

UP TO SPEED

Validate Your CSS

Just as you should make sure you've correctly written the HTML in your web pages using the W3C HTML validator (see the box on page 15), you should also check your CSS code to make sure it's kosher. The W3C provides an online tool for CSS checking as well: <http://jigsaw.w3.org/css-validator>. It operates just like the HTML validator: You can type the URL of a web page (or even just the address to an external CSS file), upload a CSS

file, or copy and paste CSS code into a web form and submit it for validation.

It's easy to make a typo when writing CSS, and one small mistake can throw all of your carefully planned designs out of whack. When your CSS-infused web page doesn't look as you expect, a simple CSS error may be the cause. The W3C CSS validator is a good first step when troubleshooting your designs.

TIP

If you have a page with an internal style sheet but want to use an external style sheet, then just cut all of the code between the `<style>` tags (without the tags themselves). Then create a new text file and paste the CSS into the file. Save the file with a .css extension—`styles.css`, for example—and link it to your page, as described next.

Once you create an external style sheet, you must connect it to the web page you wish to format. To do so, use the HTML `<link>` tag like this:

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

The `<link>` tag has two required attributes:

- **`rel="stylesheet"`** indicates the type of link—in this case, a link to a style sheet.
- **`href`** points to the location of the external CSS file on the site. The value of this property is a URL and will vary depending on where you keep your CSS file. It works the same as the `src` attribute you use when adding an image to a page or the `href` attribute of a link pointing to another page.

TIP

You can attach multiple style sheets to a web page by adding multiple `<link>` tags, each pointing to a different style sheet file. This technique is one way to organize your CSS styles, as you can see in Chapter 18.

Tutorial: Creating Your First Styles

The rest of this chapter takes you through the basic steps for adding inline styles, writing CSS rules, and creating internal and external style sheets. As you work through this book, you'll work through various CSS designs, from simple design elements to complete CSS-enabled web page layouts. To get started, download the tutorial files from https://github.com/sawmac/css_mm_4e. Download the zip archive containing the files. Each chapter's files are in a separate folder, named `02` (for Chapter 2), `03` (for Chapter 3), and so on.

NOTE In addition to a folder for each chapter's tutorial, you'll find another folder with the completed tutorial. For example, the *02_finished* folder contains the completed files for this chapter's tutorial. You can use this finished example if you get stuck and want to compare what you've written with the finished product.

Next, launch your favorite web page-editing software, whether it's a simple text editor like Notepad or TextEdit or a more full-featured editor like Sublime Text, Atom, or Dreamweaver.

Creating an Inline Style

When you type a CSS rule directly into a page's HTML, you're creating an *inline* style. Inline styles offer none of the time- and bandwidth-saving benefits of external style sheets, so the pros hardly ever use them. Still, in a pinch, if you absolutely must change the styling on a single element on a single page, then you may want to resort to an inline style. (For example, when creating HTML-formatted email messages, it's best to use inline styles. That's the only way to get CSS to work in Gmail, for one thing.) And if you do, you at least want the style to work properly. The important thing is to carefully place the style within the tag you want to format. Here's an example that shows you exactly how to do it:

1. **In your web page-editing program, open the file *02→index.html*.**

This simple-but-elegant HTML5 file contains a couple of different headings, a few paragraphs, and a copyright notice inside an `<address>` tag. You'll start by creating an inline style for the `<h1>` tag.

2. **Click Inside the opening `<h1>` tag and type `style="color: #6A94CC;"`.**

The tag should look like this:

```
<h1 style="color: #6A94CC;">
```

The `style` attribute is HTML, not CSS, so you use the equal sign after it and enclose all of the CSS code inside quotes. Only the stuff inside the quotes is CSS. In this case, you've added a property named `color`—which affects the color of text—and you've set that property to `#6A94CC`, a hexadecimal code for defining a color that's blue. (You'll learn more about coloring text on page 147.) Next, you'll check the results in a web browser.

