



HTML and CSS

8th Edition

Chapter 7: CSS Building Blocks



Objectives

- Construct style rules for webpages with CSS.
- Add CSS comments.
- Understand how inheritance works.
- Apply the cascade to conflicting style rules.
- Determine specificity, order, and importance.
- Apply values to CSS properties.
- Specify colors for CSS properties using RGB and hexadecimals.



CSS Building Blocks

- **CSS (Cascading Style Sheets):** Defines content's appearance.
- **Style sheet:** A text file containing rules for displaying elements in a webpage.
- CSS controls basic formatting, layout, print controls, and dynamic properties.
- CSS2 is the best-supported version.
 - CSS3 builds upon CSS2 with new features.
- CSS can be created outside of the webpage and then applied to many pages all at once.



Constructing a Style Rule

- Each style rule has two main parts: selector and declaration block.
- **Selector** indicates which elements will be formatted.
- **Declaration block** describes the formatting to be executed.
 - Made up of one or more property/value pairs.
 - Each property/value pair constitutes a declaration.
 - Order of declarations doesn't matter unless the same property is defined twice.
 - Left curly brace begins a declaration block, and right curly brace ends it.

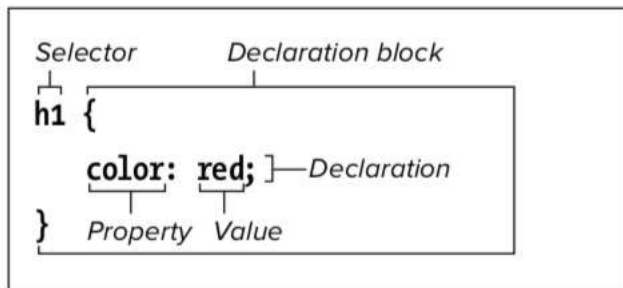


To Construct a Style Rule

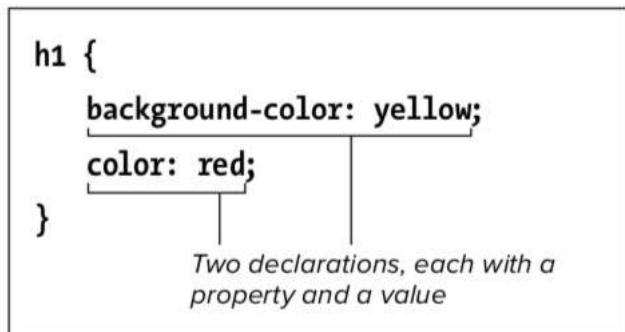
- Type **selector** to identify element(s) to format.
- Type { (an opening curly bracket) to begin declaration block.
- Type **property: value;**
 - Property names CSS property that describes formatting.
 - Value is one of a list of allowable options for that property.
- Repeat **property: value;** step as needed.
 - Enter each property: value (a declaration) on its own line.
- Type } to complete declaration block and style rule.



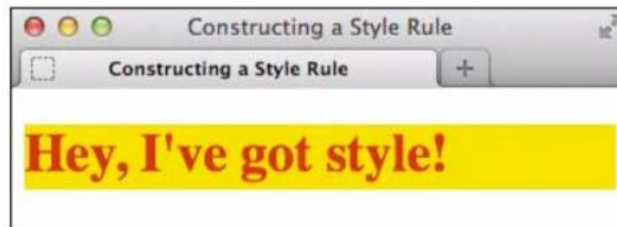
To Construct a Style Rule



Example:



```
...  
<body>  
  <h1>Hey, I've got style!</h1>  
</body>  
</html>
```





Adding Comments to Style Rules

- Comments do not display in webpages.
- Add comments to CSS to:
 - Note primary sections of style sheets.
 - Explain a particular rule or declaration.
- CSS comments help you and anyone else looking at your code.
- To add comments to style rules:
 - In your style sheet, type `/*` to begin your comment.
 - Type the comment.
 - Type `*/` to signal the end of the comment



