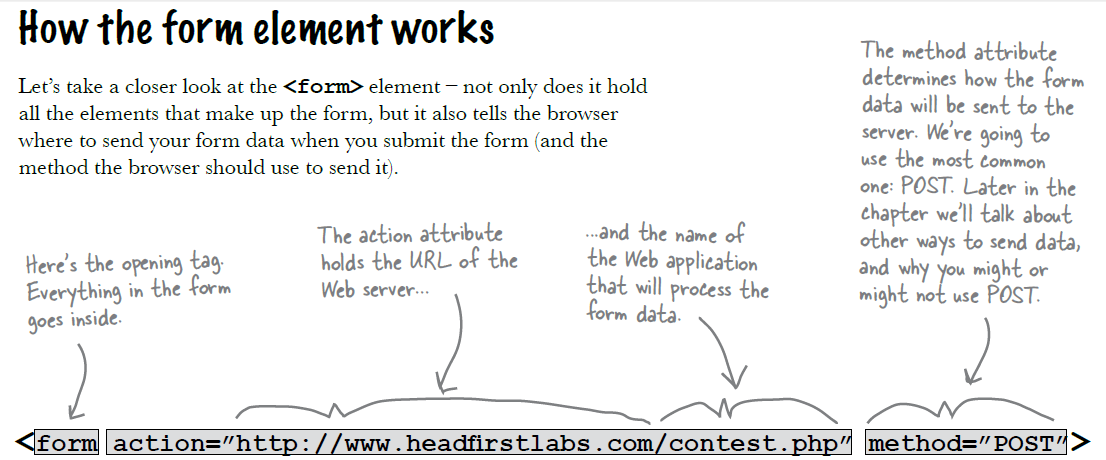
|  |  |  |  |
| --- | --- | --- | --- |
| HTML | EXPLANATION | EXAMPLE |  |
| textarea   rows=   cols=   name=     wrap=     off     virtual       physical | Text area - several lines Rows in the field. Columns in the field. Name of the field.   Control linebreaks.     Turns off linebreaks.     Shows linebreaks, but     sends text as entered.     Inserts linebreaks when     needed and even sends it. |  | <textarea cols="40" rows="5" name="myname"> Now we are inside the area - which is nice. </textarea> |
| text    size=   maxlength=   name=   value= | One line text field Characters shown. Max characters allowed. Name of the field. Initial value in the field. |  | <input type="text" size="25" value="Enter your name here!"> |
| password   size=   maxlength=   name=   value= | Password field. Characters shown. Characters allowed to enter. Name of the field. Initial value in the field. |  | Enter Password : <input type="password" size="25"> |
| checkbox   name=   value= | Choose one or more options Name of the field. Initial value in the field. |  | <input type="checkbox" name="option1" value="Milk"> Milk<br> <input type="checkbox" name="option2" value="Butter" checked> Butter<br> <input type="checkbox" name="option3" value="Cheese"> Cheese<br> |
| radio   name=   value= | Choose only one option Name of the field. Initial value in the field. |  | <input type="radio" name="group1" value="Milk"> Milk<br> <input type="radio" name="group1" value="Butter" checked> Butter<br> <input type="radio" name="group1" value="Cheese"> Cheese <hr> <input type="radio" name="group2" value="Water"> Water<br> <input type="radio" name="group2" value="Beer"> Beer<br> <input type="radio" name="group2" value="Wine" checked> Wine |
| select    name=   size=   multiple=   option   selected   value= | Drop-down menu  Name of the field. Number of items in list. Allow multiple choice if yes.   Individual items in the menu. Make an item default. Value to send if selected. |  | <select name="mydropdown"> <option value="Milk">Fresh Milk</option> <option value="Cheese">Old Cheese</option> <option value="Bread">Hot Bread</option> </select> |
| hidden   name=   value= | Does not show on the form. Name of the field. Value to send. |  | <input type="hidden" name="Language" value="English"> |
| reset    name=   value= | Button to reset all fields  Name of the button. Text shown on the button. |  | <input type="reset" value="Reset!"> |
| submit    name=   value= | Button to submit the form Name of the button. Text shown on the button. |  | <input type="submit" value="Send me your name!"> |
| image   name= | Image behaving as button Name of the image. |  | <input type="image" src="rainbow.gif" name="image" width="60" height="60"> |



