**Software Development – Training Manual - CSS**

**Trainee Guide**

**Table of Contents**

Basic CSS 3

Introduction to CSS 3

CSS Layout 4

Validations using “validator.w3.org” for HTML 7

Study & Usability of HTML Properties as per W3C Standards 8

Types of CSS 10

Advanced CSS 11

Group CSS 11

CSS Hacks 12

UL, Li & OL 13

DL, DT, DD 20

Margins & Paddings 21

Borders 26

Display Properties 28

Positioning 31

Background 38

Nested CSS 47

Firebug 50

Training Report 56

## Basic CSS

## Introduction to CSS

Save a lot of work with CSS!  
In our CSS tutorial you will learn how to use CSS to control the style and layout of multiple Web pages all at once.

**What is CSS?**

* **CSS** stands for **C**ascading **S**tyle **S**heets
* Styles define **how to display** HTML elements
* Styles were added to HTML 4.0 **to solve a problem**
* **External Style Sheets** can save a lot of work

**Styles Solved a Big Problem**HTML was never intended to contain tags for formatting a document.  
HTML was intended to define the content of a document, like:  
<h1>This is a heading</h1>  
<p>This is a paragraph.</p>  
  
When tags like <font>, and color attributes were added to the HTML 3.2 specification, it started a nightmare for web developers. Development of large web sites, where fonts and color information were added to every single page, became a long and expensive process.

To solve this problem, the World Wide Web Consortium (W3C) created CSS.

In HTML 4.0, all formatting could be removed from the HTML document, and stored in a separate CSS file.

**All browsers support CSS today.**  
CSS Saves a Lot of Work!  
  
CSS defines HOW HTML elements are to be displayed.  
  
Styles are normally saved in external .css files. External style sheets enable you to change the appearance and layout of all the pages in a Web site, just by editing one single file!

## CSS Layout

body  
{  
background-color:#d0e4fe;  
}  
h1  
{  
color:orange;  
text-align:center;  
}  
p  
{  
font-family:"Times New Roman";  
font-size:20px;  
}  
  
**CSS Syntax**

A CSS rule has two main parts: a selector, and one or more declarations:

The selector is normally the HTML element you want to style.

Each declaration consists of a property and a value.

The property is the style attribute you want to change. Each property has a value.

**CSS Example**

A CSS declaration always ends with a semicolon, and declaration groups are surrounded by curly brackets:  
  
p {color:red;text-align:center;}

To make the CSS more readable, you can put one declaration on each line, like this:

|  |
| --- |
|  |

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

**The id and class Selectors**In addition to setting a style for a HTML element, CSS allows you to specify your own selectors called "id" and "class".

**The id Selector**The id selector is used to specify a style for a single, unique element.  
The id selector uses the id attribute of the HTML element, and is defined with a "#".  
The style rule below will be applied to the element with id="para1":

**Example:**#para1  
{  
text-align:center;  
color:red;  
}  
  
**The class Selector**The class selector is used to specify a style for a group of elements. Unlike the id selector, the class.  
selector is most often used on several elements.  
  
This allows you to set a particular style for many HTML elements with the same class.   
The class selector uses the HTML class attribute, and is defined with a "."  
In the example below, all HTML elements with class="center" will be center-aligned:  
  
**Example:**.center {text-align:center;}   
  
You can also specify that only specific HTML elements should be affected by a class.  
In the example below, all p elements with class="center" will be center-aligned:

**Example:**p.center {text-align:center;}   
Do NOT start a class name with a number! This is only supported in Internet Explorer.  
  
**CSS Comments**Comments are used to explain your code, and may help you when you edit the source code at a later date. Comments are ignored by browsers.

A CSS comment begins with "/\*", and ends with "\*/", like this:

/\*This is a comment\*/  
p  
{  
text-align:center;  
/\*This is another comment\*/  
color:black;  
font-family:arial;  
}

When a browser reads a style sheet, it will format the document according to it.

## Validations using “validator.w3.org” for HTML

**Validations using “http://jigsaw.w3.org/css-validator/” for CSS**

**What is this? Do you need it?**The W3C CSS Validation Service is a free software created by the W3C to help Web designers and Web developers check Cascading Style Sheets (CSS). It can be used on this free service on the web, or downloaded and used either as a java program, or as a java servlet on a Web server.

Do you need it? If you are a Web developer or a Web designer, this tool will be an invaluable ally. Not only will it compare your style sheets to the CSS specifications, helping you find errors, typos, or incorrect uses of CSS, it will also tell you when your CSS poses some risks in terms of usability.  
  
**The explanation above is incomprehensible! Help!**Most documents on the Web are written in a computer language called HTML. This language can be used to create pages with structured information, links, and multimedia objects. For color, text, and layout, HTML uses a styling language called CSS, short for "Cascading Style Sheets". What this tool does is help people authoring CSS check, and fix if necessary, their CSS Style Sheets.  
  
**What does “Valid CSS” mean? Which version of CSS does this validator use?**According to the CSS 2.1 Specification: "The validity of a style sheet depends on the level of CSS used for the style sheet. […] valid CSS 2.1 style sheet must be written according to the grammar of CSS 2.1. Furthermore, it must contain only at-rules, property names, and property values defined in this specification."

By default, this validator checks style sheets against the grammar, properties and values defined in the CSS 2.1 specification, but other CSS profiles can be checked against by using the options.

CSS is an evolving language, and it is considered by many that “CSS” is a single grammar (the one defined in the latest specification) with a number of properties and acceptable values defined in various profiles.In a future version of this validator, the default behavior may be to check style sheets against that latest “CSS grammar” and the cloud of all standardized CSS properties and values.