Adding Comments to Style Rules

```
/*  
This is a CSS comment. It can be one line  
→ long or span several lines. This one  
→ is much longer than most. Regardless, a  
→ CSS comment never displays in the  
→ browser with your site's HTML content.  
  
Of course, you wouldn't really write a  
→ silly comment like this that merely  
→ talks about comments. The next comment  
→ is more in line with a comment's  
→ typical use.  
*/  
  
/* Set default rendering of certain HTML5  
→ elements for older browsers. */  
article,  
aside,  
figcaption,  
figure,  
footer,  
header,  
main,  
nav,  
section {  
    display: block;  
}
```

```
.byline {  
    color: green;  
    font-size: .875em;  
    text-shadow: 2px 1px 5px orange; /* IE9  
    → and earlier don't support */  
} /* You can put comments here, too! */
```



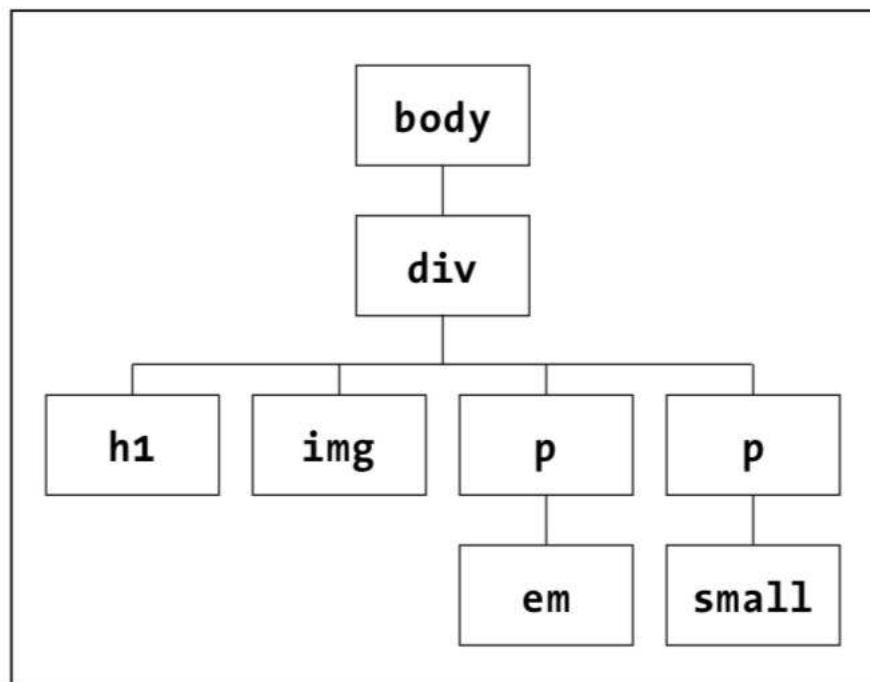

Understanding Inheritance

- **Inheritance** determines what happens if no other style rule is applied to an element.
- Many CSS properties are inherited by the descendants of the elements to which they've been applied.
- You can also use a value of *inherit* with most properties to force inheritance.
- Inheritance is the weakest rule, overruled by any other rule.



