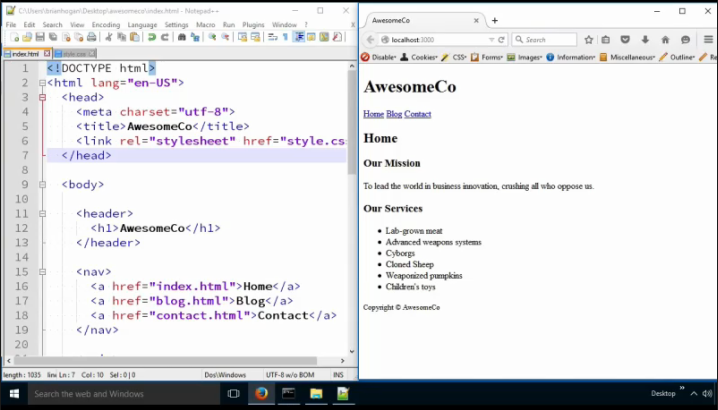
**Tutorial: AwesomeCo – Designing a Layout with CSS**

Let's get back to work on the AwesomeCo web site as we explore how to take our markup and turn it into a layout.

**Open the AwesomeCo template for this lesson.**



**Note:** The complete HTML and CSS file contents are printed out at the end of this tutorial. Please feel free to use that completed code to fix your code as you encounter any issues with this tutorial.

We're going to start things out fresh by having nothing in our stylesheet and by making a slight markup change. We're going to wrap all of our content with a **div** tag

**<div id="wrapper">**

**</div>**

We're going to use this as a logical container that we can use to constrain our page's width. It has no semantic meaning, so a div is just perfect for this. Now we are going to put this opening <div> right under the open <body> and we will close this </div> right above the end </body>. You will have to highlight all the contents inside this new div and press the tab key to move the indent over all on the content. **Add the div into the index.html file as seen below:**

**<body>**

**<div id="wrapper">**

**<header>**

**<h1>AwesomeCo</h1>**

**</header>**

**…**

**<footer>**

**<small>Copyright &copy; AwesomeCo</small>**

**</footer>**

**</div>**

**</body>**

**Save the index.html File. Apply this same change to the rest of the pages on our site.**

Next let's **open the style.css stylesheet file**, which is linked to all of the pages in our site via the link tag in the head of our web page.

The first thing we'll do is add a rule to set the background color of the page. **Type in the following bold CSS:**

**body {**

**background-color: #dddddd;**

**}**

**Note:** You can also use #ddd; This is the shorthand version of this hex color. If the 1st and 2nd letters or numbers are the same, it can be represented as just one letter or number. Same is true for the 3rd and 4th as well as the 5th and 6th characters.

This gives us a gray background. Now let's set the background color of the wrapper area. **Type in the following bold CSS:**

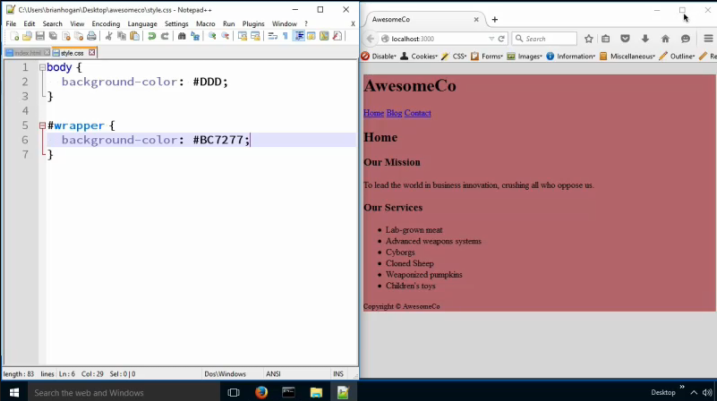
**#wrapper {**

**background-color: #BC7277;**

**}**

**Note:** # represents the id. Since we stated that the div id=”wrapper”, we reference it in CSS by #wrapper.

**Save the styles.css page and refresh the index.html page.**



**Center and Constrain Content**

Now we see we have a background color for the main background of the page and the background of the wrapper div.

