

The Basics of Cascading Style Sheets (CSS)

Irina McGuire

Graphic Designer | Front-End Web Developer

www.irinamcguire.com

December 3, 2010

CSS

Introduction

- What do you know about CSS?
- What do you hope to do with CSS?
- How familiar are you with HTML?

Examples of beautiful CSS Web sites:

www.csszengarden.com

One content, many layouts.

CSS

Presentation Summary

- What is CSS?
- CSS & HTML
- The Box Model
- Style Sheet Implementation
- CSS Rule Structure
- HTML & DIVs
- Common CSS properties
- CSS Cascade and Inheritance
- Resources



What is CSS?

CSS stands for
Cascading Style Sheet.

Typical CSS file is a text file with an extension **.css** and contains a series of commands or rules.

These rules tell the HTML how to display.

*To create a style sheet, create a file using Notepad (PC) or Text Edit (Mac), save it as a .css document and start writing the CSS code (see right).

Style.css

```
/* Styles for sitename.com*/  
  
body {  
    font-family:Arial;  
    background: #000;  
}  
  
#container {  
    text-align:left;  
    width:1020px;  
}  
  
#header {  
    height:232px;  
}  
  
#footer {  
    width: 100%;  
    padding: 0 10px;  
    margin-bottom: 10px;  
}
```

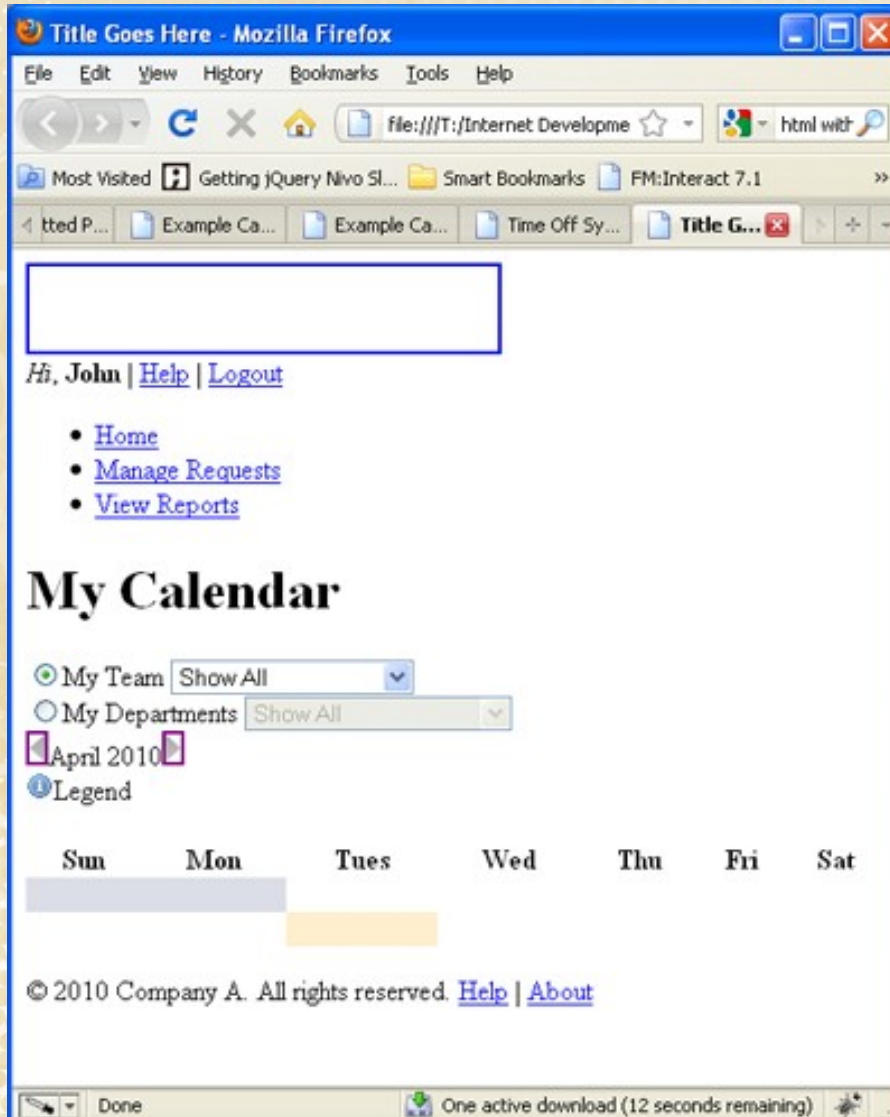
And so on....