Understanding Inheritance

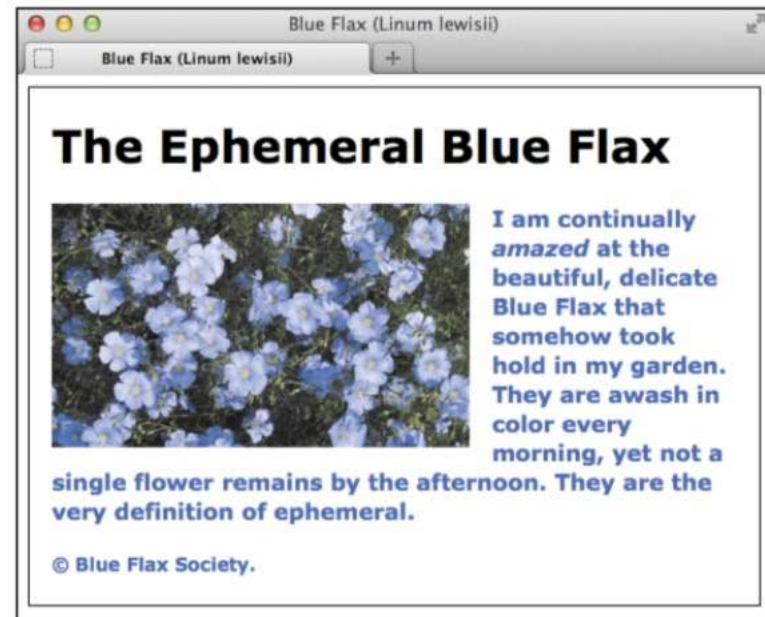
```
...  
<body>  
<div>  
  <h1>The Ephemeral Blue Flax</h1>  
  
    
  
  <p>I am continually <em>amazed</em>  
  → at the beautiful, delicate Blue Flax  
  → that somehow took hold in my garden.  
  → They are awash in color every morning,  
  → yet not a single flower remains  
  → by the afternoon. They are the very  
  → definition of ephemeral.</p>  
  
  <p><small>&copy; Blue Flax Society.  
  → </small></p>  
</div>  
</body>  
</html>
```





Understanding Inheritance

```
body {  
  font-family: Verdana, Geneva,  
  → sans-serif;  
}  
  
div {  
  border: 1px solid #000;  
  overflow: hidden;  
  padding: 0 1em .25em;  
}  
  
p {  
  color: #36c; /* a blue color */  
  font-weight: bold;  
}  
  
img {  
  float: left; /* makes text wrap it */  
  margin-right: 1em;  
}
```





Which Properties Are Inherited? (FYI)

- **Text:** color (except by the a element), direction, font, font-family, font-size, font-style, font-variant, font-weight, letter-spacing, line-height, text-align, text-indent, text-transform, visibility, white-space, word-spacing
- **Lists:** list-style, list-style-image, list-style-position, list-style-type



Which Properties Are Inherited? (FYI)

- **Tables:** border-collapse, border-spacing, caption-side, empty-cells
- **Paged media (as in printing):** Orphans, page-break-inside, widows
- **Other:** Cursor, quotes



The Cascade: When Rules Collide

- Override or complement default browser style in three ways:
 - Load styles from an external file (Recommended).
 - Insert at the top of an HTML document.
 - Apply to specific HTML element in the code.
- **Cascade:** When more than one style rule applies to an element, CSS uses this principle to determine which of the conflicting rules wins.
- Principle of cascade takes into account three characteristics: specificity, order, and importance.



The Cascade: When Rules Collide

```
p {  
  color: red;  
}  
  
.example {  
  color: blue;  
}  
  
.example.example-2 {  
  color: magenta;  
  /* negated by next rule */  
}  
  
.example.example-2 {  
  color: green;  
}
```

```
...  
<link rel="stylesheet" href="style.css" />  
</head>  
<body>  
  
<p>Here's a generic <code>p</code> element. It will be red.</p>  
  
<p class="example">Here's a <code>p</code> element with a <code>class</code> of <code>example  
→ </code>. There are two rules that could apply, but since the <code>.example</code> selector is  
→ more specific, this paragraph will be blue.</p>  
  
<p class="example example-2">Here's a <code>p</code> element with two classes: <code>example  
→ </code> and <code>example-2</code>. There are four rules that could apply to this paragraph. The  
→ first two are overruled by the more specific last two. However, because the last two have the  
→ same selector, the order breaks the tie between them: the one that appears later wins, and thus  
→ this paragraph will be green instead of magenta.</p>  
  
</body>  
</html>
```



Specificity

- Law of specificity states that the more specific the selector, the stronger the rule—and the conflicting style in the stronger rule wins.



Here's a generic `p` element. It will be red.

Here's a `p` element with a `class` of `example`. There are two rules that could apply, but since the `.example` selector is more specific, this paragraph will be blue.