<html>  
<head>  
<title>My Page</title>  
</head>  
  
<body>  
<!-- Here goes HTML -->  
<form method="post" action="http://www.echoecho.com/cgi-bin/formmail.cgi">  
<!-- Here goes form fields and HTML -->  
</form>   
<!-- Here goes HTML -->  
</body>  
</html>

To let the browser know where to send the content we add these properties to the <form> tag:

**action=address**

**method=post** or **method=get**

<table>  
 <tr>

<td> </td>

<td> </td>

</tr>  
 <tr>

<td> </td>

<td> </td>

</tr>  
</table>

http://www.echoecho.com/p.gif  
The following properties can be added to the <table> tag

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | Property | Description | | align= left center right | left align table center table right align table | | background=filename | image inserted behind the table | | bgcolor=#rrggbb | background color | | border=n | border thickness | | bordercolor=#rrggbb | border color | | bordercolordark=#rrggbb | border shadow | | cellpadding=n | distance between cell and content | | cellspacing=n | space between cells | | Nowrap | protects agains linebreaks, even though the content might be wider than the browser window. | | frame= void, above, below, lhs, rhs, hsides, vsides, box | removes all outer borders shows border on top of table shows border on bottom of table shows border on left side of table shows border on right side of table shows border on both horizontal sides shows border on both vertical sides shows border on all sides of table | | valign= top bottom | aligns content to top of cells aligns content to bottom of cells | | width= n,n n,n% | minimum width of table in pixels minimum width in percentage of window size | |

These settings can be added to both <tr> and <td> tags.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | PROPERTY | DESCRIPTION | | align= left right center | aligns content to the left of cells aligns content to the right of cells aligns content to the center of the cells | | background=filename | sets a background image for the cells | | bgcolor=#rrggbb | sets a background color for the cells | | bordercolor=#rrggbb | sets color for the border of cells | | bordercolordark=#rrggbb | sets color for the border shadow of cells | | valign= top middle bottom | aligns to the top of cells aligns to the middle of the cells aligns to the bottom of cells | | width= n n% | specify a minimum width for the cells in pixels specify a minimum width for the cells in percent of the table width | | height= n n% | minimum height of cells in pixels minimum height of cells in percentage of table height | |

These settings are only valid for <td> tags.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | PROPERTY | DESCRIPTION | | colspan=n | number of columns a cell should span | | nowrap | protects agains linebreaks, even though the content of a cell might be wider than the browser window | | rowspan=n | number of rows a cell should span | |

Links

The tags used to produce links are the <a> and </a>.   
  
The <a> tells where the link should start and the </a> indicates where the link ends.

Click <a href="http://www.yahoo.com">here</a> to go to yahoo.

Specify the target in the <a href=" ">

Click <a href="http://www.yahoo.com" style="color: rgb(0,255,0)">here</a> to go to yahoo.

You need to add a target if you want the link to open in another window or frame than the link itself is placed in.

<a href="http://www.yahoo.com" target="\_blank">

Predefined targets are:

**\_blank** loads the page into a new browser window.

**\_self** loads the page into the current window.

**\_parent** loads the page into the frame that is superior to the frame the hyperlink is in.

**\_top** cancels all frames, and loads in full browser window.

If you want to make an image work as a link, the method is exactly the same as with texts.   
  
You simply place the <a href> and the </a> tags on each side of the image.

<a href="myfile.htm"><img src="rainbow.gif" border="0"alt="Link to this page"></a>

<a href = “register.jsp”> Register </a>

Comments:

* If you place your comments in between <!-- and --> the browser will totally ignore them

Two types of elements:

1. Block elements
2. Inline elements
3. Block elements

<h1>, <h2>, ..., <h6>, <p>, and <blockquote> are all block elements.

Each block element is displayed on its own, as if it has a line break before and after it.

1. Inline elements

<q>, <a>, and <em> are inline elements.

<q> on the other hand, like all inline elements, is just displayed in the flow of the paragraph it’s in.

There are two types of elements in the world:

1. normal elements, like <p>, <h1>, and <a>,
2. empty elements, like <br> and <img>.
3. normal elements, like <p>, <h1>, and <a>,

* there should be start tag and close tag.
* **element = opening tag + content + closing tag**.

1. empty elements, like <br> and <img>.

* Elements that don’t have any HTML content by design are called empty elements.
* When you need to use an empty element, like <br> or <img>, you only use an opening tag.

You don’t switch back and forth between the two. For instance, if you just typed <a href=”mypage.html”>, that’s not an empty element – it’s an opening tag without content and a closing tag.