3. **Open the *index.html* page in a web browser.**

For example, start up your favorite web browser and choose `File→Open File` (or press `Ctrl+O` [`⌘-O`] and select the *index.html* file in the *02 tutorial* folder from your computer; or just drag the file from the desktop—or wherever you've saved the tutorial files—into an open browser window). Many HTML editors also include a “Preview in Browser” function, which, with a simple keyboard shortcut or menu option, opens the page in a web browser. It's worth checking your program's documentation to see if it includes this time-saving feature.

When you view the page in a browser, the headline is now blue. Inline styles can include more than one CSS property. You'll add another property next.

4. **Return to your HTML editor, click after the semicolon following #6A94CC, and then type font-size: 3em;.**

The semicolon separates two different property settings. The `<h1>` tag should look like this:

```
<h1 style="color: #6A94CC; font-size: 3em;">
```

5. **Preview the page in a web browser. For example, click your browser window's Reload button (but make sure you've saved the HTML file first).**

The headline is now much larger. And you've had a taste of how labor-intensive inline styles are. Making all the `<h1>` headings on a page look like this one could take days of typing and add acres of code to your HTML files.

6. **Return to your page editor and delete the entire style property, which returns the heading tag back to its normal `<h1>`.**

Next, you'll create a style sheet within a web page. (You'll find a finished version of this part of the tutorial in the `O2_finished` folder in a file named `inline-style.html`.)

Creating an Internal Style Sheet

A better approach than inline styles is using a style sheet that contains multiple CSS rules to control multiple elements of a page. In this section, you'll create a style that affects all top-level headings in one fell swoop. This single rule automatically formats every `<h1>` tag on the page.

1. **With the file `index.html` open in your text editor, click directly after the closing `</title>` tag. Then hit Enter and type `<style>`.**

The HTML should now look like the following (the stuff you've added is in bold):

```
<title>CSS: The Missing Manual -- Chapter 2</title>
<style>
</head>
```

The opening `<style>` tag marks the beginning of the style sheet. It's always a good idea to close a tag right after you type the opening tag, since it's so easy to forget this step once you jump into writing your CSS. In this case, you'll close the `<style>` tag before adding any CSS.

2. **Press Enter twice and type `</style>`.**

Now, you'll add a CSS selector that marks the beginning of your first style.

3. **Click between the opening and closing `<style>` tags and type `h1 {`.**

The `h1` indicates the tag to which the web browser should apply the upcoming style.

The weird bracket thingy after the h1 is called an *opening brace*, and it marks the beginning of the CSS properties for this style. In other words, it says, "The fun stuff comes right after me." As with closing tags, it's a good idea to type the closing brace of a style before actually adding any style properties.

4. Press Enter twice and type a single closing brace: }.

As the partner of the opening brace you typed in the last step, this brace's job is to tell the web browser, "This particular CSS rule ends here." Now time for the fun stuff.

5. Click in the empty line between the two braces. Hit the Tab key, and type color: #6A94CC;.

You've typed the same style property as the inline version—color—and set it to #6A94CC;. The final semicolon marks the end of the property declaration.

NOTE

Technically, you don't have to put the style property on its own line, but it's a good idea. With one property per line, it's a lot easier to quickly scan a style sheet and see all the properties for each style. Also, the tab is another helpful visual organizing technique (you can also insert a few spaces instead). The indentation makes it easy to discern all of your rules at a glance, since the selectors (like h1 here) line up along the left edge, with the properties spaced a bit out of the way.

6. Press Enter again and add two additional properties, like so:

```
font-size: 3em;  
margin: 0;
```

Make sure you don't leave off the semicolon at the end of each line; otherwise, the CSS won't display correctly in a browser.

Each of these properties adds a different visual effect to the headline. The first assigns a size and font to the text, while the second removes space from around the headline. Part Two of this book covers all these properties in detail.

