



CASCADING **STYLE SHEETS**

Introduction to CSS

<https://academy.evermight.com>



WHAT IS CSS?

Cascading Style Sheets (CSS) is a language that consist of declarations / rules that describe the visual representation of HTML documents in a web browser. Here are 3 different ways to incorporate CSS into HTML:

Inline

```
<!DOCTYPE html>
<html>
  <head>
  </head>
  <body>
    <p style="color:red;">hello world</p>
  </body>
</html>
```

Style Block

```
<!DOCTYPE html>
<html>
  <head>
    <style>
      p {color:red;}
    </style>
  </head>
  <body>
    <p>hello world</p>
  </body>
</html>
```

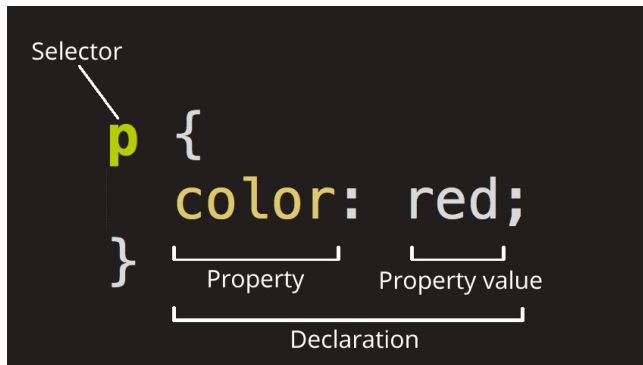
Separate File

```
/* my-style.css */
p {color:red;}

<!DOCTYPE html>
<html>
  <head>
    <link href="my-style.css" rel="stylesheet" />
  </head>
  <body>
    <p>hello world</p>
  </body>
</html>
```

In all 3 examples above, the page will show **hello world** in **red**.

ANATOMY OF CSS DECLARATION

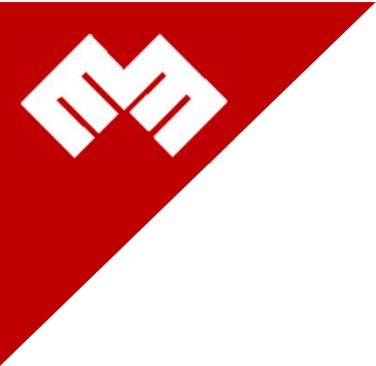


Selector – The HTML tags to style. We will learn shortly that there are many ways to express the specific HTML tag you wish to style. In this example, we are selecting ALL `<p>` tags.

Declaration – The style rule to apply. In this example, we have only one declaration. However, we can have multiple style declarations separated by the semi-colon. Each declaration is made up of a **property** and a **property value**.

Property – The style property to change. There are many types of properties to choose from, most of which are obvious like font, background, padding, margin, etc...

Property Value – The specific style you wish to apply to a property. The acceptable properties will depend on the property you are trying to change.



SELECTOR – <TAG>

As mentioned, you can select HTML tags by their tag name. Here are examples:

```
/* Anything between the forward slash and asterisk is a comment and ignored by web browsers */
```

```
/* Make all paragraph tags blue, font size 15px, font face Arial */
```

```
p {color:blue; font: 15px Arial; }
```

```
/* Make all h1 tags have underline and bold */
```

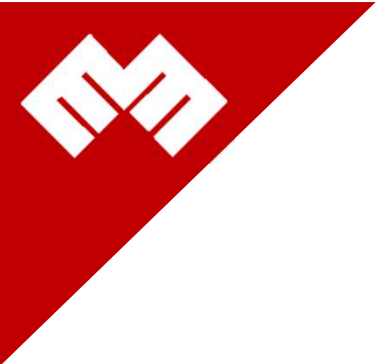
```
h1 {text-decoration:underline; font-weight:bold;}
```

```
/* Make all img tags have 10px solid yellow border, and the image should be 500px by 10px */
```

```
img {border:10px solid yellow; width:500px; height:10px;}
```

```
/* The asterisks means everything. Make everything have green background color. */
```

```
* {background-color: green; }
```



SELECTOR – ID

You can also style HTML tags by their IDs. Here are examples:

```
/* my-style.css */

/* "I love to eat pizza" will be italicized and green */
#review {
  font-style:italic;
  color:green;
}

/* "pizza.png" will be 200px by 200px and 50% opacity*/
#food {
  width: 200px;
  height: 200px;
  opacity: 0.5;
}
```

```
<!DOCTYPE html>
<html>
  <head>
    <link href="my-style.css" rel="stylesheet" />
  </head>
  <body>
    <h1>Hello World</h1>
    <p>Last modified Sept 1, 2019</p>
    <p id="review">I love to eat pizza</p>
    
  </body>
</html>
```