**List:**

1. Put each list item in an <li> element.

**<li>**Walla Walla, WA**</li>**

**<li>**Magic City, ID**</li>**

**<li>**Bountiful, UT**</li>**

**<li>**Last Chance, CO**</li>**

**<li>**Why, AZ**</li>**

**<li>**Truth or Consequences, NM**</li>**

1. Enclose your list items with either the <ol> or <ul> element.

**<ol>**

**<li>**Walla Walla, WA**</li>**

**<li>**Magic City, ID**</li>**

**<li>**Bountiful, UT**</li>**

**<li>**Last Chance, CO**</li>**

**<li>**Why, AZ**</li>**

**<li>**Truth or Consequences, NM**</li>**

**</ol>**

**u**nordered **l**ist = ul

**o**rdered **l**ist = ol

**l**ist **i**tem = li

* You should always use <ol> and <li> together (or <ul> and <li>).

Neither one of these elements really makes sense without the other.

Definition list:

**< dl>**

**<dt>**Burma Shave Signs**</dt>**

**<dd>**Road signs common in the U.S. in the 1920s

a n d **193**0s advertising shaving products.**</dd>**

**<dt>**Route 66**</dt>**

**<dd>** Most famous road in the U.S. highway

system.**</dd>**

**</dl>**

Each item in the list has a term, <dt>, and a description, <dd>.

Putting one element inside another is called “nesting”

A JPEG photo will usually be much smaller

than an equivalent quality GIF, while a GIF

logo will usually look better, and have a

smaller file size than in JPEG format.

PNG is the latest newcomer in

graphic formats, and an interesting one as

it can support both JPEG and GIF styles

of images. It also has more advanced

transparency features than GIF. Right now,

PNG is a little on the cutting edge because

not all browsers support it. But its popularity

is growing quickly. You should feel free to

use PNG, but just be aware that it won’t

work on every browser.

Use JPEG for photos and

complex graphics

Use GIF for images with solid

colors, logos, and geometric shapes.

Works best for continuous tone images,

like photographs.

Is a “lossy” format because to

reduce the file size, it throws away

some information about the image.

GIF also compresses the file to

reduce its size, but doesn’t throw

anything way. So, it is a “lossless”

format.

Can represent images with up to 16

million different colors.

Works best for images with a few

solid colors, and images with lines,

like logos, clip art, and small text in

images.

Can represent images with up to 256

different colors.

Does not support transparency. Allows one background color to be

set to “transparent” so that anything

underneath the image will show

through.

**<img src=”images/drinks.gif”>**

* The <img> element is an inline element. It doesn’t cause line breaks to be inserted before or after it.
* The src attribute specifies the location of an image file to be included in the display of the Web page.
* <img> is an empty element.

**<img src=”http://www.headfirstlabs.com/trivia/pencil.gif”**

**alt=”Pencil line 35 miles long”>**

* The alt attribute just requires a short bit of text that describes the image.
* If the image can’t be displayed, then this text is used in its place.

**<img src=”images/drinks.gif” width=”48” height=”100”>**

The width attribute tells the browser how wide the image should appear in the page.

The height attribute tells the browser how tall the image should appear in the page.

A good rule of thumb is 72 pixels to every inch, although depending on your monitor, you can have up to 120 pixels in an inch.

Assuming your monitor has 72 pixels per inch, if you want an image to be approximately 3” wide and high, you’d make it 72 (pixels) times

3 (inches) = 246 pixels, or, rounding up, 250 by 250 pixels.

make links out of images

**<a href=”html/seattle\_downtown.html”>**

**<img src=”thumbnails/seattle\_downtown.jpg” alt=”An iPod in downtown Seattle, WA”>**

**</a>**

Adding the document type definition

The document type definition tells your browser,

“I’m using HTML 4.01.” When the browser sees that, it assumes you know what you’re talking about and that you *really are* writing

HTML 4.01.

**<!DOCTYPE html PUBLIC “-//W3C//DTD HTML 4.01 Transitional//EN”**

**“http://www.w3.org/TR/html4/loose.dtd”>**

Adding a <meta> tag to specif y the content type

**<meta http-equiv=”Content-Type” content=”text/html; charset=ISO-8859-1”>**

meta” means we’re going to tell the browser something about the page...

Just like other HTML tags, the <meta> tag has attributes.

