

CSS Tutorial Part 1:

Introduction:

CSS adds style to tags in your html page. With HTML you told the browser what things were (e.g., this is a paragraph). Now you are telling the browser how things look (e.g., I want my paragraphs to have blue text with double-spacing and a purple background). To make all paragraphs have the style just mentioned, you'd add the following CSS code (I'll explain what each line means in detail later):

```
p {      background-color: purple;
         color: blue;
         line-height: 200%;
}
```

A. Adding Style to a Web Page (3 options):

You can add a style to your web page in 3 different ways:

Method 1. By including the style inline (e.g., in the tag itself in your html document), e.g.,:

`<body>`

```
<p style = "background-color: purple; color: blue; line-height: %200%;" > This is a blue paragraph with a purple background
color and double the normal space between lines</p>
```

...

Method 2. By including the style in the head section of the web page, e.g.,:

```
<head>
  <meta charset="utf-8" />
  <title> The web page </title>
  <style >
    p {      background-color: purple;
             color: blue;
             line-height: 200%;
    }
  </style>
</head>
```

****Method 3**** By attaching a "style sheet" to your web page, and then placing all your styles in that new style sheet.

To do this, you first create a brand new file (not a web page, a brand new completely blank file). You put in this brand new completely blank file they style(s) you want (e.g.,:

```
p {      background-color: purple;
         color: blue;
         line-height: 200%;
}
```

Once the file contains a style, save it as **blablabla.css**. The **blablabla** part can be some other name, if you prefer, but the **.css** must be just that.

Now you've got to attach the style file to your html file. If you skip this step, the styles you've created in the css file won't be applied to the web page you've created. To attach the css stylesheet to the web page, in the head section, add a link to the style as such:

```
<head>
  <meta charset="utf-8" >
  <title>The web page </title>

  <link rel="stylesheet" href="blablabla.css">
</head>
```

All 3 methods will work.

Method 1 allows for quick and easy changes to style; and

Method 2 has the advantage of keeping your styles within the same file as your html code. Equally it allows you to override a stylesheet (Method 3) for tags on one particular page in your web site. However, it can lead to very big html files, and can get hard to keep track of when making changes.

Method 3, when possible, is my preferred method. Method 3 allows you to create a style sheet separate from your web page(s). Once the style sheet is created, you can include it in as many web pages as you wish. Thus, if you have a web site with a large number of web pages, and you want to change the look of the entire web site, you just need to change one style sheet. In addition, you can create a number of different style sheets with very different styles. Then you can completely change the look of your web page simply by changing which style sheet you link to it. For an example of how completely you can change the look of a web page simply by changing its style sheet, visit:

<http://www.csszengarden.com/>

Thus, for our tutorials and class project, I will insist you use a separate style sheet as specified using **Method 3** and place most of your styles in the stylesheet.

B. Creating Styles:

Now you know how to include a style. But you don't know how to create a style. So we'd better learn that. You must apply a style to html tags, so first create or download an html page:

Step 1 (for method 3):

Create a separate CSS file.

- Open a new file in notepad++/wrangler/bracket
- Save it with a .css extension (I will call mine tutorial.css, but you can call it whatever you like)

Step 2:

(Note that this is still for **method 3** to attach our style to the html page)

Attach the CSS file to your html file. You must attach the css file to the html file or the styles won't show up.

1. Open the HTML file in notepad++/wrangler
2. In between the <head> and the </head> tag, add the following line (with your file name):

```
<!DOCTYPE html>
<html lang="en-US">
  <head>
    <meta charset="utf-8" >
    <title>All Things Giraffe</title>

    <link rel="stylesheet" href="tutorial.css">
  </head>
```

Note: You can attach the same CSS file to more than one html file. That way the style you create for one html file can be used on other html files as well.

Aside: You can attach more than one stylesheet to an html file as well. If you've styled the same tag in both stylesheets, the style that will be applied is the one in the stylesheet you attached below the other stylesheet.

B.1. Creating styles for Tags:

Note: you can add style to 3 different types of elements:

1. The pre-existing tags on your page (e.g., <p>, <h1>,<body>,, etc.)
2. ids you create on your web page
3. classes you create (we'll see these later)

We'll discuss 2 and 3 more later. For now, we shall just add styles to the pre-existing tags.

