More CSS

More CSS Features

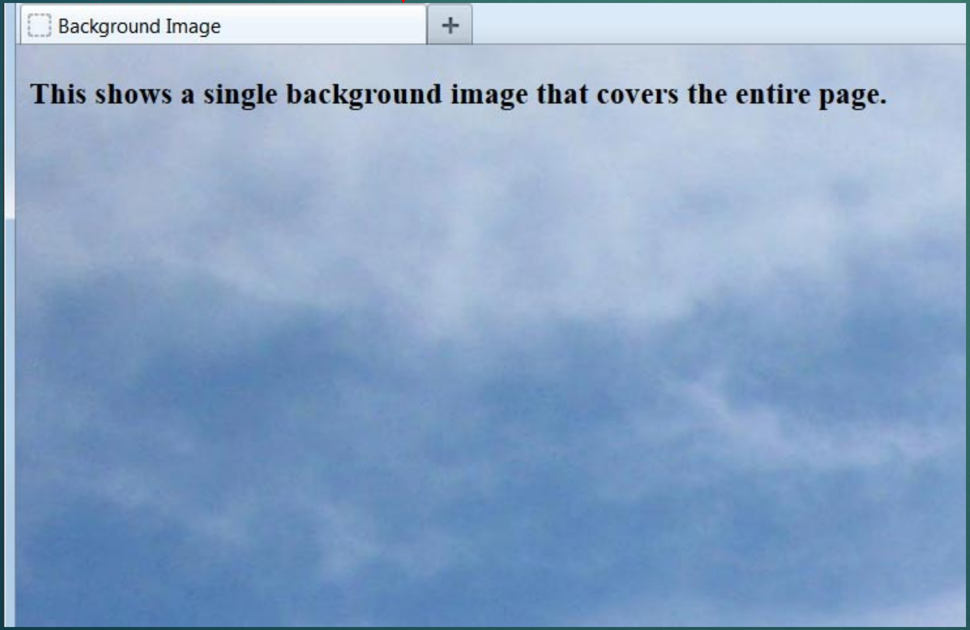
Let's take a look at some more features available to us via CSS, as well as some techniques to make our CSS declarations more precise and efficient:

- Setting images as the background for elements.
- Customizing how text links look and behave on the page.
- Centering a <div> element on a page.
- Adding CSS comments.
- Styling multiple selectors with a single CSS declaration.
- Using CSS inheritance to style elements on a page.

Setting a Background Image

Using CSS, we can place a background image on an element. By applying the style declaration to the **<body>** element, we can set the background for an entire page:

```
body {
  background-image: url('sky.jpg');
}
```



This shows a single background image that covers the entire page.

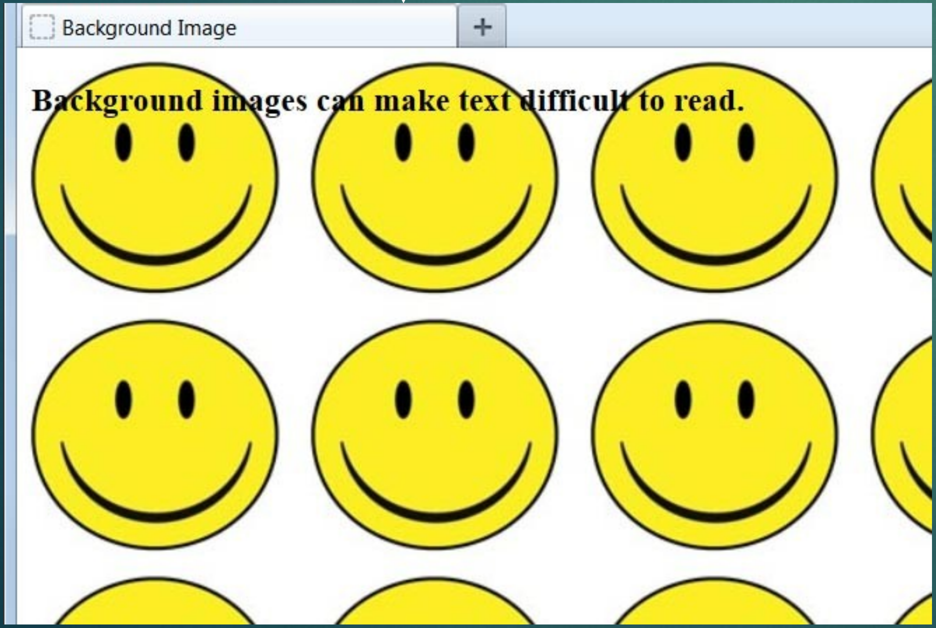
Notice that the name of the image file is surrounded by single quotes, not double quotes.