See [01-example.html](#)



SELECTOR – CLASS

You can also style HTML tags by their class. Here are examples:

```
/* my-style.css */

/* "I love to eat pizza" and "I love to eat chocolate" will be italicized and green */
.review {
  font-style:italic;
  color:green;
}

/* "pizza.png" and "dipping-sauce.png" will be 200px by 200px and 50% opacity*/
.food {
  width: 200px;
  height: 200px;
  opacity: 0.5;
}
```

```
<!DOCTYPE html>
<html>
  <head>
    <link href="my-style.css" rel="stylesheet" />
  </head>
  <body>
    <h1>Hello World</h1>
    <p>Last modified Sept 1, 2019</p>
    <p class="review">I love to eat pizza</p>
    <p class="review">I love to eat chocolate</p>
    
    
  </body>
</html>
```

See 02-example.html



SELECTOR – ATTRIBUTES

You can also style HTML tags by their attributes. Here are examples:

```
/* my-style.css */

/* Tags that have data-msg="review"
   "I love to eat pizza" and "I love to eat chocolate" will be italicized and green */
[data-msg="review"] {
  font-style:italic;
  color:green;
}

/* Tags that have class="food"
   "pizza.png" and "dipping-sauce.png" will be 200px by 200px and 50% opacity */
[class="food"] {
  width: 200px;
  height: 200px;
  opacity: 0.5;
}
```

```
<!DOCTYPE html>
<html>
  <head>
    <link href="my-style.css" rel="stylesheet" />
  </head>
  <body>
    <h1>Hello World</h1>
    <p>Last modified Sept 1, 2019</p>
    <p data-msg="review">I love to eat pizza</p>
    <p data-msg="review">I love to eat chocolate</p>
    
    
  </body>
</html>
```

See 03-example.html



SELECTOR – PSEUDO-CLASS

You can also style HTML tags by their place in HTML hierarchy. Here are examples:

```
/* my-style.css */  
  
/* first element of each child */  
:first-child {  
  color: green;  
}  
  
/* every even numbered child – could also be :nth-child(even) , or :nth-child(odd) for  
odd number */  
:nth-child(2n) {  
  color: red;  
}  
  
/* 4th child */  
:nth-child(4) {  
  font-family: cursive;  
}
```

See 04-example.html

```
<!DOCTYPE html>  
<html>  
  <head>  
    <link href="my-style.css" rel="stylesheet" />  
  </head>  
  <body>  
    <h1>Hello World</h1>  
    <p>Take these with you on vacation:</p>  
    <ul id="belongings">  
      <li class="high-priority" data-note="Expires 2020-01-01">Passport</li>  
      <li class="high-priority">Cash</li>  
      <li>Clothes</li>  
    </ul>  
    <p>Remember to:</p>  
    <ol id="remember-to">  
      <li>Notify family of departure.</li>  
      <li>Lock the house door.</li>  
      <li>Call taxi.</li>  
    </ol>  
  </body>  
</html>
```




SELECTOR – CHILDREN

You can also style HTML tags by their place in HTML hierarchy. Here are examples:

```
/* my-style.css */

/* <li> under the <body> under any tag with id="belongings"
   which happens to be Passport, Cash and Clothes */
body #belongings li {
  font-style:italic;
  color:green;
}

/* <li> under <ol>
   which happens to be Notify family of departure, Lock the house door, Call taxi */
ol li {
  color:red;
}
```

See 05-example.html

```
<!DOCTYPE html>
<html>
  <head>
    <link href="my-style.css" rel="stylesheet" />
  </head>
  <body>
    <h1>Hello World</h1>
    <p>Take these with you on vacation:</p>
    <ul id="belongings">
      <li class="high-priority" data-note="Expires 2020-01-01">Passport</li>
      <li class="high-priority">Cash</li>
      <li>Clothes</li>
    </ul>
    <p>Remember to:</p>
    <ol id="remember-to">
      <li>Notify family of departure.</li>
      <li>Lock the house door.</li>
      <li>Call taxi.</li>
    </ol>
  </body>
</html>
```



SELECTOR – JOINING SELECTORS

You can also style HTML tags by joining several selectors. Here are examples:

```
/* my-style.css */

/* <ul> and has the id="belongings"
   will have green background
*/
ul#belongings {
  background-color:green;
}

/* <li> with class="high-priority" and a data-note="Expires 2020-01-01"
   which happens to be Passport */
li.high-priority[data-note="Expires 2020-01-01"] {
  color:red;
}
```