CSS Benefits

- Separates structure from presentation
- Provides advanced control of presentation
- Easy maintenance of multiple pages
- Faster page loading
- Better accessibility for disabled users
- Easy to learn



HTML Without CSS



“HTML without CSS is like a piece of candy without a pretty wrapper.”

Without CSS, HTML elements typically flow from top to bottom of the page and position themselves to the left by default.

With CSS help, we can create containers or DIVs to better organize content and make a Web page visually appealing.



HTML & CSS

- HTML and CSS work together to produce beautiful and functional Web sites
- HTML = structure
- CSS = *style*

CSS

The Box Model



CSS works on the box model. A typical Web page consists of many boxes joined together from top to bottom. These boxes can be stacked, nested, and can float.

Header
Navigation
Content
Footer



Attaching a Style Sheet

Attach a style sheet to a page by adding the code to the <head> section of the HTML page. There are **3 ways** to attach CSS to a page:

1. External Style Sheet: Best used to control styling on multiple pages.

```
<link rel="stylesheet" type="text/css"
media="all" href="css/styles.css" />
```

2. Internal Style Sheet: Best used to control styling on one page.

```
<style type="text/css">
h1 {color: red}
</style>
```

3. Inline Style Sheet*: CSS is not attached in the <header> but is used directly within HTML tags.

```
<p style="color: red">Some Text</p>
```



CSS Rule Structure

A CSS RULE is made up of a selector and a declaration. A declaration consists of property and value.

```
selector {property: value;}
```



declaration



Selectors

A selector, here in green, is often an element of HTML.

```
body { property: value; }  
h1 { property: value; }  
em { property: value; }  
p { property: value; }
```



Properties and Values

```
body {background: purple;}  
h1 {color: green; }  
h2 {font-size: large;}  
p {color: #ff0000;} /*hexadecimal for  
red*/
```

Properties and values tell an HTML element how to display.

```
body {  
background: purple;  
color: green;  
}
```

*CSS code can be written in a linear format (above) or in a block format (below).



Grouping Selectors

Group **the same selector** with different declarations together on one line.

```
h1 {color: black;}
```

```
h1 {font-weight: bold;}
```

```
h1 {background: white;}
```

Example of grouping selectors (both are correct):

```
h1 {  
  color: black;  
  font-weight: bold;  
  background: white;  
}
```



Grouping Selectors

Group **different selectors** with the same declaration on one line.

```
h1 {color: yellow;}
```

```
h2 {color: yellow;}
```

```
h3 {color: yellow;}
```

Example of grouping selectors (both are correct):

```
h1, h2, h3 {color: yellow;}
```



