

CSS Fundamentals

- Defines the styles and design attributes of objects
 - Almost every style ends with a semicolon
 - Can format text properties, margin/spacing, background color, image properties, etc...
 - Styles are written inside a styles block at the head of the document, which is called a stylesheet
 - Style instructions are named under selectors that define which objects in a page are affected by them
- Inside the selector are style properties, like `property: value;` that make up that entry
 - Located in either the head or linked externally
- `id`
 - Uses pound to declare; `#outercontainer {`
 - The object in your page with that id (such as `div id="outercontainer"`) will automatically acquire those style settings
 - An ID can only be used on one HTML tag/object in a document
- `class`
 - Uses period to declare; `.headline`
 - Any object that has that class will take those style properties, such as `span class="headline"`
- `tag`
 - Store your style instructions under and html tag name
 - All objects of that type (tag) in the page (img for images, div for div boxes, a for hyperlinks, etc...) will have those style properties as their new

defaults

```
//Some sample code
body {
  font-size: 12pt;
  color: black;
  background-color: white;
}

a {
  color: green;
  text-decoration: none;
}

h2 {
  font-size: 24pt;
  color: purple;
}

#headshot {
  width: 200px;
  float: right;
}

.logo{
  width: 90px;
}
```

```
font-size: 18pt; //set font size
```

```
color: red; //text color
```

```
background-color: red; //background color of an object, from the body to
a paragraph to a span tag; can be named color or hex ones
```

```
text-decoration: underline or none; //can be used to remove lines
from hyperlinks
```

```
line-height: 2 or 18px; //height of a line; plain number like 2 is line spacing,  
or measurements such as 18px
```

```
text-align: right, left, center;
```

```
background-image: url(mypic.png);
```

```
float: right, left, center; //With images and block elements that are  
less than full screen width, the float css property pushes the element to  
the right or left side of the browser and lets other elements flow around it
```

```
ul a {  
  //targets all a tags inside of ul tags  
}  
  
div .infobox {  
  targets all elements with class infobox inside of a div  
}  
  
#nav * {  
  //targetest all objects inside the nav object  
}
```

Text Properties

```
font-family: Times, serif;
```

```
font-style: normal, italic, oblique;
```

```
font-weight: normal, bold, bolder, light, 100-900
```

```
font-size: 12pt, 1em, 75%, xx-small, x-small, small, medium, smaller/larger
```

```
color: green, #FFFFFF;
```

```
font: can contain values for font-variant, font-weight, font-size/lineheight,  
font-family
```

```
line-height: normal, 2 (number of lines), 200% (double space), 24pt  
(so 12pt font in a 24pt space would be double spacing)
```

```
text-decoration: none, underline-through
```

```
text-transform: none, capitalize, uppercase, lowercase
```

```
text-align: left, right, center, justify
```

```
text-indent: 24px, 5%
```

Color/Background Properties

```
background-color: transparent, red, #99ff22
```

```
background-image: none, url(images/image.gif)
```

```
background-repeat: repeat, repeat-x, repeat-y, no-repeat
```

```
background-size: Dynamic like 100% 100% or static size like 300px 400px
```

```
background-attachment: scroll, fixed
```

```
background-position: 50%, 200px, top, center, bottom, left, right
```

```
background: can contain direct values for background color, background image,  
background repeat, background attachment, background position
```