Our wrapper is a div tag, and its default mode is display-block. That means we can give it a width. For this exercise, let's give it a fixed width of 960 pixels. **Type in the following bold CSS:**

#wrapper {

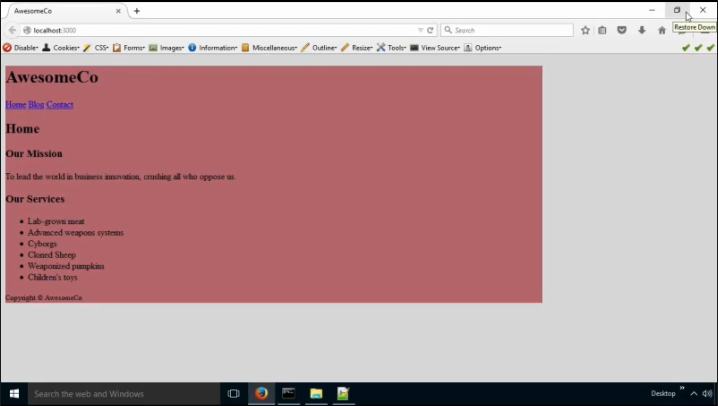
background-color: #BC7277;

**width: 960px;**

}

Now the content doesn't fill the width of the screen, but it does look a little funny. It would be better if the content of the page was centered on the screen.

**Save styles.css and refresh the index.html page.**



Since we have a div with a defined height, we can center it using automatic margins. **Type in the following bold CSS:**

#wrapper {

background-color: #BC7277;

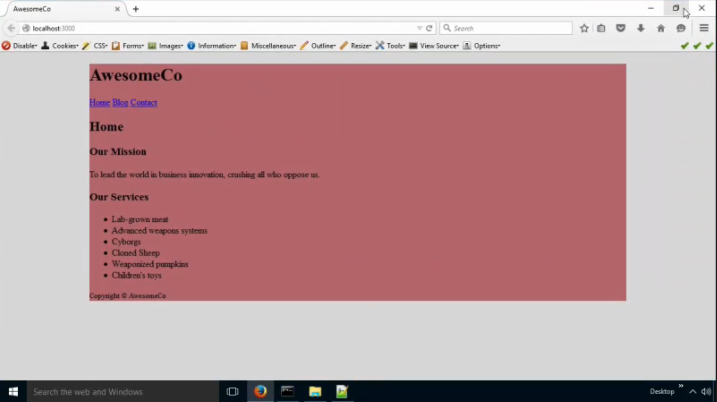
**margin: 0 auto;**

width: 960px;

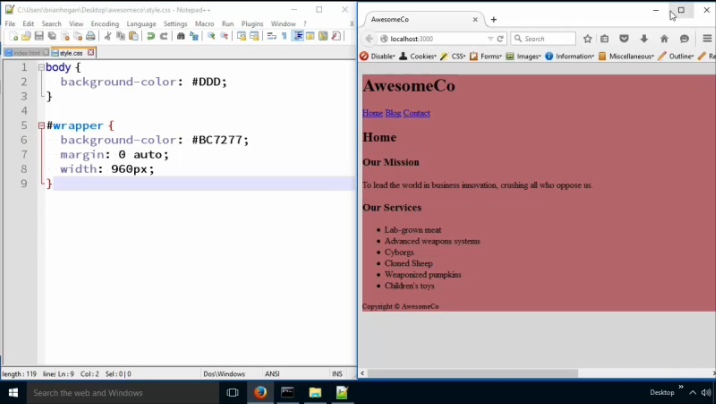
}

We're using the shorthand rule here that says, "no top and bottom margins, and automatic margins on the left and right."

**Save the styles.css page and refresh the index.html page.**



That works really well.



**Header images**

Previously, we placed the AwesomeCo logo on the page with an img tag, and this is pretty normal for web sites, but the downside to doing that is that we can't load different size header graphics based on the screen size. Later on, you'll learn how to use stylesheets to do just that.

But for now, we'll use CSS to load an image, and we'll do it by loading that image into the background of the `header` element. Then we'll hide the header's text content. This is called "image replacement."

