
CSS3

Introduction:

- (a) Here are some websites that can be referenced for this guide:
- The video I followed along to get this information: 'https://www.youtube.com/watch?v=CUxH_rWSI1k'
 - Hexadecimal color codes: '<http://goo.gl/uoGQNt>'
 - CSS Validation Service: '<http://goo.gl/qWQV0f>'

Importing:

- (a) The CSS provided is applicable for the following HTML:

```
<!DOCTYPE HTML>
<html lang="en"> <!--English Language-->
<head>
    <meta charset="UTF-8">
    <title>CSS Tutorial</title><!--Titled 'CSS Tutorial', the title can be found
    on the tab at the top of the page-->
    <link rel="stylesheet" type="text/css" href="mainstyle.css"><!--linked to a
    'stylesheet' of type 'css' called 'mainstyle.css' (which we made), the 'link'
    tag will help us connect our css to our html-->
</head>
<body>
    <h1>CSS Tutorial</h1> <!--Biggest header size-->
    <p> A Css3 Tutorial that will ...</p>
    <h2>Let's Get Started</h2><!--Second biggest header size-->
    <div>
        <h3>My Favorite Sites</h3><!--Third biggest header size-->
        <ul><!--Unordered list, aka bullets-->
            <li><!--One bullet--><a class="sitelink" href="google.com">Google</a>
            <!--'className' is 'sitelink'=link, link directs to 'google.com'-->
            </li>
            <li><a class="sitelink" href="reddit.com">Reddit</a>
            </li>
            <li><a class="sitelink" href="amazon.com">Amazon</a>
            </li>
        </ul>
    </div> <!--Divides the code out-->
    <p id="tonyquote">We wake up on a box, eat breakfast...<span id="tonyname">
    <!--paragraph with id 'tonyquote' has a span with id 'tonyname' so that we
    can target the text '- Tony Robbins'--> - Tony Robbins <span></span></p>
    <p class="sitelink">I also like <a href="youtube.com">YouTube</a></p>
    <div id="tutorials">
        <ol><!--Ordered list, aka numbered list-->
            <li>Rails</li><!--One numbered thing-->
            <li>NodeJS</li>
            <li>Android Games</li>
        </ol>
        <ul>
            <li>SASS</li>
            <li>HAML</li>
        </ul>
    </div>
    <h3>Favorite Video Games</h3>
    <p>List of my current<a href="nintendogames">favorite video games</a></p><!--
    To clarify, the 'a' tag defines a hyperlink, it needs the 'href' attribute-->
    <ul>
        <li alt="nintendo">Super Mario 3D Land</li><!--'alt' is just like any
        attribute defined as 'nintendo'-->
        <li alt="nintendo">Monster Hunter 4 Ultimate</li>
        <li alt="nintendo">Pokemon Omega Ruby</li>
        <li alt="klei entertainment">Don't Starve</li><!--here 'alt' is instead
        defined as 'klei entertainment' to differentiate the producer-->
    </ul>
</body>
</html>
```

- (b) We have a ‘heading’-style CSS file, and a ‘paragraph’-style CSS file that will be imported by our main CSS file. Here is the code to do so:

```

h1, h2 { /*Style 'h1' and 'h2'*/
    color: white; /* Text within these headers will be 'white' */
    background: black; /* Behind the text, the 'background' will be 'black' */
}

p { /* all paragraphs */
    color: blue; /* text color of 'blue' */
}

```

```

git Selection Find View Goto Tools Project Preferences
    csstut1.html • mainstyle.css • h
@import "heading.css"
@import "paragraph.css"
/* import 'heading.css' and 'paragraph.css' */

```

Selecting Elements:

- (a) We are using the html/CSS imports as from the ‘Importing’ section. We can target specific elements/tags using CSS. Here are a few of the ways we can do this:

```

@import "heading.css"
@import "paragraph.css"
/* import 'heading.css' and 'paragraph.css' */
*{ /* Everything */
    font-family: "Arial Black", Gadget, sans-serif; /* 'font-family:' is saying
        "here are the fonts that will be used in order"; by default 'Arial Black'
        will be used, if it is unavailable then we will use 'Gadget', etc...*/
}
div *{ /* all 'div' tags are targeted */
    font-family: "Comic Sans MS", cursive, sans-serif;
}
.sitelink{ /* the '.' is used to target specific classes; we are targeting a
class with id 'sitelink' */
    font-family: Georgia, serif;
}
#tonyquote{ /* '#' is used to target a specific 'id'; we are targeting the id
'tonyquote' */
    font-family: "Lucida Sans Unicode", "Lucida Grande", sans-serif;
    color: black;
}
.p.sitelink{ /* 'sitelink's that exist in 'p'aragraph tags */
    font-family: "Palatino Linotype", "Book Antiqua", Palatino, serif;
    color: black;
}
span#tonyname{ /* this is targeting the id 'tonyname' that are found in
'span's only */
    font-family: "Times New Roman", Times, serif;
    color: blue;
}

```

```

#tutorials ol li{ /* targeting the id 'tutorials' that are found in the
list-items 'li' that are found in the ordered lists 'ol' */
    color: purple;
}
#tutorials ul li{ /* same as before except we are targeting list-items 'li'
found in unordered lists 'ul' */
    color: green;
}
h3+p{ /* targets paragraphs 'p' found in headers defined as 'h3' */
    font-style: italic; /* we change the font style to 'italics' */
}
h3+p>a{ /* this is similar to before except we are targeting the link
inside of the aforementioned paragraph q */
    color:red;
}
p[class]{ /* targets all attributes of type 'class' found in paragraphs 'p' */
    background: gray;
}
p[id]{ /* same but for all paragraphs 'p' with the attribute 'id' */
    background: yellow;
}
*[alt="nintendo"]{ /* we are targeting all tags that have attributes 'alt',
but only those attributes 'alt' that are equal to 'nintendo' */
    background: orange;
}