Congratulations—you've just created an internal style sheet. The code you've added should look like the bolded text:

```
<title>CSS: The Missing Manual -- Chapter 2</title>  
<style>  
h1 {  
    color: #6A94CC;  
    font-size: 3em;  
    margin: 0;  
}  
</style>  
</head>
```

7. Save the page and preview it in a web browser.

You can preview the page by opening it in a web browser as described in step 3 on page 28, or, if the page is still open in a browser window, just click the Reload button.

Next you'll add another style.

NOTE

Always remember to add the closing `</style>` tag at the end of an internal style sheet. When you don't, a web browser displays the CSS style code followed by a completely unformatted web page—or no web page at all.

8. Back in your text-editing program, click after the closing brace of the `h1` style you just created, press Enter, and then add the following rule:

```
p {  
    font-size: 1.25em;  
    color: #616161;  
    line-height: 150%;  
    margin-top: 10px;  
    margin-left: 60px;  
}
```

This rule formats every paragraph on the page. Don't worry too much right now about what each of these CSS properties is doing; later chapters cover these properties in depth. For now, just practice typing the code correctly and get a feel for how to add CSS to a page.

9. Preview the page in a browser.

The page is starting to shape up and should look like Figure 2-3. You can see what stylistic direction the page is headed in. You can see a completed version of this tutorial by opening the `02_finished→internal-stylesheet.html` file.

The process you've just worked through is CSS in a nutshell: Start with an HTML page, add a style sheet, and create CSS rules to make the page look great. In the next part of this tutorial, you'll see how to work more efficiently, using external style sheets.

FIGURE 2-3

CSS easily formats text in creative ways, letting you change fonts, text colors, font sizes, and a lot more, as you'll see in Chapter 6.

CSS: The Missing Manual

Sed ut perspiciatis unde omnis iste natus error sit voluptatem accusantium doloremque laudantium, totam rem aperiam, eaque ipsa quae ab illo inventore veritatis et quasi architecto beatae vitae dicta sunt explicabo. Nemo enim ipsam voluptatem quia voluptas sit aspernatur aut odit aut fugit, sed quia consequuntur magni dolores eos qui ratione voluptatem sequi nesciunt. Neque porro quisquam est, qui dolorem ipsum quia dolor sit amet, consectetur,

Lorem Ipsum Dolor Sat

Sed ut perspiciatis unde omnis iste natus error sit voluptatem accusantium doloremque laudantium, totam rem aperiam, eaque ipsa quae ab illo inventore veritatis et quasi architecto beatae vitae dicta sunt explicabo. Nemo enim ipsam voluptatem quia voluptas sit aspernatur aut odit aut fugit, sed quia consequuntur magni dolores eos qui ratione voluptatem sequi nesciunt. Neque porro quisquam est, qui dolorem ipsum quia dolor sit amet, consectetur, adipisci velit, sed quia non numquam eius modi tempora incident ut labore et dolore magnum aliquam quaerat voluptatem. Ut enim ad minima veniam, quis nostrum exercitationem ullam corporis suscipit laboriosam, nisi ut aliquid ex ea commodi consequatur? Quis autem vel eum iure reprehenderit qui in ea voluptate velit esse quam nihil molestiae consequatur, vel illum qui dolorem eum fugiat quo voluptas nulla pariatur?

Sed ut perspiciatis unde omnis iste natus error sit voluptatem accusantium doloremque laudantium, totam rem aperiam, eaque ipsa quae ab illo inventore veritatis et quasi architecto beatae vitae dicta sunt explicabo. Nemo enim ipsam voluptatem quia voluptas sit aspernatur aut odit aut fugit, sed quia consequuntur magni dolores eos qui ratione voluptatem sequi nesciunt. Neque porro quisquam est, qui dolorem ipsum quia dolor sit amet, consectetur, adipisci velit, sed quia non numquam eius modi tempora incident ut labore et dolore magnum aliquam quaerat voluptatem. Ut enim ad minima veniam, quis nostrum exercitationem ullam corporis suscipit laboriosam, nisi ut aliquid ex ea commodi consequatur? Quis autem vel eum iure reprehenderit qui in ea voluptate velit esse quam nihil molestiae consequatur, vel illum qui dolorem eum fugiat quo voluptas nulla pariatur?