## Study & Usability of HTML Properties as per W3C Standards

Proper use of h1, h2, h3, h4, h5, p, ul, li tags  
 **H1 to h6 tag**Heading tags are used to create split content in parts to make things easier to read. It gets easier for the visitor to scan the document for the more important points of the article. A article with great use of heading tags could keep users reading on.  
  
But there is more. Another great advantage of headings is the power it has in search engine optimizing.  
  
The search engines have accepted the use of headings and will give more value to this text in an article. A keyword-rich heading will significant improve your search engine visibility.  
  
Use the **H1, H2...H6** tags as indicators of information hierarchy within a document, not just as formatting elements. Screen readers in particular may just scan a page for appropriate H1, H2 and H3 elements.  
  
There are six levels of headings from H1 (the most important) to H6 (the least important).  
  
A heading element briefly describes the topic of the section it introduces. Heading information may be used by user agents, for example, to construct a table of contents for a document automatically.  
  
There are six levels of headings in HTML with H1 as the most important and H6 as the least. Visual browsers usually render more important headings in larger fonts than less important ones.  
  
<h1> is the HTML element for the first-level heading of a document:

* If the document is basically stand-alone, for example Things to See and Do in Geneva, the top-level heading is probably the same as the title.
* If it is part of a collection, for example a section on Dogs in a collection of pages about pets, then the top level heading should assume a certain amount of context; just write <h1>Dogs</h1> while the title should work in any context: Dogs - Your Guide to Pets.

**<p> Tag**

**Example:**A paragraph is marked up as follows:  
<p>This is some text in a paragraph.</p>  
  
**Definition and Usage**The <p> tag defines a paragraph.  
The p element automatically creates some space before and after itself. The space is automatically applied by the browser, or you can specify it in a style sheet.  
Using the <p> tag causes the browser to automatically create a blank line below & above the text of the paragraph. The generation of these blank lines automatically by the browser is the explanation to how a paragraph is "marked" by the <p> tag & automatically interpreted by the browser as to how the  
paragraph text be displayed.  
  
Just as formatting is possible in a word processing program, paragraphs can also be formatted in HTML. The attribute used in HTML to format paragraphs is the align attribute. This attribute can be used to align the paragraph, left, right, center or to simply justify.

## Types of CSS

There are three ways of inserting a style sheet:

* **Inline CSS**
* **Embedded CSS**
* **Linked (External CSS)**

**1. Inline CSS**

An inline style loses many of the advantages of style sheets by mixing content with presentation. Use this method sparingly!

To use inline styles you use the style attribute in the relevant tag. The style attribute can contain any CSS property. The example shows how to change the color and the left margin of a paragraph:

**Example:**

<p style="color:sienna; margin-left:20px">This is a paragraph.</p>

**2. Embedded CSS (Internal Style Sheet)**An internal style sheet should be used when a single document has a unique style. You define internal styles in the head section of an HTML page, by using the <style> tag, like this:   
**Example:**

<head>

<style type="text/css">

hr {color: sienna;}

p {margin-left:20px;}

body {background-image: url("images/back40.gif");}

</style>

</head>  
  
**2. External Style Sheet**An external style sheet is ideal when the style is applied to many pages. With an external style sheet, you can change the look of an entire Web site by changing one file. Each page must link to the style sheet using the <link> tag. The <link> tag goes inside the head section:   
**Example:**

<head>

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

</head>

An external style sheet can be written in any text editor. The file should not contain any html tags. Your style sheet should be saved with a .css extension. An example of a style sheet file is shown below:

hr {color:sienna;}  
p {margin-left:20px;}  
body {background-image:url("images/back40.gif");}  
 **Note:** Do not leave spaces between the property value and the units! "margin-left:20 px" (instead of "margin-left:20px") will work in IE, but not in Firefox or Opera.

## Advanced CSS

The CSS Advanced Tutorial is for those who want to push CSS to the extreme, highlighting methods that might not be immediately obvious to the intermediate developer.

## **Group CSS**

|  |  |  |
| --- | --- | --- |
| **Advance CSS** | **Description** | **Hours** |
| **Group CSS** | Different classes which have the same css properties can be defined by using Single CSS group.  It helps to reduce the CSS file size. |  |
| **Example** | **Example of 3 difference css classes having same properties :**  .abc {font-family:Arial, Helvetica, sans-serif; font-size:12px; font-weight:bold; color:#F00;}  .xyz {font-family:Arial, Helvetica, sans-serif; font-size:12px; font-weight:bold; color:#F00;}  .pqr {font-family:Arial, Helvetica, sans-serif; font-size:12px; font-weight:bold; color:#F00;}  **These are the 3 CSS classes which has the same properties, so it can also be written as**  .abc, .xyz, .pqr {font-family:Arial, Helvetica, sans-serif; font-size:12px; font-weight:bold; color:#F00;}  **If any particular property is different from all the css or any off of css then, we can separate that single property**  .abc, .xyz, .pqr {font-family:Arial, Helvetica, sans-serif; font-size:12px; font-weight:bold; color:#F00;}  .abc {line-height:15px;}  .xyz {line-height:20px;}  .pqr {line-height:30px;} |  |