You add style to everything on the page by creating a style for the body:

Step 4: Open the tutorial.css file in textedit/notepad.

Let's start with a style for our paragraphs:

To style a specific tag, we use that tag, and then put the style in between { and }, e.g.,

```
p { style goes here...
}
```

Styling with color:

There are 17 basic colors. These are:

aqua, black, blue, fuchsia, gray, grey, green, lime, maroon,
navy, olive, purple, red, silver, teal, white, yellow.

There are actually 147 other colors that html and css recognize as standard names of colors (e.g., BurlyWood, PapayaWhip, OliveDrab (I'm not making this up)). But if you really want unlimited color choices, you'll need to use RGB values, and hex values (from the color lab). For now, let's stick with the basics so we can see what's going on.

Changing text properties:

Step 5: Change font color:

To change the font color, modify the style as such:

```
p { color: olive;
}
```

Save the css file. Load the HTML file (the web page) into a browser. The color of the font in paragraphs (and paragraphs only) should have changed to olive color.

If it didn't either your style has been defined incorrectly (did you remember the semicolon at the end of the line?) or the style sheet has been attached to the html page incorrectly.

Step 6: Changing font family:

To change the font family (e.g., arial, Helvetica, times new-roman, comic, etc.), add the following the paragraph's style:

```
p { color: olive;
font-family: arial, helvetica, sans-serif;
}
```

Save the css file. Load the HTML file (the web page) into the browser. The font in the paragraphs should have changed to arial or Helvetica font. (If not, make sure you've got a semi-colon at the end of each line and a dash between font-family.)

A note about font-family:

You can specify font-family to be any font you want. However, you want to stick with common fonts because the font must exist on everyone's computer. In the above example, we specified the font to be:

```
font-family: arial, Helvetica, sans-serif;
```

In this case, the browser will try to display arial. If arial doesn't exist on the computer, then it will try Helvetica, and if that doesn't work, it will use a default sans-serif font.

THE FONT MUST EXIST ON THE PERSON WHO LOOKS AT YOUR WEB PAGE'S COMPUTER, not yours.

As an alternative, your best bet is to use **Google Fonts** (www.google.com/fonts).

- Click the Quick-Use button
- Select a font variation. You should get a block of code in a link element.
- Copy it and
- Past it into the head of your web page before any style elements.

So it might look something like this:

```
<head>
  <meta charset = "utf-8">
  <title> My Web Page with Style </title>
  <link rel="stylesheet" type="text/css" href="http://fonts.googleapis.com/css?family=Lato:400,700,400italic"/>
  <style>
    ...
```

Step 6: Changing font size:

To change the font size, add the following the the paragraph's style:

```
p { color: olive;
```

```
font-family: arial, helvetica, sans-serif;
font-size: small;
}
```

Save the css file. Load the HTML file (the web page) into the browser. The font may (or may not) have changed in size – if the web page’s default font size is small, you may not notice a difference.

To make sure you can actually see a difference, let’s add a style for the h1 tags in your css style sheet:

```
h1 { font-size: 625%;
}
```

Save the css file. Load the HTML file (the web page) into the browser. You should definitely see a difference in the font size of the h1 elements now!

You can specify font-size as:

- xx-small
- x-small
- small
- medium
- large
- x-large
- xx-large
- smaller
- larger
- px (**avoid using!!**)
- %

% is percent of the default size. So, for instance, if the default size is small, setting:

```
font-size: 200%;
```

would result in the font being set to twice as big as the small font.

smaller and **larger** set the font size to be smaller than the default font size and larger than the default font size, respectively.

NOTE:

The text can be set with a viewport width option: By using the vw option, you’re setting the text so that it will adjust to the size of the browser window. So:

```
h1 { font-size: 10vw;
}
```

Sets the font’s size to be 10% of the viewport’s width. Try this, then resize your browser to see the font size automatically adjust.

px is the font size in pixels. You usually want to avoid using this. People with low vision often increase the font size on their web pages or apps so they can read it more easily (they use control +. Try it some night when your eyes are getting tired). With px, the font size is set and won’t increase, whereas when setting the font size using “small” or “200%”, the font size will increase. In essence, px sets the font size absolutely, so those with low vision can’t adjust it, whereas the other options set the font size relatively (relative to the default size), so if we change the default size, we change all the font sizes set relatively as well.