Nisi Ut Aliquid

Sed ut perspiciatis unde omnis iste natus error sit voluptatem accusantium doloremque laudantium, totam rem aperiam, eaque ipsa quae ab illo inventore veritatis et quasi architecto beatae vitae dicta sunt explicabo. Nemo enim ipsam voluptatem quia voluptas sit aspernatur aut odit aut fugit, sed quia consequuntur magni dolores eos qui ratione voluptatem sequi nesciunt. Neque porro quisquam est, qui dolorem ipsum quia dolor sit amet, consectetur, adipisci velit, sed quia non numquam eius modi tempora incident ut labore et dolore magnum aliquam quaerat voluptatem. Ut enim ad minima veniam, quis nostrum exercitationem ullam corporis suscipit laboriosam, nisi ut aliquid ex ea commodi consequatur? Quis autem vel eum iure reprehenderit qui in ea voluptate velit esse quam nihil molestiae consequatur, vel illum qui dolorem eum fugiat quo voluptas nulla pariatur?

Sed ut perspiciatis unde omnis iste natus error sit voluptatem accusantium doloremque laudantium, totam rem aperiam, eaque ipsa quae ab illo inventore veritatis et quasi architecto beatae vitae dicta sunt explicabo. Nemo enim ipsam voluptatem quia voluptas sit aspernatur aut odit aut fugit, sed quia consequuntur magni dolores eos qui ratione voluptatem sequi nesciunt. Neque porro quisquam est, qui dolorem ipsum quia dolor sit amet, consectetur, adipisci velit, sed quia non numquam eius modi tempora incident ut labore et dolore magnum aliquam quaerat voluptatem. Ut enim ad minima veniam, quis nostrum exercitationem ullam corporis suscipit laboriosam, nisi ut aliquid ex ea commodi consequatur? Quis autem vel eum iure reprehenderit qui in ea voluptate velit esse quam nihil molestiae consequatur, vel illum qui dolorem eum fugiat quo voluptas nulla pariatur?

Copyright 2015, Wiliam Nella

Creating an External Style Sheet

Since it groups all of your styles at the top of the page, an internal style sheet is a lot easier to create and maintain than the inline style you created a few pages ago. Also, an internal style sheet lets you format any number of instances of a tag on a page, like every `<p>` tag, by typing one simple rule. But an external style sheet gets even better—it can store all of the styles for an entire website. Editing one style in the external style sheet updates the whole site. In this section, you'll take the styles you created in the previous section and put them in an external style sheet.

1. In your text-editing program, create a new file and save it as `styles.css` in the same folder as the web page you've been working on.

External style sheet files end with the extension `.css`. The file name `styles.css` indicates that the styles contained in the file apply throughout the site. (But you can use any file name you like, as long as it ends with the `.css` extension.) Start by adding a new style to the style sheet.

2. Type the following rule into the `styles.css` file:

```
html {  
    padding-top: 25px;  
    background-image: url(images/bg_page.png);  
}
```

This rule applies to the `HTML` tag—the tag that surrounds all other `HTML` tags on the page. The `padding-top` property adds space between the top of the tag and the content that goes inside it. In other words, what you just typed will add 25 pixels of space between the top of the browser window and the page's content. The `background-image` adds a graphic file to the page's background. The CSS `background-image` property can display the graphic in many different ways—in this case, the graphic will tile seamlessly from left to right and top to bottom, covering the entire browser window. You can read more about background image properties on page 231.

3. Add a second rule following after the rule you just typed to the `styles.css` file:

```
body {  
    width: 80%;  
    padding: 20px;  
    margin: 0 auto;  
    border-radius: 10px;  
    box-shadow: 10px 10px 10px rgba(0,0,0,.5);  
    background-color: #E1EDEB;  
}
```

This rule applies to the `<body>` tag—the tag that holds all the content visible in a web browser window. There are a lot of different things going on in this style, and each of these properties is covered in-depth later in the book. But in a nutshell, this style creates a box for the page's content that's 80 percent the width of the browser window, has a little bit of space inside that moves text from the edge of the box (that's the `padding` property), and centers the box on the page (that's the `margin` property, and the particular trick of centering a page's content is discussed in the tutorial starting on page 81). Finally, the box gets a light blue background color and a transparent black drop shadow.