## CSS Hacks

|  |  |  |
| --- | --- | --- |
| **Advance CSS** | **Description** | **Hours** |
| **CSS Hacks** | CSS Hacks are used to define the CSS for particular Browser. |  |
| **Hacks for browser** | **IE 6 and below :**  \* html .xyz(class name or id name) { color: #00ff00;}  **IE 7 only:** \*:first-child+html .xyz { color: #00ff00;} OR \*+html .xyz { color: #00ff00;}  **IE8:**  . xyz {color: #0FC\0/;}  **IE9:** @media all and (min-width:0) {     #element { color:pink \0/; }  /\* IE9 \*/ }  **Mozilla:** html > body .xyz { color: #00ff00;}  **Opera versions 9 and below:** html:first-child .xyz { color: #00ff00;}  **Safari:** @media screen and (-webkit-min-device-pixel-ratio:0){ .xyz {color: #00ff00;}}  **MacIE:** /\*\\*//\*/  .xyz{ color: #00ff00;}  /\*\*/    **Examples for Browser Conditional CSS Hack** |  |

## **UL, Li & OL**

**Use of listing**  
The most common HTML lists are ordered and unordered lists:   
  
The objective of this technique is to create lists of related items using list elements appropriate for their purposes. The ol element is used when the list is ordered and the ul element is used when the list is unordered. Definition lists (dl) are used to group terms with their definitions. Although the use of this markup can make lists more readable, not all lists need markup. For instance, sentences that contain comma-separated lists may not need list markup.

When markup is used that visually formats items as a list but does not indicate the list relationship, users may have difficulty in navigating the information. An example of such visual formatting is including asterisks in the content at the beginning of each list item and using <br> elements to separate the list items.

Some assistive technologies allow users to navigate from list to list or item to item. Style sheets can be used to change the presentation of the lists while preserving their integrity.

The list structure (ul/ol) is also useful to group hyperlinks. When this is done, it helps screen reader users to navigate from the first item in a list to the end of the list or jump to the next list. This helps them to bypass groups of links if they choose to.

**How to Use?**

|  |  |  |
| --- | --- | --- |
| **Tags** | **Description** | **Hours** |
| **UL** | The <ul> tag defines an unordered list (a bulleted list). An unordered list starts with the <ul> tag. Each list item starts with the <li> tag.  The list items are marked with bullets (typically small black circles). |  |
| **Example** | |  |  | | --- | --- | | **Code** | **Result in Browser:** | | <ul>   <li>Coffee</li>   <li>Tea</li>   <li>Milk</li> </ul> | * Coffee * Tea * Milk | |  |
|  |  |  |
| **Assignment** | |  |  | | --- | --- | | **Question** | **Result should be:** | | Use "ul" Tag for  Name, Address, City, State, Zip Code | * Name * Address * City * State * Zip Code | |  |

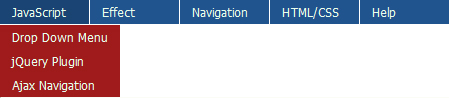
|  |  |  |
| --- | --- | --- |
| **Tags** | **Description** | **Hours** |
| **OL** | The <ol> tag is used to create an ordered list.  The list can be numerical or alphabetical.  An ordered list starts with the <ol> tag. Each list item starts with the <li> tag.  The list items are marked with numbers. |  |
| **Example** | |  |  | | --- | --- | | **Code** | **Result in Browser:** | | <ol>   <li>Coffee</li>   <li>Tea</li>   <li>Milk</li> </ol> | 1. Coffee 2. Tea 3. Milk | |  |
|  |  |  |
| **Assignment** | |  |  | | --- | --- | | **Question** | **Result should be:** | | Use "ol" Tag for  Name, Address, City, State, Zip Code | 1. Name 2. Address 3. City 4. State 5. Zip Code | |  |

|  |  |  |
| --- | --- | --- |
| **Advance CSS** | **Description** | **Hours** |
| **LI** | The <ol> tag is used to create an ordered list.  The list can be numerical or alphabetical.  List item. Used in conjunction with ul or [ol](http://www.htmldog.com/reference/htmltags/ol/) to make an unordered list or ordered list respectively. |  |
| **Example** | |  |  | | --- | --- | | **Code** | **Result in Browser:** | | <ul>   <li>Coffee</li>   <li>Tea</li>   <li>Milk</li> </ul> | * Coffee * Tea * Milk | |  |
|  |  |  |
| **Assignment** | |  |  | | --- | --- | | **Question** | **Result should be:** | | Use "Li" Tag for  Name, Address, City, State, Zip Code | * Name * Address * City * State * Zip Code | |  |

**UL, Li use for menus, why?**Because it's more semantically correct to use a list than standalone elements like span. A menu is often a "collection or list of links", therefore the <ul>-element fits nicely.  
  
Some like to say that it's easier as well as it has the most appropriate set of elements, you get a wrapper-element (the <ul>) and containing elements (<li>) "for free". But the main point is that the <ul> is the closest thing we can get to a semantically correct description/markup of a site navigation.  
  
**Tutorial**  
**In this tutorial we're going to create a professional horizontal CSS menu. First we are going to create a HTML list by using Unordered List (ul) and List Item (li) elements, and CSS (Cascading Style Sheets)  
  
Previous knowledge about some basic HTML and CSS is required. HTML elements used in this tutorial:**

* **ul (Unordered List)**
* **li (List Item)**
* **a (Anchor / Link)**

**Example:**We want create this type menu using with ul, li **HTML code (goes under body tag):**  
<ul>  
 <li> <a href="#">Home</a></li>  
 <li><a href="#">Products</a></li>  
 <li><a href="#">Services</a></li>  
 <li><a href="#">About</a></li>   
 <li><a style="background-image: none;" href="#">Contact</a></li>  
</ul>  
  