This image was large enough to take up the entire height and width of the browser window. Let's explore what will happen with smaller images.

Repeating Backgrounds

By default, the browser repeats our background image as many times as necessary, vertically and horizontally, until the entire window is filled. This is known as **tiling**. Let's use a smaller image than before:

```
body {
  background-image: url('smiley.jpg');
}
```



Background images can make text difficult to read.

If we don't want the background image to repeat over the entire page, we have three other options: tile horizontally, tile vertically, or no tiling at all.

Let's see how each looks with this image.

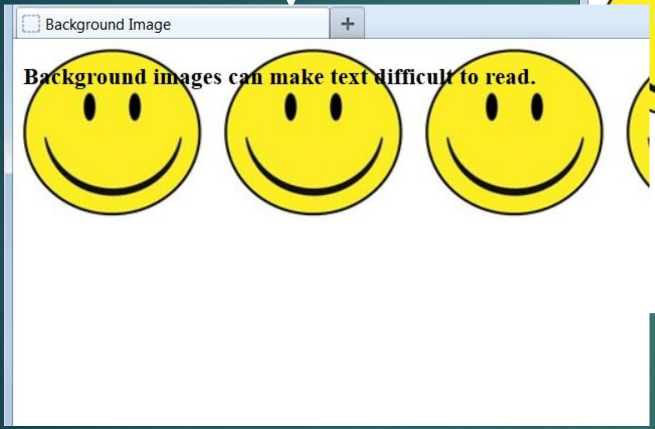
Tiling Options

By setting the **background-repeat** property, we can control how the background image tiles on the screen:

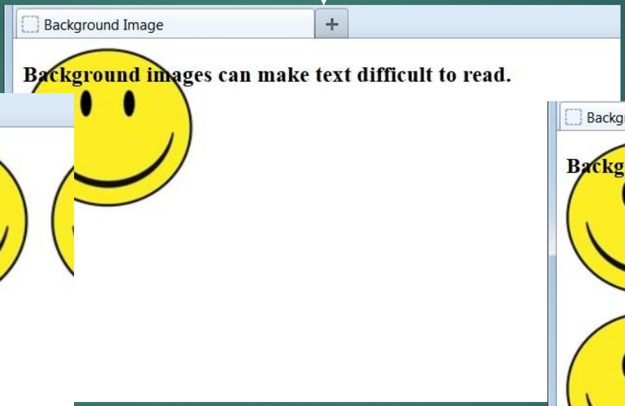
```
background-repeat: no-repeat;
```

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

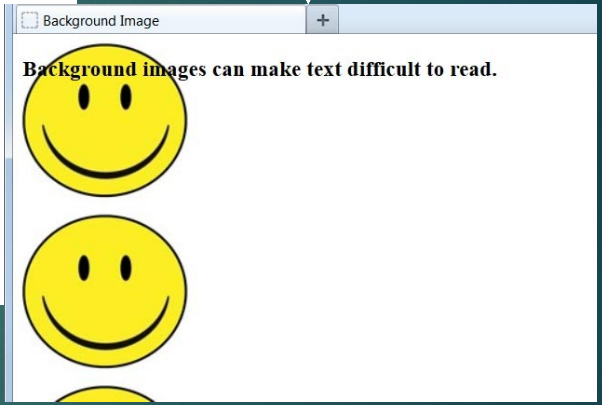
```
background-repeat: repeat-y;
```



Background images can make text difficult to read.



Background images can make text difficult to read.