Margin/Border Properties

```
margin: 10px, 5%, auto;
```

```
margin-top: 5px;  
margin-left: 5px;  
margin-right: 5px;  
margin-bottom: 5px;
```

```
//Used to define the space outside of an object
```

```
margin: 5px 5px 5px 5px;
```

```
//moves from top to right to bottom to left (top around clockwise)
```

```
margin: 5px 5px;
```

```
//top and bottom, right and left
```

```
padding: 10px, auto;
```

```
padding-top: 5px;
```

```
padding-left: 5px;  
padding-right: 5px;  
padding-bottom: 5px;
```

```
//Used to define the spacing inside of an object (like a box)
```

```
padding-top/bottom/left/right: 10%, auto
```

```
border-color: green, #FFFFFF
```

```
border-style: none, dotted, dashed, solid, double, groove, ridge, inset, outset
```

```
border-top/bottom/left/right: Can contain direct values for border width, style, color
```

```
border: 5px, red, none; Can contain direct values for border width, style, color
```

```
width: auto, 200px, 30%
```

```
height: auto, 200px
```

```
float: none, left, right
```

```
clear: none, left, right, both
```

```
display: block;  
margin: auto;
```

```
//center an image
```

Positioning Properties

```
position: fixed , absolute, relative
```

```
left: auto, 24px, 5%;
```

```
top: auto, 24px, 5%;
```

```
clip: auto, rect(100,200);
```

```
z-index: auto, number (1-100);
```

```
visibility: inherit, visible, hidden;
```

```
.classname: hover {  
}  
  
.classname: hover .subclassname {  
}  
  
//changes properties on hover  
  
//hover subclasses
```

```
p:first-letter {  
}  
//controls the actions of the first letter
```

```
p:first-line{  
}  
//controls the actions of the first line
```

```
opacity: 0.8 or 80%;  
//style to describe the opacity of an object, ranges from 0 to 1  
  
//controls transparency as well as how relatively bright something is (really controlling  
transparency)
```

```
border-radius: 10px;  
  
//rounds corners of objects
```

```
box-shadow: 10px 10px 10px #888888;  
//horizontal offset; positive puts shadow on the right, negative on the  
left  
//vertical offset; positive puts shadow below the box, a negative value puts the shadow ab  
ove the box  
//color of the shadow
```

```
text-shadow: 20px -12px 2px #6374AB;  
  
//horizontal offset;  
//vertical offset  
//blur-radius  
//color of the shadow
```

Div Layouts

```
width: controls how wide an object will be
```

```
height: height of an object; usually not defined, and let the content  
handle it
```

float: flags an object as a floating element, and allows stuff to flow around it

margin: controls if the item has a margin outside of it

padding: create spacing inside the box

border: controls the edge of the box; 3 values in one

border: solid 3px red;

```
div {  
  border: solid 1px black;  
}
```

clear: both;

//resets the float settings

display:block;

//turns a tag into a block that you can float or move

display: none;

//removes the object from the page, while **visibility:**hidden just hides it

position: fixed;

top: 0px; **left:** 0px; //sets it to the top of the browser and to the left

//locks the element in place; you must also set a location for a fixed image

```
//should also a margin to all the stuff on the page so it is not covered by the top bar, since the top bar is placed on top of everything
```

```
//
```

```
position: relative;  
top: 0px;  
left: 0px;
```

```
//part of the normal page layout, but allows you to shift its location with the top and left commands
```

```
//you can move something from its original location and cover other things
```

```
//you can control what is on top of what this with the z index property;  
//effectively, positioned in reference to its current position
```

```
z-index: 10;
```

```
//all relative values, z index 10 is above z index 9
```

```
min-height: 130px;  
max-height: 120px;  
min/max-width: 120px;
```

```
//allows us to set a minimum height for an object that can be exceeded
```

```
position: absolute;  
top: 40px; left: 200px;
```

```
//allows you to lock an item to the page but it moves with the rest of the page
```

```
//positioned in reference to something absolute, like what its in or the page itself
```

```
//the absolute positioning will work off of whatever the closest relatively positioned thing it is in; if there is none, it uses the webpage
```

```
//basically, if you put an absolute div in a relative div, it will position based off of the relative div
```

```
position: absolute (in relation to an object)
```

```

//place a div with position:absolute inside another div that is
position:relative;

//in this case, the inner div is laid out against the top/left/bottom/right coordinates of
the outer div

<div class="imgbox">
  <div class="caption">Dogs at play</div>
</div>

.caption {
  position:absolute;
  bottom: 10px;
  text-align: center;
  width: 300px;
}

.imgbox {
  height:300px; width:300px;
  background-color:pink;
  float:left;
  margin: 10px;
  position: relative;
}