```

Padding, Margin, Border:

- (a) There is a layout for a block of text. There is the the text or ‘content’ which has a border. The distance between the content and the border is called the ‘padding’. There is also the ‘margin’ which is the closest distance anything can be to the border.
- (b) Here is some HTML-code that will be CSS modified:

```

<!DOCTYPE HTML>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>CSS Tutorial</title>
    <link rel="stylesheet" type="text/css" href="stylesheet2.css">
</head>
<body>
    <p id="lorem"> <!--Paragraph with id 'lorem'-->
        |   Lorem ipsum dolor sit amet...
    </p>
</body>
</html>

```

- (c) Here is some CSS that changes the ‘padding’ of ‘lorem’:

```

#lorem{ /* targets the 'id' 'lorem' */
    padding 10px 20px 10px 20px; /* 'padding' called with 4 keys applies the
padding to all four sides of the text: top right bottom left; */
}
#lorem{
    padding 10px 20px 10px; /* with 3 keys we instead get:
    top right&left bottom */
}
#lorem{
    padding 10px 20px; /* with 2 keys we instead get: top&bottom left&right */
}
#lorem{ /* we can also do a long-hand version of this in the following way */
    padding-top: 10px;
    padding-bottom: 10px;
    padding-left: 20px;
    padding-right: 20px;
    /* all of these 'lorem' 'padding' modifiers complete the same action */
}

```

- (d) Here is some CSS that changes padding, margin, and border:

```

#lorem{
    padding-top: 10px;
    padding-bottom: 10px;
    padding-left: 20px;
    padding-right: 20px;

    margin: 10px; /* Defining pixel width is the same for padding/margin/border;
this is the last case where we can define: top&bottom&left&right */

    border-top: 5px solid #00308F; /* solid 5 pixel thick blue border on top */
    border-bottom: 5px dotted #00308F; /* dotted blue border on bottom */
    border-left: 5px dashed #00308F; /* dashed blue border on left */
    border-right: 5px double #00308F; /* double solid border on right */

    background: #F2f3F4 url("Repeat.png") repeat; /* the background has the
color '#F2f3F4' and the image 'Repeat.png'; 'repeat' means that the image is
repeated in the x&y-directions until the image fills (instead of stretching
the image); we could instead use 'repeat-y', 'repeat-x', or 'no-repeat' to
stretch the image in the x-direction, y-direction, or both directions
respectively (repeat in y, x, no direction) */
}

```

Font Properties:

- (a) Before we touched on ‘font-style’ by creating italics for text with “font-style: italic;”. ‘style’ is the property, but ‘italic’ is the type of ‘style’. There are many other properties and types:

```

#fsNormal{ font-style: normal; } /* style returns to normal */
#fsItalic{ font-style: italic; } /* style becomes italicized */
#fsOblique{ font-style: oblique; } /* style becomes obliquized */
#fvNormal{ font-variant: normal; } /* variant returns to normal */
#fvSmallCaps{ font-variant: small-caps; } /* uppercase letters are just bigger
uppercase letters while lowercase letters are smaller uppercase letters */
#fwNormal{ font-weight: normal; } /* returns font weight to non-bold/light */
#fwBold{ font-weight: bold; } /* weight becomes bold */
#fwBolder{ font-weight: bolder; } /* weight becomes bolder than bold */
#fwLighter{ font-weight: lighter; } /* weight becomes lighter than normal */
#fszMedium{ font-size:medium; } /* size returns to medium, this is bigger in a
tag such as 'h1' and smaller in a tag such as 'p' */
#fszXXSmall{ font-size: xx-small } /* font size becomes smaller than x-small */
#fszSmall{ font-size: small; } /* font size becomes small */
#fszLarge{ font-size: large; } /* font size becomes large */
#fszPrct{ font-size: 200%; } /* font size is specifically 2x bigger */
#fsz25pt{ font-size: 25pt; } /* specify size to '25pt'-size */
#lhNormal{ line-height: normal; } /* return 'line-height' to normal, this
essentially defines the vertical height for a line, or distance between them */
#lh25pt{ line-height: 25pt; } /* change it to 25pt font size */
#lh200prct{ line-height: 200%; } /* line doubles in height */

```

- (b) To understand font properties it is important to understand a few unit types; these consist of units that describe size of font:

Pixels (px): Unit the represents pixels on a device

Points (pt): Used in print media in which 72 points equals 1 inch

Ems (em): An em is equal to the current font size

Percent (%): The current font size is equal to 100%

Positioning:

- (a) The first thing to learn is the difference between ‘inline’ and ‘block’ elements.

