

# What is HTML?

HTML is a **markup** language for **describing** web documents (web pages).

- HTML stands for **Hyper Text Markup Language**
- A markup language is a set of **markup tags**
- HTML documents are described by **HTML tags**
- Each HTML tag **describes** different document content

## HTML Example

A small HTML document:

```
<!DOCTYPE html>
<html>
<head>
<title>Page Title</title>
</head>
<body>

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

</body>
</html>
```

## Example Explained

- The **DOCTYPE** declaration defines the document type to be HTML
- The text between **<html>** and **</html>** describes an HTML document
- The text between **<head>** and **</head>** provides information about the document
- The text between **<title>** and **</title>** provides a title for the document
- The text between **<body>** and **</body>** describes the visible page content
- The text between **<h1>** and **</h1>** describes a heading
- The text between **<p>** and **</p>** describes a paragraph

## HTML Tags

HTML tags are **keywords** (tag names) surrounded by **angle brackets**:

<tagname>content</tagname>

- HTML tags normally come **in pairs** like <p> and </p>
- The first tag in a pair is the **start tag**, the second tag is the **end tag**
- The end tag is written like the start tag, but with a **slash** before the tag name



The start tag is often called the **opening tag**. The end tag is often called the **closing tag**.

---

## Web Browsers

The purpose of a web browser (Chrome, IE, Firefox, Safari) is to read HTML documents and display them.

The browser does not display the HTML tags, but uses them to determine how to display the document:



# HTML Page Structure

Below is a visualization of an HTML page structure:

```
<html>
  <head>
    <title>Page title</title>
  </head>

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



Only the `<body>` area (the white area) is displayed by the browser.

---

## The `<!DOCTYPE>` Declaration

The `<!DOCTYPE>` declaration helps the browser to display a web page correctly.

There are different document types on the web.

To display a document correctly, the browser must know both type and version.

The doctype declaration is not case sensitive. All cases are acceptable:

```
<!DOCTYPE html>
<!DOCTYPE HTML>
<!doctype html>
<!Doctype Html>
```

## HTML Editors

# Write HTML Using Notepad or TextEdit

HTML can be edited by using professional HTML editors like:

- Microsoft WebMatrix
- Sublime Text

However, for learning HTML we recommend a text editor like Notepad (PC) or TextEdit (Mac).

We believe using a simple text editor is a good way to learn HTML.

Follow the 4 steps below to create your first web page with Notepad.

## Step 1: Open Notepad

To open Notepad in Windows 7 or earlier:

Click **Start** (bottom left on your screen). Click **All Programs**. Click **Accessories**. Click **Notepad**.

To open Notepad in Windows 8 or later:

Open the **Start Screen** (the window symbol at the bottom left on your screen). Type **Notepad**.

---

## Step 2: Write Some HTML

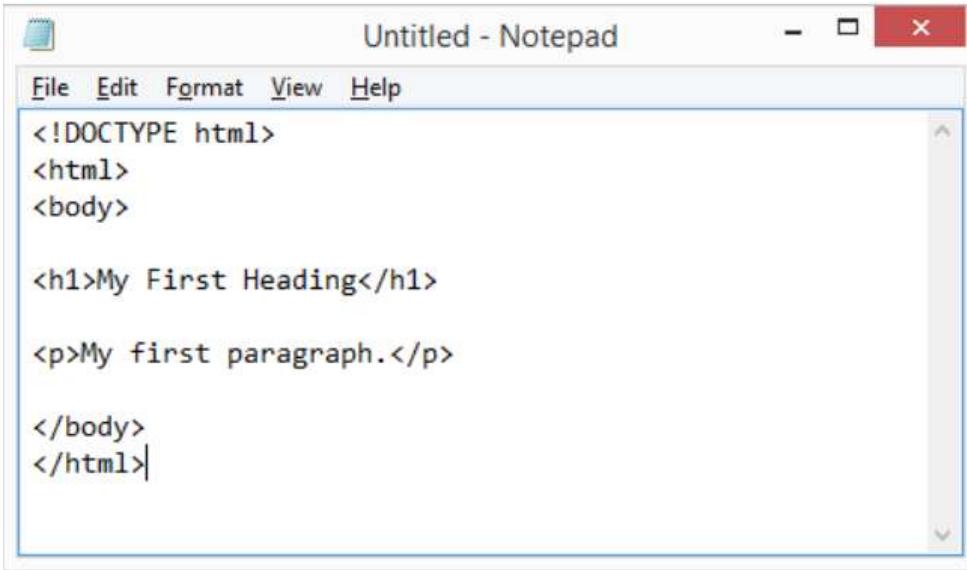
Write or copy some HTML into Notepad.

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

<h1>My First Heading</h1>

<p>My first paragraph.</p>

</body>
</html>
```