First, we'll give the header a white background, and then we'll load in the image using the background-image property. **Type in the following bold CSS:**

**header {**

**background-color: #fff;**

**background-image: url(awesomeco\_logo.png);**

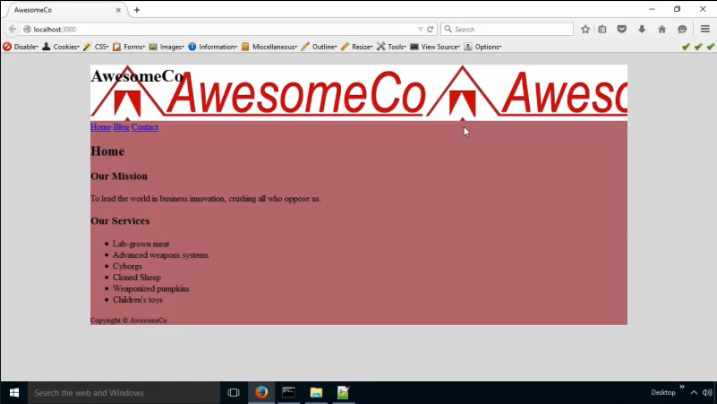
**height: 100px;**

**}**

**Note:** the url is a function in CSS that takes the file location and file name; much like the img src attribute takes the location of a file.

We'll also give the header a height of 100px which should be tall enough to fit the image.

**Save the styles.css and refresh the index.html page.**



But when we look at our page, it looks like the image is repeating. We need to stop that. We use the background-repeat property, with the value of no-repeat. We don't use none or 0 - we use no-repeat. Let's also add a bottom border to that header. **Type in the following bold CSS:**

header {

background-color: #fff;

background-image: url(awesomeco\_logo.png);

**background-repeat: no-repeat;**

**border-bottom: 1px solid #ddd;**

height: 100px;

}

We then use some positioning tricks to hide the h1 from the screen. Notice the AwesomeCo text in the header:



**Type in the following bold CSS:**

**header > h1 {**

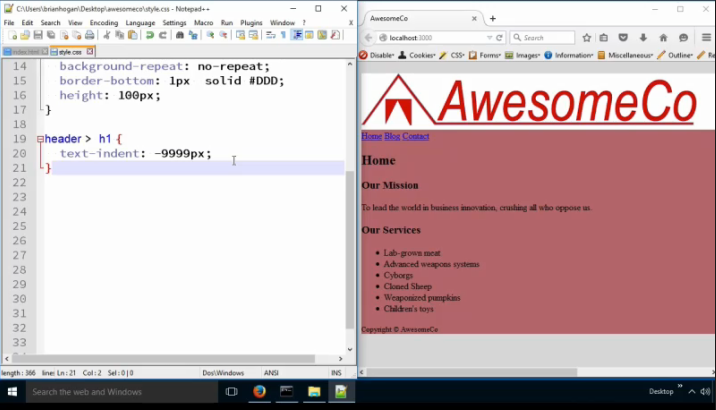
**text-indent: -9999px;**

**}**

**Note**: > means the h1 is a direct child of header.

This is an acceptable accessibility solution. We push the header’s h1 tag off the page to the left side of the screen. This technique should only be used for providing alternative text to images loaded via CSS. It should never be used to load hidden keywords or text intended to spam or trick people, as the search engines are now smart enough to figure that out.

**Save the styles.css and refresh the index.html.**



**Align nav and main section**

For this site, let's put the navigation on the left-hand side of the page, and then the main content can sit next to that content.

To do this, we'll use floats.

First, let's float the navigation. Then we'll give it a width of 15%. That 15% is the width of the CONTAINING element, which is 960 pixels. It's not 15% of the whole screen. **Type in the following bold CSS:**

**nav {**

**float: left;**

**width: 15%;**

**}**

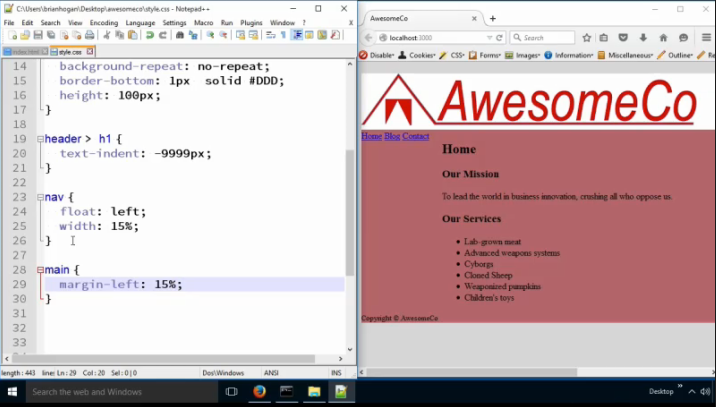
Now we can push the main section's content over 15% by using a left margin. This will make the main section appear to sit next to the menu. **Type in the following bold CSS:**