- ‘Inline’ elements don’t require break ‘
’ statements:

```

a, abbr, address, area, audio, b, cite, code, del,
details, dfn, command, datalist, em, font, i, iframe,
img, input, ins, kbd, label, legend, link, mark, meter,
nav, optgroup, option, q, small, select, source, span,
strong, sub, summary, sup, tbody, td, time, var

```

ii. ‘Block’ elements require break ‘
’ statements:

```
article, aside, blockquote, body, br, button, canvas,
caption, col, colgroup, dd, div, dl, dt, embed, fieldset,
figcaption, figure, footer, form, h1 - h6, header,
hgroup, hr, li, map, object, ol, output, p, pre, section,
table, tbody, textarea, tfoot, th, thead, tr, ul, video
```

(b) Here is some HTML to use CSS on:

```
<!DOCTYPE HTML>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>CSS Tutorial</title>
  <link rel="stylesheet" type="text/css" href="stylesheet3.css">
</head>

<body>

<div id="parent-div">
  <div id="default">Default</div>
  <div id="centered">Centered</div>
  <div id="centered-text">Centered Text</div>
</div>

<!-- Demonstrate Absolute Positioning -->

<div id="top-left-pos">Top Left
  <div id="bottom-right-tl-parent">Bottom Right Parent</div>
</div>

<div id="top-right-pos">Top Right
  <div id="bottom-left-tr-parent">Top Right Parent</div>
</div>

<p id="fixed-pos">
  I'm staying right here
</p>

<p>
  Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do
  eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut
  enim ad minim veniam, quis nostrud exercitation ullamco laboris
  nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor
  in reprehenderit in voluptate velit esse cillum dolore eu fugiat
  nulla pariatur. Excepteur sint occaecat cupidatat non proident,
  sunt in culpa qui officia deserunt mollit anim id est laborum.
</p>
```

```

<p>
    Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do
    eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut
    enim ad minim veniam, quis nostrud exercitation ullamco laboris
    nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor
    in reprehenderit in voluptate velit esse cillum dolore eu fugiat
    nulla pariatur. Excepteur sint occaecat cupidatat non proident,
    sunt in culpa qui officia deserunt mollit anim id est laborum.
</p>

<p>
    Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do
    eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut
    enim ad minim veniam, quis nostrud exercitation ullamco laboris
    nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor
    in reprehenderit in voluptate velit esse cillum dolore eu fugiat
    nulla pariatur. Excepteur sint occaecat cupidatat non proident,
    sunt in culpa qui officia deserunt mollit anim id est laborum.
</p>

<div id="relative-1">Relative Layout</div>
<div id="relative-2">Relative Layout 2</div>

<p>
    Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do
    eiusmod tempor incididunt ut labore et dolore magna aliqua. 
    enim ad minim veniam, quis nostrud exercitation ullamco laboris
    nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor
    in reprehenderit in voluptate velit esse cillum dolore eu fugiat
    nulla pariatur. Excepteur sint occaecat cupidatat non proident,
    sunt in culpa qui officia deserunt mollit anim id est laborum.
</p>

<div id="crop"></div>
</body>
</html>

```

Here are a few new ways to position elements:

```

#parent-div{ /* choose element with id 'parent-div' */
    background: #B2bEb5;
    border: 0.1em solid black; /* has 'border' with thickness 0.1 elements, it
        is a 'solid' border that is 'black' */
}
#default{ /* choose element with id 'default' */
    background: #DBE9F4;
}
#centered{ /* choose element with id 'centered' */
    background: #89cff0;
    width: 50%; /* text will only use 50% of the normal width on the page */
    margin: auto; /* same space on right and left of the text */
}
#centered-text{
    text-align: center; /* centers the text similarly to 'margin: auto' */
}

```

(c) Here is some CSS that covers ‘absolute’ positioning:

```

#top-left-pos{
    background-color: #89cff0;
    border: 0.1em solid black;
    position: absolute; /* The element is positioned relative to its first
    positioned (not static) ancestor element, elements are static by default
    so 'top-left-pos' will show up in the top-left corner of the window because
    the window's position is fixed */
    width: 200px; /* 200 pixel width */
    height: 100px; /* 100 pixel height */
    z-index: 4; /* out of the page, so this means that this image will show
    over things with z-index < 4 */
} /* note that the top-left of the screen is the default position */
#bottom-right-tl-parent{
    background-color: #dbe9f4;
    position: absolute; /* since 'top-left-pos' is an ancestor element, this
    will be positioned based off of it */
    bottom: 0px; /* 0 pixels away from the bottomside of 'top-left-pos' */
    right: 0px; /* 0 pixels away from the rightside of 'top-left-pos' */
}
#top-right-pos{
    background-color: #89cff0;
    border: 0.1em solid black;
    position: absolute; /* same thing happens with this as 'top-left-pos' */
    width: 200px;
    height: 100px;
    top: 0px; /* 0 pixels away from the topside of the window */
    right: 0px; /* 0 pixels away from the rightside of the window */
} /* note that 'top-left-pos' has a 'z-index' of 4 while the default is
0, so 'top-left-pos' could be on top of 'top-right-pos' */
#bottom-left-tr-parent{
    background-color: #dbe9f4;
    position: absolute; /* since 'top-right-pos' is an ancestor element, this
    will be positioned based off of it */
    bottom: 0px; /* 0 pixels away from the bottomside of 'top-right-pos' */
    left: 0px; /* 0 pixels away from the leftside of 'top-right-pos' */
}

```

- (d) Here is some CSS that covers ‘fixed’ positioning:

```

#fixed-pos{
    position: fixed; /* a 'fixed' position implies that the '<div>' 
    'fixed-pos' will always show up on the window, even when scrolling
    through the page; the position is relative to the window */
    top: 0;
    left: 0;
    width: 100%; /* 'fixed' position is relative to the window size, so
    this implies that '<div>' 'fixed-pos' has the width of the window*/
    background: #dbe9f4;
    border: 0.1em solid black;
    margin: 0; /* no margin */
}

