

Introduction to CSS

The Anatomy of CSS

CSS stands for **Cascading Stylesheets**. CSS contains the **rules** that define how to display **HTML** elements on a webpage. **HTML** stands for **Hyper Text Markup Language** and comprises **markup tags** that describe the content of a webpage. While HTML defines the *structure* of the content, CSS defines the *style* of the content (hence, “stylesheet”). You will be introduced to HTML and CSS in much greater depth in Lab #4. For now, you just need to know the basic anatomy, and how you leverage this syntax to style multiscale basemaps using the mapping variant of CSS: CartoCSS.

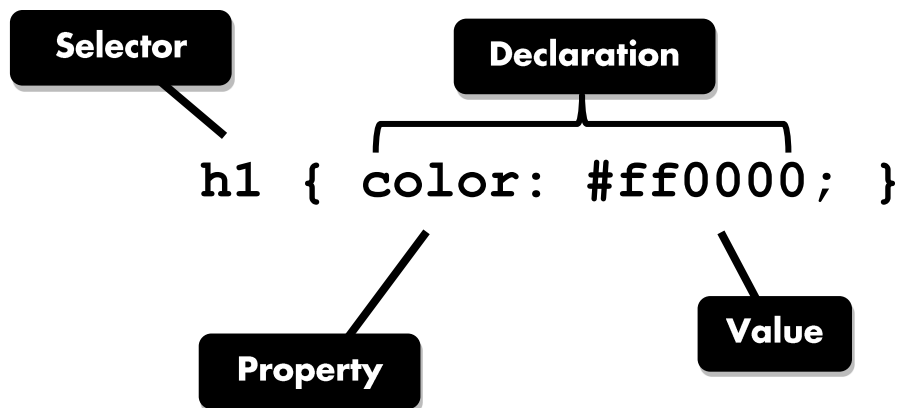


Figure 1: The Anatomy of CSS

A CSS rule is composed of two parts: a selector and a declaration (Figure 1). The **selector** determines the screen object to which the style will be applied. In typical webpage design, the selector refers to an HTML element. The style applied in Figure 1 is applied to the primary heading (<h1>) element—like a webpage title—using h1 as a selector. The **declaration** then indicates a style rule applied to the HTML element that was identified by the selector. The declaration is composed of a **property** (color in Figure 1) and a **value** (the hexadecimal #ff0000 in Figure 1, a bright red). Properties therefore are attributes of an HTML element that are used to manipulate the style of that element across the associated HTML document. It is possible to make multiple declarations for a single HTML element, with each declaration written on a separate line for readability (Example 12). All declarations end with a semicolon. Example 1 include comments, which are delimited in CSS using the /* and */ character sets.

```
h1 {
  color: #ff0000; /* assigns a color value to the h1 element */
  font-size: 30px; /* assigns the type size for the heading */
}
```

Example 1: Making Multiple Declarations for the Primary Heading Element

Inline and Internal CSS Style Rules

There are three ways to declare style rules for an HTML element: inline, internal, and external style rules. **Inline** styles are written directly into the HTML element as an attribute of that element ([Example 2](#)). Style rules added inline to the HTML document **override** (i.e., take precedent over) all other style rules. Note that inline styles require use of quotations and the equals (=) character to separate the style value from the style property. It is not recommended to make use of inline styles regularly, as they violate the web convention to separate the style (CSS) from the structure (HTML).

```
<h1 style="color:#ff0000">My Name</h1>
```

Example 2: An Inline Style Rule

Internal styles reside within the head (<head>) element of an HTML document itself using the style (<style>) element ([Example 3](#)). Internal styles only apply to the given webpage containing the style element.

```
<head>
  <style type="text/CSS">
    h1 {
      color: #ff0000;
      font-size: 30px;
    }
  </style>
</head>
```

Example 3: An Internal Style Rule

Inheritance and Cascading Style Rules

Before introducing external style rules, it is important to introduce two concepts fundamental to CSS: inheritance and cascading style rules. **Inheritance** describes the process by which some values of a parent element (such as the `color` property in [Figure 1](#)) are assigned to its children. For instance, the strong () element—used for emphasis within text—in [Example 4](#) will inherit the color `#ff0000` declared for the parent paragraph (<p>) element because a unique color was not assigned to the strong element itself (assuming a color rule for strong was not defined within an internal or external stylesheet).

```
<p style="color:#ff0000">It's important to understand <strong>inheritance
</strong> and <strong>cascading</strong> when writing CSS.</p>
```

Example 4: Inheritance of Styles from Parent Elements

The idea of **cascading style rules** arises because more than one rule can apply to a given HTML element, but only one can be assigned upon rendering. Rules are applied to elements in the following order, with '1' taking the lowest priority and '4' overriding all others:

1. browser default
2. external stylesheet
3. internal stylesheet
4. inline styles