See [06-example.html](#)

```
<!DOCTYPE html>
<html>
  <head>
    <link href="my-style.css" rel="stylesheet" />
  </head>
  <body>
    <h1>Hello World</h1>
    <p>Take these with you on vacation:</p>
    <ul id="belongings">
      <li class="high-priority" data-note="Expires 2020-01-01">Passport</li>
      <li class="high-priority">Cash</li>
      <li>Clothes</li>
    </ul>
    <p>Remember to:</p>
    <ol id="remember-to">
      <li>Notify family of departure.</li>
      <li>Lock the house door.</li>
      <li>Call taxi.</li>
    </ol>
  </body>
</html>
```



SELECTOR – SHARING STYLES

You can also style HTML tags by joining several selectors. Here are examples:

```
/* my-style.css */

/* <h1> OR <ul id="belongings"> */
h1, ul#belongings {
  background-color:green;
}

/* <h1> OR <p> OR <li class="high-priority" data-note="Expires 2020-01-01"> */
h1,
p,
li.high-priority[data-note="Expires 2020-01-01"] {
  color:red;
}
```

See [07-example.html](#)

```
<!DOCTYPE html>
<html>
  <head>
    <link href="my-style.css" rel="stylesheet" />
  </head>
  <body>
    <h1>Hello World</h1>
    <p>Take these with you on vacation:</p>
    <ul id="belongings">
      <li class="high-priority" data-note="Expires 2020-01-01">Passport</li>
      <li class="high-priority">Cash</li>
      <li>Clothes</li>
    </ul>
    <p>Remember to:</p>
    <ol id="remember-to">
      <li>Notify family of departure.</li>
      <li>Lock the house door.</li>
      <li>Call taxi.</li>
    </ol>
  </body>
</html>
```



RULES OF SPECIFICITY

CSS, like plain language, has the notion of **general statements** vs. **specific statements**.

Using a real world analogy, imagine the following scenario:

Statements:

- George is a human that broke his leg
- Doctor says George should not walk for 3 months
- Humans should walk to remain healthy
- Humans should breathe to remain alive

Questions:

- Should George walk right now to stay healthy?
- Should George breathe right now to remain alive?

Answer:

Although there is a **general statement** that humans should exercise to remain healthy, there is a **specific statement** that George broke his leg and has been advised to not walk for 3 months which **overrides** the more **general statement** that walking at this moment will improve George's health. However, there is no **specific statement** that states George should not breathe, hence George will inherit the more **general statement** to breathe.



RULES OF SPECIFICITY

If you apply multiple style declarations to the same tag, the more specific rule, followed by the most recent rule takes precedence.

```
/* my-style.css */
/* all <li class="high-priority" data-note="Expires 2020-01-01">
   are white text with red background */
li.high-priority[data-note="Expires 2020-01-01"] {
  background-color:red;
  color:white;
}

/* all <li class="high-priority"> is red text */
li.high-priority {
  color:red;
}

/* all <li> are gray text */
li {
  color:gray;
}
```

```
<!DOCTYPE html>
<html>
  <head>
    <link href="my-style.css" rel="stylesheet" />
  </head>
  <body>
    <h1>Hello World</h1>
    <p>Take these with you on vacation:</p>
    <ul id="belongings">
      <li class="high-priority" data-note="Expires 2020-01-01">Passport</li>
      <li class="high-priority">Cash</li>
      <li>Clothes</li>
    </ul>
    <p>Remember to:</p>
    <ol id="remember-to">
      <li>Notify family of departure.</li>
      <li>Lock the house door.</li>
      <li>Call taxi.</li>
    </ol>
  </body>
</html>
```



RULES OF SPECIFICITY

Selector Type	HTML	CSS
inline	<code><p style="color:red; padding:0px;"></code>	
id	<code><p id="intro"></code>	<code>#intro {color:red; padding:0px;}</code>
class	<code><p class="intro"></code>	<code>.intro {color:red; padding:0px;}</code>
pseudo class*	<code><p></code>	<code>:hover {color:red; padding:0px;}</code>
attributes	<code><p data-blah="hello"></code>	<code>[data-blah="hello"] {color:red; padding:0px;}</code>
tag	<code><p></code>	<code>p {color:red; padding:0px;}</code>

*pseudo class – there are only a handful of other pseudo classes such as:

`:nth-child([num])` - select the [num]th child element

`:first-child` – select the first child element

`:last-child` – select the last child element

`:visited` – select `<a>` that have been visited