Instead of recreating the work you did earlier, just copy the styles you created in the previous section and paste them into this style sheet.

4. Open the `index.html` page that you've been working on and copy all of the text inside the `<style>` tags. (Don't copy the `<style>` tags themselves.)

Copy the style information the same way you'd copy any text. For example, choose `Edit→Copy` or press `Ctrl+C` (`⌘-C`).

5. In the `styles.css` file, paste the style information by selecting `Edit→Paste` or pressing `Ctrl+V` (`⌘-V`).

An external style sheet never contains any HTML—that's why you didn't copy the `<style>` tags.

6. Save `styles.css`.

Now you just need to clean up your old file and link the new style sheet to it.

7. Return to the `index.html` file in your text editor and delete the `<style>` tags and all the CSS rules you typed in earlier.

You no longer need these styles, since they're in the external style sheet you're about to attach. In this tutorial, you'll take a dip into the exciting world of web fonts. You'll learn all about web fonts starting on page 126, but the basic idea is that you can use nearly any font you want in a web page—even a font that your site's visitors don't have installed on their own computers—simply by providing a link to that font file. There are many different ways to use web fonts, but in this example, you'll use Google's web font service.

8. In the space where the styles used to be (between the closing `</title>` tag and the closing `</head>` tag), type the following:

```
<link href='http://fonts.googleapis.com/css?family=Varela+Round'  
      rel='stylesheet'>
```

Again, don't worry about the details yet. All you need to know for now is that when a web browser encounters this link, it downloads a font named Varela Round from a Google server, and your CSS styles can freely use it.

Next, you'll link to the external style sheet you created earlier.

9. After the `<link>` tag you added in the previous step, type:

```
<link href="styles.css" rel="stylesheet">
```

The `<link>` tag specifies the location of the external style sheet. The `rel` attribute simply lets the browser know that it's linking to a style sheet.

NOTE

In this example, the style sheet file is in the same folder as the web page, so using the file's name for the `href` value provides a simple "document-relative" path. If it were in a different folder from the page, then the path would be a bit more complicated. In either case, you'd use a *document-* or *root-relative* path to indicate where the file is. The routine is the same as when you create a link to a web page or set a path to an image file when using the HTML `` tag. (For a brief primer on document- and root-relative links, visit www.kirupa.com/html5/all_about_file_paths.htm.)

10. Save the file and preview it in a web browser.

You'll see the same text styles for the `<h1>` and `<p>` tags that you created in the internal style sheet on page 29. In addition, there's now a speckled, tan background (the background image you applied on the `<html>` tag), as well as

a light-colored, greenish-blue box. That box is the `<body>` tag, and its width is 80 percent that of the browser window. Try resizing the browser window and notice that the box changes width as well. There's also a drop shadow on the box; you can see through the drop shadow to the speckled background. That's thanks to a special color type—`rgba` color—that includes a transparency setting (you'll read about it on page 149). Also notice that the corners of the box are rounded, thanks to the `border-radius` property.

NOTE

If the web page doesn't have any formatting (for example, the top headline isn't big, bold, and dusty brown), then you've probably mistyped the code from step 6 or saved the `styles.css` file in a folder other than the one where the `index.html` file is. In this case, just move the `styles.css` into the same folder.

Now, you'll use the web font you linked to in step 8.

11. In your text editor, return to the `styles.css` file. For the `h1` style, add the following two lines:

```
font-family: 'Varela Round', 'Arial Black', serif;  
font-weight: normal;
```

The finished style should look like this (additions in bold):

```
h1 {  
    font-family: 'Varela Round', 'Arial Black', serif;  
    font-weight: normal;  
    color: #6A94CC;  
    font-size: 3em;  
    margin: 0;  
}
```

If you preview the page now, you'll see the new font, Varela Round, for the headline.