**main {**

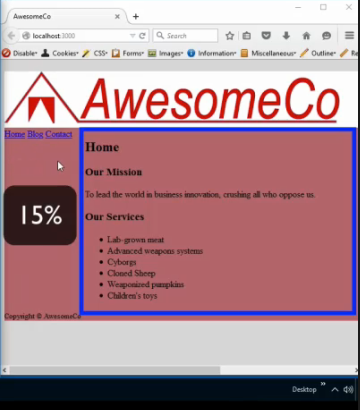
**margin-left: 15%;**

**}**

**Save the styles.css file and refresh the index.html.**



Notice that the main is moved over 15% in the container and the nav takes up 15% of the container.



Let's color in the main section by adding a background color. **Type in the following bold CSS:**

main {

**background-color: #ffffff; /\* added \*/**

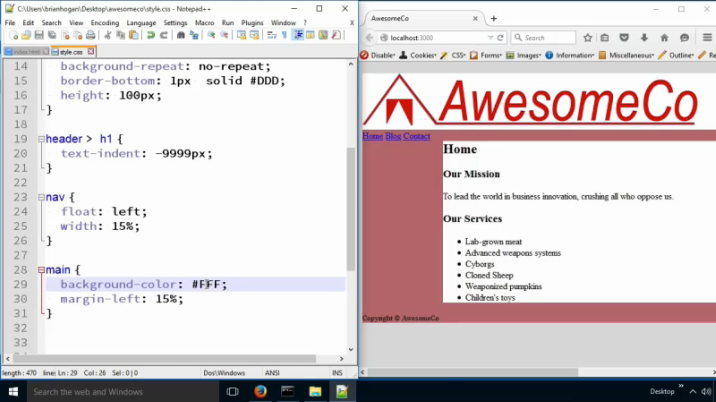
margin-left: 15%;

}

**Note:** /\* and \*/ is the starting and ending comments statements used in CSS. This is optional.

Changing the main background color makes the text in the main easier to read.

**Save the styles.css and refresh the index.html.**



We'll also add some padding to keep the text away from the edge of the white box. This will have the added advantage of ensuring that the color fills out the box. Without the padding, we'd run into that margin collapsing issue we learned about before. **Type in the following bold CSS:**

main {

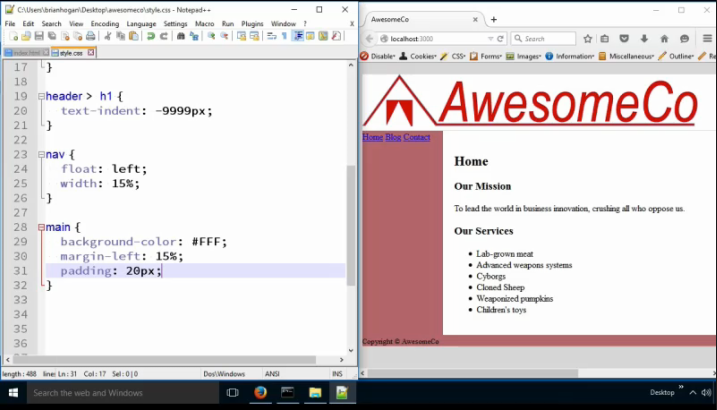
background-color: #ffffff;

margin-left: 15%;

**padding: 20px; /\*added \*/**

}

**Save the styles.css and refresh the index.html.**



So now the sidebar is red because the background color of the wrapper is bleeding through. Regions of a page are transparent unless we color them. So, we've colored our main section and that gives the appearance of colored columns.

Unfortunately, the text in the footer's pretty hard to read. Let's fix that by making the color of the footer text white, and while we're doing that, let's center the text in the footer. **Type in the following bold CSS:**

