

Girl Develop It Albuquerque

Intro to HTML & CSS: Day Two

Supplemental Material

January 2016

Hello!

Welcome to Girl Develop It Albuquerque's Introduction to HTML & CSS class. This packet contains material and notes that supplement the lecture slides.

Your Instructors

Rochelle Lewis

Front End Web Developer and Web Development Educator
CNM STEMulus Deep Dive Coding Bootcamp

GitHub: [rlewis2892](#)

Twitter: [@RochelleUXUI](#)

Web: www.rochellelewis.com

Email: hello@rochellelewis.com

Josh Garcia

Web Developer and Co-Founder of Eleven Online

Web: 11online.us

Email: josh@11online.us

What We'll Cover

Lecture slides and content are available at:

<http://gdi.rochellelewis.com/>

1. What is CSS
2. CSS Rules and Syntax
3. CSS Selectors
4. Ids vs. Classes
5. Descendant Selectors
6. CSS Properties and Values
 - a. Color
 - b. Background
 - c. Font
 - d. Shorthand
7. Connecting HTML and CSS
 - a. Inline
 - b. Embedded
 - c. Linked
8. "Cascading"

CSS Day Two

Today we'll be adding CSS to our HTML file that we created in Class One. With CSS, we can add some style to the simple HTML web page that we created previously.

We'll be using our Text Editor program (Atom or Sublime Text) to create a new CSS file in our project folder.

What is CSS?

CSS stands for "Cascading Styles Sheets", and is the "style language" of the web. It has a different **syntax** and different **rules** than HTML. CSS allows you to add visual style to the HTML elements on your page. Colors, borders, backgrounds, font styles, and more.

CSS works in conjunction with HTML, but it is not HTML. It is a different language.

CSS Rules & Syntax



- A "block" of CSS code is a **rule**.
- The rule starts with a **selector**.
- It has sets of **properties** and **values**.
- A property-value pair is a **declaration**.

Selectors

The **Selector** is how we choose an HTML element, or elements, to apply a CSS style rule to.

Declarations

- **Declarations** contain the **property** and **value** of a CSS style you plan use on an HTML element, or elements.
- Declarations always end with a semicolon.
- Declaration groups are surrounded by curly brackets.

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

CSS Selectors

The **Selector** is how we choose an HTML element, or elements, to apply CSS declarations (properties and values) to. We will explore 4 kinds of CSS selectors today:

1. Element Selector
2. id Selector
3. class Selector
4. Descendant Selector

Element Selector

To apply CSS to a particular type of HTML element, simple type that element as the **selector** in your CSS.

Selects ALL <p> HTML elements:

```
p {  
    property: value;  
}
```

Selects ALL HTML elements:

```
img {  
    property: value;  
}
```

id Selector

To apply CSS to HTML elements with a specific **id**, type the id name preceded by a # (hash or pound) symbol as the selector.

Selects all elements with an id of "footer":

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

The associated HTML.

```
<p id="footer">Copyright 2011</p>
```

Class Selector

To apply CSS to HTML elements with a specific **class**, type the class name preceded by a . (period) as the selector.

To apply CSS to Selects all elements with a class of "warning".

```
.warning {  
  color: red;  
}
```

The associated HTML.

```
<p class="warning">Run away!</p>
```

Descendant Selector

To select specific **child elements** in your HTML, type them together separated by a space as the selector.

Selects all elements that are inside <p> tags.

```
p em {  
  color: yellow;  
}
```

The associated HTML.

```
<p>This is <em>important.</em></p>
```

The descendant selector also works with id and class selectors as well. The following example selects all HTML elements with a class name "red" that are inside HTML elements with an id of "warning".

```
#warning .red {  
  color: red;  
}
```

The associated HTML.

```
<p id="warning">This is some <span class="red">red text.</span></p>
```

Ids vs. Classes

Ids should only apply to ONE unique element per webpage. For example, a web page has only one footer.

- The # (hash symbol) is how you tell CSS "this is an id."

Many elements can have the same **class**. For example, there can be many "warnings" on each web page.

- The . (period) is how you tell CSS "this is a class name."

A common practice is to use **classes** primarily to style web pages, and **ids** on a limited basis, and only when necessary.

