


Intro to CSS

 <https://github.com/learn-co-curriculum/dsc-intro-to-css>  <https://github.com/learn-co-curriculum/dsc-intro-to-css/issues/new/choose>

Introduction

Browsers combine the content (HTML) and presentation (CSS) layers to display web pages. CSS is the language for *styling* web pages.

CSS instructions live apart from the HTML elements and have a different look and feel ("syntax"). CSS directives give web pages their specific look and feel. If you have ever been impressed by how a website can be displayed on a desktop browser while the same content looks great on a mobile device, you have CSS to thank for it!

One of the most amazing displays of the power of CSS is the [CSS Zen Garden](http://www.csszengarden.com/)  [\(http://www.csszengarden.com/\)](http://www.csszengarden.com/) where people take the *exact same* HTML page and use CSS to create *wildly* different pages. Take a look!

The Beauty of CSS Design

A demonstration of what can be accomplished through CSS-based design. Select any style sheet from the list to load it into this page. Download the example [html](#) file and [css](#) file

So What is This About?

There is a continuing need to show the power of CSS. The Zen Garden aims to excite, inspire, and encourage participation. To





All of the differences you see are due to CSS! Astounding, right!?

Objectives

You will be able to:

- List the components of CSS
- Declare CSS properties and values

Recognize The Differences Between HTML And CSS

HTML and CSS play two different roles. When you write HTML, it's important to focus on structure, hierarchy, and meaning — the "marking-up" of content. Questions in the mind of an HTML author are:

- Is it best to list these members' names with numbers or bullets?
- Does this menu belong in the navigation in the header?
- Should this additional reference be an aside or a separate section?

These questions deal with structure, hierarchy, and meaning, which are concerns of the content layer (HTML).

When defining the presentation layer (CSS), here are the questions to ask yourself:

- Do I want the header menu to be stationary or do I want it to scroll with the browser window?

- How do I want the content to display inside of a container? For example, does it fill the whole area, edge-to-edge? Is there white space around the content and/or the container?
- How large should an **H1** be relative to an **H2** ? What about an **H3** ?
- What properties should links have? Underline or no underline? Which color for the normal state versus the hover state? Should the visited link state be different?
- How should the content appear when on a desktop machine versus a mobile device?

As you ask yourself these questions, your focus is on the *aesthetic* quality of the page. For each bit of *content*, we can define a **presentation rule** that will change the way the HTML is displayed.

List the Components of CSS

For each **presentation rule**, there are 3 things to keep in mind:

1. What is the specific HTML we want to style?
2. What are the qualities we want to modify (e.g. the properties of text in a paragraph)?
3. *How* do we want to modify the qualities of the element (e.g. font family, font color, font size, line height, letter spacing, etc.)?

Once you've decided what to modify and how we can start writing CSS rules.

CSS **selectors** are a way of declaring which HTML elements you wish to style. Selectors can appear a few different ways:

- The type of HTML element(**h1** , **p** , **div** , etc.)
- The value of an element's **id** or **class** (`<p id='idvalue'></p>` , `<p class='classname'></p>`).
- The value of an element's attributes (`value="hello"`)
- The element's relationship with surrounding elements (a **p** within an element with class of **.infobox**)

For example, if you want the body of the page to have a black background, your selector syntax may be **html** or **body** . For anchors, your selector would be **a** . A few more examples are listed below:


```
/*
The CSS comment syntax is text between "slash-star" and "star-slash"
*/

/*
selects all anchor tag elements in the document (e.g. <a href="page-link.html">Page Link<,
*/
a

/*
selects all headers of type h3 in the document (e.g. <h3>Type selectors</h3>)
*/
```

h3

```
/*
selects all paragraph elements in the document (e.g. <p>Type selectors are used
to...</p>)
*/
n
```

[Type selectors documentation](https://developer.mozilla.org/en-US/docs/Web/CSS/Type_selectors)  (https://developer.mozilla.org/en-US/docs/Web/CSS/Type_selectors)

The element type `class` is a commonly used selector. Class selectors are used to **select all elements that share a given class name**. The class selector syntax is: `.classname`. Prefix the class name with a `'`(period).

```
/*
select all elements that have the 'important-topic' classname (e.g. <h1 class='important-t
and <h1 class='important-topic'>)
*/
.important-topic

/*
select all elements that have the 'welcome-message' classname (e.g. <p class='helpful-hin
and <p class='helpful-hint'>)
*/
.helpful-hint
```

You can also use the `id` selector to style elements. However, **there should be only one element with a given id** in an HTML document. This can make styling with the ID selector ideal for one-off styles. The `id` selector syntax is: `#idvalue`. Prefix the id attribute of an element with a `#` (which is called "octothorpe," "pound sign", or "hashtag").

```
/*
selects the HTML element with the id 'main-header' (e.g. <h1 id='main-header'>)
*/
#main-header

/*
selects the HTML element with the id 'welcome-message' (e.g. <p id='welcome-message'>)
*/
#welcome-message
```

[id selectors documentation](https://developer.mozilla.org/en-US/docs/Web/CSS/ID_selectors)  (https://developer.mozilla.org/en-US/docs/Web/CSS/ID_selectors)

Declare CSS Properties and Values

Each element has a list of qualities that can be styled. CSS "property" names identify those qualities. For text styling, examples of property names include `color` , `text-align` and `line-height` .

CSS Property Values are directly related to property names. If you are working with the `color` property, the value could be a named color such as `red` , or `#660000` . Some properties have their values set with words, others with numbers, and some can take both.

A CSS property name with a CSS property value is a **CSS declaration**. To apply a CSS declaration like `color: blue` to a specific HTML element, you need to combine your CSS declaration with a CSS selector. The association between one or more CSS declarations and a CSS selector is called a **CSS declaration block**. CSS declarations (one or more) that applied to a specific selector are wrapped by curly braces (`{ }`). Each declaration inside a declaration block **must** be separated by a semi-colon (`;`).

Below is a sample CSS declaration block.

```
selector {  
  color: blue;  
}  
/*  
This is a css declaration for a selector  
'color' is a property name and 'blue' is a css property value  
!!!!!! CSS declarations must end with a semi-colon (;) !!!!!  
*/
```

Here's a more complete example declaration block.

```
/*  
The CSS declaration block below:  
* Will apply to all `h1` elements  
* Will change the text color to blue  
* Will set the font family to Georgia  
*/  
h1 {  
  color: blue;  
  font-family: Georgia;  
}
```

Summary

With the combination of HTML and CSS, you are able to define content, structure, and style for websites. In this lesson, you learned about the components of CSS and how to declare CSS

properties and values in declaration blocks. For example, you can use a CSS selector like `h1` or `p` paired with a declaration block to change the display of that element.

How do you feel about this lesson?



Have specific feedback?

[Tell us here! \(https://github.com/learn-co-curriculum/dsc-intro-to-css/issues/new/choose\)](https://github.com/learn-co-curriculum/dsc-intro-to-css/issues/new/choose)