Here's a `p` element with two classes: `example` and `example-2`. There are four rules that could apply to this paragraph. The first two are overruled by the more specific last two. However, because the last two have the same selector, the order breaks the tie between them: the one that appears later wins, and thus this paragraph will be green instead of magenta.

Selector	Corresponding HTML
<code>p { ... }</code>	<code><p>...</p></code>
<code>.someClass { ... }</code>	<code><p class="someClass">...</p></code>
<code>.someClass.someOtherClass { ... }</code>	<code><p class="someClass someOtherClass">...</p></code>
<code>#someID { ... }</code>	<code><p id="someID">...</p></code> <code><p id="someID" class="someClass">...</p></code> <code><p id="someID" class="someOtherClass">...</p></code> <code><p id="someID" class="someClass someOtherClass">...</p></code>

This table lists a few selectors, from least specific (an element name, at the top) to most specific (an ID, at the bottom)



Order and Importance

- **Order:** When two rules have equal specificity, then rules that appear later in the style sheet take precedence.
 - Rules applied inline in the HTML element are considered to appear after—and thus take precedence over—equally specific rules in an external style sheet or embedded at top of HTML document.
- **Importance:** Override whole system by declaring a particular rule more important than the others by adding `!important` at the end of the declaration.

```
p { color: orange !important; }.
```



A Property's Value

- Each CSS property has different rules about what values it can accept.
 - **Inherit value:** Use to explicitly specify that the value for the property be the same as the parent element's value.
 - **Predefined values:** Most CSS properties have a few predefined values.
 - **Lengths and percentages:** Many CSS properties take a length for their value. All length values must state quantity and unit, with no spaces between.



Code for Preset Value

A preset value

```
border: none;
```



Lengths and Percentages

- Length types can be relative or absolute units.
- Only use absolute lengths when size of output is known.
- Ems, pixels, and percentages are most common.
 - Ems are relative to some other value.
 - Pixels (px) are not relative to other style rules.
 - Pixels aren't necessarily the same size on two devices.
 - Percentage values are generally relative to their parent element's value.



Code for Length and Percentage

A length

```
font-size: 24px;
```

A percentage

```
font-size: 80%;
```



Bare Numbers

- Very few CSS properties accept a value in the form of a number without a unit.
 - Most common are line-height.
- Don't confuse numbers and integers with length.
 - A number or integer has no unit (like px).

```
line-height: 1.5;
```

A number



URLs

- Some CSS properties allow URL of another file to be specified, particularly images.
 - Use **url(*file.ext*)**
- Relative URLs should be relative to style sheet, not HTML document.
- Quotation marks around the file name aren't required.
- No space between the word url and the opening parentheses.
- White space between parentheses and address allowed, but not required.



Code for Specifying a URL

A URL

```
background: url(bg_flax.jpg);
```





CSS Colors

















- Specify colors for CSS properties in several ways.
 - Value can be one of the predefined color keywords.
 - Easiest method.
 - In practice, more common to define CSS colors with the hexadecimal format or RGB format.
 - Can also specify a color with the HSL format, and the level of color transparency with RGBA and HSLA.
 - All of which are new in CSS3.



CSS Colors

Sixteen Predefined Colors

Note that the color names and hex values are not case-sensitive, not even in XHTML.

			
Aqua #00FFFF	Black #000000	Blue #0000FF	Fuchsia #FF00FF
			
Gray #808080	Green #008000	Lime #00FF00	Maroon #800000
			
Navy #000080	Olive #808000	Purple #800080	Red #FF0000
			
Silver #C0C0C0	Teal #008080	White #FFFFFF	Yellow #FFFF00

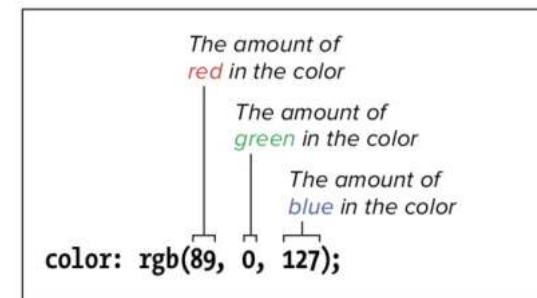
The full list is available at www.w3.org/TR/css3-color/#svg-color.



