{WebCodeChicks}
</spartanburg>

HTML and CSS

Tools

There are many tools to get you coding. A good code editor will take you far and fast in your coding career. Try some of these out and pick the one that best fits you.

Atom - atom.io/ - FREE for now
Atom, by Github, is a relatively new addition to the text editor options but it is making quite an impression so far. Atom is free, open source, and highly customizable. Plugin friendly.
Mac, Win 7 & 8 , Linux

Brackets - brackets.io/ - FREE
Brackets is another free and open source editor by a big name. Adobe, to be exact. Brackets comes with some unique and exciting features. One of those being Extract, which allows you to extract information directly from PSDs such as colors, fonts, gradients, and measurements as clean CSS with out contextual code hints. Plugin friendly.
Mac, Win, Linux

Sublime Text 3 - sublimetext.com/3 - FREE/$70 to remove nag popup
ST3 is a popular and beautiful, feature rich option for code writing and editing. Customized snippets lets you code your way getting things done faster. Plugin friendly.
Mac, Win, Linux

Tinker Online
CodePen
Codeanywhere
JS Bin
JSFiddle

Other IDEs
An Integrated Development Environment (IDE) is a programming environment that has been packaged as an application program, typically consisting of a code editor, a compiler, a debugger, and a graphical user interface (GUI) builder.

Komodo Edit - FREE
NetBeans - FREE
WebStorm $99/Free
PHPStorm $199/Free

More on Code Editors

These days most good editor have plugins or extension that extend the features and capabilities that the software developer had in mind. Here are just a few plugins to look for on your coding path:

- Emmet
- LiveReload
- Color coding for errors
- Color picker
- Store and use code snippets

HTML

HTML is the structure of a webpage.

HTML stands for **H**yper**T**ext Markup **L**anguage. Yeah and what does that mean?

- **HyperText** is the method by which you move around on the web — by clicking on special text called hyperlinks which take you to the next page. The fact that it is hyper just means it is not linear — i.e. you can go to any place on the Internet whenever you want by clicking on links — there is no set order to do things in.
- **Markup** is what HTML tags do to the text inside them. They mark it as a certain type of text (italicized text, for example).
- It is a Language, having code-words and syntax like any other language.

THE HTML STRUCTURE:

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4    <meta charset="UTF-8">
5    <title>Document</title>
6  </head>
7  <body>
8
9  </body>
10 </html>
```

Things to notice:
Each one of the “tags” is encapsulated by < and >. Some of them have pairs of tags which the second has a / after the angle brackets. The / designates the ending tag.

Explanation of lines:

1. Instructs the browser what version of the markup language is being used. It just is a declaration tag.
2. The starting of the html tag instructions. Notice on line #10 there is a closing tag. These two tags essentially start and stop/end the html instructions on the page. lang=“en” doesn't help the browser to display

the page, but it is useful for search engines, screen readers, and other things that might read and try to interpret the page to English, besides human beings.

3. Starts the head section that includes the title of the document, scripts, styles, and meta information aka information about the web page.
4. http://en.wikipedia.org/wiki/UTF-8 - enjoy the read! Needs to be there, wish I could explain - deep computer science stuff.
5. The contents of the title tag are shown in the top of the browser in the tab for that page.
6. Head closing tag.
7. Everything between the opening and closing body tags are shown in the main browser window. This is where most of your code will be placed - line 8.
9. Ending of body tag.
10. Ending of html tag

Resources

BOOK

HTML&CSS Design and Build Websites by Jon Duckett

WEBSITE

http://learn.shayhowe.com/html-css/

Everyday Tags/Elements

(The word “tag” and “element” are both used interchangeably .
 is the opening tag, is the closing tag. The element is “ and the text between ”)

There are tags that have default styling from the browser and there are tags that are simply for semantic purposes - they are just a name and have no styling.

Some tags are block level elements and some are inline elements.

Styled Tags

Heading tags: h1, h2, h3, h4, h5, h6 go from big to small in size. They have margin and bold applied yet their size is different.

Paragraph tag: By default, the browser will show each paragraph on a new line with top and bottom margin.

DIV tag: tags are for “divisions” in a page. They are often used as a container tag that can then be manipulated with CSS from classes and ids. Sectioning tags have no default styling applied to them.

Other tags that can be used: **<section>**,**<article>**,**<nav>**,**<header>**,**<footer>** including their closing tags.

Bold and Strong tags: look the same in the browser. tag is for WIA-Aria - visually impaired and should be used rather than the bold tag.

Italic and Emphasis tags: are much like the bold and strong tags. Emphasis is standard for WIA-Aria visually impaired and should be used to emphasize text.

Lists - Unordered and Ordered: inside these tags are listed items, each item is surrounded by these tags.

Links - also known as the “anchor” tag <a> allow you to move from one page to another. Inside the opening a tag an **href** attribute references the link to the other page to go to:

Go to WebCodeChicks!

the text in-between make the link on the page.

Images - <img> Does not have a closing tag and has several attributes: **src** which tells the browser where to find the image, **alt** provides a text description of the image, and **title** which provides more info about the image:

<img src=“dog.jpg” alt=“Dog in the ocean” title=“A photo of my dog in the ocean at the beach last year.”>

Image formats: .jpg .gif .png .svg

Comments
Commenting your code is helpful to label code and make notes. In HTML comments are not visible in the browser window but can be visible in the code.
<!– comment goes here –>

ID Attribute
Every tag can have an Id. Ids uniquely identify that element from other elements on the page and no two elements can have the same value - otherwise they are not unique. Ids will be styled in the css document.
** About page introduction paragraph **

Class Attribute
Every tag can have a class attribute. Classes can be use multiple times on the page. Class names will be styled in the css document.
This paragraph has gray background.
This other paragraph also has gray background.

CSS

CSS stands for **C**ascading **S**tyle **S**heets. CSS defines how HTML elements are to be displayed.

CSS works by associating rules with HTML elements that specify how the content should appear.

Styles can be declared in three ways: internally, inline and externally. The most common and best practice is to use them externally in a separate file.

Internal Style Sheets - Not Recommended

The Code:

```
<!DOCTYPE html>
<html lang="en">
<head>
      <meta charset="UTF-8">
      <title>Document</title>
      <style>
      body {
            background-color: lightblue;
      }
      </style>
</head>
<body>

</body>
</html>
```

The CSS stye rules are placed within the head tag surrounded by style tags.

Inline Styles - Not Recommended

The style rule is declared in the html tag itself:

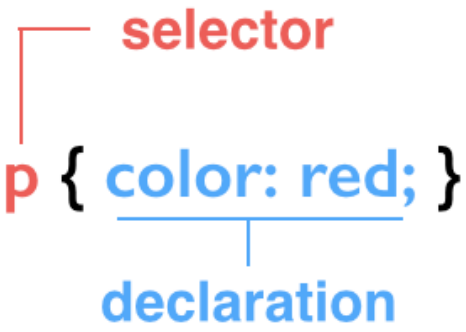
```
<p style="background:#ccc; color:#fff; border: solid black 1px;">
```

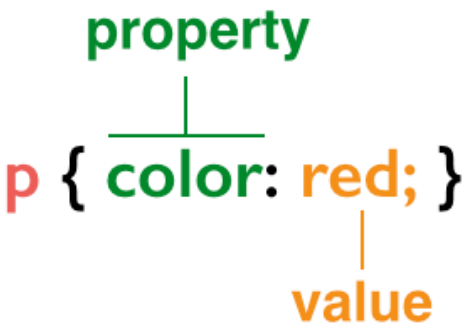
External Stylesheets - Recommended

The **<link>** tag is used in an HTML document in the **<head>** section to tell the browser where to find the CSS file used to style the page. It does not need a closing tag. Styles are normally saved in a file with the name **style** with the **.css** extension and saved in a folder labeled **css**. More than one .css file can be linked.

rel - specifies the kind of relationship between the HTML page and the file it is linked to.
href - specifies the path to the css file

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>WCC Basics</title>
  <link rel="stylesheet" href="style.css">
</head>
<body>
</body>
</html>
```





The key to understanding how CSS works is to imagine that there is an invisible box around every HTML element. This is called the “Box Model.”

CSS declarations (rules), sit inside curly brackets and each is made up of two parts: a property and a value, separated by a colon and ends with a semi-colon.

Declaration indicates how the (HTML) elements should be styled.
Property indicates what you want to change. For example, color, font, width, height and border.

For declaring several rules, it is good coding practice to put each declaration on a separate line:

```
h1 {
  color: lightblue;
  font-family: Helvetica;
  text-decoration: underline;
}
```

Selectors indicate which (HTML) element the rule applies to. The same rule can apply to more than one element if you separate the element name s with a comma:

```
h1, h2, h3 { font-family: Arial; }
```

This rule states that all h1 tags text color will be lightblue, Helvetica font, and underlined.