```

- (e) Here is some CSS that covers ‘relative’ positioning:

```

#relative-1{
    background-color: #dbe9f4;
    border: 0.1em solid black;
    height: 100px; /* 100 pixels tall */
}
#relative-2{
    position: relative; /* this means that 'relative-2' will be positioned
    relative to where it should have been */
    left: 20px; /* 20 pixels from the leftside */
    top: 30px; /* 30 pixels from the topside */
    width: 300px;
    background-color: yellow;
    border: 0.1em solid black;
}
img[src="littleBrain.png"]{ /* targeting an 'img' (image) with 'src'
    'littleBrain.png' */
    float: left; /* float positions relative to the container;
    'littleBrain.png' is inside of a paragraph, so it gets positioned on
    the 'left' side of the paragraph */
    padding: 10px; /* 10 pixels between the element and the border */
}

```

Clipping:

- (a) We're going to be clipping a few brain images found in the following HTML:

```
<!DOCTYPE HTML>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>CSS Tutorial</title>

  <link rel="stylesheet" type="text/css" href="stylesheet4.css">
</head>

<body>

  <div class="little-brain-clip">
    
  </div>

  <div class="little-brain-small overflow-visible">
    
  </div>

  <div class="little-brain-small overflow-hidden">
    
  </div>

  <div class="little-brain-small overflow-scroll">
    
  </div>

</body>
</html>
```

- (b) In the following image there is CSS that will clip images as well as :

```
div.little-brain-clip{ /* '.' targets the class name that follow it */
  position:absolute; /* absolute position relative to webpage */
  width: 100px; /* <div> is 100 pixels wide */
  clip:rect(0px, 50px, 50px, 0px); /* CSS functions go in the clockwise
direction (top, right, bottom, left); this function clips based off of
the sides of the image: the top-side is 0 pixels away from the top of
the image, the right-side is 0 pixels away from the left of the image,
the bottom-side is 50 pixels from the top of the image, and the left-
side is 0 pixels from the left of the image */
}

div.little-brain-small{ /* note that there are 3 images with the class name
'little-brain-small' */
  position:absolute;
  width:100px; /* <div> is 100 pixels wide */
  height:50px; /* <div> is 50 pixels wide */
}

div.overflow-visible{
  top: 100px; /* div is 100 pixels away from the nearest positioned
ancestor which is the webpage, the former <div> size is 100 pixels,
so this will be right below the image above it */
  overflow:visible; /* anything that doesn't fit within 'width:100px'
and 'height:50px' will still be visible */
}

div.overflow-hidden{
  top:200px;
  overflow:hidden; /* anything that doesn't fit within 'width:100px'
and 'height:50px' will be completely hidden */
}

div.overflow-scroll{
  top:300px;
  overflow:scroll; /* anything that doesn't fit within 'width:100px'
and 'height:50px' won't be visible but can be scrolled through
to be seen */
} /* NOTE: This works because we have altered the size of the '<div>', but
not the size of the image. Since the image is not affected, it may or may
not fit within the '<div>' causing 'overflow' to occur */
```

HTML5 Layouts:

- (a) The following HTML contains a lot of tags and ids that will be experimented with. This

HTML will be used for this section as well as every section up to ‘Buttons’:

```
<!DOCTYPE HTML>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>CSS Tutorial</title>

  <link rel="stylesheet" type="text/css" href="stylesheet5.css">
</head>

<body>
<!-- A div wrapper is often used to define the size and position
of the page divs and spans should be used only for styling and
not structural reasons -->
<div id="wrapper">

  <header>
    <a href="http://localhost/css/cssstut4.html">
      
    </a>
  </header>

  <nav id="horzNav">
    <p>
      <a href="google.com">Google</a>
      <a href="youtube.com">YouTube</a>
      <a href="yahoo.com">Yahoo</a>
    </p>
  </nav>

  <!-- main surrounds the main content of the document
You can only have one main -->
<main id="content">

  <!-- We surround blocks of content in a document with section tags -->
  <section>
    <!-- Section elements should have a heading specific to it -->
    <h1>Interesting</h1>

    <!-- An article is a self contained block of content that is complete on
its own with a heading and content -->
    <article>

      <h2>Body Language</h2>
      <p id="para-1">In this article, I'll cover <span>Arm and Hand Body
Language Analysis</span>. I know doubt know someone that reads this ar
will say to themselves, <span id="common-quote">"I just feel more comf
crossed in front of me."</span> Body Language Analysis is used because
has been proven to work for many people.
    </p>
  </article>
</section>
</main>
</div>
```

```

<p id="para-2">
    If you feel comfortable with a specific body position it is because
    you are feeling a certain way. Ask yourself, when would you feel
    comfortable with your arms crossed while enjoying yourself with a
    loved one? Have you ever seen anyone cross their arms in excitement?
    Just a few things to think about, as you read this article.
</p>
<p id="para-3">
    This article was created because one of my readers asked for it. If
    you have any <span>questions</span> leave them in the comment section
    I will gladly answer any questions that I can.
</p>
<p id="para-4">
    When shaking someones hand, if your hand is on top you are trying
    to dominate and vice versa.
</p>
<p id="para-5">
    An equal handshake, in which both people shake in a vertical manor,
    with equal pressure, is the best way to generate good rapport.
</p>

<dl id="buttons">
    <dt class="site-button"><a href="google.com">Google</a></dt>
    <dt class="site-button"><a href="youtube.com">YouTube</a></dt>
    <dt class="site-button"><a href="yahoo.com">Yahoo</a></dt>
</dl>

<dl id="kids-pics">
    <dt><a>
        </a></dt>
    <dt><a>
        </a></dt>
    <dt><a>
        </a></dt>
</dl>

</article>
</section>
</main>

<div id="sidebar">
    <ol id="sidebar-list">
        <li>Random 1</li>
        <li>Random 2</li>
        <li>Random 3</li>
    </ol>

    <ol id="sidebar-list-2">
        <li>Random 1</li>
        <li>Random 2</li>
        <li>Random 3</li>
    </ol>

    <h3>Favorite Movies</h3>
    <h4>District 9</h4>
    <h4>The Matrix</h4>
    <h4>The Cabin in the Woods</h4>

    <h3>Others</h3>
    <h4>Minority Report</h4>
    <h4>Lord of the Rings</h4>
    <h4>Fargo</h4>
</div>

<div id="filler"></div>

<footer>
    <!-- copyright is normally placed in a small tag -->
    <small>copyright 2014</small>
</footer>
</div> <!-- END OF WRAPPER -->
</body>
</html>