A screenshot of the Windows Notepad application. The window title is "Untitled - Notepad". The menu bar includes "File", "Edit", "Format", "View", and "Help". The main text area contains the following HTML code:

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

<h1>My First Heading</h1>

<p>My first paragraph.</p>

</body>
</html>
```

## Step 3: Save the HTML Page

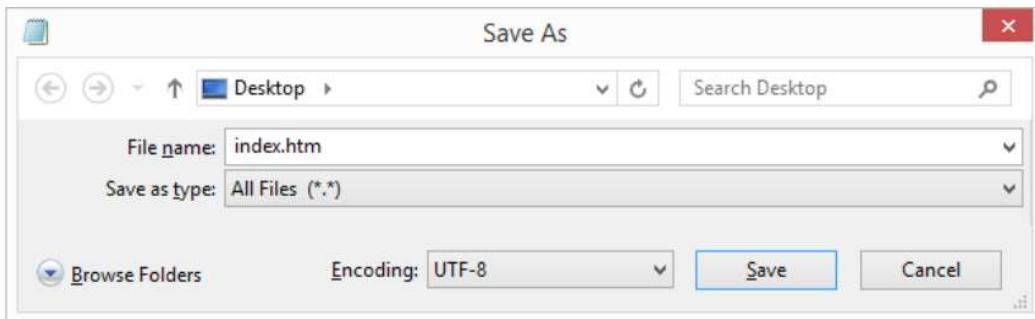
Save the file on your computer.

Select **File > Save as** in the Notepad menu.

Name the file "index.html" or any other name ending with html or htm.

UTF-8 is the preferred encoding for HTML files.

ANSI encoding covers US and Western European characters only.



## Step 4: View HTML Page in Your Browser

Open the saved HTML file in your favorite browser. The result will look much like this:



To open a file in a browser, double click on the file, or right-click, and choose open with.

## HTML Basic Examples

# HTML Headings

HTML headings are defined with the **<h1>** to **<h6>** tags:

#### Example

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

# HTML Paragraphs

HTML paragraphs are defined with the **<p>** tag:

#### Example

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

# HTML Links

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

#### Example

```
<a href="http://www.w3schools.com">This is a link</a>
```

The link's destination is specified in the **href attribute**.

Attributes are used to provide additional information about HTML elements.

---

## HTML Images

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

The source file (**src**), alternative text (**alt**), and size (**width** and **height**) are provided as **attributes**:

Example

```

```

## HTML Attributes

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

## The lang Attribute

- The document language can be declared in the **<html>** tag.
- The language is declared in the **lang** attribute.
- Declaring a language is important for accessibility applications (screen readers) and search engines:

```
<!DOCTYPE html>  
<html lang="en-US">  
<body>  
  
<h1>My First Heading</h1>  
<p>My first paragraph.</p>  
  
</body>  
</html>
```

- The first two letters specify the language (en). If there is a dialect, use two more letters (US).

# Summary

- All HTML elements can have **attributes**
- The HTML **title** attribute provides additional "tool-tip" information
- The HTML **href** attribute provides address information for links
- The HTML **width** and **height** attributes provide size information for images
- The HTML **alt** attribute provides text for screen readers
- At W3Schools we always use **lowercase** HTML attribute names
- At W3Schools we always **quote** attributes with double quotes

## Headings Are Important

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

## 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>
```

## HTML Line Breaks

The HTML **<br>** element defines a **line break**.

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

**Example**

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

The **<br>** element is an empty HTML element. It has no end tag.

## The HTML **<pre>** Element

The HTML **<pre>** element defines preformatted text.

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

Example

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

    My Bonnie lies over the sea.

    My Bonnie lies over the ocean.