**CSS for the menu (goes under head tag or external style sheet):**ul {  
list-style-type: none;  
background-image: url(navi\_bg.png);  
height: 80px;  
width: 663px;  
margin: auto;  
}  
  
li a{  
list-style-type: none;  
border-right: 1px solid #562b2b;  
height: 80px;  
padding: 0px 30px;  
display: inline-block;  
}  
  
li a:hover{  
color:#ffffff;  
}  
  
li a .last{  
border-right: 0px;  
}

**Use of UL, Li for JavaScript, JQuery & CSS Drop down menus using UL, LI**You can use ul, li for JavaScript, JQuery & CSS Drop down to this way.  
 **Example:**This is screenshot for result ****  
First include this script in your head tag:  
<script src=http://ajax.googleapis.com/ajax/libs/jquery/1.2.6/jquery.js type="text/javascript"></script>  
  
Now you can create main menu structure using HTML unordered list. Using of an unordered list for navigation is a very good practice. First of all it is semantically correct; it is also handy for changing/editing especially in compound treelike cases.

|  |
| --- |
| <ul id="**jsddm**">  <li><a href="#">JavaScript</a>  <ul>  <li><a href="#">Drop Down Menu</a></li>  <li><a href="#">jQuery Plugin</a></li>  <li><a href="#">Ajax Navigation</a></li>  </ul>  </li>  <li><a href="#">Effect</a>  <ul>  <li><a href="#">Slide Effect</a></li>  <li><a href="#">Fade Effect</a></li>  <li><a href="#">Opacity Mode</a></li>  <li><a href="#">Drop Shadow</a></li>  <li><a href="#">Semitransparent</a></li>  </ul>  </li>  <li><a href="#">Navigation</a></li>  <li><a href="#">HTML/CSS</a></li>  <li><a href="#">Help</a></li>  </ul> |

The difference between this list from any other - the identifier [ id="**jsddm**" ].  
Styles. Align elements of the first level in a row. Setup for all a-tags display:block style. Hide all inner unordered lists (layers). Decorate everything else.

|  |
| --- |
| #jsddm  { margin: 0;  padding: 0}    #jsddm li  { float: left;  list-style: none;  font: 12px Tahoma, Arial}  #jsddm li a  { display: block;  background: #20548E;  padding: 5px 12px;  text-decoration: none;  border-right: 1px solid white;  width: 70px;  color: #EAFFED;  white-space: nowrap}  #jsddm li a:hover  { background: #1A4473}    #jsddm li ul  { margin: 0;  padding: 0;  position: absolute;  visibility: hidden;  border-top: 1px solid white}    #jsddm li ul li  { float: none;  display: inline}    #jsddm li ul li a  { width: auto;  background: #9F1B1B}    #jsddm li ul li a:hover  { background: #7F1616} |

So, the script. Bind event handlers with a first-level li-nodes. If mouse over the button - close old layer (if it visible) and open new. When mouse out from button - turn on the timer to 500 milliseconds to close all layers. That's all :)  
  
Use the following function in head tag inside  
<script type="text/javascript">**Script or function goes here**</script>

|  |
| --- |
| var timeout = 500;  var closetimer = 0;  var ddmenuitem = 0;  function jsddm\_open()  { jsddm\_canceltimer();  jsddm\_close();  ddmenuitem = $(this).find('ul').css('visibility', 'visible');}  function jsddm\_close()  { if(ddmenuitem) ddmenuitem.css('visibility', 'hidden');}  function jsddm\_timer()  { closetimer = window.setTimeout(jsddm\_close, timeout);}  function jsddm\_canceltimer()  { if(closetimer)  { window.clearTimeout(closetimer);  closetimer = null;}}  $(document).ready(function()  { $('#jsddm > li').bind('mouseover', jsddm\_open)  $('#jsddm > li').bind('mouseout', jsddm\_timer)});  document.onclick = jsddm\_close; |

## **DL, DT, DD**

**How and when should it be used?**  
Definition lists, created using the DL element, generally consists of a series of term/definition pairs (although definition lists may have other applications). Thus, when advertising a product, one might use a definition list:  
Definition lists consist of two parts: a **term** and a **description**. To mark up a definition list, you need three HTML elements; a container <dl>, a definition term <dt>, and a definition description <dd>  
 **Layout of DL, DT & DD  
Example**:

|  |  |  |
| --- | --- | --- |
| **Advance CSS** | **Description** | **Hours** |
| **DL, DT, DD** | The <dd> tag is used to describe an item in a definition list.  The <dd> tag is used in conjunction with <dl> (defines the definition list) and <dt> (defines the item in the list).  Inside a <dd> tag you can put paragraphs, line breaks, images, links, lists, etc. |  |
| **Example** | |  |  | | --- | --- | | **Code** | **Result in Browser:** | | <dl>   <dt>Coffee</dt>     <dd>- black hot drink</dd>   <dt>Milk</dt>     <dd>- white cold drink</dd> </dl> | Coffee  - black hot drink  Milk  - white cold drink | |  |
|  |  |  |
| **Assignment** | |  |  | | --- | --- | | **Question** | **Result should be:** | | Use "DL, DT, DD " Tag for this  CSS - Cascading Style Sheets  HTML - Hypertext Markup Language  JS - Java Script | CSS  - Cascading Style Sheets  HTML  - Hypertext Markup Language JS  - Java Script | |  |

## **Margins & Paddings**

**Difference between margin & padding**Margins and padding can be confusing to the novice Web designer. After all, in some ways, they seem like the same thing: white space around an image or object.