```

(b) Here are a few ways to combine the HTML with CSS:

```

a:hover{ color: blue; } /* Anytime a link is hovered over */
a:active{ color: red; } /* Anytime a link is the active link */
a: visited{ color: #989898; } /* If the link has been visited */
body{ /* Targeting the body tag (only one body tag) */
    background: #5d8aa8;
}
div#wrapper{ /* div tag with id 'wrapper' */
    margin: 0 auto; /* centers the text between the margin */
    background: white;
    width: 800px;
    height: 1500px;
}
header{ /* tag 'header' that contains the image */
    display: block; /* displays like a block rather than inline, this means
        that the header is separate, but since it is just an image, this doesn't
        affect the appearance at all */
    padding-bottom: 10px; /* no padding except for the bottom */
}
#websiteLogo{ /* targets id 'websiteLogo', the image */
    padding: 5px 0px 0px 5px; /* padding on the topside and leftside */
}
#horzNav{ /* targets id 'horzNav', block with urls */
    display: block;
    padding: 5px 20px 5px 0px;
    background-color: black;
}
#horzNav a{ /* targets links in 'horzNav' */
    text-decoration: none; /* no underline etc, looks better */
    padding: 0px 0px 0px 20px;
    color: white;
}
#horzNav a:hover{ /* occurs when hovering the links */
    color: #89cfff;
}
main{
    padding: 10px 10px 20px 20px;
    float: left; /* text stays on left side of main */
    width: 600px; /* main, which text is contained in has 600 pixel width */
}
#sidebar{ /* id 'sidebar' is a div tag */
    padding: 20px 10px 0px 10px;
    float: right; /* will be right of main */
    width: 150px; /* column compared to main */
    height: auto; /* height according to other things, main in this case */
    background: #DBE9F4;
}
h2{
    font-size: 1em;
}

```

Alignment:

```

#para-1{
    text-align: left; /* already aligned left so no effect */
    word-spacing: .25em; /* space between words */
}
#para-2{
    text-align: right; /* aligns right */
}
#para-3{
    text-align: center; /* center align */
}
#para-4{
    text-indent: 2em; /* first word in paragraph gets indented */
}
#para-5{
    letter-spacing: .5em; /* big spacing between every letter */
}
footer{
    clear: both; /* nothing shows up next to footer (nothing
        can be on side of it) */
    background: black;
    color: white; /* white text */
    padding: 10px;
}

```

Lists:

```

#sidebar-list{
    list-style-type: lower-latin; /* letters to list, a, b, c, */
    list-style-position: outside; /* Text, not a,b,c, are aligned */
}
#sidebar-list-2{
    list-style-type: upper-roman; /* I, II, III, is list type*/
    list-style-position: inside; /* I, II, III, not text, are aligned */
}

```

Pseudo Classes:

```

p#para-1:before{ /* put before text in paragraph 'para-1' */
    content: "*"; /* add '*' */
}
p#para-1:after{ /* put after text in paragraph 'para-1' */
    content: "@"; /* add '@' */
}

```

Counter Increment:

```

h3{
    font-size: .9em;
    counter-increment: num; /* creates counter 'num' */
}
h4{
    font-size: .7em;
    counter-increment: sub 2; /* creates counter 'sub' that
        increments by 2 at a time */
}
h3:before{
    content: counter(num) " "; /* writes 'num' before an
        h3-tag text */
}
h4:before{
    content: counter(num) "." counter(sub) " "; /* writes
        'num.sub' before every h4-tag text */
}
#para-2:first-letter{ /* first letter in text */
    font-size: 25pt;
}
#para-3:first-line{ /* first line in text, not sentence */
    background-color: yellow;
}
span:first-child{ /* first span in document */
    background: orange;
}
p:last-child{ /* last paragraph in document */
    background: #fae7b5;
}
span:nth-child(2){ /* second span in document */
    background: #ffe4c4;
}
h3:hover{
    background-color: orange;
}

```

Cursors:

```

background-color: orange;
}
#common-quote{
    cursor:pointer; /* cursor changes to pointer type, there are also
        'crosshair', 'move', 'text', 'wait', 'progress', 'help', and 'url(samp.cur)' */
}
#para-4{
    outline: 2px solid blue; /* very similar to 'border', except
        it's only on the outside of the element */
}

```

Outline:

```

#para-4{
    outline: 2px solid blue; /* very similar to 'border', except
        that 'outline' never has something around it (surrounds border) */
}
#para-5{
    border: 2px solid black;
}


/* targets 'buttons' list */


