximum value for an input field
maxlength	Specifies the maximum number of character for an input field
min	Specifies the minimum value for an input field
pattern	Specifies a regular expression to check the input value against
readonly	Specifies that an input field is read only (cannot be changed)
required	Specifies that an input field is required (must be filled out)

size	Specifies the width (in characters) of an input field
step	Specifies the legal number intervals for an input field
value	Specifies the default value for an input field

Example

```
<form>
  Quantity:
  <input type="number" name="points" min="0" max="100" step="10" value="30">
</form>
```

HTML Input Attributes

The value Attribute

The **value** attribute specifies the initial value for an input field:

Example

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

The readonly Attribute

The **readonly** attribute specifies that the input field is read only (cannot be changed):

Example

```
<form action="">
First name:<br>
<input type="text" name="firstname" value="John" readonly>
<br>
Last name:<br>
<input type="text" name="lastname">
</form>
```

Try it Yourself »

The readonly attribute does not need a value. It is the same as writing readonly="readonly".

The disabled Attribute

The **disabled** attribute specifies that the input field is disabled.

A disabled element is un-usable and un-clickable.

Disabled elements will not be submitted.

Example

```
<form action="">
First name:<br>
<input type="text" name="firstname" value="John" disabled>
<br>
Last name:<br>
<input type="text" name="lastname">
</form>
```

The disabled attribute does not need a value. It is the same as writing disabled="disabled".

The size Attribute

The **size** attribute specifies the size (in characters) for the input field:

Example

```
<form action="">
First name:<br>
<input type="text" name="firstname" value="John" size="40">
<br>
Last name:<br>
<input type="text" name="lastname">
</form>
```

The maxlength Attribute

The **maxlength** attribute specifies the maximum allowed length for the input field:

Example

```
<form action="">
First name:<br>
<input type="text" name="firstname" maxlength="10">
<br>
Last name:<br>
<input type="text" name="lastname">
</form>
```

With a maxlength attribute, the input control will not accept more than the allowed number of characters.

The attribute does not provide any feedback. If you want to alert the user, you must write JavaScript code.

Input restrictions are not foolproof. JavaScript provides many ways to add illegal input. To safely restrict input, restrictions must be checked by the receiver (the server) as well.

HTML5 Attributes

HTML5 added the following attributes for <input>:

- autocomplete
- autofocus
- form
- formaction
- formenctype
- formmethod
- formnovalidate
- formtarget
- height and width
- list
- min and max
- multiple
- pattern (regexp)
- placeholder
- required
- step

and the following attributes for <form>:

- autocomplete
- novalidate

The novalidate Attribute

The novalidate attribute is a <form> attribute

When present, novalidate specifies that form data should not be validated when submitted.

Example

Indicates that the form is not to be validated on submit:

```
<form action="action_page.php" novalidate>
  E-mail: <input type="email" name="user_email">
  <input type="submit">
</form>
```

The height and width Attributes

The height and width attributes specify the height and width of an <input> element.

The height and width attributes are only used with <input type="image">.


Always specify the size of images. If the browser does not know the size, the page will flicker while images load.

Example

Define an image as the submit button, with height and width attributes:

```
<input type="image" src="img_submit.gif" alt="Submit" width="48" height="48">
```

The min and max Attributes

- The min and max attributes specify the minimum and maximum value for an <input> element.
- The min and max attributes work with the following input types: number, range, date, datetime, datetime-local, month, time and week.

Example

<input> elements with min and max values:

```
Enter a date before 1980-01-01:
<input type="date" name="bday" max="1979-12-31">

Enter a date after 2000-01-01:
<input type="date" name="bday" min="2000-01-02">

Quantity (between 1 and 5):
<input type="number" name="quantity" min="1" max="5">
```

The required Attribute

- The required attribute is a boolean attribute.
- When present, it specifies that an input field must be filled out before submitting the form.
- The required attribute works with the following input types: text, search, url, tel, email, password, date pickers, number, checkbox, radio, and file.

Example

A required input field:

Username: <input type="text" name="usrname" required>

The step Attribute

- The step attribute specifies the legal number intervals for an <input> element.
- Example: if step="3", legal numbers could be -3, 0, 3, 6, etc.
- **Tip:** The step attribute can be used together with the max and min attributes to create a range of legal values.
- The step attribute works with the following input types: number, range, date, datetime, datetime-local, month, time and week.

Example

An input field with a specified legal number intervals:

<input type="number" name="points" step="3">