Comments in CSS

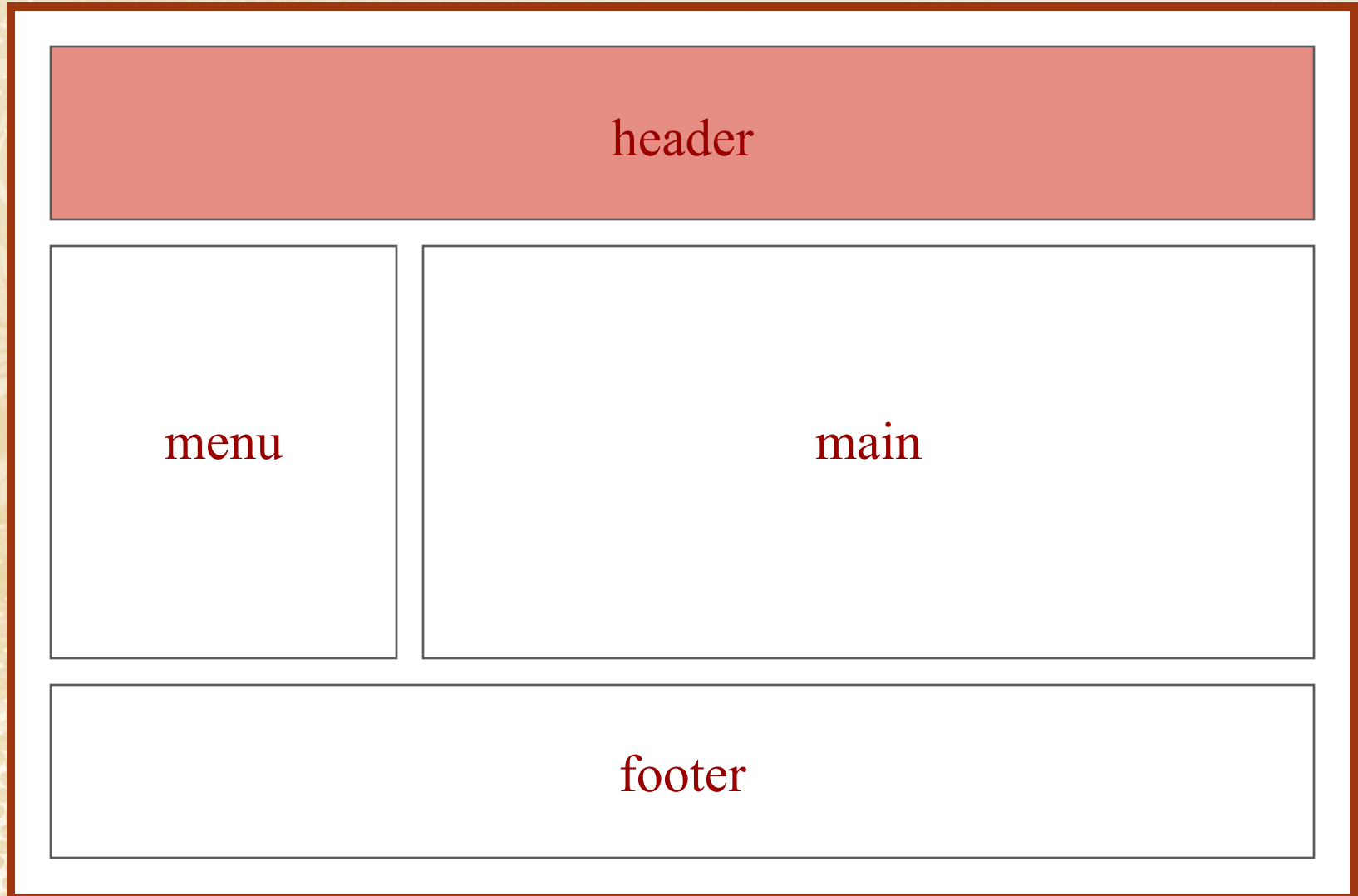
- Explain the purpose of the coding
- Help others read and understand the code
- Serve as a reminder to you for what it all means
- Starts with `/*` and ends with `*/`

```
p {color: #ff0000;} /*Company Branding*/
```



Typical Web Page (Browser)

Container



Typical Web Page (HTML)

Typical HTML Web page is made up of containers (boxes) or DIVs. Each DIV is assigned an ID or a Class.

```
<div id="container">  
    <div id="header">Insert Title</div>  
    <div id="main">content  
        <div id="menu">content</div>  
    </div>  
    <div id="footer">content</div>  
</div>
```



Typical Web Page (CSS)

The CSS file uses the same DIV/ID/Class names as the HTML and uses them to style the elements.

```
#container {property: value;}
```

```
#menu {property: value;}
```

```
#main {property: value;}
```

```
#footer {property: value;}
```



IDs and Classes

- **IDs (#)** are unique and can only be used once on the page
- **Classes (.)** can be used as many times as needed