```

Buttons:

```

dt.site-button{ /* all the items (buttons) in the list */
    text-align: center;
    margin-bottom: 5px;
}
dt.site-button a{ /* targets links in the button list */
    display: block; /* displays as a 'block' rather than inline */
    color: white;
    text-decoration: none;
}
dt.site-button a:link{ /* targets text in link in button list */
    background: blue;
    border: 5px solid blue;
}
dt.site-button a:hover{ /* targets same as before but on hover */
    background: green;
    border: 5px solid blue;
}

```

Text Decoration:

- (a) Similarly to before, here is some HTML that will be edited with CSS for the next few sections. It will be used for this section and up until section ‘Tables’:

```

<!DOCTYPE HTML>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>CSS Tutorial</title>
    <link rel="stylesheet" type="text/css" href="stylesheet6.css">
</head>

<body>

<p>
    <span id="underline">Underline </span>
    <span id="line-through">line through </span>
    <span id="overline">overline </span>
    <span id="lowercase">LOWERCASE </span>
    <span id="uppercase">uppercase </span>
    <span id="capitalize">capitalize </span>
</p>

<p id="whitespace-pre">
    White space      will     be      preserved
</p>

<p id="never-wrap">
    Lorem ipsum dolor sit amet, consectetur adipiscing elit. Cras accumsan tort
</p>

<p id="preserve-wrap">
    Lorem ipsum dolor sit amet, consectetur adipiscing elit.      Cras accum
</p>

<p id="left-right">
    I prefer left to right
</p>

<p id="right-left">
    I prefer right to left
</p>

```

```

<table>
  <caption>Auto Table</caption>
  <tr>
    <td>Random</td>
    <td>More Random Words</td>
    <td>Even More Random Words</td>
  </tr>
</table>

<table id="fixed">
  <caption>Auto Table</caption>
  <tr>
    <td>Random</td>
    <td>More Random Words</td>
    <td>Even More Random Words</td>
  </tr>
</table>

<table id="spacing">
  <caption>Auto Table</caption>
  <tr>
    <td>Random</td>
    <td></td>
    <td>Even More Random Words</td>
  </tr>
</table>

</body>
</html>

```

- (b) Here are a few ways to decorate text:

```

#underline{
  text-decoration: underline; /* puts line under text */
}
#line-through{
  text-decoration: line-through; /* puts line through text */
}
#overline{
  text-decoration: overline; /* puts line on top of text */
}
#lowercase{
  text-decoration: lowercase; /* changes all letters to lowercase */
}
#uppercase{
  text-decoration: uppercase; /* changes all letters to uppercase */
}
#capitalize{
  text-decoration: capitalize; /* capitalizes the first letter of each word */
}

```

Whitespace:

```

#whitespace-pre{
  white-space: pre; /* html doesn't keep more than one 'space' between words, this will
  'preserve' those 'spaces' by not ignoring any of them */
}
#never-wrap{
  white-space: nowrap; /* deletes all line breaks which makes your text one line */
}
#preserve-wrap{
  white-space: pre-wrap; /* keeps all white space and puts line breaks in */
}
#left-right{
  direction: ltr; /* essentially left alignment */
}
#right-left{
  direction: rtl; /* elements are placed one by one from right to left,
  if only one text element then this is just right alignment */
}

```

Tables:

```

table{ /* targets all tables */
    width: 600px;
    caption-side: top; /* caption shows up on the top of the table */
    text-align: center; /* every column is centered */
    border: 2px solid black; /* table has a border all around it */
}
td{
    border: 2px solid black; /* every cell has its own border */
}
table#fixed{ /* targets a table with id 'fixed' */
    table-layout: fixed; /* every table entry is the same size */
}
table#spacing{
    border-spacing: 10px 15px; /* spacing between 'table' border and 'td' border */
    empty-cells: hide; /* deletes the borders in each 'spacing' cell */
    border-collapse: collapse; /* deletes outer table border and fuses inner cell borders */
}

```

Shadows:

- (a) Similarly to before, here is some HTML that will be edited with CSS for the next few sections. It will be used for this section and up until section ‘Background Image’:

```

<!DOCTYPE HTML>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>CSS Tutorial</title>
    <link rel="stylesheet" type="text/css" href="stylesheet7.css">
</head>
<body>
    <div id="div1"></div>
    <div id="div2"></div>
    <div id="div3"></div>
    <div id="div4">CSS3</div>
    <div id="div5">Is Awesome</div>
    <div id="div6"></div>
    <div id="div7"></div>
    <div id="div8"></div>
    <div id="div9"></div>
    <div id="div10"></div>
    <div id="div11">Rounded</div>
    <div id="div12">Circle</div>
    <div id="div13">Tab</div>
    <div id="div14"></div>
    <div id="div15"></div>
</body>
</html>

```

- (b) Here are a few ways to shade text:

```

div#div1, div#div2, div#div3{ /* target 'div1', 'div2', and 'div3' in div tags */
    height: 100px;
    width: 100px;
    margin: 10px;
    border: 3px solid black;
}
#div1{
    box-shadow: 5px 5px 5px 2px gray; /* shadow around 'div1', first two entries are offsets,
    the first is the rightwards shift and the second is the downwards shift; the shadow
    has a thickness of the fourth entry '2px' in every direction (so it won't show up on
    the topside or leftside), and there is a blur of 5px (third entry); the shadow color is
    gray so it will be difficult to see the blur since it blurs into white */
}
#div2{
    box-shadow: 5px 5px 5px 2px gray inset; /* essentially the same thing except that the
    shadow/blur are 'inset' which means that they are inside of 'div2' */
}
#div3{
    box-shadow: 0 0 10px 5px yellow; /* no offset, but a large shadow/blur, so we have a
    'glow' effect */
}
div#div4, div#div5{
    height: 30px;
    width: 30px;
    font: 20pt Arial;
}
#div4{
    text-shadow: 2px 2px 3px gray; /* same as before except drop shadow for text */
}
#div5{
    text-shadow: 0 0 3px yellow; /* same as before except a glow for text */
}