NOTE

If you don't see the new font—it should have rounded tips on the ends of all the letters as pictured in Figure 2-4—then one of a couple of things could be wrong. If you're not connected to the Internet, you won't be able to download the font from Google; second, you may have mistyped either the `<link>` tag (see step 8 on the previous page), or the `font-family` declaration (line 2 above).

To demonstrate how useful it can be to keep your styles in their own external file, you'll attach the style sheet to another web page.

12. Open the file `02→another_page.html`.

This page contains some of the same HTML tags—`h1`, `h2`, `p`, and so on—as the other web page you've been working on.

13. Click after the closing `</title>` tag and press Enter.

You'll now link to both the web font and the external style sheet.

14. Type the same <link> tags you did in steps 8 and 9.

The web page code should look like this (the code you just typed appears in bold):

```
<title>Another Page</title>
<link href='http://fonts.googleapis.com/css?family=Varela+Round'
      rel='stylesheet'>
<link href="styles.css" rel="stylesheet">
</head>
```

15. Save the page and preview it in a web browser.

Ta-da! Just two lines of code added to the web page is enough to instantly transform its appearance. To demonstrate how easy it is to update an external style sheet, you'll do so by editing one style and adding another.

16. Open the `styles.css` file and add the CSS declaration `font-family: "Palatino Linotype", Baskerville, serif;` at the beginning of the `p` style.

The code should look like this (the bold text is what you've just added):

```
p {
    font-family: "Palatino Linotype", Baskerville, serif;
    font-size: 1.25em;
    color: #616161;
    line-height: 150%;
    margin-top: 10px;
    margin-left: 60px;
}
```

In this case, you're not using a web font, but relying on the site visitor to already have one of the fonts listed on his machine (you'll learn all about using fonts on page 121). Next, create a new rule for the `<h2>` tag.

17. Click at the end of the `p` style's closing `}`, press Enter, and add the following rule:

```
h2 {
    color: #B1967C;
    font-family: 'Varela Round', 'Arial Black', serif;
    font-weight: normal;
    font-size: 2.2em;
    border-bottom: 2px white solid;
    background: url(images/head-icon.png) no-repeat 10px 10px;
    padding: 0 0 2px 60px;
    margin: 0;
}
```

Some of these CSS properties you've encountered already. Some are new—like the border-bottom property for adding a line underneath the headline. And

some—like the background property—provide a shortcut for combining several different properties—in this case, the background-image and background-repeat—into a single property. Don't worry about the specifics of these properties; you'll learn them all in great detail in upcoming chapters (Chapter 6 covers font properties; Chapter 8 covers backgrounds; Chapter 7 covers padding and margins).

The styles you've created so far affect mainly tags—the h1, h2, and p—and they affect every instance of those tags. In other words, the p style you created formats every single paragraph on the page. If you want to target just one paragraph, you need to use a different kind of style.

18. Click at the end of the h2 style's closing }, press Enter, and add the following rule:

```
.intro {  
    color: #666666;  
    font-family: 'Varela Round', Helvetica, sans-serif;  
    font-size: 1.2em;  
    margin-left: 0;  
    margin-bottom: 25px;  
}
```

If you preview the *index.html* page in a web browser, you'll see that this new style has no effect...yet. This type of style uses a *class selector*, which formats only the specific tags you apply the class to. In order for this new style to work, you need to edit some HTML.

19. Save the file *styles.css* and switch to the *index.html* file in your text editor. Locate the opening <p> tag following the <h1> tag and add class="intro" so the opening tag looks like this:

```
<p class="intro">
```

You don't have to add a period before the word *intro* as you did when you created the style in step 18 (you'll learn why in the next chapter). This little extra HTML applies the style to the first paragraph (and only that one paragraph).

Repeat this step for the *another_page.html* file—in other words, add class="intro" to the first <p> tag on that page.