And we’re going to tell it something more about the content type ofthe page.

The content attribute is where we specify the content type information.

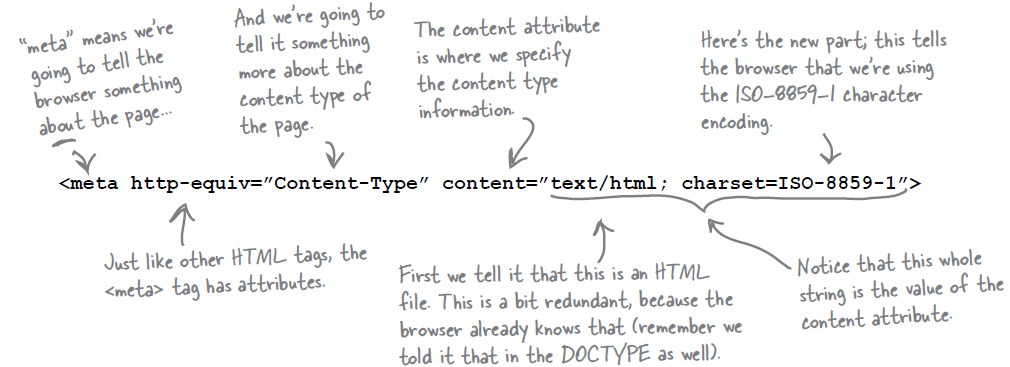
First we tell it that this is an HTML file. This is a bit redundant, because the

browser already knows that (remember we told it that in the DOCTYPE as well).

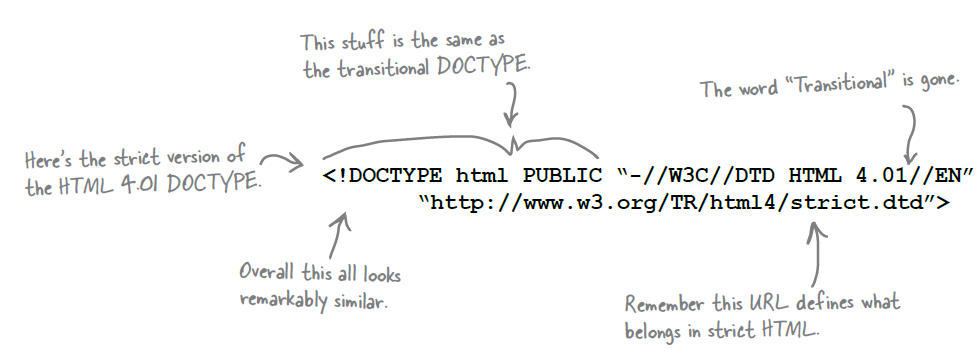
Here’s the new part; this tells the browser that we’re using the ISO-8859-1 character

encoding.

Notice that this whole string is the value of the content attribute



Changing the DOCTYPE to strict

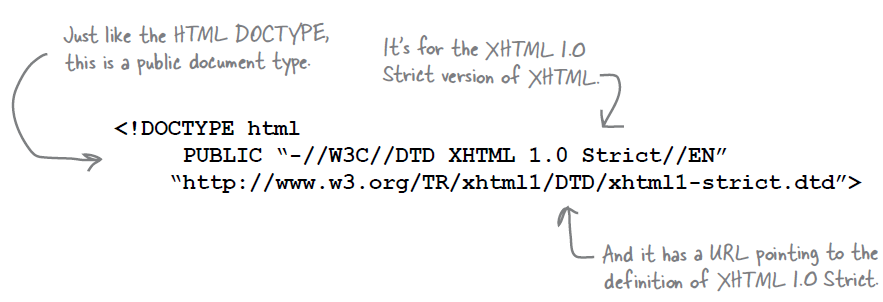


eXtensible HTML – otherwise known as XHTML

Going from strict HTML to XHTML 1.0 in three steps

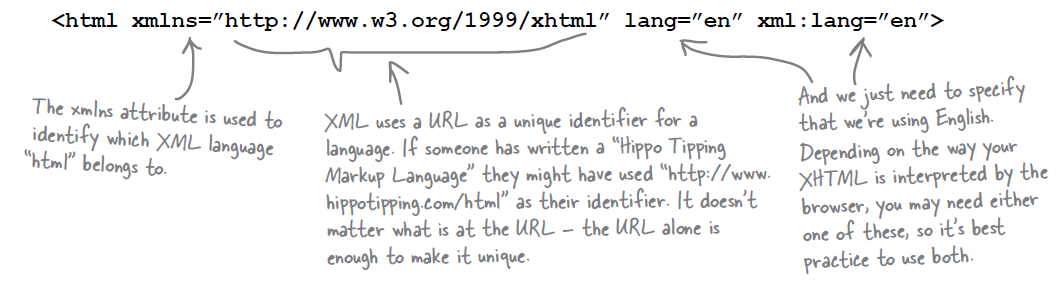
1. Change your DOCTYPE to XHTML 1.0 Strict.