Earlier IDs and classes were introduced. Here is how to style them:
Ids will use the # character followed by the name of the id:

```
#about-intro {
  font-weight: bold;
  font-style: italic;
}
```

Classes start with a . followed by the name:

```
.gray-bg {
  background-color: gray;
}
```

Comments are seen if anyone looks at the source code and are not displayed on the webpage.

Commenting in CSS
In CSS, the comment tags are different from HTML:

```
/* This is a comment in CSS */
```

Some commonly used properties:

:active	font-size	margin-(top,right, bottom, left)
background	font-weight	overflow
background-color	height	padding-(top,right, bottom, left)
background-image	font-family	text-align
border-(top, right, bottom, left)	font-size	text-decoration
box-shadow	font-weight	text-shadow
box-sizing	height	text-transform
color	:hover	:visited
display	letter-spacing	width
float	:link	
font-family	list-style-type	

Size

Pixels

Pixels have been around for quite some time and are commonly used with a handful of different properties. As an absolute unit of measurement, they don't provide too much flexibility. Pixels are, however, trustworthy and great for getting started.

The default base font size is 16 pixels which is referring to the body font size rendered by the browser. There are 96 pixels in an inch, but may vary depending on different devices.

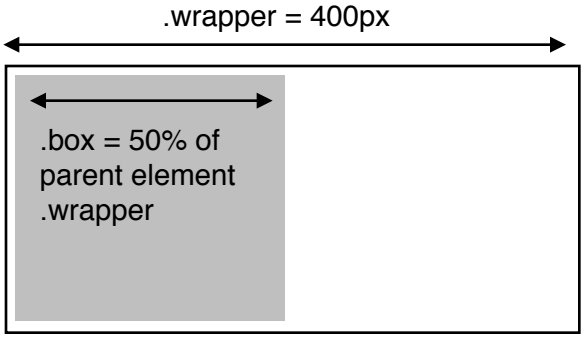
Example:

```
p {
  font-size: 14px;
}
```

Percentages

This gets a little complex, with percentages. They are not fixed units of measurement, and rely on the length of another measurement. Percentages are known as "relative value".

```
.wrapper {
   width: 400px;
}
.box {
  width: 50%;
  background: #999;
}
```



In the example above, the class .box is 50% of the total width of the class .wrapper. This is how percentages work, they must have a “parent” element to set their percentages off of.

Em

Em is another relative value and is calculated on an element’s font size. So, for example, if an element has a font size of 16 pixels and a width set to 5em, the width would equal 80 pixels (16 pixels multiplied by 5).
1em = 16px
2em = 32px
…
5em = 80px
When a font size is not explicitly stated for an element, the em unit will be relative to the font size of the closest parent element with a stated font size. So it is good practice to set the body font size to 16px and the parent size to work from.

Positioning with Floats

Essentially, the float property allows us to take an element, remove it from the normal flow of a page, and position it to the left or right of its parent element.

An <img> element floated to the side of a few paragraphs of text, for example, will allow the paragraphs to wrap around the image as necessary.

```
img {
  float: left;
}
```

(Note this means all images will float left. Adding a class or id would give the specific image specificity.)

The float property accepts a few values; the two most popular values are **left** and **right,** which allow elements to be floated to the left or right of their parent element which contains them.

The float comes with a few pitfalls. Often margin and padding property values aren’t interpreted correctly, causing them to blend into the floated element; other properties can be affected, too.

The Clear Fix - both

The both value, however, will clear both left and right floats and is often the most ideal value.

Positioning with Inline-Block

We can also position content is by using the display property in conjunction with the inline-block value. The inline-block method is primarily helpful for laying out pages or for **placing elements next to one another within a line.**

Uniquely Positioning Elements

Every now and then we'll want to precisely position an element, but floats or inline-block elements won't do the trick. Floats, which remove an element from the flow of a page, often produce unwanted results as surrounding elements flow around the floated element. Inline-block elements, unless we're creating columns, can be fairly awkward to get into the proper position.

The relative value for the position property allows elements to appear within the normal flow a page, leaving space for an element as intended while not allowing other elements to flow around it; however, it also allows an element's display position to be modified with the box offset properties.

```
div {
  height: 100px;
  width: 100px;
}

.offset {
  left: 20px;
  position: relative;
  top: 20px;
}
```

Absolute Positioning

The absolute value for the position property is different from the relative value in that an element with a position value of absolute **will not appear within the normal flow of a document**, and the original space and position of the absolutely positioned element **will not be preserved.**

Additionally, absolutely positioned elements **are moved in relation to their closest relatively positioned parent element.** Should a relatively positioned parent element not exist, the absolutely positioned element **will be positioned in relation to the <body> element.**