```

RGB and HSL:

```

div#div6, div#div7, div#div8, div#div9, div#div10{
    height: 100px;
    width: 100px;
    float: left;
}
#div6{
    background: rgb(0, 0, 205); /* the 'rgb()' function sets color based off of the rgb color
    scale, 256 possibilities for each color, this would be blue */
}
#div7{
    background: rgb(0, 0, 205);
    opacity: .5; /* makes the background translucent */
}
#div8{
    background: rgba(0, 0, 205, .5); /* the 'rgba()' function is the same as 'rgb()' except
    that there is a 4th term which determines the opacity, this is the same as the previous
    block of code, but cleaner as it is a one-liner, (a stands for alpha which is opacity) */
}
#div9{
    background: hsl(205, 100%, 50%); /* the 'hsl()' function has 3 inputs, the first goes up to
    the number 360, where there are 360 different 'hues' to choose from, the second is the
    'saturation' percentage, and the last is the 'lightness' percentage, hence 'h.s.l' */
}
#div10{
    background: hsla(205, 100%, 50%, .5); /* this is the same as 'hsl()' but with an extra
    4th term that determines the opacity */
}

```

Border Radius:

```

#div11{
    border-radius: 30px; /* this makes the border curved, the easiest way to imagine it
    is that the curve begins 30 pixels in from the corner in each direction */
}
#div12{
    border-radius: 50px; /* makes a circle because 'div12' is 100 pixels across */
}
#div13{
    border-top-right-radius: 30px; /* only affects the 'top-right' corner of the div
    instead of the whole div */
}

```

Background Image:

```

div#div14, div#div15{
    height:92px;
    width: 80px;
    float: left;
    background-image: url(nttlogo.png); /* inserts image behind div */
}
#div14{
    background-size: 200% 200%; /* fits the top quarter of the image in the div since we're
        making the image bigger by a factor of 4 */
}
#div15{
    background-size: 50% 50%; /* fits 4 images in the div since we're making the image smaller by
        a factor of 4 */
}

```

Gradient:

- (a) Similarly to before, here is some HTML that will be edited with CSS for the next few sections. It will be used for this section and up until section ‘Animations’:

```

<!DOCTYPE HTML>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>CSS Tutorial</title>
    <link rel="stylesheet" type="text/css" href="stylesheet8.css">
</head>

<body>

<div id="div1"></div>
<div id="div2"></div>
<div id="div3"></div>
<div id="div4"></div>

<div id="div5"></div>
<div id="div6"></div>
<div id="div7"></div>
<div id="div8"></div>

<div id="div9"></div>
<div id="div10"></div>

</body>
</html>

```

- (b) Here are a few ways to create background gradients:

```

/* Browser Prefixes are as follows:
moz : Firefox
ms : IE
o : Opera
webkit : Chrome / Safari
*/


#div1, div#div2, div#div3, div#div4{
    height: 100px;
    width: 100px;
    margin: 10px;
    border: 3px solid black;
}
#div1{
    background: -moz-linear-gradient(45deg, red, yellow, blue);
    background: -webkit-linear-gradient(45deg, red, yellow, blue);
    background: -ms-linear-gradient(45deg, red, yellow, blue);
    background: -o-linear-gradient(45deg, red, yellow, blue);
    background: linear-gradient(45deg, red, yellow, blue);
} /* these all do the same thing but for different browsers, the first input is the direction
of the gradient, the next inputs are the colors in the gradient, there is no color limit; in this
case we have a square which begins with a red color in the bottom-left of the square, becomes
yellow by the middle of it, and becomes blue by the top right of the square*/
#div2{
    background: -moz-linear-gradient(left top, red 20%, yellow 30%, blue 60%);
    background: -webkit-linear-gradient(left top, red 20%, yellow 30%, blue 60%);
    background: -ms-linear-gradient(left top, red 20%, yellow 30%, blue 60%);
    background: -o-linear-gradient(left top, red 20%, yellow 30%, blue 60%);
    background: linear-gradient(left top, red 20%, yellow 30%, blue 60%);
} /* degrees are more specific, but we can instead just state what part of the background
that the red color starts from here it is the 'left top' part; we can also state what percent
of the image should have what colors, note that this does not need to add to 100% */
#div3{
    background: -moz-repeating-linear-gradient(left top, red 20%, yellow 30%, blue 60%);
    background: -webkit-repeating-linear-gradient(left top, red 20%, yellow 30%, blue 60%);
    background: -ms-repeating-linear-gradient(left top, red 20%, yellow 30%, blue 60%);
    background: -o-repeating-linear-gradient(left top, red 20%, yellow 30%, blue 60%);
    background: repeating-linear-gradient(left top, red 20%, yellow 30%, blue 60%);
} /* this makes the gradient cycle through the colors multiple times 'repeating' */
#div4{
    background: -moz-radial-gradient(left top, red 20%, yellow 30%, blue 60%);
    background: -webkit-radial-gradient(left top, red 20%, yellow 30%, blue 60%);
    background: -ms-radial-gradient(left top, red 20%, yellow 30%, blue 60%);
    background: -o-radial-gradient(left top, red 20%, yellow 30%, blue 60%);
    background: radial-gradient(left top, red 20%, yellow 30%, blue 60%);
} /* this is a gradient where the red starts in the middle of the background and changes
to yellow and then blue as the color moves away from the center */