**<!DOCTYPE html PUBLIC “-//W3C//DTD XHTML 1.0 Strict//EN” “http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd”>**

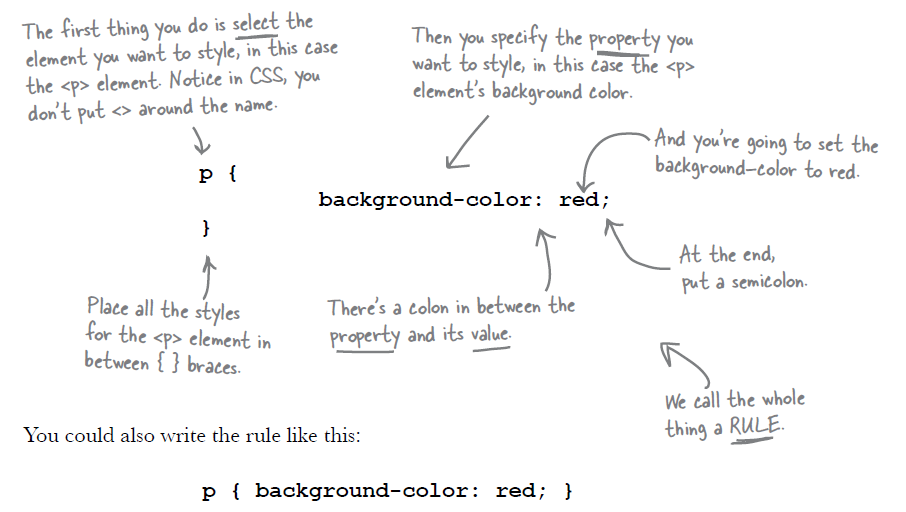
****

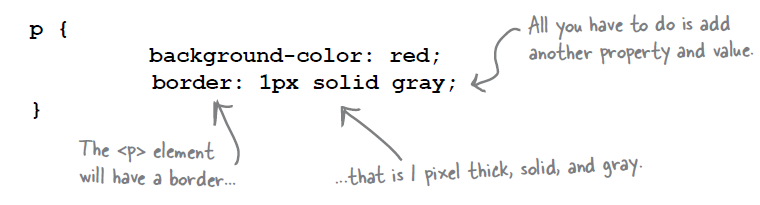
1. Add the xmlns, lang and xml:lang attributes to your <html> element.

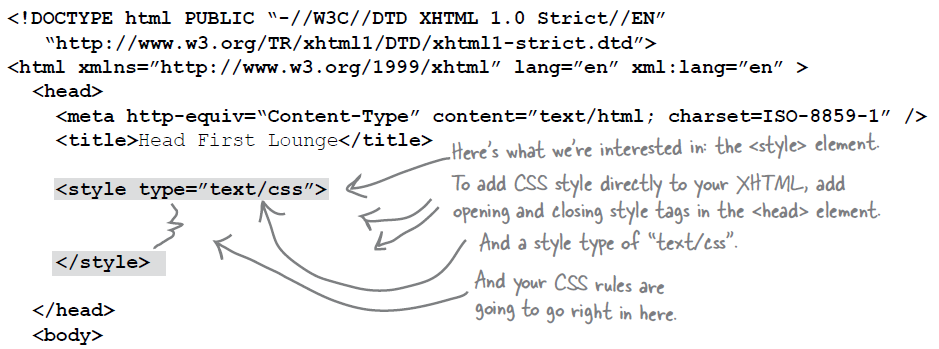
**<html xmlns=”http://www.w3.org/1999/xhtml” lang=”en” xml:lang=”en”>**



