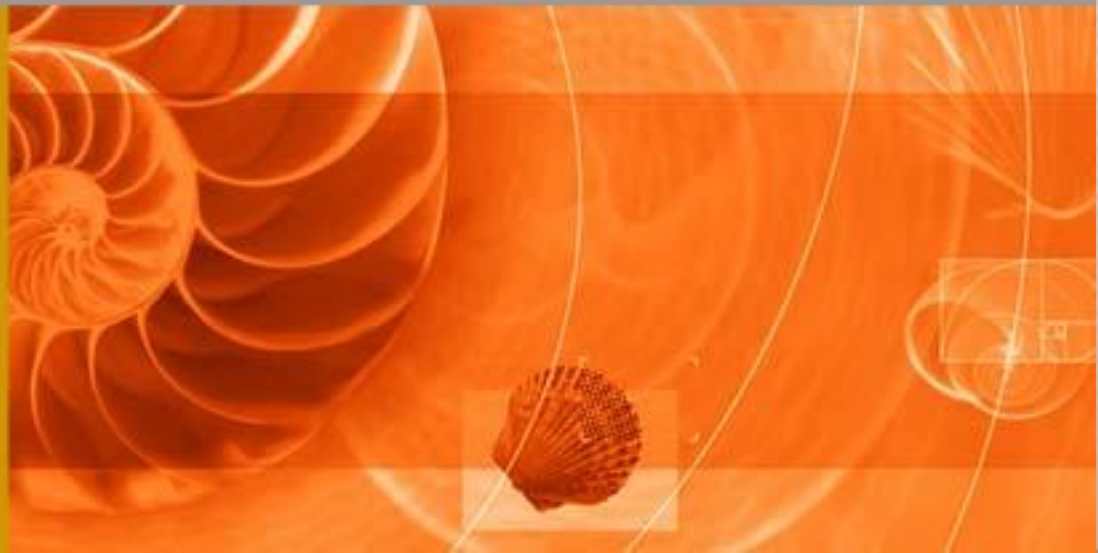


Fundamentals
of **WEB
DESIGN**



Classes, IDs, & Style Tag

Cascading Style Sheets

- Up until now we have applied style rules to HTML using what is called inline styles, which adds style to HTML tags via the style attribute.
- Now we will learn how to use style sheets to apply style across a Web page or Web site.
- Most of the examples will be similar to what you have already learned except that we will be applying the style rules differently.

- The more tags, attributes, and properties there are in a Web page, the larger the page is, thus, the slower it will load.
- Inline styles take up at least as much space as the HTML attributes they replace, thus, they are bulky and will cause the pages to load slower than they might.
- Style sheets will give you a way to define style rules once, then recycle them as needed. This can drastically reduce the page size, thereby increasing the speed at which the page loads.

About Style Sheets

- A Style sheet is a collection of style rules that apply to the entire Web page or Web site instead of just individual tags.
- Style sheets make updating and changing your style rules easier since they are all defined in a single location.
- For example, say you wanted all `<h1>` tags within a Web page or Web site to be 24 point and blue. Using inline styles you would need to apply style rules every time you used the `<h1>` tag.
- With style sheets, you can define the rule once and apply it to all `<h1>` tags within the page or site.

The <style> tag

- The <style> tag allows you to define style rules for the entire Web page.
- It is defined within the <head> section of the Web page and can contain all of the style rules for that page.
- The collection of style rules within the style tags is considered an embedded style sheet.

The <style> tag

- In the example below, the style rules for the <body> tag and the "font-size" property for the <p> tag have been defined within the <style> tag. When the page is displayed, the <body> and <p> tags will inherit the style rules defined within the <style> tag.

```
<head>
<title>Style Tag Example</title>

<style>
body {background-color : #ffffff; color : #000000;}
p    {font-size : 20px;}
</style>

</head>
```

This text will display at the browser default size.

This text will display with a height of 20 pixels.

This text will display at the browser default size.

Backward Compatibility

- In browsers that do not know about CSS, the style rules within the opening and closing `<style>` tags will be displayed.
- This is because, like all other unknown tags, the `<style>` tags would be ignored by non-compatible browsers and the content left over would be considered regular content.
- To keep this from happening, you should capture the style rules within an HTML comment tag as shown in the example below.

Selector

```
<style>
<!--
body {background-color : #ffffff; color : #000000;}
p    {font-size : 20px;}
-->
</style>
```

Declaration

Embedded Style Sheets

- An embedded style sheet is simply style rules that have been applied to a Web page using the <style> tag.
- This should not be confused with using the "style" attribute within the HTML tags which is referred to as inline styles.
- Below is another example using an embedded style sheet. This time the <body>, <div>, <p>, and style rules are defined within the <style> tag.

```
<style>
<!--
body {background-color : #ffffff;}
div  {background-color: #ccffcc;
      border    : solid 1px #000000;
      color     : #000000;
      padding   : 8px;
      position  : relative;
      width    : 350px;
      height   : 300px;}
p    {font-size : 14pt;}
img  {float     : left;
      margin    : 15px 15px 15px 15px;}
-->
</style>
```

CSS Comments

- A CSS comment begins with "/*", and ends with "*/".
- CSS comments, which are ignored by the browser, are used to explain your CSS code.
- Comments can help you when you edit your source code at a later date.

```
<style>
/* this is a comment */
<!--
body {background-color : #ffffff; color : #000000;}
p    {font-size : 20px;}
-->
</style>
```

Different Styles Across The Same Tag

- The "class" attribute in HTML can be used to allow style rules to be applied to specific tags. By defining separate classes, you can choose which rules apply to which tags without having to apply the style directly to the tags.