Padding is the space inside the border between the border and the actual image or cell contents. In the image, the padding is the yellow area around the contents. Note that padding goes completely around the contents: there is padding on the top, bottom, right and left sides. **Margin**  
The margin clears an area around an element (outside the border). The margin does not have a background color, and is completely transparent.

The top, right, bottom, and left margin can be changed independently using separate properties. A shorthand margin property can also be used, to change all margins at once.  
  
**Margin - Shorthand property**The margin property can have from one to four values.  
**margin:25px 50px 75px 100px;**top margin is 25px  
right margin is 50px  
bottom margin is 75px  
left margin is 100px

**margin:25px 50px 75px;**top margin is 25px  
right and left margins are 50px  
bottom margin is 75px  
  
**margin:25px 50px;**top and bottom margins are 25px  
right and left margins are 50px  
  
**margin:25px;**all four margins are 25px

|  |  |  |
| --- | --- | --- |
| **Advance CSS** | **Description** | **Hours** |
| **Margin** | The CSS margin properties define the space around elements. |  |
| **Example** | |  |  | | --- | --- | | **Code** | **Result in Browser:** | | margin-top:10px; margin-bottom:10px; margin-right:10px; margin-left:10px; |  | |  |
|  |  |  |
| **Assignment** | |  |  | | --- | --- | | **Question** | **Result should be:** | | Take 3 boxes which have 1 main box. Main box width: 720px  3 boxes: 200px/box and place the text inside each box with use “padding” property.  It should be 15px padding inside in edge of each box.  \*Make sure you have 720px total width only. |  | |  |

**Padding**  
The padding clears an area around the content (inside the border) of an element. The padding is affected by the background color of the element.

The top, right, bottom, and left padding can be changed independently using separate properties. A shorthand padding property can also be used, to change all paddings at once.   
  
**Padding - Shorthand property**The padding property can have from one to four values.  
**padding:25px 50px 75px 100px;**  
top padding is 25px  
right padding is 50px  
bottom padding is 75px  
left padding is 100px  
  
**padding:25px 50px 75px;**top padding is 25px  
right and left paddings are 50px  
bottom padding is 75px  
  
**padding:25px 50px;**top and bottom paddings are 25px  
right and left paddings are 50px  
  
**padding:25px;**all four paddings are 25px

|  |  |  |
| --- | --- | --- |
| **Advance CSS** | **Description** | **Hours** |
| **Padding** | The CSS padding properties define the space between the element border and the element content. |  |
| **Example** | |  |  | | --- | --- | | **Code** | **Result in Browser:** | | padding-top:10px; padding-bottom:10px; padding-right:10px; padding-left:10px; | Xxxxx xxxxxxx  Xxxxx xxxxxxx  Xxxxx xxxxxxx | |  |
|  |  |  |
| **Assignment** | |  |  | | --- | --- | | **Question** | **Result should be:** | | Take 3 boxes which have 1 main box. Main box width: 720px  3 boxes: 200px/box and place the text inside each box with use “padding” property.  It should be 15px padding inside in edge of each box.  \*Make sure you have 720px total width only. | Xxxxx xxxxxxx  Xxxxx xxxxxxx  Xxxxx xxxxxxx | |  |

**CSS optimization for margin & padding**CSS is very important component for design and layout of web pages. These includes all the syles like colors, size, position, backgrounds, fonts etc. to enrich the look and feel of you pages, but it is very easy for CSS to grow to a huge size if you are not following proper guidelines to optimize this. By doing so you are not only making you css cleaner but also decreasing the response size of file.  
  
Here are some tip for optimizing the CSS for margin & padding: **For Margin**margin-top: 8px;  
margin-right: 4px;  
margin-bottom: 8px;  
margin-left: 4px;  
  
You can use optimize property for margin  
margin: 8px 4px 8px 4px; OR margin: 8px 4px;

**For Padding**padding-top: 8px;  
padding -right: 4px;  
padding -bottom: 8px;  
padding -left: 4px;  
  
You can use optimize property for margin  
padding: 8px 4px 8px 4px; OR padding: 8px 4px;

## Borders

How to use border?

The CSS border properties allow you to specify the style and color of an element's border.

border-style values:

|  |
| --- |
|  |

**You can use border directly in html to this way**

|  |  |  |
| --- | --- | --- |
| **Advance CSS** | **Description** | **Hours** |
| **Example** | |  |  | | --- | --- | | **Code** | **Result in Browser:** | | <table border="1"> <tr> <th>Header 1</th> <th>Header 2</th> </tr> <tr> <td>row 1, cell 1</td> <td>row 1, cell 2</td> </tr> <tr> <td>row 2, cell 1</td> <td>row 2, cell 2</td> </tr> </table> | |  |  | | --- | --- | | **Header 1** | **Header 2** | | row 1, cell 1 | row 1, cell 2 | | row 2, cell 1 | row 2, cell 2 | | |  |

**Also you can use border by CSS to this way**

|  |  |  |
| --- | --- | --- |
| **Advance CSS** | **Description** | **Hours** |
| **Example** | |  |  | | --- | --- | | **Code** | **Result in Browser:** | | <table class="red\_border"> <tr> <th>Header 1</th> <th>Header 2</th> </tr> <tr> <td>row 1, cell 1</td> <td>row 1, cell 2</td> </tr> <tr> <td>row 2, cell 1</td> <td>row 2, cell 2</td> </tr> </table> | |  |  | | --- | --- | | Header 1 | Header 2 | | row 1, cell 1 | row 1, cell 2 | | row 2, cell 1 | row 2, cell 2 | | |  |
|  |  |  |
| **CSS** | **.** red\_border{border:1px solid #ff0000;} |  |