1. All empty tags should end in “ />”, not “>”.







external style sheet

**<html xmlns=”http://www.w3.org/1999/xhtml” lang=”en” xml:lang=”en”>**

**<head>**

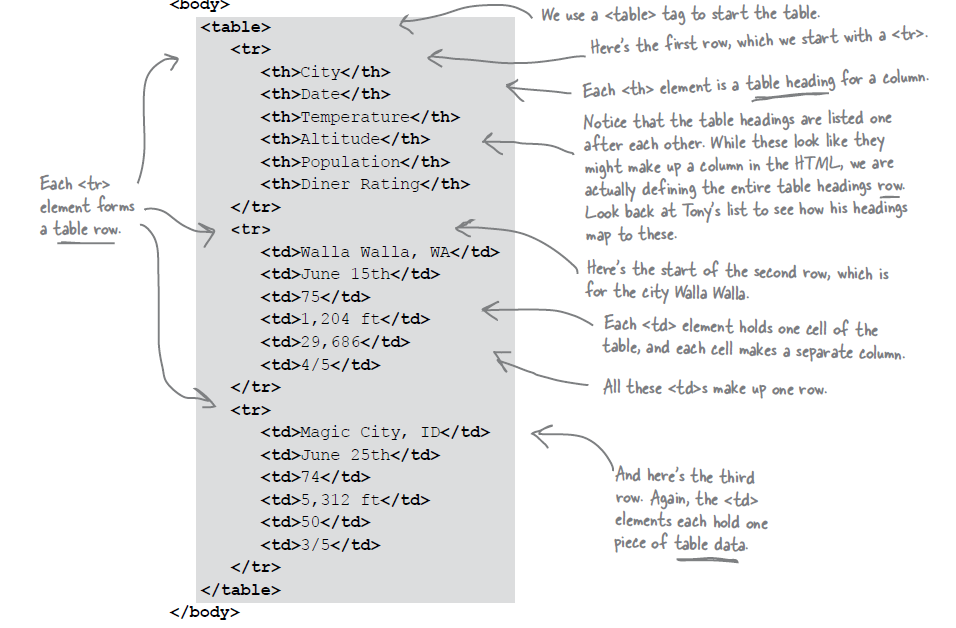
**<meta http-equiv=“Content-Type” content=”text/html; charset=ISO-8859-1” />**

**<title>**Head First Lounge Elixirs**</title>**

**<link type=”text/css” rel=”stylesheet” href=”../lounge.css” />**

**</head>**

**<body>**



**DIV and SPAN**

**(**a div usually needs at least a non-breaking space (&nbsp;) in order to have a width or height.)

The <div> tag defines a division or a section in an HTML document.

The <div> tag is used to group block-elements to format them with CSS.

By default, browsers always place a line break before and after the <div> element. However, this can be changed with CSS.

The <div> tag also supports the [Global Attributes in HTML](http://www.w3schools.com/tags/ref_standardattributes.asp).

The <div> tag also supports the [Event Attributes in HTML](http://www.w3schools.com/tags/ref_eventattributes.asp).

Most browsers will display the <div> element with the following default values:

div {  
    display: block;  
}

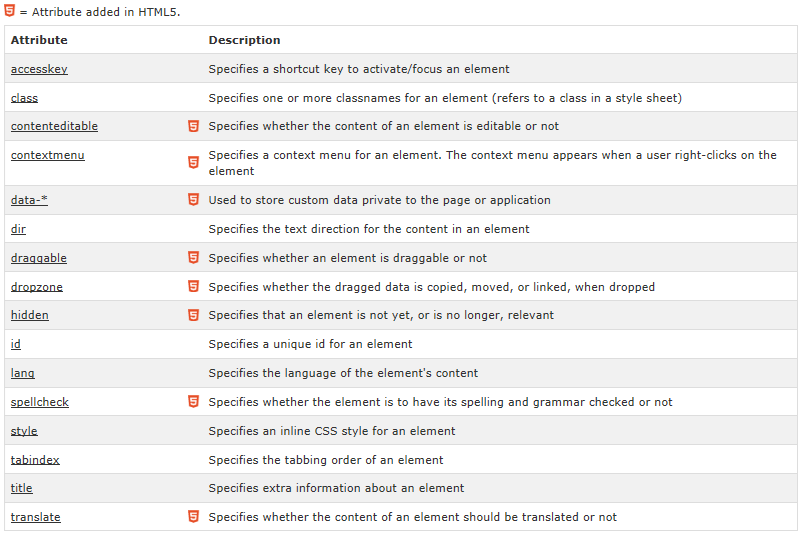
The difference between the div tag and the span tag is that the div tag is used with block-level elements whilst the span tag is used with inline elements.

