**qwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmrtyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmrtyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmrtyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmrtyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmrtyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmrtyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmrtyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnm**

|  |
| --- |
| HTML & CSS  Assignment-1  8/14/2015  DHARANI MANICKAM (Trainee id: 044) |

***HTML***

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

* HTML stands for **H**yper **T**ext **M**arkup **L**anguage
* A markup language is a set of **markup tags**
* HTML documents are described by **HTML tags**
* Each HTML tag **describes** different document content

## *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
* HTML tags are not case sensitive: <P> means the same as <p>.

## *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>*

* 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 Headings*

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

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

***Hr and br tag:***

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

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

***pre Tag:***

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

***Address tag***

## HTML <address> for Contact Information

* The HTML **<address>** element defines contact information (author/owner) of a document or article.
* The <address> element is usually displayed in italic. Most browsers will add a line break before and after the element.

*<address>  
Written by Jon Doe.<br>   
Visit us at:<br>  
Example.com<br>  
Box 564, Disneyland<br>  
USA  
</address>*

***Abbreviation tag***

* The HTML **<abbr>** element defines an abbreviation or an acronym.
* Marking abbreviations can give useful information to browsers, translation systems and search-engines.

*<p>The <abbr title="World Health Organization">WHO</abbr> was founded in 1948.</p>*

***Bidirectional tag***

* The HTML **<bdo>** element defines bi-directional override.
* The <bdo> element is used to override the current text direction:

*<bdo dir="rtl">This text will be written from right to left</bdo>*

***HTML comments***

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

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

# *HTML Styles*

* Every HTML element has a **default style** (background color is white and text color is black).
* Changing the default style of an HTML element, can be done with the **style attribute**.

style="property:value"

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

*<body style="background-color:lightgrey">  
  
<h1>This is a heading</h1>  
<p>This is a paragraph.</p>  
</body>*

* 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 and etc.

***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

***Bold text & Important text***

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

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

*<p>This text is normal.</p>  
  
<p><b>This text is bold</b>.</p>  
  
<p><strong>This text is strong</strong>.</p>*

***Italic text & Emphasized text***

The HTML **<i>** element defines italic text, without any extra importance.  
The HTML **<em>** element defines emphasized text, with added semantic importance.

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

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

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

***Small text***

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

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

***Mark text***

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

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

***Delete text***

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

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

***Insert text***

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

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

***Subscripting text***

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

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

***Superscripting text***

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

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

## *HTML Links – Hyperlinks*

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

HTML links are hyperlinks.

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

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

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

*<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 |

Note:

* 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 link to the bookmark

## *HTML Images*

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:

*<img src="url" alt="some\_text">*

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

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

## *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:

*<img src="planets.gif" alt="Planets" usemap="#planetmap" style="width:145px;height:126px;">  
  
<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*

## *Defining HTML Tables*

*<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>*

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.

## *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:

*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:

*<table style="width:100%">  
  <tr>  
    <th>Firstname</th>  
    <th>Lastname</th>   
    <th>Points</th>  
  </tr>  
</table>*

## *Table Cells that Span Many Columns*

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

*<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:

*<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>*

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

*<caption>Monthly savings</caption>*

# *HTML Lists*

## *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):

*<ul style="list-style-type:disc">  
  <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 |

***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:

*<ol type="1">>  
  <li>Coffee</li>  
  <li>Tea</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 |

***HTML Description Lists***

HTML also supports description lists.

A description list is a list of terms, with a description of each term.

The **<dl>** tag defines the description list, the **<dt>** tag defines the term (name), and the **<dd>** tag describes each term:

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

***Nested HTML Lists***

List can be nested (lists inside lists):

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

# *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.

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:

*<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:

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

***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 |

|  |  |
| --- | --- |
|  |  |

***HTML code :***

*<body>  
  
<header>  
<h1>City Gallery</h1>  
</header>  
  
<nav>  
London<br>  
Paris<br>  
Tokyo<br>  
</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>*

***CSS code:***

*<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;   
}*

# *HTML Responsive Web Design*

* RWD stands for Responsive Web Design
* RWD can deliver web pages in variable sizes
* RWD is a must for tablets and mobile devices

# *HTML Entities*

Reserved characters in HTML must be replaced with character entities.

Characters, not present on your keyboard, can also be replaced by entities.

Some Other Useful HTML Character Entities

|  |  |  |  |
| --- | --- | --- | --- |
| **Result** | **Description** | **Entity Name** | **Entity Number** |
|  | non-breaking space | &nbsp; | &#160; |
| < | less than | &lt; | &#60; |
| > | greater than | &gt; | &#62; |
| & | ampersand | &amp; | &#38; |
| ¢ | cent | &cent; | &#162; |
| £ | pound | &pound; | &#163; |
| ¥ | yen | &yen; | &#165; |
| € | euro | &euro; | &#8364; |
| © | copyright | &copy; | &#169; |
| ® | registered trademark | &reg; | &#174; |

# *HTML Forms*

HTML forms are used to collect user input.