## Display Properties

**Property Values**

|  |  |
| --- | --- |
| **Value** | **Description** |
| none | The element will generate no box at all |
| block | The element will generate a block box (a line break before and after the element) |
| inline | The element will generate an inline box (no line break before or after the element). This is default |
| inline-block | The element will generate a block box, laid out as an inline box |
| inline-table | The element will generate an inline box (like <table>, with no line break before or after) |
| list-item | The element will generate a block box, and an inline box for the list marker |
| run-in | The element will generate a block or inline box, depending on context |
| table | The element will behave like a table (like <table>, with a line break before and after) |
| table-caption | The element will behave like a table caption (like <caption>) |
| table-cell | The element will behave like a table cell |
| table-column | The element will behave like a table column |
| table-column-group | The element will behave like a table column group (like <colgroup>) |
| table-footer-group | The element will behave like a table footer row group |
| table-header-group | The element will behave like a table header row group |
| table-row | The element will behave like a table row |
| table-row-group | The element will behave like a table row group |
| inherit | Specifies that the value of the display property should be inherited from the parent element |

**Difference between inline-block, block & none display properties  
block:** Takes up the full width available, with a new line before and after. All blocks show on separate line (display:block;)  
**inline-block:** Elements, objects & texts width auto show in block and all block show in a single line (display: inline-block;)  **none:** Element will not be displayed (display:none;)  
  
IE 7 not supported to inline-block property of Display.  
All browser support to block and none property of Display.  
Major difference between inline-block, block & none display properties that is inline-block, block showing elements but none property nor show element.

**Assignments**

| **Session** | **Description** | **Result should be like** |
| --- | --- | --- |
| inline-block | You have some text  “Back, Like and Submit” we need border and background. Use display “inline-block” property | Back  I Like  Submit |
| none | You have 3 boxes. When you implement that you look like this  We want hide red box but remaining both boxes placement should be same.  How to use this property?  Please implement. |  |
| block | You have “Join Us” link  We need border and background, but width and hit area should be 200px. | Join Us |
| list-item | You have some words  USA, UK, India, Egypt, Japan, China  Use “list-item” property How to use this property?  Please implement. | * USA * UK * India * Egypt * Japan * China |
| Inline | You have some words list   * USA * UK * India * Egypt * Japan * China   Use “inline” property How to use this property?  Please implement. | USA UK India Egypt Japan China |

## Positioning

* Position types
* Use of Position
* How to built layer in html using positions
* Z-index (Overlapping elements)

The CSS positioning properties allow you to position an element. It can also place an element behind another, and specify what should happen when an element's content is too big.  
  
Elements can be positioned using the top, bottom, left, and right properties. However, these properties will not work unless the position property is set first. They also work differently depending on the positioning method.  
  
**There are four different positioning methods.  
Static Positioning**HTML elements are positioned static by default. A static positioned element is always positioned according to the normal flow of the page.Static positioned elements are not affected by the top, bottom, left, and right properties.  
  
**Fixed Positioning**An element with fixed position is positioned relative to the browser window.  
It will not move even if the window is scrolled:

|  |
| --- |
| **Assignment:** |
| <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">  <html>  <head>  <style type="text/css">  p.pos\_fixed  {  position:fixed;  top:30px;  right:5px;  color:#ff0000;  }  </style>  </head>  <body>  <p class="pos\_fixed">fixed content</p>  <p><b>Note:</b> IE7 and IE8 supports the fixed value only if a  !DOCTYPE is specified.</p>  <p>Some text</p><p>Some text</p><p>Some text</p><p>Some text</p><p>Some text</p><p>Some text</p><p>Some text</p><p>Some text</p><p>Some text</p><p>Some text</p><p>Some text</p><p>Some text</p><p>Some text</p><p>Some text</p><p>Some text</p><p>Some text</p>  </body>  </html> |
| **Result should as below** |
|  |

Note: IE7 and IE8 support the fixed value only if a !DOCTYPE is specified.  
Fixed positioned elements are removed from the normal flow. The document and other elements behave like the fixed positioned element does not exist.  
  
Fixed positioned elements can overlap other elements.

**Relative Positioning**A relative positioned element is positioned relative to its normal position.

|  |
| --- |
| **Assignment** |
| <html>  <head>  <style type="text/css">  h2.pos\_left  {  position:relative;  left:-20px;  }  h2.pos\_right  {  position:relative;  left:20px;  }  </style>  </head>  <body>  <h2>This is a heading with no position</h2>  <h2 class="pos\_left">This heading is moved left according to its normal position</h2>  <h2 class="pos\_right">This heading is moved right according to its normal position</h2>  <p>Relative positioning moves an element RELATIVE to its original position.</p>  <p>The style "left:-20px" subtracts 20 pixels from the element's original left position.</p>  <p>The style "left:20px" adds 20 pixels to the element's original left position.</p>  </body>  </html> |
| **Result should be as below** |
|  |

The content of relatively positioned elements can be moved and overlap other elements, but the reserved space for the element is still preserved in the normal flow.  
  
**Absolute Positioning**An absolute position element is positioned relative to the first parent element that has a position other than static. If no such element is found, the containing block is <html>:

|  |
| --- |
| **Assignment** |
| <html>  <head>  <style type="text/css">  h2  {  position:absolute;  left:100px;  top:150px;  }  </style>  </head>  <body>  <h2>This is a heading with an absolute position</h2>  <p>With absolute positioning, an element can be placed anywhere on a page. The heading below is placed 100px from the left of the page and 150px from the top of the page.</p>  </body>  </html> |
| **Result should be as below** |
|  |

Absolutely positioned elements are removed from the normal flow. The document and other elements behave like the absolutely positioned element does not exist. Absolutely positioned elements can overlap other elements.  
  