Elements housed within a <div> tag acquire any styles or properties applied to the master div element.

Placing <div> elements inside of other <div> elements allows these elements to be further subdivided.

**HTML Global Attributes**

* The global attributes below can be used on **any** HTML element.



**Global Event Attributes**

HTML 4 added the ability to let events trigger actions in a browser, like starting a JavaScript when a user clicks on an element. For eg: onclick, onload, onselect, onblur, onchange, onfocus etc.

**Question: How to use HTML color codes?**

With HTML color codes you can set the color of web site background, color of text, cells in tables and much more.

* Using HTML color codes for web site background color:

**<body style="background:#80BFFF">**

* Using HTML color codes for setting font/text color:

**<span style="color:#80BFFF">**

* Using HTML color codes for table background color:

**<table style="background:#80BFFF">**

* Using HTML color code for link color:

**<a style="color:#80BFFF">**

**Question: Divide an html page into 3 vertical parts using div**

Answer:

* for it to work, no whitespace between the div elements.

<div style="width: *100%*; height: *100%*">

<div style="float: *left*; width: *40%*; height:*100%*; background: *red*">Left Stuff</div>

<div style="float: *left*; width: *20%*; height:*100%*; background: *blue*">Middle Stuff</div>

<div style="float: *left*; width: *40%*; height:*100%*; background: *red*">Right Stuff</div>

<br style="clear: *left*;" />

</div>

Use of float:

It helps the three containers sit side-by-side: "With CSS float, an element can be pushed to the left or right, allowing other elements to wrap around it."

**Divide web pages in columns and rows:**

<body>

<div style="width: *100%*; height: *100%*">

<div style = "width: *100%*; height: *20%*; background:*#FFFFCC* ;color:*red*; align:*center*">

Header

</div>

<div style = "width: *100%*; height: *70%*">

<div style="float: *left*; width: *10%*; height:*100%*; background: *red*">

Left Stuff

</div>

<div style="float: *left*; width: *70%*; height:*100%*; background: *#CCCCFF*">

Hi

</div>

<div style="float: *left*; width: *20%*; height:*100%*; background: *red*">

Right Stuff

</div>

<div style="clear: *left*;" />

</div>

<div style = "width: *100%*; height: *10%*; background:*green* ;color:*red*;align:*center*">

Footer

</div>

</div>

</body>

**Design Menu**

**1.**

**Steps:**

1. **create the list item**

<div id = “menu” >

<ul >

<li > <a href = "home.html">Home</a></li>

<li > <a href = "myaccount.html">My Account</a></li>

<li > <a href = "mywork.html">My Work</a></li>

<li > <a href = "contactus.html">Contact Us</a></li>

</ul>

</div>

1. modify the font of the menu list
2. make list horizontal

Change the positioning of the text so it fits into the menu and runs horizontally (inline).

#menu li{

display: inline;

padding: 20px;

}

1. remove the text underline and change the color of the text

#menu a{

text-decoration: none;

color: #00F;

Padding: 8px 8px 8px 8px

}

1. add hover effect for the text.

#menu a:hover{

color: white;

background-color: purple;

}

For e.g.:

<html>

<head>

<style>

body {background-color:#8080FF}

#menu{ background-color:#000080; width: 100%; text-align: right; margin-top : 0px; }

#menu li{ display: inline; padding: 8px}

#menu a{text-decoration: none;color: red;}

#menu a:hover{ color:#000033 ; background-color:#8080FF}

</style>

</head>

<body>

<div id = "menu">

<ul>

<li> <a href = "#">Home</li> </a>

<li> <a href = "#">Account</li> </a>

<li> <a href = "#">Gallery</li> </a>

<li> <a href = "#">News Update</li> </a>

<li> <a href = "#">Contact Us</li> </a>

</ul>

</div>

</body>

</html>

Eg: 2:

div.horizontal

{

margin: 10px;

width:100%;

height:30px;

background-color:#98bf21;

}

div.horizontal ul

{

list-style-type:none;

margin:0;

padding:0;

}

div.horizontal li

{

float:left;

}

div.horizontal a

{

display:block;

width:100px;

}

div.horizontal a:link,div.horizontal a:visited

{

font-weight:bold;

color:#FFFFFF;

background-color:#98bf21;

text-align:center;

padding:4px;

text-decoration:none;

text-transform:uppercase;