HTML Code:

```
<h1 id="mainHeading">Names</h1>
```

```
<p class="name">Joe</p>
```

CSS Code:

```
#mainHeading {color: green}
```

```
.name {color: red}
```



CSS Box Properties

- Background-color
- Width
- Padding
- Margin
- Border-width
- Border-color
- Border-style



HTML

div id="header"

div id="content"

div id="footer"

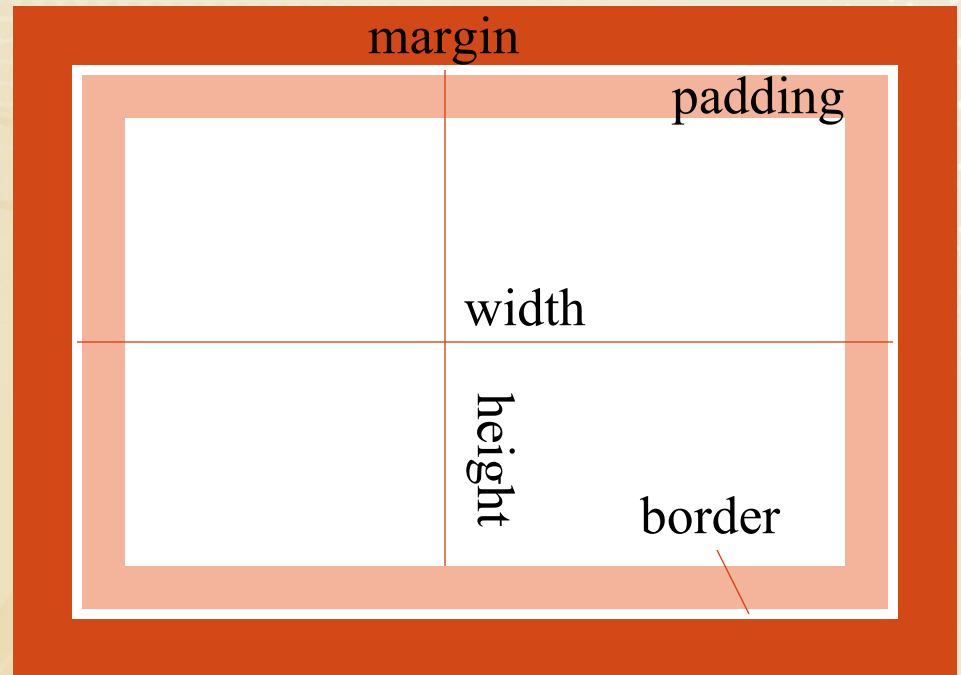
CSS

```
#content {  
background-color: #ccc;  
margin-bottom: 10px;  
border: 1px dashed blue;  
color: #fff;  
width: auto;  
}
```

CSS

Common CSS Layout Properties

- Width
- Height
- Float
- Clear
- Border
- Padding
- Margin



CSS

Width & Height

Width and height define the width and height of an element.

```
div id="box"
```

```
#box {width="50px" }
```

```
#box {width="50em" }
```

```
#box {width="100%" }
```

```
#box {width="auto" }
```

```
#box {height="auto" }
```

*Width and height can be specified in pixels, ems, percentages or set to auto



Float: (left, right)

Float property makes elements float to the right or left of the screen, positioned where they are in the HTML. Floating allows word wrapping.

```
div id="box"
```

Here is some text which wraps around the box floated to the left.

```
#box {float:left; margin-right: 10px;}
```



Clear: (left, right, both)

When elements are floated, they wrap around each other to form a “caravan.” The **clear** property detaches an element from the “caravan” and allows it to start on a new line.

```
div id="box1"
```

```
div id="box2"
```

```
div id="box3"
```

```
#box3 { background-color: white; border:  
1px solid #000; clear: both; }
```

Border (top, right, bottom, left)

```
div id="box"
```

You can define the entire border or only the top, bottom, left, or right. You can also define the border using one declaration. The code could be any of the following:

```
#box {  
border-top: red dotted 1px;  
border-bottom: red dotted 1px;  
border-left: red dotted 1px;  
border-right: red dotted 1px;  
}
```

```
#box {  
border: red dotted 1px;
```