CSS Properties and Values

Many CSS properties have self-explanatory names:

- background-color
- font-family
- font-size
- color
- width
- height

A comprehensive list of all available CSS Properties can be found here:

<https://developer.mozilla.org/en-US/docs/Web/CSS/Reference>

Color

The **color** property changes the color of text.

```
p {  
  color: red;  
}
```

Colors can be set via:

- Color name
`color: red;`
- Hexadecimal value - a hash followed by 6 digits
`color: #ff0000;`

- RGB and RGBA values - rgb/a, followed by a series of digits for R, G, B and A.

```
color: rgb(255, 0, 0);
color: rgba(255, 0, 0, 1);
```

- ...and more.

For a list of the web-friendly **color names** that are currently widely supported, refer to the Standard X11 Color Scheme names here:

https://en.wikipedia.org/wiki/X11_color_names#Color_name_chart

For comprehensive documentation featuring all of the ways to use the **color property**, refer to MDN's documentation here: <https://developer.mozilla.org/en-US/docs/Web/CSS/color>

Background Color

The background-color property changes the color of the background of an HTML element.

```
.warning {
  background-color: #ff0000;
}
```

Background color can also be set using the **background** property as well.

```
div {
  background: #000000;
}
```

Font Styles

The **font-family** property defines which font is used.

```
p {
  font-family: Arial;
}
```

With font-family, you can declare font styles in the following ways:

- By declaring a specific font name - "Helvetica", "Times New Roman", "Garamond", etc.
- By the generic font style name - serif, sans-serif, monospace, etc.
- Using a comma-separated list of different font names. This will create "fallbacks" that the browser can load in case a specific font style is unavailable.

```
p {
  font-family: "Helvetica", Arial, sans-serif;
}
```

The **font-size** property changes the size of an element's text.

```
p {  
  font-size: 12px;  
}  
  
h1 {  
  font-size: 100%;  
}  
  
div {  
  font-size: 1.5em;  
}
```

Font sizes are typically set using one of the following **units of measure**:

- Pixels - **px**
- **em** - a relative unit of measure
- Percentage - %

You can set more than one font style property at a time using the **font property**:

```
p {  
  font-style: italic;  
  font-weight: bold;  
  font-size: 10px;  
  font-family: sans-serif;  
}
```

OR

```
p {  
  font: italic bold 10px sans-serif;  
}
```

Full documentation on the **font property** can be found here:

<https://developer.mozilla.org/en-US/docs/Web/CSS/font>

Connecting HTML & CSS

There are three ways to add CSS to your HTML file:

1. Inline CSS
2. Embedded CSS
3. Linked CSS

Inline CSS

```
<p style="color:red;">Some text here.</p>
```

- Inline CSS is added directly to the HTML element, in your HTML file
- Uses the **style** HTML attribute
- Difficult to use and maintain in large projects
- **NOT preferred! Generally considered a bad practice.**

Embedded CSS

```
<head>
  <style type="text/css">
    p {
      color: blue;
      font-size: 12px;
    }
  </style>
</head>
```

- CSS is written inside the HTML <head> element, in your HTML file.
- Uses the HTML <style> tag.
- These CSS rules apply **ONLY** to the page that they are written in.
- Generally not a preferred method to add CSS to web sites.

Linked CSS

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

- A separate CSS file is created, and linked to a web page in the HTML <head> tag.
- Uses the HTML <link /> tag.
- One CSS file can be shared across many web pages at a time.
- Reduces file size & bandwidth.
- Easy to scale and maintain in larger projects.
- **Preferred by developers everywhere!**

Cascading

CSS styles "cascade" down until changed.

CSS:

```
p {
  color: blue;
  font-family: 'Helvetica', sans-serif;
}

.red {
  color: red;
}

.green {
  color: green;
}

#special {
  background: #000000;
}
```

HTML:

```
<h2>My Heading</h2>
<p class="green">Paragraph One</p>
<p class="red">Paragraph Two</p>
<p class="red" id="special">Paragraph Three</p>
```

Class Activity

- Create a new file inside your HTML project folder named **style.css**
- Add a <link/> tag in the <head> of your HTML document that loads your new CSS file.
See the “linked CSS” example.
- Add CSS rules to change the text colors, background colors, and fonts of different parts of your HTML content.
- Try using **ids** and **classes** to change specific HTML elements