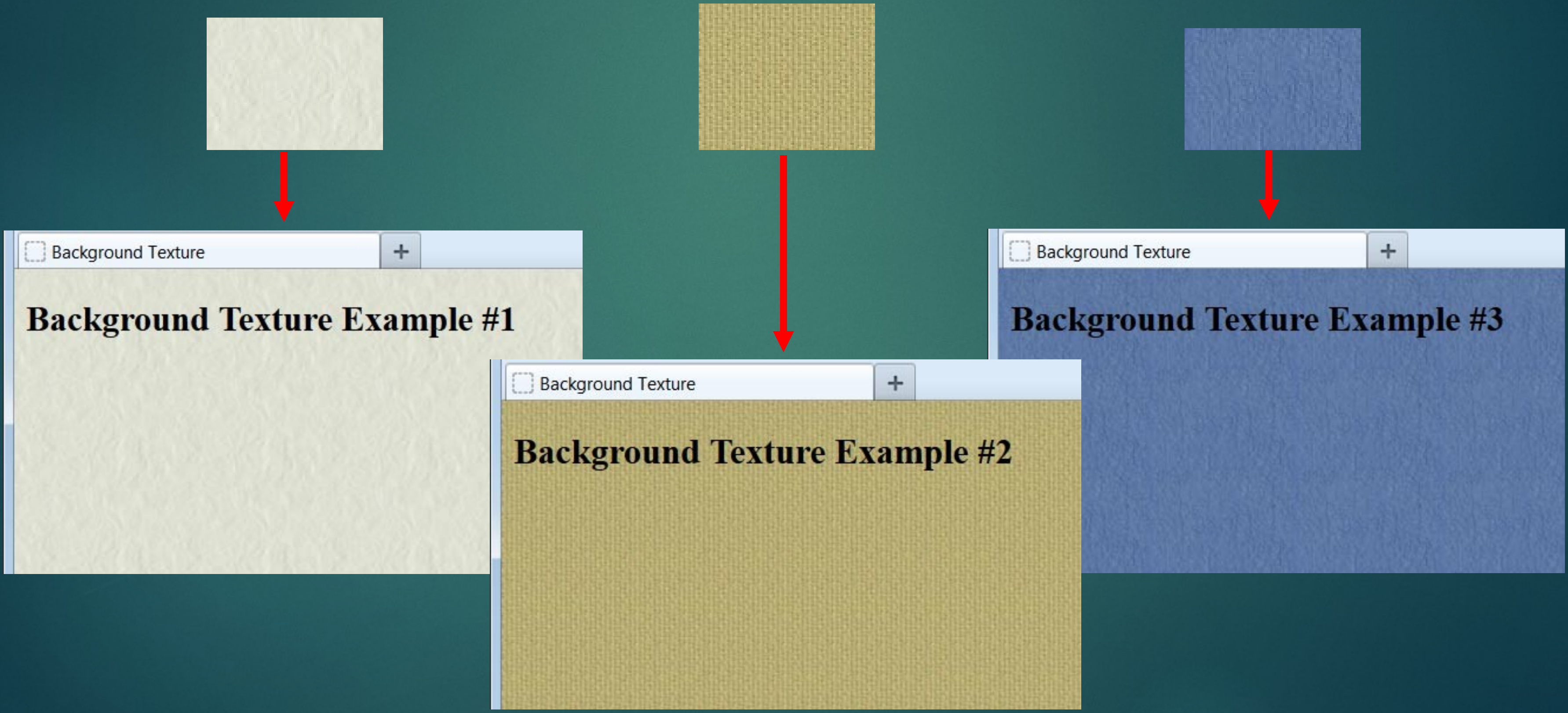


Background images can make text difficult to read.

An image such as this smiley face does not make for a pleasant background, especially when tiled across the page. Not only is it distracting but it also makes text difficult to read. Many web designers will use a small graphic file that looks nearly seamless when it repeats horizontally and vertically.

Background Textures

A repeated image that looks seamless is often referred to as a **texture** background and can make a website look more polished. Here are three sample textures:



Background Texture Example #1

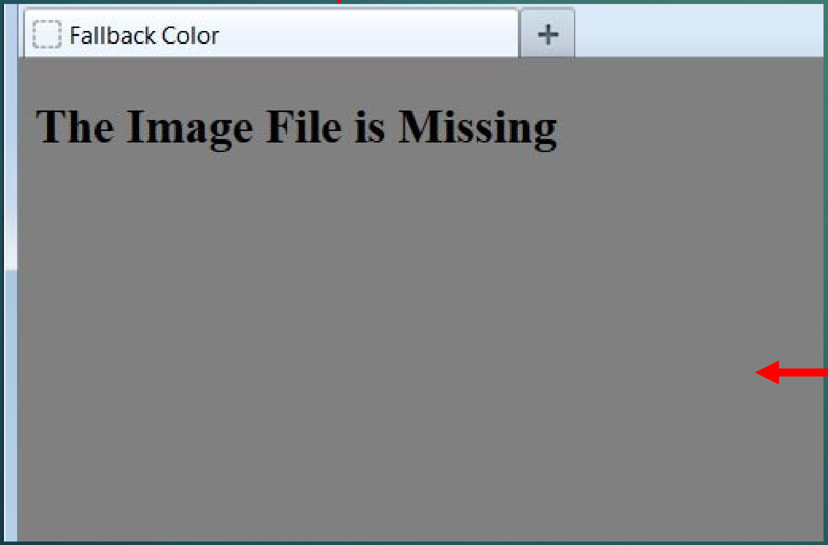
Background Texture Example #2

Background Texture Example #3

Setting a Fallback Color

Even if we set an image to be our page background, it's a good practice to set the background-color property also:

```
body {
  background-image: url('texture.jpg');
  background-color: gray;
}
```



The Image File is Missing

If the background image is found, the color will not show (unless the image has transparent portions.)

If for any reason the image file cannot be found, the browser will fall back to our color declaration.

Styling Text Links

Like other text on the page, link text (between the opening <a> and closing </a> tags) can be styled using CSS. Some of the properties that can be set are:

- font-family
- font-size
- font-style
- font-weight
- color
- background-color
- text-decoration

Link States

The <a> element is unusual in that it can be in one of four different conditions - or **states** - and we can define separate styles for each of these states. The four states are:

1. a:link = unvisited link (default state)
2. a:visited = previously visited link
3. a:hover = link being moused over
4. a:active = link being clicked

Have you ever noticed upon returning to a web page after clicking one of its links that the text link you clicked earlier is now a different color? That's due to most browsers automatically styling unvisited and visited links differently.

Let's see how we can override browser defaults and customize how text links look and behave on our web pages.

Example CSS Link Styling

```
<head>
  <style type="text/css">
    a:link {
      color: blue;
    }
    a:visited {
      color: green;
    }
    a:hover {
      color: orange;
    }
    a:active {
      color: red;
    }
  </style>
</head>
...
<a href="http://www.google.com">
  Link to Google
</a>
```

Link to Google

Link to Google

Link to Google

Link to Google

As we mouse over the link but before clicking on it.

As we are actually clicking on the link.

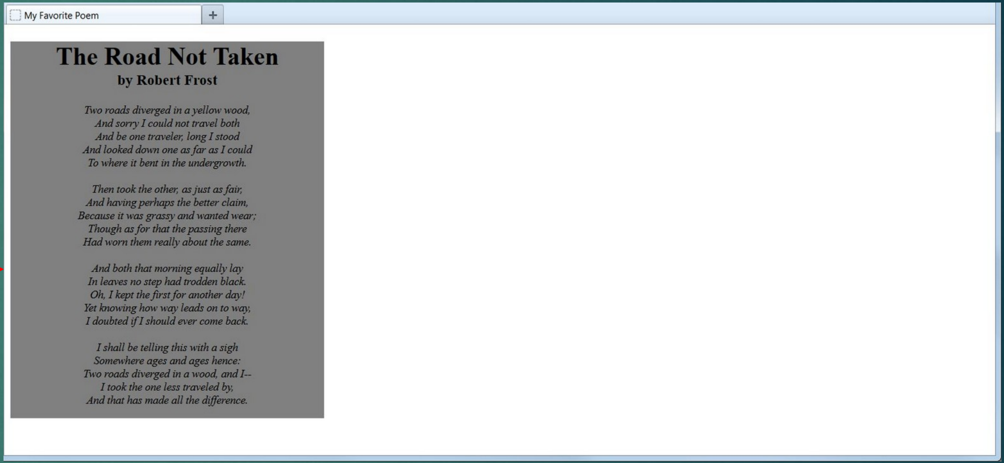
Warning #1: Do not put a space after a colon in the selector section. Doing so will result in the style declaration being ignored.

Warning #2: Always list these properties and states in the order shown. Mixing their order on your style sheet can result in some of the styling not being applied.