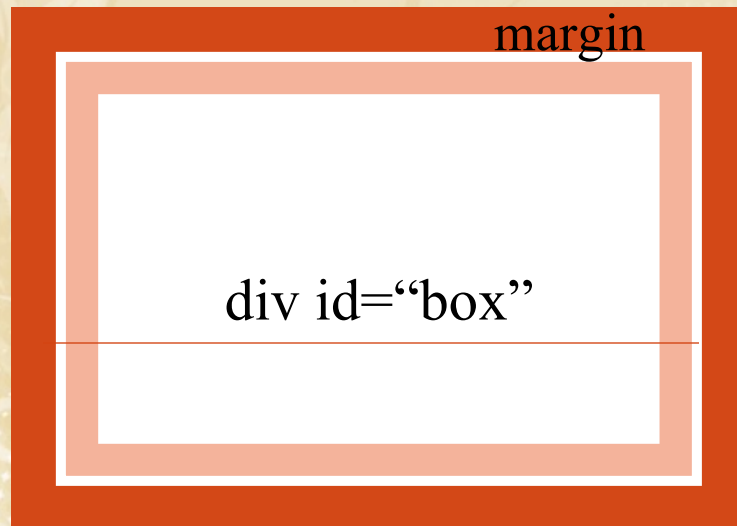
```
#box {  
border-color: red;  
border-style: dotted;  
border-width: 2px;
```



Margin (top, right, bottom, left)

Margin is the space outside the text/content and the border. You can use margin for all around the element or specify each side of the rectangle separately.

The code could be any of the following:



```
margin: 10px;
```

or

```
margin: 10px 10px;
```

or

```
margin: 10px 10px 10px 10px;
```

or

```
margin-left: 10px;
```

```
margin-right: 10px;
```

```
margin-bottom: 10px;
```

```
margin-top: 10px;
```



Text Properties

MAIN HEADING

Gravida lacinia velit.
Vivamus tortor enim,
tincidunt at, pellentesque ut,
iaculis eu, quam.

To style the main heading in the paragraph above, we assigned a class the HTML tag.

```
.mainHeading {  
  color: red;  
  letter-spacing: 5px;  
  text-transform: uppercase;  
  word-spacing: 15px;  
  text-align: left;  
  font-family: Times;  
  text-decoration: underline;  
  font-size: 12px;  
  font-style: italic;  
  font-weight: bold;  
}
```

```
<h3 class="mainHeading">Main Heading</h3>
```



CSS Colors

Standard

- White
- Black
- Blue
- Fuchsia
- Gray
- Green
- Lime
- Aqua

Hexadecimal

- #ffffff
- #fff
- #cccf0f3

CSS

Styling Links

The links property defines how inactive, hovered, active, and visited link states appear to the user.

```
a:link {color: red; text-decoration: none; border-bottom: 1px dashed red; background: white;}
```

```
a:visited {color: yellow;}
```

```
a:active {color: green;}
```

```
a:hover {color: orange;}
```