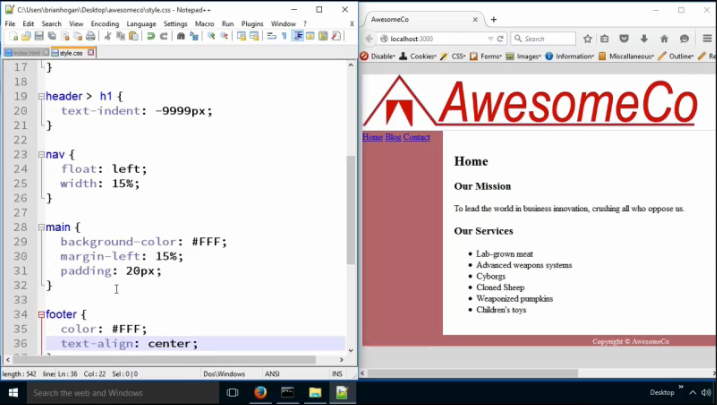
**footer {**

**color: #fff;**

**text-align: center;**

**}**

**Save the styles.css and refresh the index.html.**

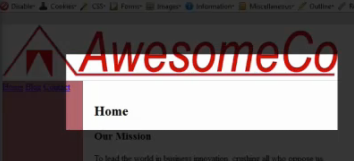


Things are starting to look pretty good now.

It looks like there's a lot of space above the headings. When you're developing your site, you really

want to watch out for little things like this - it makes a difference in how professional your work looks.

Good enough may be good enough to get a grade, but is it really good work you would be proud of?



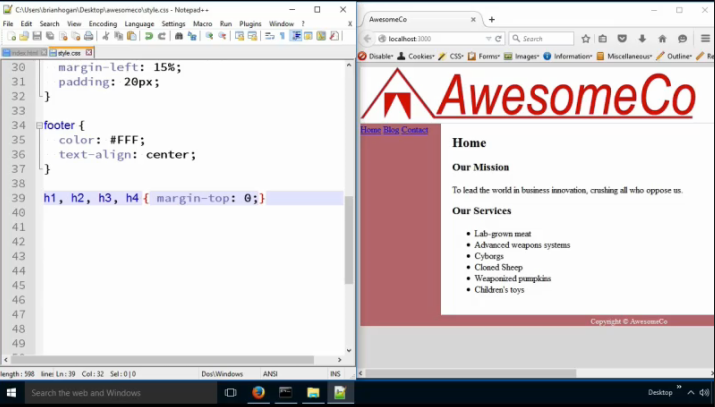
We can fix this issue by adjusting margins on the headers. **Type in the following bold CSS:**

**h1, h2, h3, h4 {**

**margin-top: 0;**

**}**

**Save the styles.css and refresh the index.html.**



**Note:** It is ok to put the CSS declaration property on the same line as the selector if you only have one declaration.

**nav Buttons**

Ok, now let's look at making the navigation look nicer. We'll turn the navigation into buttons.

To do that, we'll turn the navigation items from regular links into links inside an unordered list. This is a very common pattern in web development. A navigation menu is a list of links, and so it makes a lot of sense to embed them in a list. **Go to the index.html page and type in the following bold tags into the nav in the HTML:**

<nav>

**<ul>**

**<li>**<a href="index.html">Home</a>**</li>**

**<li>**<a href="blog.html">Blog</a>**</li>**

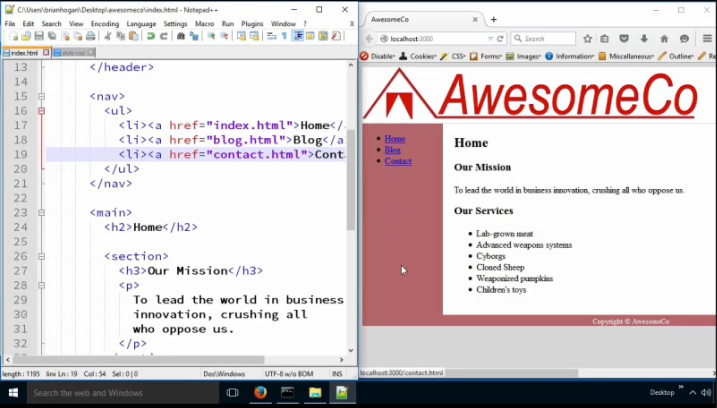
**<li>**<a href="contact.html">Contact</a>**</li>**

**</ul>**

</nav>

Notice we added the ul and li(s) to the a tag. Fix your indentation.

**Save the index.html page and refresh it.**



Notice the bullet points in the nav now.