RGB and Hexadecimal

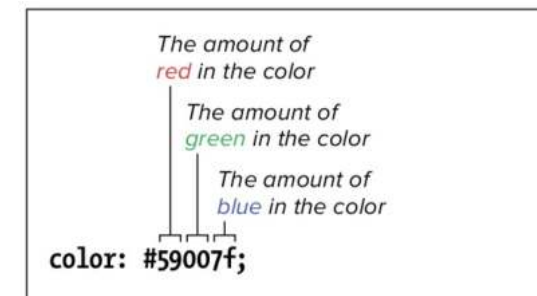
RGB

- Construct your own color by specifying its amount of red, green, and blue.
- Give the values for each of these contributing colors as a number from 0–255, a percentage, or a hexadecimal representation of the number.



Hexidecimal

- Convert those numerical values to hexadecimals, join them together, and prefix the value with a #
 - Most common method.





HTML and CSS

8th Edition

Chapter 8: Working with Style Sheets



Objectives

- Create CSS style sheets.
- Apply CSS to HTML pages using external style sheets.
- Understand the basic style rule of location.
- Identify !important and user style sheet exceptions.
- Employ media-specific style sheets.
- Define persistent, preferred, and alternate style sheets.
- View other designers' CSS code.



Working with Style Sheets

- Create a style sheet file and apply CSS to multiple webpages, a single page, or an individual HTML element.
- **Three** methods:
 - **External Style Sheets** (preferred choice)
 - **Embedded Style Sheets**
 - **Inline Styles** (least desirable)



Creating an External Style Sheet

- **External style sheet:** A separate page that holds all CSS styles to be loaded into one or more pages.
 - Preferred method.
- Set each page on the site to load the same external sheet.
 - Ensures each page has the same settings and a consistent look.



To Create an External Style Sheet

- Create new text document in text editor.
- Define style rules for webpages.
 - Option: Include CSS comments.
- Save document in text-only format in desired directory.
 - Document extension must be .css for a Cascading Style Sheet.



Linking to External Style Sheets

- Load a style sheet into HTML pages to apply style rules to content.
- Best way to do so is to link to the style sheet.
 - Link element goes inside head section of HTML document.
 - Style rule (a solid red border four pixels thick) is applied to each img element.
 - Other documents can link to same external style sheet.



To Link an External Style Sheet

- Type **<link rel="stylesheet"** in head section of each HTML page.
- Type a space and then **href="url.css"**
 - *url.css* is the location and name of the CSS style sheet.
- Type a space and the final **/>**
 - Or, type no space and simply **>**
 - HTML5 allows both approaches and they work exactly the same.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8" />
  <title>El Palau de la Música</title>
  <link rel="stylesheet"
  → href="style.css" />
</head>
<body>
<article>
  <h1>El Palau de la Música</h1>

  <p>I love the <span lang="es">Palau de la
  → Música</span>. It is ornate and gaudy
  → and everything that was wonderful
  → about modernism. It's also the home
  → of the <span lang="es">Orfeó Català
  → </span>, where I learned the benefits
  → of Moscatell.</p>
</article>
</body>
</html>
```



To Link an External Style Sheet

Name	Size	Kind
▼ css	--	Folder
style.css	171 bytes	CSS
▼ img	--	Folder
palau.jpg	23 KB	JPEG image
sunflowers.jpg	16 KB	JPEG image
tickets.jpg	4 KB	JPEG image
example.html	647 bytes	HTML Document

one common way to organize a site.