Step 7: Changing font style:

To change the font style, add the following to the h1 style:

```
h1 { font-size: 625%;
font-style: italic;
}
```

Save the css file. Load the HTML file (the web page) into the browser. The h1 element’s font should be italic.

You can also set the font style to:

- normal
- oblique (no one uses this one – it looks just like italic)

Step 8: Changing font weight:

To change the font weight, add the following to the h1 style:

```
h1 { font-size: 625%;  
      font-style: italic;  
      font-weight: bold;  
}
```

Save the css file. Load the HTML file (the web page) into the browser. The h1 element's font should be bold.

You can also set the font weight to:

- normal
- bold
- bolder
- lighter
- 100
- 200
- 300
- 400 *(same weight as normal)*
- 500
- 600
- 700 *(same weight as bold)*
- 800
- 900

Step 9: Changing font family (to Times new roman):

Let's make sure the font family of our h1 elements is Times new roman. To do this, we must put quotes around "Times New Roman" so that it is interpreted as one word. Add to the h1 style:

```
h1 { font-size: 625%;  
      font-style: italic;  
      font-weight: bold;  
      font-family: "times new roman", serif;  
}
```

Save the css file. Load the HTML file (the web page) into the browser. You may not see a difference in the h1 elements because the default font is times new roman.

Changing text properties and line height:

Step 10: Change text alignment:

Let's change the text alignment of the h1 elements to the right:

```
h1 { font-size: 625%;  
      font-style: italic;  
      font-weight: bold;  
      font-family: "times new roman", serif;  
      text-align: right;  
}
```

And let's change the text alignment of our paragraphs to be lined up on the left:

```
p { color: olive;  
     font-family: arial, helvetica, sans-serif;  
     font-size: small;  
     text-align: left;  
}
```

Save the css file. Refresh the html file in the browser. You should see a different alignments – the h1 element should be aligned to the right, and the paragraphs should be left.

You can also set text-align to:

- center
- left
- right

Step 11: Change text indent

Text indent changes the indent of the first line of an element (e.g., think paragraphs – the first line is often indented a bit). Let's do that:

```
p {
    color: olive;
    font-family: arial, helvetica, sans-serif;
    font-size: small;
    text-align: left;
    text-indent: 25px;
}
```

Save the css file. Refresh the html file in the browser. You should see a different alignments – the h1 element should be aligned to the right, and the paragraphs should be left.

Step 12: Text-Shadow

You can add a shadow to text by specifying the horizontal shadow, the vertical shadow, and the color as follows (this is a new property):

```
p {
    text-shadow: 3px 2px blue;
}
```

Text-decoration

Add to a style:

```
text-decoration: underline;
```

You can also set the text decoration to be:

- blink (I think, meaning I really hope!!! this has been deprecated on most browsers now)
- overline
- line-through
- underline (this one can quickly become confusing, because links on your web page are usually underlined, so if you set other text to be underlined, people looking at your web page will assume that text is a link as well.)

Text indent changes the indent of the first line of an element (e.g., think paragraphs – the first line is often indented a bit). Let's do that:

Step 12: Change line height:

This controls the height between two lines of text in a paragraph. In this case we're using 120% of the default height between two lines. To specify line-height for the paragraph style, add:

```
p {
    color: olive;
    font-family: arial, helvetica, sans-serif;
    font-size: small;
    text-align: left;
    text-indent: 25px;
    line-height: 120%;
}
```

Save the css file. Reload the html file in your browser and see the results of line-height. Feel free to play with it a bit.

Borders, Margins, Padding, Background Images

Adding a border and border styles:

Add a border:

You can modify a border's style, width, and color. You can also specify that a tag's style only has a border on one, two, or three sides. Try adding a border around your h1 elements:

```
h1 {  
    border-style: solid;  
    border-width: 4px;  
    border-color: green;  
}
```

Save the css file and look at the html file in the browser. You should see a green border around your h1 elements that is 4 pixels wide and solid.

Other border **styles** include:

- solid,
- dotted,
- dashed,
- double,
- groove,
- ridge,
- inset,
- outset.

Border width is specified in pixels, and can be anything you like. You can also use preset widths, but I almost always use pixel widths (e.g., border-width: 3px; (note there's no space between the 3 and the px)):

- thin
- medium
- thick

Border color can be any of the standard colors, or you can specify the color using hex values or rgb values (which we will discuss later).