20. Save all the files and preview both the *index.html* and *another_page.html* files in a web browser.

Notice that the appearance of both pages changes, based on the simple edits you made to the CSS file. Close your eyes and imagine your website has a thousand pages. Aaaahhhhhh, the power.

You've got one last change to make. If you look at the very bottom of the page in your browser, you'll see the copyright notice. It's a little small, and it isn't lined up with the paragraphs above. Also, it would look better if it shared the same formatting as the other paragraphs.

21. In your text editor, return to the *styles.css* file. Locate the style that has the *p* selector. Type a comma, a space, and *address*.

The style looks like this:

```
p, address {  
    font-family: "Palatino Linotype", Baskerville, serif;  
    font-size: 1.25em;  
    color: #616161;  
    line-height: 150%;  
    margin-top: 10px;  
    margin-left: 60px;  
}
```

You haven't changed any of the style's properties—you've just changed the selector. In fact, you've just created a group selector. A group selector is a very efficient way to apply the same styling to a bunch of different page elements, and you'll learn more about them on page 49. In this case, the style applies to two tags: the *<p>* tag and the *<address>* tag.

22. Close *styles.css* file and reload the *index.html* file in your web browser.

The finished page should now look like Figure 2-4. (You'll find a completed version of this tutorial in the *02_finished* folder.)

For added practice, spend a few minutes playing around with the *styles.css* file. Try different values for the style sheet properties. For example, try a different number for the width property of the body style, or try different numbers for the font sizes.

FIGURE 2-4

Using an external style sheet, you can update an entire site's worth of web pages by editing a single CSS file. In addition, by moving all of the CSS code out of an HTML document and into a separate file, you cut down on the file size of your web pages, so they load faster.

What Have We Here?

Sed ut perspiciatis unde omnis iste natus error sit voluptatem accusantium doloremque laetus, ut enim
voleret, neque ipsa sumit ab illis invenire debet et quasi architecto beatus sit deus non implorens.
Quem enim operum iniquitatem quis voluntate et experientia sit nisi sit fugit, sed quis
conscientiam suam habet et illius voluntate non videntur. Quibus sumit cursum est, qui
quoniam ipsorum suis doloribus amens, conseruant amicos nulli, sed quis non sumit quae non modi tempora
volent ut lassus et dolores tristitia diram ducant voluptates.

• Another Page...

Sed ut perspiciatis unde omnis iste natus error sit voluptatem accusantium doloremque laetus, ut enim
voleret, neque ipsa sumit ab illis invenire debet et quasi architecto beatus sit deus non implorens. Nam
enim ipsorum iniquitatem quis voluntate et experientia sit nisi sit fugit, sed quis conscientiam magis dolens et
qui voluntate iniquitatem non videntur. Quibus sumit cursum est, qui dolent ipsorum quis doloribus amens,
conseruant amicos nulli, sed quis non sumit quae non modi tempora volent ut lassus et dolores magnam
dolorum quamvis iniquitatem. Tunc autem vel cum non reprobaretur qui in ea
voluptate vel cum ipsius nichil resistentem conseruant, vel ilium qui dolent cum fugit quis voluptas nulla
possunt?

• To Link a Style Sheet To...

Sed ut perspiciatis unde omnis iste natus error sit voluptatem accusantium doloremque laetus, ut enim
voleret, neque ipsa sumit ab illis invenire debet et quasi architecto beatus sit deus non implorens. Nam
enim ipsorum iniquitatem quis voluntate et experientia sit nisi sit fugit, sed quis conscientiam magis dolens et
qui voluntate iniquitatem non videntur. Quibus sumit cursum est, qui dolent ipsorum quis doloribus amens,
conseruant amicos nulli, sed quis non sumit quae non modi tempora volent ut lassus et dolores magnam
dolorum quamvis iniquitatem. Tunc autem vel cum non reprobaretur qui in ea
voluptate vel cum ipsius nichil resistentem conseruant, vel ilium qui dolent cum fugit quis voluptas nulla
possunt?

Copyright 2010, Prentice Hall