```
...
<head>
  <meta charset="UTF-8" />
  <title>El Palau de la Música</title>
  <link rel="stylesheet"
  → href="css/style.css" />
</head>
...
```

style.css is located in a subfolder named css

```
img {
  border: 4px solid red;
}
```

Style.css



Creating an Embedded Style Sheet

- **Embedded style sheet:**
 - Second way to apply CSS to a page.
 - Sets the styles directly in HTML document.
 - Typically goes in the page's head.
- **Because styles are only in HTML file:**
 - CSS won't apply to other pages.
 - Lacks same benefits.



To Create an Embedded Style Sheet

- Type **<style>** in the head section of your HTML document.
- Define as many style rules as desired.
- Type **</style>** to complete the embedded style sheet.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8" />
  <title>El Palau de la Música</title>
  <style>
    img {
      border: 4px solid red;
    }
  </style>
</head>
<body>
<article>
  <h1>El Palau de la Música</h1>

  <p>I love the <span lang="es">Palau de
  → la Música</span>. It is ornate and
  → gaudy and everything that was
  → wonderful about modernism. It's also
  → the home of the <span lang="es">Orfeó
  → Català </span>, where I learned the
  → benefits of Moscatell.</p>
</article>
</body>
</html>
```





Applying Inline Styles

- Inline styles are the third way to apply CSS to HTML.
- Least desirable option:
 - Mixes content (HTML) and presentation (CSS).
 - Only affects one element.
 - Means sifting through HTML pages to make simple changes.
- Inline styles:
 - Aren't intended for regular use.
- Use to perform a quick test before placing a style in an external style sheet.



To Apply Styles Inline

- Type **style=""** within HTML element to be formatted.
 - Add it to the start tag of non-void elements.
- Create a style rule without curly brackets or selector.
 - Selector isn't needed since you're placing it directly inside the element.
 - If your rule has more than one declaration, separate each one by typing ; (a semicolon).
- Type the final quote mark " .

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8" />
  <title>El Palau de la Música</title>
</head>
<body>
  ...
  

  
  ...
</body>
</html>
```





The Cascade and the Order of Styles

- Basic rule: With all else equal, the later the style appears, the greater its precedence or importance.
- How that plays out:
 - Relationship between an embedded style sheet and any linked external style sheets depends on their relative positions in HTML.
 - Inline styles take precedence, overriding any conflicting styles applied elsewhere.
 - Exception: A style marked !important always wins, whether first or last in order.



The Cascade and the Order of Styles

- Relationship between an embedded style sheet and any linked external style sheets depends on their relative positions in HTML.

```
img {
  border: 4px solid red;
}

/*
That is a shorter way of writing this:

img {
  border-width: 4px;
  border-style: solid;
  border-color: red;
}
*/
```

Style.css

1

```
...
<head>
  <title>El Palau de la Música</title>

  <link rel="stylesheet" href="style.
css" />

  <style>
    img {
      border-style: dashed;
    }
  </style>
</head>
...
```



The embedded style sheet comes last. Therefore, its styles will have precedence over the ones in **style.css**

2

```
...
<head>
  <title>El Palau de la Música</title>
  <style>
    img {
      border-style: dashed;
    }
  </style>
  <link rel="stylesheet"
  → href="style.css" />
</head>
...
```



The linked style sheet (**style.css**) comes last and has precedence over rules in the **style** element



The Inspiration of Others: CSS

- **To view others' CSS code:**
 - First view the page's HTML code.
 - If the CSS code is in an embedded style sheet, you'll be able to see it already.
 - If the CSS is in an external style sheet, locate the reference to it in the HTML and click the file name.
 - The style sheet displays in the browser window.
 - You can copy it from there and paste it into your text editor if you like.



View CSS in Browser

```
Source of: http://htmlcssvqs.com/8ed/examples/chapter-08/li...
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4     <meta charset="UTF-8" />
5     <title>El Palau de la Música</title>
6     <link rel="stylesheet" href="style.css" />
7 </head>
8 <body>
9 <article>
10     <h1>El Palau de la Música</h1>
11
12     
14
15     
17
18     <p>I love the <span lang="es">Palau de la
```

Line 8, Col 7