Including Images

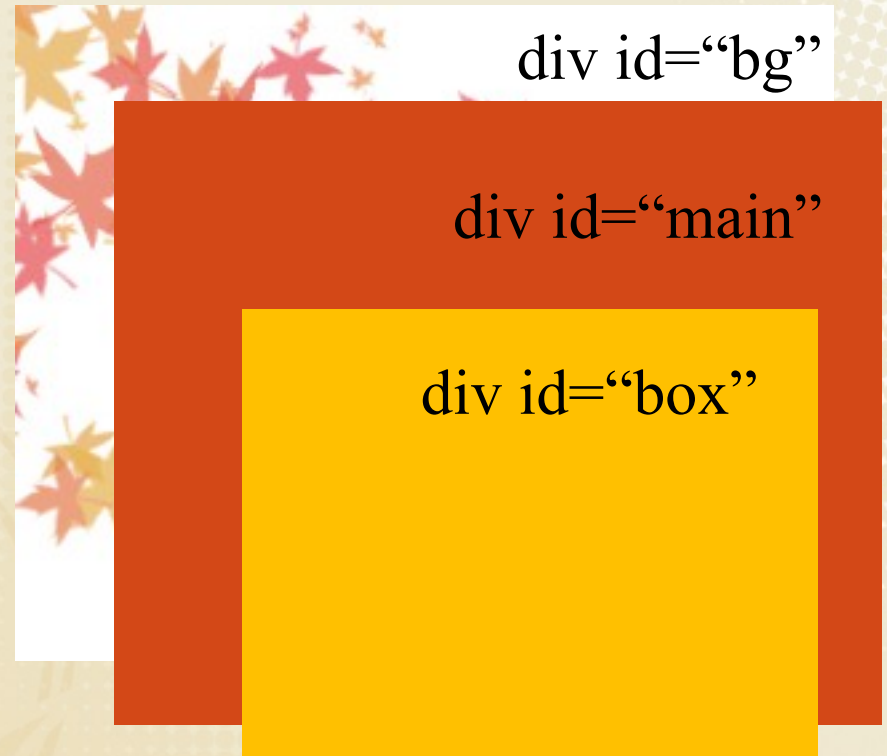
Properties for working with images include:

- Background-image
- Background-repeat
- Background-position
- Background-attachment



Layering

Background colors and images are layered like sheets of paper one on top of the other.



```
#bg {background:url(leaves.jpg) no-repeat top left}
```

```
#main {background-color: red}
```

```
#box {background-color: yellow}
```



Background-Image

The background-image property sets an image in the background of an element.



Background images and colors are layered.



If not transparent, the last one listed in the CSS file is visible.

```
li {  
  background-image:url (flower.jpg);  
  padding-left: 10px;  
}
```



Background-Repeat



The background-repeat property sets an image in the background of an element and tiles, or repeats, it. Tiling is the default.

```
li {  
background-image:url (flower.jpg);  
background-repeat:no-repeat;  
}
```

- Possible Values >
- repeat
 - repeat-x (horizontal)
 - repeat-y (vertical)
 - no-repeat



Image Positioning

The background-position property positions the image using either combined keywords (top, bottom, left, right, and center); length values; or percentage values.

```
background-position: right top;  
/*can also use number values*/
```

```
background-attachment: fixed; /*  
*can also use 'scroll'*/
```

left

center

top

top



left

center

right

bottom

bottom

bottom

The background-attachment property fixes or scrolls an image in the browser window. Values include *fixed* and *scroll*.



The Power of Cascade

When multiple styles or style sheets are used, they start to cascade and sometimes compete with one another due to CSS's inheritance feature. Any tag on the page could potentially be affected by any of the tags surrounded by it.

So, which one wins? Nearest Ancestor Wins.

4. Inline style or directly applied style

5. The last style sheet declared in the <header> section



Saving Time with Inheritance

In a nutshell, **inheritance** (not the money you get from your grandma) is the process by which CSS properties applied to one tag are passed on to nested tags.

For example, the paragraph tag will inherit the same styling as the body tag because `<p>` is always located inside `<body>`.

```
<body style="font-family: Arial">  
    <p>This text will be Arial as well</p>  
</body>
```

So, instead of styling each paragraph separately, you can define the font color in the `<body>`, and everything inside will have that color.



Resources

- http://www.w3schools.com/css/css_reference.asp (list of all CSS properties)
- <http://www.w3schools.com/css/>
- <http://www.glish.com/css/>
- <http://www.html.net/tutorials/css/>
- <http://blog.html.it/layoutgala/>

Great Book

“CSS: The Missing Manual” - by David Sawyer McFarland

CSS Galleries

<http://www.cssbeauty.com/gallery/>

www.cssdrive.com

<http://www.css-website.com>



Thank You

I hope you enjoyed this presentation and learned some basic CSS. Good luck with creating beautiful and functional Web sites.

CSS