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

## HTML Styles

### The HTML Style Attribute

Setting the style of an HTML element, can be done with the **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 Background Color

The **background-color** property defines the background color for an HTML element:

This example sets the background for a page to lightgrey:

Example

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

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

</body>
```

### HTML Text Color

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

#### Example

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

## HTML Fonts

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

#### Example

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

## HTML Text Size

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

#### Example

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

## HTML Text Alignment

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

#### Example

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

## HTML Text 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 **<strong>** 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 **<em>** element defines *emphasized* text, with added semantic importance.

### Example

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

## 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~~**Deleted** Formatting

The HTML **<del>** element defines **deleted** (removed) 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 <sub>Subscript</sub> Formatting

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

Example

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

## HTML <sup>Superscript</sup> Formatting

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

Example

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

## HTML Comments

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

---

## HTML Comment Tags

You can add comments to your HTML source by using the following syntax:

```
<!-- Write your comments here -->
```

## HTML Colors

### Color Names

With CSS, colors can be set by using color names:

Example

Color	Name
Red	Red
Orange	Orange
Yellow	Yellow
Cyan	Cyan
Blue	Blue

### RGB (Red, Green, Blue)

With HTML, RGB color values can be specified using this formula:  
rgb(red, green, blue)

Each parameter (red, green, and blue) defines the intensity of the color between 0 and 255.

For example, rgb(255,0,0) is displayed as red, because red is set to its highest value (255) and the others are set to 0. Experiment by mixing the RGB values below:

Example

Color	RGB
Red	rgb(255,0,0)
Yellow	rgb(255,255,0)
Green	rgb(0,255,0)
Cyan	rgb(0,255,255)
Blue	rgb(0,0,255)

## Hexadecimal Colors

With HTML, RGB values can also be specified using hexadecimal color values in the form: #RRGGBB, where RR (red), GG (green) and BB (blue) are hexadecimal values between 00 and FF (same as decimal 0-255).

For example, #FF0000 is displayed as red, because red is set to its highest value (FF) and the others are set to the lowest value (00).

Example

Color	HEX
Red	#FF0000
Yellow	#FFFF00
Green	#00FF00
Cyan	#00FFFF
Blue	#0000FF

## HTML Links - Hyperlinks

HTML links are hyperlinks.

A hyperlink is a text or an image you can click on, and jump to another document.

## HTML Links - Syntax

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

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

Example

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

# 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 - Colors

When you move the mouse over a link, two things will normally happen:

- The mouse arrow will turn into a little hand
- The color of the link element will change

By default, a link will appear like this (in all browsers):

- An unvisited link is underlined and blue
- A visited link is underlined and purple
- An active link is underlined and red

You can change the default colors, by using styles:

Example

```
<style>
a:link {color:green; background-color:transparent; text-decoration:none}
a:visited {color:purple; background-color:transparent; text-decoration:none}
a:hover {color:red; background-color:transparent; text-decoration:underline}
a:active {color:yellow; background-color:transparent; text-decoration:underline}
</style>
```

## 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.w3schools.com/" target="_blank">Visit  
W3Schools!</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

## HTML Links - Image as Link

It is common to use images as links:

#### Example

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

## HTML Links - Create a Bookmark

HTML bookmarks are used to allow readers to jump to specific parts of a Web page.

Bookmarks are practical if your website has long pages.

To make a bookmark, you must first create the bookmark, and then add a link to it.

When the link is clicked, the page will scroll to the location with the bookmark.

## Example

First, create a bookmark with the id attribute:

```
<h2 id="tips">Useful Tips Section</h2>
```

Then, add a link to the bookmark ("Useful Tips Section"), from within the same page:

```
<a href="#tips">Visit the Useful Tips Section</a>
```

Or, add a link to the bookmark ("Useful Tips Section"), from another page:

Example

```
<a href="html_tips.html#tips">Visit the Useful Tips Section</a>
```

## HTML Images

### HTML Images Syntax

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

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

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

```

```

---

### The alt Attribute

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

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

If a browser cannot find an image, it will display the alt text:

Example

```

```

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

You can use **width** and **height** attributes. Here, the values are specified in pixels by default:

Example

```

```

# 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 to store images in a sub-folder. You must then include the folder name in the src attribute:

Example

```

```

# 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

```

```

# Animated Images

The GIF standard allows animated images:

Example

```

```

# Using an Image as a Link

To use an image as a link, simply nest the `<img>` tag inside the `<a>` tag:

Example

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

# Image Maps

Use the `<map>` tag to define an image-map. An image-map is an image with clickable areas.

The name attribute of the `<map>` tag is associated with the `<img>`'s `usemap` attribute and creates a relationship between the image and the map.

The `<map>` tag contains a number of `<area>` tags, that defines the clickable areas in the image-map:

**Example**