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

*<form>  
.  
form elements  
.  
</form>*

## The <input> Element

The <input> element has many variations, depending on the **type** attribute.

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

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

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

*<form>  
User password:<br>  
<input type="password" name="psw">  
</form>*

**<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:

*<form action="action\_page.php">  
First name:<br>  
<input type="text" name="firstname" value="Mickey">  
<input type="submit" value="Submit">  
</form>*

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

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

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

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

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

*<form>  
<input type="checkbox" name="vehicle2" value="Car"> I have a car   
</form>*

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

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

The **<fieldset>** element groups related data in a form.

The **<legend>** element defines a caption for the <fieldset> element.

## *HTML5 Input Types(Newly added)*

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

***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 |

***Action Attribute***

The **action attribute** defines the action to be performed when the form is submitted.

<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"****>*

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*

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 HTML <meta> Element***

* The <meta> element is used to specify page description, keywords, author, and other metadata.
* Meta data is used by browsers (how to display content), by search engines (keywords), and other web services.

1. Define keywords for search engines:

*<meta name="keywords" content="HTML, CSS, XML, XHTML, JavaScript">*

1. Define a description of your web page:

*<meta name="description" content="Free Web tutorials on HTML and CSS">*

1. Define the character set used:

*<meta charset="UTF-8">*

1. Define the author of a page:

*<meta name="author" content="Hegel Refines">*

1. Refresh document every 30 seconds:

*<meta http-equiv="refresh" content="30">*

## *mailto link in HTML*

The mailto link is written like regular link with extra parameters inside the href attribute:

*<a href="mailto:name@email.com">Link text</a>*

# *HTML5:*

|  |
| --- |
| * Reach multiple devices Desktop, mobile, tablet, TV * Powerful and modular Documents, multimedia, interactivity * Multi-application eBooks, user interfaces, games * Standard scheduled for 2014 |

## *New HTML5 Elements*

* The most interesting new elements are:
* New **semantic** elements like <header>, <footer>, <article>, and <section>.
* New form **control attributes** like number, date, time, calendar, and range.
* New **graphic** elements: <svg> and <canvas>.
* New **multimedia** elements: <audio> and <video>.

## *CSS:*

* **CSS** stands for **C**ascading **S**tyle **S**heets
* CSS defines **how HTML elements are to be displayed**
* Styles were added to HTML 4.0 **to solve a problem**
* CSS saves a lot of work
* External Style Sheets are stored in **CSS files**

## *CSS Syntax*

A CSS rule set consists of a selector and a declaration block:



The selector points to the HTML element you want to style.

The declaration block contains one or more declarations separated by semicolons.

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

## *CSS Comments*

A CSS comment starts with /\* and ends with \*/. Comments can also span multiple lines:  
*/\* this is a single-line comment \*/  
/\* this is  
a multi-line  
comment \*/*

***CSS Selectors***

CSS selectors allow you to select and manipulate HTML elements.

CSS selectors are used to "find" (or select) HTML elements based on their id, class, type, attribute, and more.

***The element Selector***

The element selector selects elements based on the element name.

You can select all <p> elements on a page like this: (all <p> elements will be center-aligned, with a red text color)

*p {  
    text-align: center;  
    color: red;  
}*

***The id Selector***

* The id selector uses the id attribute of an HTML element to select a specific element.
* An id should be unique within a page, so the id selector is used if you want to select a single, unique element.
* To select an element with a specific id, write a hash character, followed by the id of the element.
* The style rule below will be applied to the HTML element with id="para1":

*#para1 {  
    text-align: center;  
    color: red;  
}****The class Selector***

* The class selector selects elements with a specific class attribute.
* To select elements with a specific class, write a period character, followed by the name of the class:
* In the example below, all HTML elements with class="center" will be center-aligned:

*.center {  
    text-align: center;  
    color: red;  
}*

* You can also specify that only specific HTML elements should be affected by a class.

*p.center {  
    text-align: center;  
    color: red;  
}*

***Grouping Selectors***

* you can group the selectors, to minimize the code.
* To group selectors, separate each selector with a comma.
* In the example below we have grouped the selectors from the code above:

*h1, h2, p {  
    text-align: center;  
    color: red;  
}*

## *Three Ways to Insert CSS*

There are three ways of inserting a style sheet:

* External style sheet
* Internal style sheet
* Inline style

## *External Style Sheet*

* With an external style sheet, you can change the look of an entire website by changing just one file!
* Each page must include a reference to the external style sheet file inside the <link> element. The <link> element goes inside the head section:

*<head>  
<link rel="stylesheet" type="text/css" href="mystyle.css">  
</head>*

## *Internal Style Sheet*

* An internal style sheet may be used if one single page has a unique style.
* Internal styles are defined within the <style> element, inside the head section of an HTML page:

*<head>  
<style>  
body {  
    background-color: linen;  
}  
  
h1 {  
    color: maroon;  
    margin-left: 40px;  
}   
</style>  
</head>*

## *Inline Styles*

* An inline style may be used to apply a unique style for a single element.