Different Styles Across The Same Tag

- The example below applies the blue class to two <p> tags and the red class to the other.
- The "font-size" is also set for all <p> tags, thus, effects all of them.
- The class rule is defined by placing a period in front of the name for the class when defining the style rules.

```
<style>
<!--
body {background-color : #ffffff; color : #000000;}
p    {font-size : 20px;}
.red {color : red;}
.blue {color : blue;}
-->
</style>

</head>

<body>
<p class="blue">
This text will display in blue.
</p>

<p class="red">
This text will display in red.
</p>

<p class="blue">
This text will display in blue.
</p>
```

Same Style Across Different Tags

- The "class" attribute can also be used to apply the same style to different tags without having to define the style twice.
- The example below is the same as the last one, but a <h1> tag has been added with the red class applied to it.

```
<style>
<!--
body {background-color : #ffffff; color : #000000;}
p    {font-size : 20px;}
.red {color : red;}
.blue {color : blue;}
-->
</style>

</head>

<body>
<h1 class="red">
This Header Will Display Red
</h1>

<p class="blue">
This text will display in blue.
</p>

<p class="red">
This text will display in red.
</p>

<p class="blue">
This text will display in blue.
</p>
```

Style Grouping

- Grouping allows you to define a set of style rules that are for multiple tags.
- In this example, the "font-size" rule has been defined for the <p> tag, then the "color:#00ff00" rule defined for both the <p> and <h1> tags. Thus, unless a class is associated with one of these tags, the color for both tags will be green, as seen in this example.

```
<style>
<!--
body {background-color : #ffffff; color : #000000;}
p    {font-size : 20px;}
p, h1 {color : #00ff00;}
.red {color : #ff0000;}
.blue {color : #0000ff;}
-->
</style>
```

Defining Style For Only One Tag

- The "id" attribute in HTML can be used to allow style rules to be applied to one specific tag.
- Each "id" defined should only be applied to one HTML tag.
- This example applies the green "id" to the first <p> tag, blue to the next, and red to the last.
- The "font-size" is also set for all <p> tags, thus, effects all of them.
- The "id" rule is defined by placing a "#" in front of the name for the "id" when defining the style rules.

```
<style>
<!--
body {background-color : #ffffff; color : #000000;}
p {font-size : 20px;}
#green {color : #00ff00;}
#red {color : #ff0000;}
#blue {color : #0000ff;}
-->
</style>

</head>

<body>

<p id="green">
This text will display in green.
</p>

<p id="blue">
This text will display in blue.
</p>

<p id="red">
This text will display in red.
</p>
```

External Style Sheets

- You are not limited to inline styles and embedded styles.
- You can also create a separate file that contains the style rules then link to it from within your html document.
- In this file you do not include the `<style>` tags, only the style rules.

External Style Sheets

- In this example the style from one of our previous examples has been placed in a file called mystyles.css then linked to your HTML document from within the <head> section.
- Both the external style sheet and the HTML file linking to it have been listed below.

External style sheet: (mystyles.css)

```
body {background-color : #ffffff; color : #000000;}
p    {font-size : 20px;}
p, h1 {color : #00ff00;}
.red {color : #ff0000;}
.blue {color : #0000ff;}
```

HTML file linking to the mystyles.css:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html lang="en-us">

<head>
<title>Style Tag Example</title>

<link rel="stylesheet" type="text/css" href="mystyles.css">
```

External Style Sheets

- The power of external style sheets lies in the fact that you can define style rules once in a .css file, then link to that file within every page of a Web site.
- If you need to change a style rule, you can change it once in the .css file and it effects every page in the Web site.

Style Inheritance

- Style inheritance refers to the priority order in which style rules are applied.
- Inline styles take precedence over embedded styles, and embedded styles take precedence over external styles.

Style Inheritance

- The example below has an external file with the `<p>` rule set to #00ff00 "green", the embedded `<p>` rule set to red, and an inline `<p>` rule set to blue. Notice how the results are colored to better understand how the inheritance works.

This Header Will Display Green

This paragraph will display red even though the .css file says green, because the embedded style rules take precedence.

This paragraph will display blue even though the .css file says green and the embedded style rules says red, because the inline style takes precedence over both.

```
<body>
<h1>
This Header Will Display Green
</h1>

<p>
This paragraph will display red even though<br />
the .css file says green, because the embedded<br />
style rules take precedence.
</p>

<p style="color:blue">
This paragraph will display blue even though<br />
the .css file says green and the embedded<br />
rules says red, because the inline style takes<br />
takes precedence over both.
</p>

<br />
<br />
</body>
```

External style sheet: (mystyles.css)

```
body {background-color : #ffffff; color : #000000;}
p {font-size : 20px;}
p, h1 {color : #00ff00;}
.red {color : #ff0000;}
.blue {color : #0000ff;}
```

HTML file linking to the mystyles.css:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html lang="en-us">

<head>
<title>Style Tag Example</title>

<link rel="stylesheet" type="text/css" href="mystyles.css">

<style>
<!--
body {background-color : #ffffff; color : #000000;}
p {font-size : 20px; color : red}
-->
</style>
```