Next, we take off the margin and padding from the list and we remove the bullets. **Go back into the styles.css file and type in the following bold CSS:**

**nav > ul {**

**list-style: none;**

**margin: 0;**

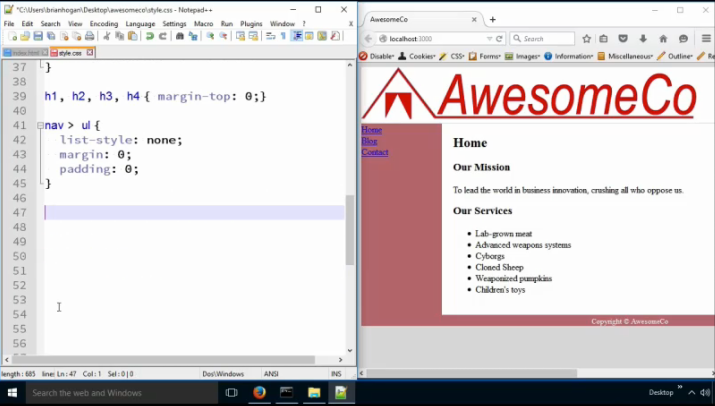
**padding: 0;**

**}**

**Note:** list-style: none removes the bullets.

Notice we're saying, "only do this to the unordered list in the nav section."

**Save the styles.css and refresh the index.html.**



Next, we apply the border and width to the list item, not the link itself. We also apply a little auto-margin magic here to center the buttons in the sidebar. **Type in the following bolded CSS:**

**nav > ul > li {**

**border: 1px solid #333;**

**margin: 10px auto;**

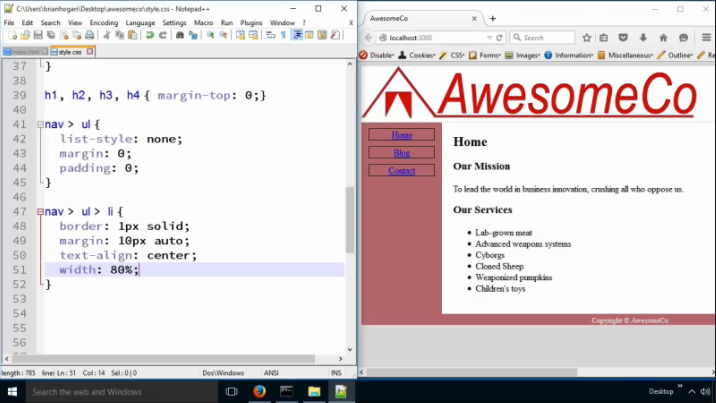
**text-align: center;**

**width: 80%;**

**}**

**Note:** nav>ul>li means to find the li inside of the ul that is inside of a nav tag. The margin: 10px auto; means that the top/bottom margin is 10px and the left/right is centered in that container. The width of 80% is 80% of the 15% wide nav. This gives some padding to the left and right of the button.

**Save the styles.css and refresh the index.html.**



Then we add color and definition to the individual links. **Type in the following bolded CSS:**

**nav >ul > li > a {**

**background-color: #600;**

**color: #fff;**

**display: block;**

**text-decoration: none;**

**}**

**Note:** If we use display: block, then the background color fills out the container, stretching to fill it. Text-decoration: none removes the underline on an ‘a’ link. The ‘a’ tag is an inline element and they don’t by nature have a width attribute. We can change this by changing the ‘a’ tag to a block element.

Finally, we'll add interactivity to the site by adding hover effects on the menu.

We have to remember that we must add support for focus too, because not everyone has a mouse. **Type in the following bolded CSS:**

**nav > ul > li > a:hover,**

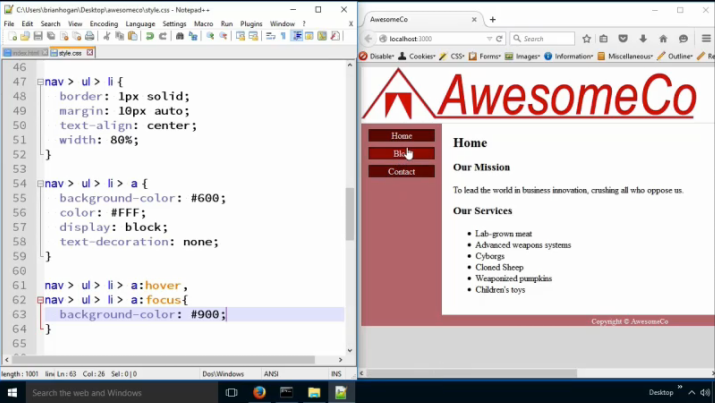
**nav > ul > li > a:focus {**

**background-color: #900;**

**}**

**Note:** The colon means that the hover is a Pseudo-Class of the selector ‘a’. We will be covering this later in the course.