*<h1 style="color:blue;margin-left:30px;">This is a heading.</h1>*

# *CSS Background*

CSS background properties are used to define the background effects of an element:

* background-color
* background-image
* background-repeat
* background-attachment
* background-position

***Background Color***

* The background-color property specifies the background color of an element.
* The background color of a page is set like this:

*body {  
    background-color: #b0c4de;  
}*

## *Background Image*

* The background-image property specifies an image to use as the background of an element.
* By default, the image is repeated so it covers the entire element.
* The background image for a page can be set like this:

*body {  
    background-image: url("paper.gif");  
}*

## *Background Image - Repeat Horizontally or Vertically*

* By default, the background-image property repeats an image both horizontally and vertically.
* Some images should be repeated only horizontally or vertically, or they will look strange, like this:
* If the image is repeated only horizontally (repeat-x), the background will look better:

*body {  
    background-image: url("gradient\_bg.png");  
    background-repeat: repeat-x;  
}*

## *Background Image - Set position and no-repeat*

* The position of the image is specified by the background-position property:

*body {  
    background-image: url("img\_tree.png");  
    background-repeat: no-repeat;  
    background-position: right top;  
}*

* The background-attachment property sets whether a background image is fixed or scrolls with the rest of the page.

# *CSS Text*

# *Text Color*

The color property is used to set the color of the text.

With CSS, a color is most often specified by:

* a HEX value - like "#ff0000"
* an RGB value - like "rgb(255,0,0)"
* a color name - like "red"

## *Text Alignment*

* The text-align property is used to set the horizontal alignment of a text.
* Text can be centered, or aligned to the left or right, or justified.
* When text-align is set to "justify", each line is stretched so that every line has equal width, and the left and right margins are straight (like in magazines and newspapers).

*h1 {  
    text-align: center;  
}  
  
p.date {  
    text-align: right;  
}  
  
p.main{ text-align: justif*

## *Text Decoration*

* The text-decoration property is used to set or remove decorations from text.
* The text-decoration property is mostly used to remove underlines from links for design purposes:

*a {  
    text-decoration: none;  
}*

*It can also be used to decorate text:*

*h1 {  
    text-decoration: overline;  
}  
  
h2 {  
    text-decoration: line-through;  
}  
  
h3 {  
    text-decoration: underline;  
}*

## *Text Transformation*

* The text-transform property is used to specify uppercase and lowercase letters in a text.
* It can be used to turn everything into uppercase or lowercase letters, or capitalize the first letter of each word.

*p.uppercase {  
    text-transform: uppercase;  
}  
  
p.lowercase {  
    text-transform: lowercase;  
}  
  
p.capitalize {  
    text-transform: capitalize;  
}*

# *CSS Font*

## *Font Family*

* The font family of a text is set with the font-family property.

*p {*

*font-family: "Times New Roman", Times, serif; }*

## *Font Style*

The font-style property is mostly used to specify italic text.

This property has three values:

* normal - The text is shown normally
* italic - The text is shown in italics
* oblique - The text is "leaning" (oblique is very similar to italic, but less supported)

*p.normal { font-style: normal; }  
  
p.italic { font-style: italic; }  
  
p.oblique { font-style: oblique; }*

## *Font Size*

The font-size property sets the size of the text.

*h1 {  
    font-size: 40px;  
}*

## *CSS Font Properties*

|  |  |
| --- | --- |
| **Property** | **Description** |
| [font](http://www.w3schools.com/cssref/pr_font_font.asp) | Sets all the font properties in one declaration |
| [font-family](http://www.w3schools.com/cssref/pr_font_font-family.asp) | Specifies the font family for text |
| [font-size](http://www.w3schools.com/cssref/pr_font_font-size.asp) | Specifies the font size of text |
| [font-style](http://www.w3schools.com/cssref/pr_font_font-style.asp) | Specifies the font style for text |
| [font-variant](http://www.w3schools.com/cssref/pr_font_font-variant.asp) | Specifies whether or not a text should be displayed in a small-caps font |
| [font-weight](http://www.w3schools.com/cssref/pr_font_weight.asp) | Specifies the weight of a font |

# *CSS Links*

* a:link - a normal, unvisited link
* a:visited - a link the user has visited
* a:hover - a link when the user mouses over it
* a:active - a link the moment it is clicked

*/\* unvisited link \*/  
a:link { color: #FF0000; }  
  
/\* visited link \*/  
a:visited { color: #00FF00; }  
  
/\* mouse over link \*/  
a:hover { color: #FF00FF; }  
  
/\* selected link \*/  
a:active { color: #0000FF; }*

When setting the style for several link states, there are some order rules:

* a:hover MUST come after a:link and a:visited
* a:active MUST come after a:hover

# *CSS Layout - float and clear*

* The float property specifies whether or not an element should float.
* The clear property is used to control the behavior of floating elements.