You can shorthand the border properties as follows:

```
border: 2px dashed purple;
```

border images:

You can also use an image as a border. There are many ways to control the border image, but you most likely want to pick something that is square and borderish looking. To make the border an image, do the following:

```
h2 {  
    border: 15px solid transparent;  
    padding: 10px;  
    border-image:url("borderpic.jpg") 30 round;  
}
```

The image must be in the same folder as the html file you are working on, or you must specify the path to the image like you did with all other images and links. You can specify how you want the border with the following: round, stretch (will take one border and stretch it to the size of the element it is bordering, or repeat (the image will repeat around the edges),

Adjusting a border on one side:

You can also add a border or set border properties on individual sides. So, for instance, you could say:

```
h2 {  
    border-bottom-style: dashed;  
    border-bottom-width: 2px;  
    border-bottom-color: purple;  
}
```

You could also say:

```
h3 {  
    border-top-style:dotted;  
    border-right-style:solid;  
    border-bottom-style:dotted;  
    border-left-style:solid;  
}
```

Feel free to try this and see how it looks.

Rounding borders:

You can also round the border corners using border-radius:

```
h2 {  
    border: 2px solid purple;  
    border-radius: 20px;  
}
```

If you want to round the radius on only one side:

```
b2 {    2px solid purple;  
    border-top-left-radius: 30px;  
}
```

ASIDE:

Wanna try something cool?

Try the following web page (you should be able to copy and paste this into a new web page and save it, then run it):

```
<!doctype html>  
<html lang = "en">  
  <head>  
    <meta charset = "utf-8">  
    <title> test </title>  
    <style>  
      p{ color: blue;  
          width: 300px;  
          height: 200px;  
          background-color: blue;  
          border: 1px solid black;  
          animation: mymove 5s infinite;  
        }  
      @keyframes mymove {  
        50% {border-top-left-radius:190px;}  
      }  
    </style>  
  </head>  
  <body>  
    <p>Secret Message!</p>  
  </body>  
</html>
```

Adding Padding and Margins:

Padding and margins are different. To clarify the difference:

- Padding is the space between the text and the border (inside the element).
- Margin is the space between the border and the other elements on the page (outside the element)

You can adjust both separately, and again, you can set the padding and margin for the top, left, bottom, and right separately.

```
h1 {    border-style: solid;  
        border-width: 4px;  
        border-color: green;  
        padding-left: 10px;  
        padding-right: 10px;  
        padding-bottom: 10px;  
        padding-top: 40px;  
        margin: 0px;  
}
```

With this style, you should see padding inside the border between the border and the text, and you should see no margins between the border and other elements.

Special Note: Margins: auto; for centering

Sometimes you want to center an element. For instance, you may create a header or a special paragraph that you want centered on the page. To center an element (not just text, but an entire element) you can set the margins to auto. This will automatically adjust the margins on both sides to be equal, so that the element stays centered even when you resize your browser. To try this, first set the width of an element to be a specific size (in pixels). To do this you'd use:

```
width: 500px;
```

Include this in a style you create for some element (I'll use h2):

```
h2 {
  width: 500px;
  border-style: solid;
  border-width: 5px;
  border-color: green;
  padding: 15px;
  margin: auto;
}
```

Try including this in your css stylesheet. Load your html page into a browser and see what your h2 elements look like. Resize the browser. See how the h2 elements stay centered?

Width, height, max-width:

So you've seen width – it sets the width of an element. There's also a height property:

```
height: 500px;
```

This sets the height of an element.

There's also a max-width property. This forces the browser to keep an element at a max width even if the browser is resized to be smaller than that width. Try it on an element in your web page, then resize your web page to see exactly what it does.

Adding background color and image:

Add a background color:

You can also add a background color to individual elements by using the style: background-color.

```
h1 {
  border-style: solid;
  border-width: 4px;
  border-color: green;
  padding: 10px;
  margin: 0px;
  width: 800px;
  background-color: #F39642;
}
```

Now the whole h1 should have a background color

OPACITY:

For color (background-color, color, border-color, etc.) you can use the name of the color (blue, green etc.) or you can use a hex value (#F39642) or you can use RGB values (12,120,255).