}

div.horizontal a:hover,div.horizontal a:active

{

background-color:#7A991A;

}

<div class="horizontal">

<ul>

<li><a href="#">Home</a></li>

<li><a href="#">News</a></li>

<li><a href="#">Articles</a></li>

<li><a href="#">Forum</a></li>

<li><a href="#">Contact</a></li>

<li><a href="#">About</a></li>

</ul>

</div>

<div style="text-align:center;">

<p>☎ (24) 3347-3110 | (24) 8119-1085&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;✉ @Html.ActionLink(Localization.Contact, MVC.Home.ActionNames.Contact, MVC.Home.Name)</p>

</div>

1. For header we can create a header image with logo and text and background color. Then we can include the image at the appropriate location.
2. How to put on div element over another div element.

Html 5:

The DOCTYPE declaration for HTML5 is very simple:

<!DOCTYPE html>

HTML5 Example:

<!DOCTYPE html>  
<html>  
<head>  
<meta charset="UTF-8">  
<title>Title of the document</title>  
</head>  
  
<body>  
Content of the document......  
</body>  
  
</html>

Html

HTML stands for Hyper Text Markup Language, which is the most widely used language on Web to develop

web pages.

HTML was created by Berners-Lee in late 1991 but "HTML 2.0" was the first standard HTML specification

which was published in 1995. HTML 4.01 was a major version of HTML and it was published in late 1999.

Though HTML 4.01 version is widely used but currently we are having HTML-5 version which is an extension

to HTML 4.01, and this version was published in 2012.

**HTML Document Structure:**

Document declaration tag

<html>

<head>

Document header related tags

</head>

<body>

Document body related tags i.e. all content which has to display.

</body>

</html>

**The doctype declaration:**

According to the version for the html we define the doctype

For example:

For html5:

<!DOCTYPE html>

HTML Basic tags

Heading tags:

* HTML have six levels of headings, which use the elements

**<h1>, <h2>, <h3>, <h4>, <h5>, and <h6>**

<!DOCTYPE html>

<html>

<head>

<title>Heading Example</title>

</head>

<body>

<h1>This is heading 1</h1>

<h2>This is heading 2</h2>

<h3>This is heading 3</h3>

<h4>This is heading 4</h4>

<h5>This is heading 5</h5>

<h6>This is heading 6</h6>

</body>

</html>

Paragraph tag

The **<p>** tag offers a way to structure your text into different paragraphs. Each

paragraph of text should go in between an opening <p> and a closing </p>

<!DOCTYPE html>

<html>

<head>

<title>Paragraph Example</title>

</head>

<body>

<p>Here is a first paragraph of text.</p>

<p>Here is a second paragraph of text.</p>

<p>Here is a third paragraph of text.</p>

</body>

</html>

**Line Break Tag**

Whenever you use the **<br />** element, anything following it starts from the

next line. This tag is an example of an **empty** element, where you do not need

opening and closing tags, as there is nothing to go in between them.

<!DOCTYPE html>

<html>

<head>

<title>Line Break Example</title>

</head>

<body>

<p>Hello<br />

You delivered your assignment on time.<br />

Thanks<br />

Mahnaz</p>

</body>

</html>

Use a nonbreaking space entity **&nbsp;**

HTML documents consist of a tree of the elements and they specify how HTML documents should be built, and what kind of content should be placed in what part of an HTML document.

Html Attributes:

An attribute is used to define the characteristics of an HTML element and is

placed inside the element's opening tag. All attributes are made up of two

parts: a **name** and a **value**:

The name is the property you want to set. For example, the paragraph

<p> element in the example carries an attribute whose name is **align**,

which you can use to indicate the alignment of paragraph on the page.

The value is what you want the value of the property to be set and

always put within quotations. The below example shows three possible

values of align attribute: **left, center** and **right**.

**Core Attributes**

The four core attributes that can be used on the majority of HTML elements

(although not all) are:

id

title

class

style

**The id Attribute**

The **id** attribute of an HTML tag can be used to uniquely identify any element

within an HTML page. There are two primary reasons that you might want to

use an id attribute on an element:

If an element carries an id attribute as a unique identifier it is possible

to identify just that element and its content.

If you have two elements of the same name within a Web page (or style

sheet), you can use the id attribute to distinguish between elements

that have the same name.

class attribute:

The value of the attribute may also be a space-separated list of class names.

For example:

class="className1 className2 className3"