```

More Image Transformations:

- (a) In the following code are some image transformations which includes scaling, rotating, skewing, and translating:

```

div#div5, div#div6, div#div7, div#div8{
    height: 100px;
    width: 100px;
    margin: 20px;
    border: 3px solid black;
}
#div5:hover{
    -webkit-transform: scale(2);
    -moz-transform: scale(2);
    -o-transform: scale(2);
    -ms-transform: scale(2);
} /* upon hovering over the div, it will increase in size by a factor of 2 */
#div6:hover{
    -webkit-transform: rotate(45deg);
    -moz-transform: rotate(45deg);
    -o-transform: rotate(45deg);
    -ms-transform: rotate(45deg);
} /* upon hovering over the div, it will rotate the div by 45 degrees */
#div7:hover{
    -webkit-transform: skew(20deg, 10deg);
    -moz-transform: skew(20deg, 10deg);
    -o-transform: skew(20deg, 10deg);
    -ms-transform: skew(20deg, 10deg);
} /* the x-axis is skewed by 20 degrees and the y-axis is skewed by 10 degrees; for the x-skew of 20 degrees imagine that the top of the image shifts left and the bottom of the image shifts right so that the top-left/bottom-right angles are 70 degrees; for the y-skew of 10 degrees imagine that the leftside shifts up and the right shifts down so that the top-left/bottom-right angles are 80 degrees */
#div8:hover{
    -webkit-transform: translate(20px, 20px);
    -moz-transform: translate(20px, 20px);
    -o-transform: translate(20px, 20px);
    -ms-transform: translate(20px, 20px);
} /* moves image to the right 20 pixels and down 20 pixels */

```

Animations:

```

div#div10{
    height: 100px;
    width: 100px;
    margin: 20px;
    border: 3px solid black;
    background: red;
}
div#div10{
    width: 100px;
    height: 100px;
    background-color: red;
    position: relative;
    -webkit-animation-name: example;
    -webkit-animation-duration: 4s;
    -webkit-animation-delay: 2s;
    animation-name: example; /* the animation is called 'example' */
    animation-duration: 4s; /* animation takes 4s */
    animation-delay: 2s; /* when the page is loaded it waits 2s before animating */
}
@-webkit-keyframes example {
    0%   {background-color: red; left:0px; top:0px;}
    25%  {background-color: yellow; left:0px; top:0px;}
    50%  {background-color: blue; left:200px; top:200px;}
    75%  {background-color: green; left:0px; top:200px;}
    100% {background-color: red; left:0px; top:0px;}
}
@keyframes example { /* occurs for animation 'example' */
    0%   {background-color: red; left:0px; top:0px;} /* starts red and original position */
    25%  {background-color: yellow; left:200px; top:0px;} /* becomes yellow and shifts 200 pixels to the right */
    50%  {background-color: blue; left:200px; top:200px;} /* becomes blue and shifts 200 pixels down */
    75%  {background-color: green; left:0px; top:200px;} /* becomes green and shifts back 200 pixels to the left */
    100% {background-color: red; left:0px; top:0px;} /* turns back red and shifts back 200 pixels up */
}

```

Flex Box:

- (a) Here is a big chunk of Html that is used to describe the ‘flex box’:

```

<!DOCTYPE HTML>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>CSS Tutorial</title>

  <!-- You can provide many alternative style sheets -->
  <link rel="stylesheet" type="text/css" media="screen"
  href="stylesheet9.css" title="Default Style">

  <link rel="stylesheet" type="text/css" media="screen"
  href="large-text.css" title="Large Text Style">

  <link rel="stylesheet" type="text/css" media="handheld"
  href="handheld.css" title="Mobile Device Style">

  <link rel="stylesheet" type="text/css" media="print"
  href="print.css" title="Printer Style">
</head>

<body>

<div id="lorem">
  <h2>Lorem Ipsum</h2>
  <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla</p>
</div>

<div id="container">
  <div id="div1">
    <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit</p>
  </div>
  <div id="div2">
    <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit</p>
  </div>
  <div id="div3">
    <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit</p>
  </div>
</div>

</body>
</html>

```

(b) Here is the CSS that talks about flex boxes:

```

div#lorem{
  height: 150px;
  width: 450px;
  border: 2px solid black;
  overflow: hidden;
}
div#lorem{
  -webkit-columns: 4;
  -ms-columns: 4;
  -o-columns: 4;
  -moz-columns: 4;
  columns: 4;
} /* splits paragraph tag into 4 columns, if the 4 columns don't fit, the last column will be
hidden since we have 'overflow: hidden' in the previous css block */
div#lorem{
  -webkit-column-rule: 2px dashed black;
  -ms-columns: 2px dashed black;
  -o-columns: 2px dashed black;
  -moz-columns: 2px dashed black;
  columns: 2px dashed black;
} /* same except each column has a 2 pixel thick vertical dashed black line dividing them */
#container{
  height: 150px;
  width: 450px;
  border: 2px solid black;
}
#container{
  display: -webkit-box; /* defines 'container' as a flexible box */
}
#div1{
  -webkit-box-flex: 2; /* only works for chrome/safari (lazy) */
}
#div2{
  -webkit-box-flex: 2; /* takes up 2 units out of 2+2+3 total units */
}
#div3{
  -webkit-box-flex: 3; /* takes up 3 units */
}
/* these flexible boxes make it possible to resize the screen, allowing for the divs to resize
while the window is resized */

```