IF YOU USE RGB VALUES: you can add a 4th field called the alpha channel that affects the opacity. Try the following:

```
h1 {
  border-style: solid;
  border-width: 4px;
  border-color: green;
  padding: 10px;
  margin: 0px;
  width: 800px;
```

```
        background-color: rgba(12,120,55,1.0);
    }
```

Save and load in a browser and view.

Now try:

```
h1 {    border-style: solid;
        border-width: 4px;
        border-color: green;
        padding: 10px;
        margin: 0px;
        width: 800px;
        background-color: rgba(12,120,55,0.3);
    }
```

The h1 should now be partially transparent – it will look faded (make sure you have an h1 element inside the body of your web page!). If you use 0.0 as the alpha channel, the color will be completely transparent, and if you use 1.0, the color will be completely opaque (not transparent at all). This is new, and kinda neat to play with.

You can also set the opacity of an image/ element using the opacity property. You can use it with an image or an element. Note that if you use it with an element, all of the element will be set to that opacity, including the text. So, while the background will fade, so will the text. To use the opacity property, do the following:

```
h1 {    border-style: solid;
        border-width: 4px;
        border-color: green;
        padding: 10px;
        margin: 0px;
        width: 800px;
        background-color: rgb(12,120,55);
        opacity: 0.5;
    }
```

See the difference between just setting the background-color's opacity and setting the entire element's opacity?

Add a background image:

Color is cool, but images are cooler. You can add a background image to the style of a tag. To do this, go find a good background image on the web. Make sure you place the image in the same folder as the css file you're working on (or follow the rules for getting the browser to locate the image. Now add it to your element:

```
h1 {    border-style: solid;
        border-width: 4px;
        border-color: green;
        padding: 10px;
        width: 500px;
        margin: auto;
        background-color:#F39642;
        background-image: url("savannahbg.jpg");
    }
```

The h1 elements should now have a background image.

NOTE about background images: they're fun, but if you're using it with something that has text (a paragraph, an h1 element, etc.) **MAKE SURE THE TEXT IS READABLE** on top of the background image.

The default is for the background image to "tile" or repeat again and again across and down in the background (like tiles on your bathroom wall). If you use a small picture as your background image, you'll see this.

Background-repeat:

If you want a background image (that other elements float on top of), but you don't want it to tile, or you only want it to tile in one direction, you can set the background-repeat property.

If you only want the background image to show up once, you'd specify that you don't want it to repeat using no-repeat:

```
background-repeat: no-repeat;
```

If you specify a repeat, you might want to set the background image's position using background-position:

```
background-position: left top;
```

You can use the following to set the background image's position:

- left top
- left center
- left bottom
- right top
- right center
- right bottom
- center top
- center center
- center bottom

And you can set the background image's position using x% y% (the top left is 0% 0% and the bottom right is 100% 100%

```
background-position: 20% 40%
```

Or if you want to specify using pixels:

```
background-position: 50px 100px;
```

If you wish to have the background image tile (repeat), but only in one direction, you can use:

```
background-repeat: repeat-y;
```

or

```
background-repeat: repeat-x;
```

You can use background-position to specify the position for the repeat-x and repeat-y as well.

You can shorthand the whole background as follows:

```
background: red url("coolpic.jpg") no-repeat right top;
```

Background-size

You can specify the size of the background image. The default size is the actual size of the image (if you just clicked on it, its size in pixels. If it is smaller than the element it is acting as a background image for, it will tile throughout the space. If it is larger, it will just get cut off.

However, you can specify the background image's size in a few ways. The first is to use background-size:

```
background-size: 200px 150px;
```

In this case you've set the width to 200px and the height to 150px. If you only set one, the width gets set and the height is adjusted automatically.

You can also set the size using percentages, again with the width being set first and the height being the second number. If you only set one, the width is set and the height is adjusted automatically:

```
background-size: 50% 70%;
```

If you want to make sure the background covers the entire background of your area, use "cover":

```
background-size: cover;
```

And if you want to make sure that all of a large image is visible, use "contain":

```
background-size: contain;
```

Adding default style to the entire page:

Modifying the body's style:

If you want your entire web page to have a default background color, specific font, font color, etc. you can style the body tag. For example you'd add to your css style sheet the following style:

```
body {  
    background-color: brown;  
    font-size: small;  
    font-family: arial, helvetica, sans-serif;  
}
```

Now the default for the entire web page is to have a brown background with a small font that's arial. You can override this default by styling specific elements.