```


<map name="planetmap">
  <area shape="rect" coords="0,0,82,126" alt="Sun" href="sun.htm">
  <area shape="circle" coords="90,58,3" alt="Mercury"
  href="mercur.htm">
  <area shape="circle" coords="124,58,8" alt="Venus"
  href="venus.htm">
</map>
```

## HTML Tables

### BASIC TABLES

Tables are defined with the `<table>` tag.

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

### ROWS:

To add rows to your table use the `<tr>` and `</tr>` tags.

**Example:**

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

### COLUMNS:

You can divide rows into columns with `<td>` and `</td>` tags:

**Example:**

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

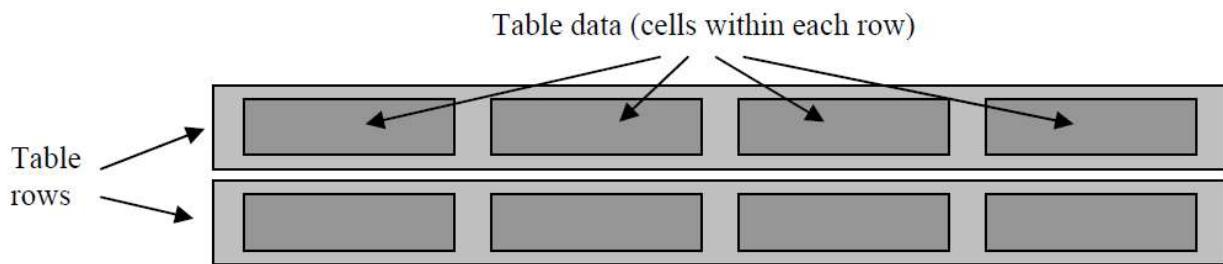
```

<td>This is row one, left side.</td> <td>This is row one, right side.</td>
</tr>
<tr>
    <td>This is row two, left side.</td> <td>This is row two, right side.</td>
</tr>
</table>

```

**Result:**

This is row one, left side.	This is row one, right side.
This is row two, left side.	This is row two, right side.



## TABLE ATTRIBUTE

- **The border**

This will make it easier to see what's happening.

```

<table border="1">
    <tr><td>Content goes here</td></tr>      =      Content goes here
    </table>

```

# Defining HTML Tables

**Example**

```

<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>
</table>

```

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>
```

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

Example

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

# 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>
```

## 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>
```

### Cell that spans two columns:

Name	Telephone	
Bill Gates	555 77 854	555 77 855

## 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>
```

## Cell that spans two rows:

Name:	Bill Gates
Telephone:	555 77 854
	555 77 855

## HTML Lists

Example of an unordered list and an ordered list in HTML:

Unordered List:

- Item
- Item
- Item
- Item

Ordered List:

1. First item
2. Second item
3. Third item
4. Fourth item

Lists are used to present information in an easy to read style. There are many types of lists:

- UNORDERED LISTS (Bullets)
- ORDERED LISTS (Numbers).
- COMBINED LISTS.
- DEFINED LISTS (Dictionary).

### Unordered List Types:

Type="square"

- Milk
- Toilet Paper

Type="disc"

- Milk
- Toilet Paper

Type="circle"

- Milk
- Toilet Paper

### Ordered List Types:

Lower-Case Letters    Upper-Case Letters    Lower-Case Numerals    Upper-Case Numerals

- |               |               |               |               |
|---------------|---------------|---------------|---------------|
| a. Find a Job | A. Find a Job | i. Find a Job | I. Find a Job |
| b. Get Money  | B. Get Money  | ii. Get Money | II. Get Money |

c. Move Out      C. Move Out      iii. Move Out      III. Move Out

## COMBINED LISTS.

In this one we can combined unordered and order lists, or may it combined unordered or order together.