**Save the styles.css and refresh the index.html.**



Now, when you mouse over the buttons, the color should change slightly.

Look at our selectors! This is getting out of hand. The descendant selector is expensive, but at this point we’ve got a lot of extra code in here that we’re repeating. It’s s not a huge performance concern for us to shorten these selectors. We can just say that all the anchors in the navbar should have their color changed on hover and focus. **Replace the previous CSS with the following:**

nav a:hover,

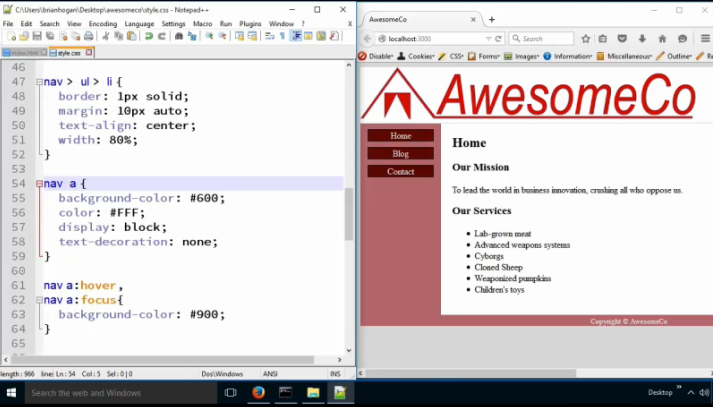
nav a:focus {

background-color: #900;

}

**Note**: We can use a descendent selector instead of a child selector. The space instead of the > represents the descendent selector.

**Save the styles.css and refresh the index.html.**



**Eye Candy**

Before we call it quits, let’s do a couple of things to make the site look really nice. First, let’s round the corners of the buttons in the sidebar. **Find the nav>ul>li selector and add:**

nav > ul > li {

**border-radius: 5px;**

}

Next, let's add a gradient to the background. Right now, our background color is #ddd. Let's make it a gradient, so it transitions from one color to another.

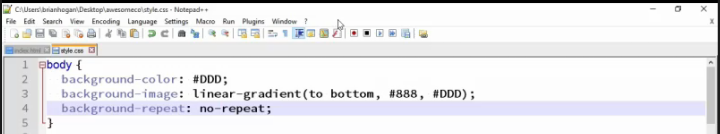
Here's how we do that. We apply a gradient as a background image. We'll say that we want a linear gradient that goes "to the bottom". It starts out at the color #888, a medium grey, and ends at #ddd, our current background color. **Find the body selector and add:**

body {

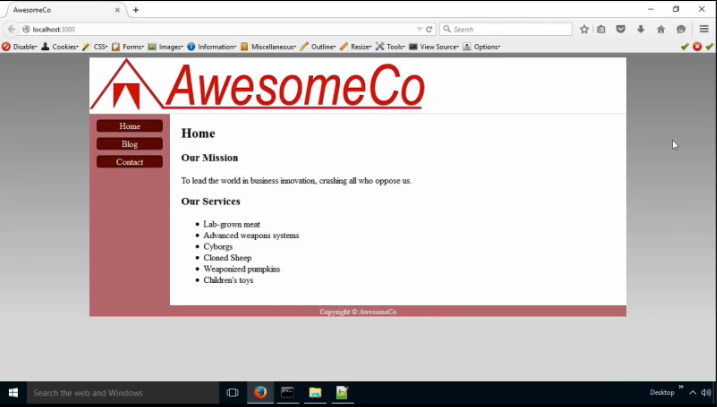
**background-image: linear-gradient(to bottom, #888, #DDD);**

**background-repeat: no-repeat;**

}



**Save the styles.css and refresh the index.html.**



**Update all .html Pages**

Make sure you apply the wrapper id to the wrapper div on the Blog and Contacts pages

In addition, add the ul and li(s) to the nav on each page.

Remember to indent the code, highlight the code and press the tab key to insert in parent elements.

**Note:** You may have to add div>footer to the footer css so the footer CSS does not enact on the footer in the article found on the blog.html page:

**div>footer {**

**color: #FFF;**

**text-align: center;**

**}**

Notice you have two footers on the blog.html page. One is a page footer and the other is an article footer.