```

```

visibility: hidden;
visibility: visible;

```

```

//hides objects, and allows you to show them with hover

```

CSS Driven Drop Down Menus

```

//start by putting all of the names of links into an unordered list
<div id="menu">
  <ul>
    <li> Webdev</li>
    <ul>
      <li>Acad275</li>
      <li>Acad276</li>
    </ul>
    <li> W3Schools</li>
    <ul>
      <li>HTML</li>

```

```
        <li>CSS</li>
      </ul>
    </ul>
    <li>Zen Garden CSS Web Site</li>
  </ul>
</div>
```

```
//What we need to do is:
```

```
// Remove all margins and paddings from the menu/div, the ULs in the list and the LIs in the list
```

```
// Remove the bullets from all list items
```

```
// Change the list items so that they stack/float against each other (horizontally).
```

```
// Tell each list item that it can "float" against each other, and set margins around each
```

```
#menu{
  padding:0;
  margin:0;
}
```

```
#menu ul{
  padding:0;
  margin:0;
}
```

```
#menu li{
  margin: 0;
  padding:0;
}
```

```
//its actually much simpler to use the * object to target all objects inside the menu
```

```
#menu * {
  padding:90px;
  margin:0px;
}
```

```
//now we need to address the positioning, bullets, and stacking of items
```

```
#menu li{
  position: relative;
  float: style;
```

```
list-style: none;
}
```

```
//now we need style instructions to:

//set a fixed width of 100px

//height of 30px

//change the anchors to BLOCK elements

//remove the underline

//set a text-align of center and line-height of 30px

//set a yellow background for the "buttons" and a purple text color

#menu li a{
  width:100px;
  height: 30px;
  display: block;
  text-decoration:none;
  background-color: purple;
  color: yellow;
  text-align: center;
  line-height: 30px;
}
```

```
//also sets a hover instruction to set the background color to a highlight color
#menu li a:hover{
  background-color: red;
}
```

```
//items are laid out across not up and down, so we need to set the level2 menus to absolute
positioning

//this will allow them to go straight down

#menu ul ul{
  position: absolute;
  top: 30px;
}
```

```

//now we must

//set the visibility of the sub menus to hidden... and add an instruction to hover of the
parents to set their visibility to visible

#menu ul ul{
  position: absolute;
  top: 30px;
  visibility: hidden;
}

#menu ul li:hover ul{
  visibility: visible;
}

```

Creating Label Column Layouts

```

//the way you can create two column layouts with divs is by putting a
span inside of it

```

```

//this shows the HTML necessary
<span class="label">Name:</span> Patrick Dent
<span class="label">Name:</span> dent@usc.edu
<span class="label">Name:</span> (213) 821-1400

```

```

.label{
  float:left;
  width:150px
}

```

```

//this is the necessary CSS to correctly create a two column layout for inputs and labels
.label{
  float:left;
  width:150px
}

```

```

//if you want to use newer HTML elements you can use the label tag that will be a standard
for setting up columns

```

```

//uses the same CSS

```

```
<label>Name:</label> Patrick Dent  
<label>Name:</label> dent@usc.edu  
<label>Name:</label> (213) 821-1400
```

```
//CSS for using the label tag and creating a two column layout for inputs and  
labels  
label{  
  float:left;  
  width:150px  
}
```