Different Screen Sizes

Because we don't know what the screen resolution will be for our visitors, it can be challenging to get our pages to display attractively for all viewers:

```
.textbox {
  width: 400px;
  height: 480px;
  background-color: gray;
  text-align: center;
}
...
<div class="textbox">
  <h1>The Road Not Taken</h1>
  <h3>by Robert Frost</h3>
  ...
</div>
```



The Road Not Taken

by Robert Frost

Two roads diverged in a yellow wood,
And sorry I could not travel both
And be one traveler, long I stood
And looked down one as far as I could
To where it bent in the undergrowth.

Then took the other, as just as fair,
And having perhaps the better claim,
Because it was grassy and wanted wear;
Though as for that the passing there
Had worn them really about the same.

And both that morning equally lay
In leaves no step had trodden black.
Oh, I kept the first for another day!
Yet knowing how way leads on to way,
I doubted if I should ever come back.

I shall be telling this with a sigh
Somewhere ages and ages hence:
Two roads diverged in a wood, and I--
I took the one less traveled by,
And that has made all the difference.

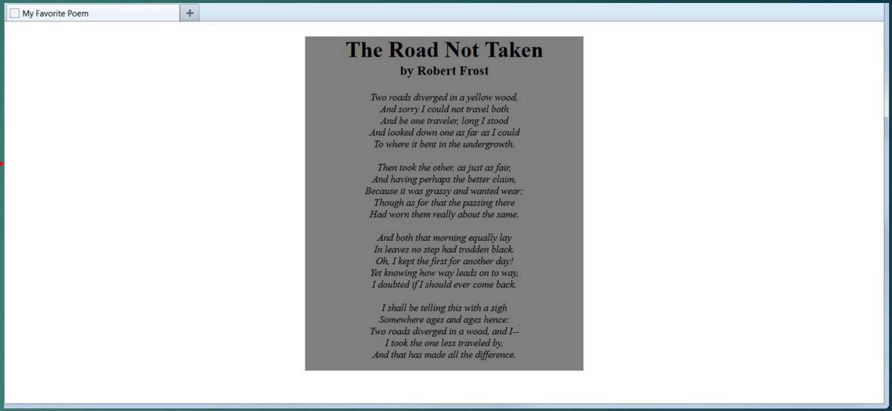
1280x720 Screen

If we tried to set a large left margin for this <div> to center it on a wide monitor screen such as this one, then viewers with smaller screens would have the content cut off and would have to scroll right to view it.

Centering a <div>

A technique to center a <div> on the screen, regardless of the screen resolution, is to set the right and left margins to **auto**:

```
.textbox {
  width: 400px;
  height: 480px;
  margin: 0px auto;
...
```



My Favorite Poem

The Road Not Taken

by Robert Frost

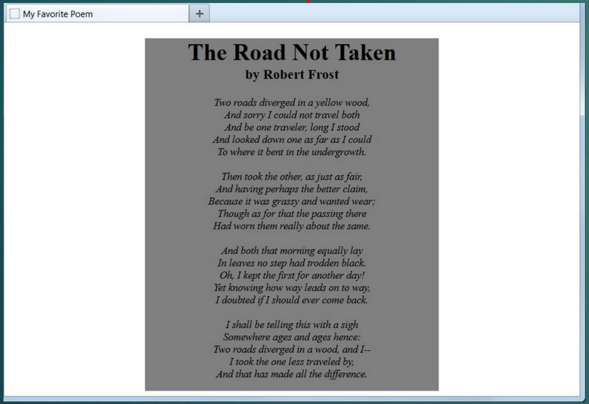
Two roads diverged in a yellow wood,
And sorry I could not travel both
And be one traveler, long I stood
And looked down one as far as I could
To where it bent in the undergrowth.

Then took the other, as just as fair,
And having perhaps the better claim,
Because it was grassy and wanted wear;
Though as for that the passing there
Had worn them really about the same.

And both that morning equally lay
In leaves no step had trodden black.
Oh, I kept the first for another day!
Yet knowing how way leads on to way,
I doubted if I should ever come back.

I shall be telling this with a sigh
Somewhere ages and ages hence:
Two roads diverged in a wood, and I--
I took the one less traveled by,
And that has made all the difference.