```
<HTML>
  <HEAD>
    <TITLE> SAFETY TIPS </TITLE>
  </HEAD>
  <BODY>
    <H2 ALIGN="CENTER"> FREE ADVICES FOR STUDENS </H2>
    <OL>

      <LI>Read, read and read
      <LI>Write more and more
      <LI>Don't give up never

      <UL>
        <LI>Like this subject
        <LI>Define your aim
        <LI>Make a plan
      </UL>

      <LI>Work hard
      <LI>Use your Mind

    </OL>
  </BODY>
</HTML>
```

## DEFINED LISTS.

Make definition lists as seen in dictionaries using the `<dl>` tag. These lists displace the term word just above the definition itself for a unique look. It's wise to bold the terms to displace them further.

- `<dl>` - defines the start of the list
- `<dt>` - definition term
- `<dd>` - defining definition

## HTML Code:

```
<dl>
<dt><b>Fromage</b></dt>
<dd>French word for cheese.</dd>
<dt><b>Voiture</b></dt>
<dd>French word for car.</dd>
</dt>
```

#### HTML Code:

**Fromage**  
French word for cheese.  
**Voiture**  
French word for car.

## HTML Block and Inline Elements

Every HTML element has a default display value depending on what type of element it is. The default display value for most elements is block or inline.

---

## Block-level Elements

A block-level element always starts on a new line and takes up the full width available (stretches out to the left and right as far as it can).

The `<div>` element is a block-level element.

Examples of block-level elements:

- `<div>`
  - `<h1>` - `<h6>`
  - `<p>`
  - `<form>`
- 

## Inline Elements

An inline element does not start on a new line and only takes up as much width as necessary.

This is an inline <span> element inside a paragraph.

Examples of inline elements:

- <span>
  - <a>
  - <img>
- 

## The <div> Element

The <div> element is a **block-level element** that is often used as a container for other HTML elements.

The <div> element has no required attributes, but **style** and **class** are common.

When used together with CSS, the <div> element can be used to style blocks of content:

### Example

```
<div style="background-color:black; color:white; padding:20px;">

<h2>London</h2>
<p>London is the capital city of England. It is the most populous
city in the United Kingdom, with a metropolitan area of over 13
million inhabitants.</p>

</div>
```

## The <span> Element

The <span> element is an **inline element** that is often used as a container for some text.

The <span> element has no required attributes, but **style** and **class** are common.

When used together with CSS, the <span> element can be used to style parts of the text:

### Example

```
<h1>My <span style="color:red">Important</span> Heading</h1>
```

## HTML Layouts

Websites often display content in multiple columns (like a magazine or newspaper).

The screenshot shows a website layout titled "City Gallery". On the left, there is a sidebar with three items: "London", "Paris", and "Tokyo". The main content area is titled "London" and contains the following text:  
London is the capital city of England. It is the most populous city in the United Kingdom, with a metropolitan area of over 13 million inhabitants.  
Standing on the River Thames, London has been a major settlement for two millennia, its history going back to its founding by the Romans, who named it Londinium.

Copyright © W3Schools.com

## HTML Layout Using <div> Elements



The `<div>` element is often used as a layout tool, because it can easily be positioned with CSS.

This example uses four `<div>` elements to create a multiple column layout:

### Example

```
<body>

<div id="header">
<h1>City Gallery</h1>
</div>

<div id="nav">
London<br>
Paris<br>
Tokyo
</div>

<div id="section">
<h1>London</h1>
```

```

<p>London is the capital city of England. It is the most populous
city in the United Kingdom,
with a metropolitan area of over 13 million inhabitants.</p>
<p>Standing on the River Thames, London has been a major settlement
for two millennia,
its history going back to its founding by the Romans, who named it
Londinium.</p>
</div>

<div id="footer">
Copyright © W3Schools.com
</div>

</body>

```

The CSS:

```

<style>
#header {
    background-color:black;
    color:white;
    text-align:center;
    padding:5px;
}
#nav {
    line-height:30px;
    background-color:#eeeeee;
    height:300px;
    width:100px;
    float:left;
    padding:5px;
}
#section {
    width:350px;
    float:left;
    padding:10px;
}
#footer {
    background-color:black;
    color:white;
    clear:both;
    text-align:center;
    padding:5px;
}
</style>

```

## Website Layout Using HTML5

HTML5 offers new semantic elements that define different parts of a web page:



- <header> - Defines a header for a document or a section
- <nav> - Defines a container for navigation links
- <section> - Defines a section in a document
- <article> - Defines an independent self-contained article
- <aside> - Defines content aside from the content (like a sidebar)
- <footer> - Defines a footer for a document or a section
- <details> - Defines additional details
- <summary> - Defines a heading for the <details> element

This example uses <header>, <nav>, <section>, and <footer> to create a multiple column layout:

#### Example