**Overlapping Elements**When elements are positioned outside the normal flow, they can overlap other elements.  
The z-index property specifies the stack order of an element (which element should be placed in front of, or behind, the others).  
  
An element can have a positive or negative stack order:

|  |
| --- |
| **Assignment** |
| <html> <head> <style type="text/css"> img { position:absolute; left:0px; top:0px; z-index:-1; } </style> </head> <body> <h1>This is a heading</h1> <img src="w3css.gif" width="100" height="140" /> <p>Because the image has a z-index of -1, it will be placed behind the text.</p> </body> </html> |
| **Result should be as below** |
|  |

An element with greater stack order is always in front of an element with a lower stack order.  
  
Note: If two positioned elements overlap, without a z-index specified, the element positioned last in the HTML code will be shown on top.

**All CSS Positioning Properties**The number in the "CSS" column indicates in which CSS version the property is defined (CSS1 or CSS2).

|  |  |  |  |
| --- | --- | --- | --- |
| **Property** | **Description** | **Values** | **CSS** |
| bottom | Sets the bottom margin edge for a positioned box | auto *length %* inherit | 2 |
| clip | Clips an absolutely positioned element | *shape* auto inherit | 2 |
| cursor | Specifies the type of cursor to be displayed | *url* auto crosshair default pointer move e-resize ne-resize nw-resize n-resize se-resize sw-resize s-resize w-resize text wait help | 2 |
| left | Sets the left margin edge for a positioned box | auto *length %* inherit | 2 |
| overflow | Specifies what happens if content overflows an element's box | auto hidden scroll visible inherit | 2 |
| position | Specifies the type of positioning for an element | absolute fixed relative static inherit | 2 |
| right | Sets the right margin edge for a positioned box | auto *length %* inherit | 2 |
| top | Sets the top margin edge for a positioned box | auto *length %* inherit | 2 |
| z-index | Sets the stack order of an element | *number* auto inherit | 2 |

## Background

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

• 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 defined in the body selector:

|  |
| --- |
| **Assignment:** |
| <html>  <head>  <style type="text/css">  body { background-color:#b0c4de; }  </style>  </head>  <body>  <h1>My CSS web page!</h1>  <p>Hello world! This is a W3Schools.com example.</p>  </body>  </html> |
| **Result should be as below:** |
|  |

**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"

Look at CSS Color Values for a complete list of possible color values.  
In the example below, the h1, p, and div elements have different background colors:

|  |
| --- |
| **Assignment:** |
| <html>  <head>  <style type="text/css"> h1 { background-color:#6495ed; }  p { background-color:#e0ffff; }  div{ background-color:#b0c4de; }  </style>  </head  <body>  <h1>CSS background-color example!</h1> <div> This is a text inside a div element.  <p>This paragraph has it's own background color.</p>  We are still in the div element. </div> </body>  </html> |
| **Result should be as below:** |
|  |

**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');}

|  |
| --- |
| **Assignment:** |
| <html>  <head>  <style type="text/css">  body {background-image:url('paper.gif');}  </style>  </head>  <body>  <h1>Hello World!</h1>  </body>  </html> |
| **Result should be as below:** |
|  |

**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:   
  
**Example:**

body

{

background-image:url('gradient2.png');

}

|  |
| --- |
| **Assignment:** |
| <html>  <head>  <style type="text/css">  body  {  background-image:url('gradient2.png');  }  </style>  </head>  <body>  <h1>Hello World!</h1>  </body>  </html> |
| **Result should be as below** |
|  |

**If the image is repeated only horizontally (repeat-x), the background will look better:**body  
{ background-image:url('gradient2.png'); background-repeat:repeat-x; }

|  |
| --- |
| **Assignment:** |
| <html>  <head>  <style type="text/css">  Body { background-image:url('gradient2.png'); background-repeat:repeat-x; }  </style>  </head>  <body> <h1>Hello World!</h1> </body> </html> |
| **Result should be as below:** |
|  |

**If the image is repeated vertically (repeat-y)**

|  |
| --- |
| **Assignment:** |
| <html>  <head>  <style type="text/css">  body  { background-image:url('gradient3.png'); background-repeat:repeat-y; }  </style>  </head> <body>  <h1>Hello World!</h1>  </body> </html> |
| **Result should be as below** |
|  |

**Background Image - Set position and no-repeat**RemarkWhen using a background image, use an image that does not disturb the text.  
Showing the image only once is specified by the background-repeat property:

|  |
| --- |
| **Assignment:** |
| <html> <head> <style type="text/css">  body { background-image:url('img\_tree.png'); background-repeat:no-repeat; }  </style> </head> <body>  <h1>Hello World!</h1> <p>W3Schools background image example.</p> <p>The background image is only showing once, but it is disturbing the reader!</p>  </body> </html> |
| **Result should be as below:** |
|  |

In the example above, the background image is shown in the same place as the text. We want to change the position of the image, so that it does not disturb the text too much.