Cascading also logically implies that style rules are interpreted by the web browser in order, from top to bottom within a stylesheet. Subsequent rules override previously defined ones. [Example 5](#) provides a simple example of how one rule can be overridden by another; in the example, the primary heading (<h1>) element would be rendered with the hexadecimal #ff0000 and a font size of 20px.

```
<style type="text/CSS">
  h1 {
    color: #ff0000;
    font-size: 30px;
  }
  h1 {
    color: #0000ff;
    font-size: 20px;
  }
</style>
```

Example 5: Overriding Internal Styles

External Style Rules

The recommended method for declaring style rules is use of an **external** stylesheet that is linked to each HTML document in your web portfolio using the link (<link>) element. External stylesheets allow you to keep the styles in one place and apply those uniformly across all HTML documents constituting your web portfolio. External stylesheets are written in plain text within a text editor (e.g., NotePad++) and are saved within a text file using the extension .css.

To create an external stylesheet, open a new document in a text editor and save the file within the css directory, naming it *style.css*. Add the code in [Example 6](#) to *screen.css* to provide an initial style rule for the primary heading (<h1>) element.

```
h1 {
  color: #ff0000;
  font-size: 30px;
}
```

Example 6: External Styles located in style.css

You will need to link the *screen.css* document to the *index.html* document before the style rule will be added to the primary heading element. To associate these two files, you will need to add a link

(<link>) element to your *index.html* document, along with some additional attributes (**Example 7**); the **relationship** attribute indicates that the linked file is a stylesheet and the **href** attribute indicates the directory in which the *screen.css* file is located. **Example 7** should be added within the head (<head>) element of your *index.html* document. Save the changes to both *index.html* and *style.css*; the primary heading in your updated webpage now should be red (or whichever other color you specified in the external stylesheet).

```
<!DOCTYPE html>
<html lang="en">
<head>
  <link rel="stylesheet" href="css/style.css" />
</head>
<body>
  <h1>Red Heading, 30pt</h1>
</body>
</html>
```

Example 7: Associating the HTML and CSS Documents through the <link> Tag

Unique Identifiers, Classes, and Descendant Selectors

External style rules often are applied differentially to one or a subset of HTML elements of a given type. A unique style rule within a linked stylesheet can be applied to an HTML element in two ways: (1) give the HTML element a **unique identifier** (*id* attribute) to style the element unique, or (2) identify the HTML element as part of a broader **class** (*class* attribute) of elements that are styled consistently. Referencing a unique identifier in CSS uses the hashtag notation (#) similar with referencing source map layers in Mapbox Studio Classic, while referring to the class in CSS uses the dot notation (.). Importantly, multiple classes can be defined for a single HTML element, but the element can only have one unique identifier. As described in the *Primer on Bootstrap*, much of the Bootstrap responsive design framework is built upon custom CSS classes: <http://getbootstrap.com/css/>.

Descendant selectors can be used to apply a unique style to a given HTML element only when it is nested within another HTML element of a particular type, with a particular ID, or that is part of a particular class. Example 8 presents three different style rules for divider (<div>) elements and paragraph (<p>) elements nested within divider elements. By default, all <div> elements are given a background-color of white and all <p> elements within these <div> elements make use of a gray text color. One specific <div> element, given the name div-main, is given a gray background-color to differentiate it from all other elements, with the nested <p> elements given a full black color to maintain legibility. Finally, there is a div-special class created for additional <div> elements that are visually separated from the div-main element and regular <div> elements. The div-special class also changes the color for nested <p> elements to compensate for the shift in background-color.

```
div {
    background-color: #000;
}

p {
    color: #999;
}

#div-main {
    background-color: #666;
}

#div-main p {
    color: #fff;
}

.div-special {
    background-color: #333;
}

.div-special p {
    color: #ccc;
}
```

Example 8: External Styles by Unique Identifiers, Classes, and Descendant Selectors.

The CSS Box Model

Many of the styles applied to HTML elements through CSS conform to the ***CSS box model***. Content elements (headings, paragraphs, footers, etc.) can be visualized as a rectangular box positioned somewhere on the webpage that has three footprint properties: a border, a margin, and padding.

- The ***border*** wraps around the content included in the HTML element and can have color values applied to it (i.e., it can be made visible in the layout).
- The ***margin*** is transparent and clears an area around the outside of the border. The margin is useful for pushing other HTML elements away from a given element without moving the given element's content.
- The ***padding*** clears space around the content within the element and assumes the background color of the element itself. The padding is useful for pushing the HTML element content away from surrounding elements without moving the surrounding elements.

The footprint relationship of the border, margin, and padding properties are illustrated in [Figure 2](#). The border, margin, and padding properties are used to create more or less whitespace between HTML elements; such an approach is recommended over inline use of paragraph or line break elements, continuing with the separation of content and form/style through external stylesheets. It is worth noting that HTML5 also includes an ***outline*** property, although it is not yet widely supported among modern browsers.

Explore the full set of CSS style rules at the w3schools reference: <http://www.w3schools.com/cssref/default.asp>



Figure 2: The CSS Box Model