**Completed HTML & CSS**

Below is the completed HTML and CSS code for all the pages:

**Contents in the styles.css file:**

body {

background-color: #ddd;

}

h1, h2, h3, h4 {

margin-top: 0px

}

#wrapper {

background-color: #BC7277;

border-radius: 5px;

box-shadow: 5px 5px 5px #555;

margin: 0px auto;

overflow: hidden;

width: 960px;

}

header > h1 {

background-color: #fff;

background-image: url(awesomeco\_logo.png);

background-repeat: no-repeat;

border-bottom: 1px solid #300;

height: 100px;

margin-bottom: 0;

text-indent: -9999px;

}

nav {

float: left;

width: 15%;

}

main {

background-color: #fff;

margin-left: 15%;

padding: 20px;

}

footer {

color: #FFF;

text-align: center;

}

nav > ul {

list-style: none;

margin: 0;

padding: 0;

}

nav > ul > li {

border: 1px solid #333;

border-radius: 5px;

margin: 10px auto;

text-align: center;

width: 80%;

}

nav > ul > li > a {

background-color: #600;

color: #fff;

display: block;

text-decoration: none;

}

nav > ul > li > a:hover,

nav > ul > li > a:focus {

background-color: #900;

}

body {

background-image: linear-gradient(to bottom, #888, #DDD);

background-repeat: no-repeat;

}

**Contents in the index.html file:**

<!DOCTYPE html>

<html lang="en-US">

<head>

<meta charset="utf-8">

<title>AwesomeCo</title>

<link rel="stylesheet" href="style.css">

</head>

<body>

<div id="wrapper">

<header>

<h1>AwesomeCo</h1>

</header>

<nav>

<ul>

<li><a href="index.html">Home</a></li>

<li><a href="blog.html">Blog</a></li>

<li><a href="contact.html">Contact</a></li>

</ul>

</nav>

<main>

<h2>Home</h2>

<section>

<h3>Our Mission</h3>

<p>

To lead the world in business innovation,

crushing all who oppose us.

</p>

</section>

<section>

<h3>Our Services</h3>

<ul>

<li>Lab-grown meat</li>

<li>Advanced weapons systems</li>

<li>Cyborgs</li>

<li>Cloned Sheep</li>

<li>Weaponized pumpkins</li>

<li>Children's' toys</li>

</ul>

</section>

</main>

<footer>

<small>Copyright &copy; AwesomeCo</small>

</footer>

</div>

</body>

</html>

**Contents in the blog.html file:**

<!DOCTYPE html>

<html lang="en-US">

<head>

<meta charset="utf-8">

<title>Blog - AwesomeCo</title>

<link rel="stylesheet" href="style.css">

</head>

<body>

<div id="wrapper">

<header>

<h1>AwesomeCo</h1>

</header>

<nav>

<ul>

<li><a href="index.html">Home</a></li>

<li><a href="blog.html">Blog</a></li>

<li><a href="contact.html">Contact</a></li>

</ul>

</nav>

<main>

<h2>Blog</h2>

<section>

<article>

<header>

<h3>My first post</h3>

</header>

<section>

<p>

This is the text of my first blog post.

</p>

</section>

<footer>

<small>Posted by me on January 13th, 2045</small>

</footer>

</article>

</section>

</main>

<footer>

<small>Copyright &copy; AwesomeCo</small>

</footer>

</div>

</body>

</html>

**Contents in the contact.html file:**

<!DOCTYPE html>

<html lang="en-US">

<head>

<meta charset="utf-8">

<title>Contact - AwesomeCo</title>

<link rel="stylesheet" href="style.css">

</head>

<body>

<div id="wrapper">

<header>

<h1>AwesomeCo</h1>

</header>

<nav>

<ul>

<li><a href="index.html">Home</a></li>

<li><a href="blog.html">Blog</a></li>

<li><a href="contact.html">Contact</a></li>

</ul>

</nav>

<main>

<h2>Contact</h2>

<section>

<p>

AwesomeCo Headquarters<br>

123 Fake Street<br>

Eau Claire, WI 54703<br>

</p>

<p>

<a href="mailto:info@awesomeco.com">info@awesomeco.com</a>

</p>

</section>

</main>

<footer>

<small>Copyright &copy; AwesomeCo</small>

</footer>

</div>

</body>

</html>