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

|  |
| --- |
| **Assignment:** |
| <html> <head> <style type="text/css">  body { background-image:url('img\_tree.png'); background-repeat:no-repeat; background-position:right top; margin-right:200px; } </style> </head>  <body> <h1>Hello World!</h1> <p>W3Schools background no-repeat, set postion example.</p> <p>Now the background image is only show once, and positioned away from the text.</p> <p>In this example we have also added a margin on the right side, so the background image will never disturb the text.</p> </body> </html> |
| **Result should be as below:** |
|  |

**Background - Shorthand propert**  
To shorten the code, it is also possible to specify all the properties in one single property. This is called a shorthand property.  
The shorthand property for background is simply "background":  
  
**Example:**body {background:#ffffff url('img\_tree.png') no-repeat right top;}  
When using the shorthand property the order of the property values are:

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

**All CSS Background Properties**

|  |
| --- |
|  |

## Nested CSS

**How to build nesting css**To minimize the code, you can group selectors.  
Separate each selector with a comma.  
In the example below we have grouped the selectors from the code above:  
h1,h2,p { color:green; }  
  
**Nesting Selectors**It is possible to apply a style for a selector within a selector.  
In the example below, one style is specified for all p elements, one style is specified for all elements with class="marked", and a third style is specified only for p elements with class="marked":

p { color:blue; text-align:center; }  
.marked { background-color:red; }  
.marked p { color:white; }  
  
**Follow of html structure to use nested css**

|  |
| --- |
| **Assignment:** |
| <html> <head> <style type="text/css"> h1,h2,p { color:green; } </style> </head <body> <h1>Hello World!</h1> <h2>Smaller heading!</h2> <p>This is a paragraph.</p> </body> </html> |
| **Result should be as below** |
|  |

|  |
| --- |
| **Assignment:** |
| <html> <head> <style type="text/css"> p { color:blue; text-align:center; } .marked { background-color:red;} .marked p { color:white;} </style> </head>  <body> <p>This is a blue, center-aligned paragraph.</p> <div class="marked"> <p>This p element should not be blue.</p> </div> <p>p elements inside a "marked" classed element keeps the alignment style, but has a different text color.</p> </body> </html> |
| **Result should be as below** |
|  |

## Firebug

* Usage of Firebug
* Edit css properties using firebug
* Try to fix css issues using firebug

**Usage of Firebug**Firebug integrates with Firefox to put a wealth of development tools at your fingertips while you browse. You can edit, debug, and monitor CSS, HTML, and JavaScript live in any web page...  
  
The most popular and powerful web development tool

* Inspect HTML and modify style and layout in real-time
* Use the most advanced JavaScript debugger available for any browser
* Accurately analyze network usage and performance
* Extend Firebug and add features to make Firebug even more powerful
* Get the information you need to get it done with Firebug.

**One Click to Start Firebug**After you install Firebug you will see a gray 'Firebug' in the Firefox status bar (along the bottom of the window). Click on the Bug to start. Firebug will open in the bottom of your web page.

|  |
| --- |
|  |

**Instant HTML, CSS, and DOM Views**As soon as you open Firebug you can examine the Web page markup, styling, and object model. Three tabs -- **HTML**, **CSS**, and **DOM** -- activate as soon as you open Firebug. If you are a HTML author or CSS designer, this may be the entire Firebug you need. As long as you don't enable other panels, the overhead caused by using the Firebug should be small: by default Firebug keeps all the expensive features off.

|  |
| --- |
|  |

**Enabling Panels**To work on Javascript code or to study the network action on a site, you need to enable one or more Firebug panels. If you select the Console, Script, or Net panels, you will see the tab is grey and the panel says "disabled". Each panel tab has a small menu control for enabling the panel. All the panels can be enabled or disabled using the context menu (right click) on the Firebug status bar icon.

|  |
| --- |
|  |

**Firebug For HTML Authors and CSS Designers**If your primary work in Firebug is with HTML or CSS, leave all the active-listener panels Disabled. Firebug will have minimal overhead when you are not using it. You probably want enable the Console to look for errors. Enable one site at a time or enable the Console (only) for all sites, then selectively disable it for sites like your email

|  |
| --- |
|  |

**Firebug for Javascript Developers**For Javascript you'll want the Console and Script panels enabled. You may was well run with these panels fully enabled.

|  |
| --- |
|  |

**Firebug for Site Engineers**If your primary job is monitoring load times and looking out for 404s, activate the Net panel and perhaps the Console.

|  |
| --- |
|  |

If your primary job is monitoring load times and looking out for 404s, activate the Net panel and   
perhaps the Console.  
  
**Edit css properties using firebug  
  
Edit HTML**

|  |
| --- |
|  |

**HTML Layout**

|  |
| --- |
|  |

**Edit CSS**

|  |
| --- |
|  |

**Console firebug**

|  |
| --- |
|  |

**Document Object Mode**

|  |
| --- |
|  |

**Try to fix css issues using firebug**

## Training Report

**Instructor Name: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Trainee Name: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Date: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Time: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

|  |  |
| --- | --- |
| **Session No/Name** | **Conducted (Yes/No)** |
| Session 1 |  |
| Session 2 |  |
| Session 3 |  |
| Session 4 - Assessment |  |

**Assessment Report:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Sr. no** | **Desired Output** | **Code** | **Coding Standards** | **Marks** |
| 1 |  |  |  |  |
| 2 |  |  |  |  |
| 3 |  |  |  |  |
| 4 |  |  |  |  |
| 5 |  |  |  |  |
| 6 |  |  |  |  |
| 7 |  |  |  |  |
| 8 |  |  |  |  |
| 9 |  |  |  |  |
| 10 |  |  |  |  |
| 11 |  |  |  |  |
| 12 |  |  |  |  |
| 13 |  |  |  |  |
| 14 |  |  |  |  |
| 15 |  |  |  |  |
|  | **Total Marks:** | | |  |

Instructor Feedback:

|  |  |
| --- | --- |
| **Sign:** | **Sign:** |
| *(Trainee: )* | *(Instructor: )* |