The float Property

* In its simplest use, the float property can be used to wrap text around images.
* The following example specifies that an image should float to the right in a text:

*img {  
    float: right;  
    margin: 0 0 10px 10px;  
}*

***The clear Property***

* The clear property is used to control the behavior of floating elements.
* Elements after a floating element will flow around it. To avoid this, use the clear property.
* The clear property specifies on which sides of an element floating elements are not allowed to float:

*div {  
    clear: left;  
}*

# *CSS Layout - The position Property*

The position property specifies the type of positioning method used for an element (static, relative, fixed or absolute).

***Absolute***  
The value absolute generates an absolutely positioned box that’s positioned relative to its containing block. The position can be specified using one or more of the properties [top](http://reference.sitepoint.com/css/top), [right](http://reference.sitepoint.com/css/right), [bottom](http://reference.sitepoint.com/css/bottom), and [left](http://reference.sitepoint.com/css/left). Absolutely positioned boxes are removed from the flow and have no effect on later siblings. Margins on absolutely positioned boxes never collapse with margins on other boxes.

***Fixed***  
The value fixed generates an absolutely positioned box that’s positioned relative to the initial containing block (normally the viewport). The position can be specified using one or more of the properties [top](http://reference.sitepoint.com/css/top), [right](http://reference.sitepoint.com/css/right),[bottom](http://reference.sitepoint.com/css/bottom), and [left](http://reference.sitepoint.com/css/left). In the print media type, the element is rendered on every page.

***relative***  
The value relative generates a positioned box whose position is first computed as for the normal flow. The generated box is then offset from this position according to the properties [top](http://reference.sitepoint.com/css/top) or [bottom](http://reference.sitepoint.com/css/bottom) and/or [left](http://reference.sitepoint.com/css/left) or[right](http://reference.sitepoint.com/css/right). The position of the following box is computed as if the relatively positioned box occupied the position that was computed before the box was offset. This value cannot be used for table cells, columns, column groups, rows, row groups, or captions.

***static***  
The value static generates a box that isn’t positioned, but occurs in the normal flow. The properties [top](http://reference.sitepoint.com/css/top),[right](http://reference.sitepoint.com/css/right), [bottom](http://reference.sitepoint.com/css/bottom), [left](http://reference.sitepoint.com/css/left), and [z-index](http://reference.sitepoint.com/css/z-index) are ignored for static boxes.

***CSS Display***

* Every element on a web page is a rectangular box. The display property in CSS determines just how that rectangular box behaves. There are only a handful of values that are commonly used:

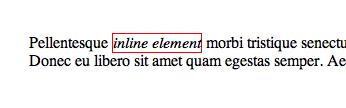
*display: inline; /\* Default of all elements, unless UA stylesheet overrides \*/*

*display: inline-block; /\* Characteristics of block, but sits on a line \*/*

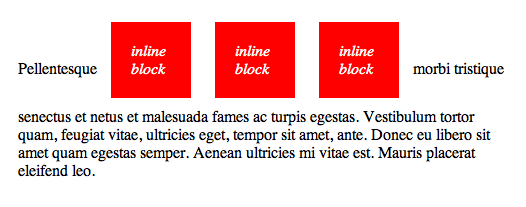
*display : block;*

* The default value for all elements is inline. Most "User Agent stylesheets" (the default styles the browser applies to all sites) reset many elements to "block". Let's go through each of these, and then cover some of the other less common values.

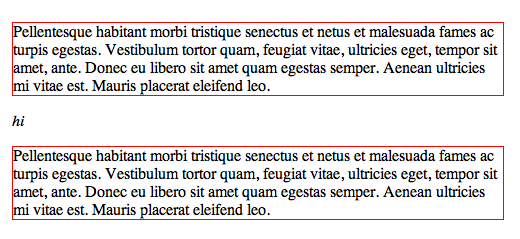
#### Inline

*   
  The <em> element has a 1px red border. Notice it sits right inline with the rest of the text.

#### Inline Block



#### Block



* The two elements with the red borders are <p>s which are block level elements. The<em> element in between them doesn't sit inline because the blocks break down below inline elements.