1280x720 Screen



My Favorite Poem

The Road Not Taken

by Robert Frost

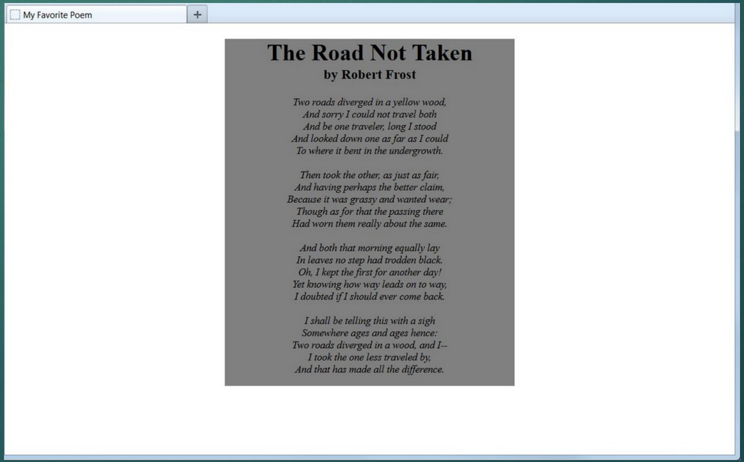
Two roads diverged in a yellow wood,
And sorry I could not travel both
And be one traveler, long I stood
And looked down one as far as I could
To where it bent in the undergrowth.

Then took the other, as just as fair,
And having perhaps the better claim,
Because it was grassy and wanted wear;
Though as for that the passing there
Had worn them really about the same.

And both that morning equally lay
In leaves no step had trodden black.
Oh, I kept the first for another day!
Yet knowing how way leads on to way,
I doubted if I should ever come back.

I shall be telling this with a sigh
Somewhere ages and ages hence:
Two roads diverged in a wood, and I--
I took the one less traveled by,
And that has made all the difference.

800x600 Screen



My Favorite Poem

The Road Not Taken

by Robert Frost

Two roads diverged in a yellow wood,
And sorry I could not travel both
And be one traveler, long I stood
And looked down one as far as I could
To where it bent in the undergrowth.

Then took the other, as just as fair,
And having perhaps the better claim,
Because it was grassy and wanted wear;
Though as for that the passing there
Had worn them really about the same.

And both that morning equally lay
In leaves no step had trodden black.
Oh, I kept the first for another day!
Yet knowing how way leads on to way,
I doubted if I should ever come back.

I shall be telling this with a sigh
Somewhere ages and ages hence:
Two roads diverged in a wood, and I--
I took the one less traveled by,
And that has made all the difference.

1024x768 Screen

CSS Comments

Just as we added useful comments to our XHTML markup, we can add comments to our CSS code:

```
.infobox {
  height: 200px;
  width: 150px; /*Do not make wider!*/
  background-color: teal;
}
/*
This footer section needs to be updated
   next January.
*/
.footer {
  text-align: center;
  /*font-weight: bold;*/
  font-size: 24px;
  color: brown;
}
```

Everything between the /* and */ will be disregarded, even if the comments span multiple lines.

CSS comments are often used when experimenting with styles or troubleshooting problems. Rather than deleting style declarations and then retyping them again, we can just "comment them out" temporarily, evaluate the effect, and then restore the style by removing the /* and */ characters.

Multiple Selectors

We can make our CSS code leaner and prevent extra typing by styling multiple selectors in the same declaration:

```
h1 {
  color: blue;
  text-align: center;
}
.summary {
  color: blue;
}
#header {
  color: blue;
  text-align: center;
}
a:link {
  color: blue;
}
```

=

```
h1, .summary, #header, a:link {
  color: blue;
}
h1, #header {
  text-align: center;
}
```

Just place a comma between each selector and the style will be applied to the entire list. Each of the CSS code samples above accomplishes the same thing.

CSS Inheritance

Some elements automatically inherit the CSS styling of the element that contains them. For example, the following properties affecting text will be inherited by **child** elements from their **parent** elements.

- font-family
- font-size
- font-style
- font-weight
- color
- text-align

We can exploit this fact to make our styling easier. For example, if we have a <div> that contains <h3>, <p>, <ul>, and <li> elements, and we want to style all the elements to have the same font and color, we don't need to style each separately. We can just style the <div> directly and all the elements will inherit the style.

Likewise, if we wanted the entire page to have the same text styling, we could style the <body> element itself.

A child element will inherit a style <u>only if there is not a more specific style applied to it</u>. As designers, we always have the ability to override inheritance when we need to.