```
<body>

<header>
<h1>City Gallery</h1>
</header>

<nav>
London<br>
Paris<br>
Tokyo
</nav>

<section>
<h1>London</h1>
<p>London is the capital city of England. It is the most populous city in the United Kingdom, with a metropolitan area of over 13 million inhabitants.</p>
<p>Standing on the River Thames, London has been a major settlement for two millennia, its history going back to its founding by the Romans, who named it Londinium.</p>
</section>
<footer>
Copyright © W3Schools.com
</footer>
</body>
```

```
The CSS:  
<style>  
header {  
    background-color:black;  
    color:white;  
    text-align:center;  
    padding:5px;  
}  
nav {  
    line-height:30px;  
    background-color:#eeeeee;  
    height:300px;  
    width:100px;  
    float:left;  
    padding:5px;  
}  
section {  
    width:350px;  
    float:left;  
    padding:10px;  
}  
footer {  
    background-color:black;  
    color:white;  
    clear:both;  
    text-align:center;  
    padding:5px;  
}  
</style>
```

---

## HTML Layout Using Tables



The `<table>` element was not designed to be a layout tool.  
The purpose of the `<table>` element is to display tabular data.

Layout can be achieved using the `<table>` element, because table elements can be styled with CSS:

### Example

```
<body>  
  
<table class="lamp">  
  <tr>  
    <th>
```

```


</th>
<td>
    The table element was not designed to be a layout tool.
</td>
</tr>
</table>

</body>

```

The CSS:

```

<style>
table.lamp {
    width:100%;
    border:1px solid #d4d4d4;
}
table.lamp th, td {
    padding:10px;
}
table.lamp th {
    width:40px;
}
</style>

```

## HTML frames

Frames can divide the screen into separate windows.



Each of these windows can contain an HTML document.

A file that specifies how the screen is divided into frames is called a frameset.

If you want to make a homepage that uses frames you should:

- make an HTML document with the frameset
- make the normal HTML documents that should be loaded into each of these frames.

When a frameset page is loaded, the browser automatically loads each of the pages associated with the frames.

An iframe is used to display a web page within a web page.

**HTML Code:**

```
<html>
<head>
</head>
<frameset cols="30%,*">
<frame src="menu.html">
<frame src="content.html">
</frameset>
</html>
```

- **frameset** - The parent tag that defines the characteristics of this frames page.  
Individual frames are defined inside it.
- **frameset cols="#%, \*"-** Cols(columns) defines the width that each frame will have. In the above example we chose the menu (the 1st column) to be 30% of the total page and used a "\*", which means the content (the 2nd column) will use the remaining width for itself.
- **frame src=""** -The location of the web page to load into the frame.

#### **adding a banner or title frame**

Add a row to the top for a title and graphics with the code as follows:

**HTML Code:**

```
<html><head></head>
<frameset rows="20%,*">
<frame src="title.html">
<frameset cols="30%,*">
<frame src="menu.html">
<frame src="content.html">
</frameset>
</html>
```

# Iframe

## Iframe Syntax

The syntax for adding an iframe is:

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

The **src** attribute specifies the URL (web address) of the iframe page.

---

## Iframe - Set Height and Width

Use the **height** and **width** attributes to specify the size.

The attribute values are specified in pixels by default, but they can also be in percent (like "80%").

Example

```
<iframe src="demo_iframe.htm" width="200" height="200"></iframe>
```

## Iframe - Remove the Border

By default, an iframe has a black border around it.

To remove the border, add the style attribute and use the CSS border property:

Example

```
<iframe src="demo_iframe.htm" style="border:none"></iframe>
```

With CSS, you can also change the size, style and color of the iframe's border:

Example

```
<iframe src="demo_iframe.htm" style="border:5px dotted red"></iframe>
```

## Use iframe as a Target for a Link

An iframe can be used as the target frame for a link.

The **target** attribute of the link must refer to the **name** attribute of the iframe:

#### Example

```
<iframe src="demo_iframe.htm" name="iframe_a"></iframe>
<p><a href="http://www.w3schools.com" Target="iframe_a">W3Schools</a></p>
```

## HTML Forms

### The <form> Element

HTML forms are used to collect user input.

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

```
<form>
.
.
</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

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

This is how it will look like in a browser:

First name:

Last name:

## Radio Button Input

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

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

### Example

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

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

First name:

Last name:

---

## 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">
```

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

- Male
- Female
- Other

---

## 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

```
<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>
```

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:

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

```
<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>
```

## 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:

Last name:

---

## HTML Form Attributes

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

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