Color

```
color: red;  
  
//Named colors
```

```
color: #FFFFFF;  
  
//Hex codes
```

```
color: rgb(99,9,99)
```

```
color: rgba(99,0,99,0.5)
```

```
//adds a second factor to the color, which is defining the alpha (opacity) of the color from 0 to 1
```

```
color: hsl(298,26%,100%)
```

```
//uses hue, saturation, and lightness to drive color  
//hue is measured in degrees from the color wheel  
//saturation is percentage of how deep the color is  
//lightness is percentage of how bright the color is
```

```
color: hsla(298, 26%, 100%, 0.5);  
  
//hsl but with an alpha (opacity) value
```

Font

```
@font-face{  
  font-family: Tuffy;  
  src: url('Tuffy.ttf'),  
       url('Tuff.eot'); /*IE9*/  
}
```

//allows you to embed fonts with a stylesheet entry
//you need multiple file formats of each font because not all browsers support all fonts
//font-family names the font family, and the src tells where to find the different necessary fonts

```
p{  
  font-family: Tuffy;  
}  
  
//display fonts
```

```
link rel="stylesheet" href="http://openfontlibrary.org/face/gamaliel"  
type="text/css"/>
```

```
<div style="font-family: Gamaliel">Test paragraph that uses the Gamaliel font,  
loaded remotely from the Open Font Library.</div>>
```

//can also load external stylesheets with necessary fonts

```
@font-face {  
  font-family: 'Gamaliel';  
  src: url('/assets/fonts/gamaliel/e0c9601992c81cd08b8d449411bc0ac6/252299ad933b2acb  
62e561c828c86f75/GamalielMedium.otf')  
  format('opentype');  
  font-weight: normal;
```

```
font-style: normal;
}

//this is the stylesheet we just linked to
```

Multiple Columns Through CSS

```
.col3{
  columns: 3;
  -webkit-columns: 3;
  -moz-columns: 3;
}

//adding a column is as simple as this
//column is a shorthand for column count, but there are many different
properties
//column-count, column-fill, column-gap, column-width
```

```
.col3b{
  columns: 3;
  -webkit-columns: 3;
  -moz-columns: 3;
  column-rule: red 2px solid;
  -webkit-column-rule: red 2px solid;
  -moz-column-rule: red 2px solid;
}

//column rule parameters
//for example, this code would give us a red line in between the two columns as a result o
f the column rule instruction
```

Background Images

```
.nav2 {
  background-color: lightblue;
  background-image: url('blueback.jpg');
  background-position: center;
  background-repeat: repeat-y;
}

//some example code for background instructions
```

```
.nav2 {  
background: url('blueback.jpg') repeat-y center;  
}  
//you can compile all of the background image tags together
```

```
background-size: 400px 200px;  
  
//allows you to change the size of a background image, and prevent it from being clipped  
  
//you should scale to the size of its container, or a different size if you prefer  
  
//could also set to 100%
```

```
#field{  
width:400px; height:200px;  
background-repeat:no-repeat;  
background-image: url('moon.gif'), url('bat.gif'), url('pumpkinfield.jpg') ;  
background-size: 70px 70px, 90px 90px, 100%;  
background-position: left top, 150px 10px, center;  
}  
//to overlay images, simply create the background image tags in order
```

Transitions

- An instruction in an object that tells the browser that if one or more css properties are changed for an object (say in hover or focus), that property should transition

```
transition-property: width;  
  
//sets the property that will receive the transition
```

```
transition-duration: 2s;  
  
//sets the time that the transition will take
```

```
transition-timing-function: linear, ease, ease-in, ease-out, ease-in-out;

//linear: same speed from start to finish

//ease: slow start, then fast, then end slowly

//ease-in: slow start

//ease-out: slow end

//ease-in-out: slow start and slow end

//sets the speed curve for how a transition will take effect
```

```
transition-delay: 0s;

//sets the time when the transition effect will start
```

```
transition: width 2s linear 0s, height 2s;

//simplification of transition elements

//at minimum, you must define a property and a delay

//can also separate multiple transitions in one line with commas

//can also use all for all transitions

//if you want to animate in a visibility change, you have to use opacity with CSS
```