



Cascading Style Sheets (CSS)




CSS Rule

```
<head>
  <meta charset="UTF-8">
  <title>First Style</title>

  <!-- New HTML Element: style -->
  <style></style>
</head>

<body>
  <h1>Glorious Color</h1>
</body>
```



```
<head>
  <meta charset="UTF-8">
  <title>First Style</title>

  <!-- New HTML Element: style -->
  <style>
    h1 {
      color: white;
      background-color: darkviolet;
    }
  </style>
</head>


<body>
  <h1>Glorious Color</h1>
</body>
```



CSS Rule

```
h1 {  
    color: white;  
    background-color: darkviolet;  
}
```

Colors



```
color: red;  
color: #FF0000;  
color: #FF0000FF;  
color: rgb(255, 0, 0);  
color: rgba(255, 0, 0, 1);  
color: hsl(0, 100%, 50%);  
color: hsla(0, 100%, 50%, 1);
```

```
/* Keyword */  
/* RGB Hexadecimal */  
/* RGBA Hexadecimal */  
/* RGB */  
/* RGBA */  
/* HSL */  
/* HSLA */
```

Typography

Font Size

- Most common units: pixels (px), percent (%), em and rem
- We'll be focusing on pixels to start

```
font-size: 35px;
```

Fonts

SERIF

Serif fonts have extra details on the end of the main strokes of the letters.

EXAMPLES:

Georgia

Times

Times New Roman

SANS-SERIF

Sans-serif fonts have straight ends to letters and therefore have a much cleaner design.

EXAMPLES:

Arial

Verdana

Helvetica

Fonts

MONOSPACE

Every letter in a monospace typeface is the same width. (Non-monospace fonts have different widths.)

EXAMPLES:

Courier

Courier New

CURSIVE

Cursive fonts either have joining strokes or other cursive characteristics, such as handwriting styles.

EXAMPLES:

Comic Sans MS

Monotype Corsiva

FANTASY

Fantasy fonts are usually decorative fonts and are often used for titles. They're not designed for long bodies of text.

EXAMPLES:

Impact

Haettenschweiler

Font Family

- Specifies which font to use
 - Every computer has different fonts installed
- Can use a generic family
 - serif, sans-serif, monospace, cursive, fantasy
- Can use a font that is stored on visitor's computer
 - See fonts on your machine: workmark.it

```
font-family: "Georgia";  
font-family: sans-serif;
```



Font Stacks

- Different operating systems have different fonts
- Safe stacks: cssfontstack.com

```
font-family: "Helvetica Neue", "Helvetica", "Arial", sans-serif;
```



Embedding Fonts

- Font files loaded over the internet
- Uses @font-face (font file support)
- Web font services: fonts.google.com, Adobe Typekit



Font Weight

- Can be a number (100, 200, 300, ..., 900)
- Can be a keyword: normal (400), bold (700)
- Not all fonts are available in all weights

Raleway, 100

Raleway, 200

Raleway, 300

Raleway, 400

Raleway, 500

Raleway, 600

Raleway, 700

Raleway, 800

Raleway, 900

```
font-weight: 100;
```

Text Alignment

Left

Jelly beans tootsie roll carrot cake bonbon donut
cotton candy lollipop powder sesame snaps. Candy
canes gingerbread sugar plum cheesecake chupa
chups jelly beans sesame snaps chocolate
cheesecake. Croissant halvah powder tart donut.

Right

Croissant cotton candy candy canes pastry. Cookie
candy jujubes chocolate cake chocolate. Macaroon
danish lollipop. Chocolate cake powder sugar plum
pastry macaroon.

Center

Liquorice gingerbread sweet toffee marshmallow
croissant sweet roll lollipop. Dragée candy pudding.
Topping cheesecake cake sugar plum powder
tootsie roll. Sesame snaps liquorice cookie.

Justify

Soufflé dessert lemon drops lollipop powder.
Lollipop dessert ice cream bear claw danish chupa
chups pie gummi bears jelly-o. Halvah lemon drops
dessert. Topping powder gummies toffee powder
apple pie chocolate bar candy tart.

```
/*  
    Possible values:  
    justify, left, right, center  
*/  
text-align: center;
```


Font Style

- Can be a keyword: normal, italic, oblique
 - Italic – more like cursive
 - Oblique – normal font, but slanted
- If font doesn't have italic/oblique, browser will try to make it

```
font-style: italic;
```



Line Height

- [Leading](#)
- Space between lines
- Best practice: unit-less number, based on font size

```
line-height: 1.4;
```

Typographic Tools

- [gridlover](#)
- [typescale](#)



Text References

- MDN's [Text Styling Fundamentals](#)
- w3schools [text](#) and [font](#) pages

Selectors

Element Selector

```
h1 {  
    font-size: 50px;  
}  
h2 {  
    font-size: 35px;  
}  
p {  
    font-size: 25px;  
}
```

Combining Selectors

```
h1, h2, h3, h4, h5, h6 {  
    font-family: "Raleway", sans-serif;  
}
```



ID Selector

- Targets 1x *unique* element
- Valid IDs:
 - Must have at least 1 character
 - No spaces
 - Best to stick to letters and dashes

```
<style>
  #title {
    font-size: 40px;
    font-style: italic;
  }
</style>
```

```
<body>
  <h1 id="title">Artist Statement</h1>
</body>
```




Class Selector

- Targets *multiple* elements
- Same rules for valid id names apply to class names

```
<style>
  .completed {
    text-decoration: line-through;
  }
</style>
```

```
<body>
  <h1>To Do</h1>
  <ol>
    <li class="completed">Wake up</li>
    <li class="completed">Eat</li>
    <li>Run</li>
    <li>Sleep</li>
  </ol>
</body>
```

| SELECTOR | MEANING | EXAMPLE |
|---------------------------|---|--|
| UNIVERSAL SELECTOR | Applies to all elements in the document | <code>* {}</code> Targets all elements on the page |
| TYPE SELECTOR | Matches element names | <code>h1, h2, h3 {}</code> Targets the <code><h1></code> , <code><h2></code> and <code><h3></code> elements |
| CLASS SELECTOR | Matches an element whose <code>class</code> attribute has a value that matches the one specified after the period (or full stop) symbol | <code>.note {}</code> Targets any element whose <code>class</code> attribute has a value of <code>note</code> <code>p.note {}</code> Targets only <code><p></code> elements whose <code>class</code> attribute has a value of <code>note</code> |
| ID SELECTOR | Matches an element whose <code>id</code> attribute has a value that matches the one specified after the pound or hash symbol | <code>#introduction {}</code> Targets the element whose <code>id</code> attribute has a value of <code>introduction</code> |
| CHILD SELECTOR | Matches an element that is a direct child of another | <code>li>a {}</code> Targets any <code><a></code> elements that are children of an <code></code> element (but not other <code><a></code> elements in the page) |
| DESCENDANT SELECTOR | Matches an element that is a descendent of another specified element (not just a direct child of that element) | <code>p a {}</code> Targets any <code><a></code> elements that sit inside a <code><p></code> element, even if there are other elements nested between them |
| ADJACENT SIBLING SELECTOR | Matches an element that is the next sibling of another | <code>h1+p {}</code> Targets the first <code><p></code> element after any <code><h1></code> element (but not other <code><p></code> elements) |
| GENERAL SIBLING SELECTOR | Matches an element that is a sibling of another, although it does not have to be the directly preceding element | <code>h1~p {}</code> If you had two <code><p></code> elements that are siblings of an <code><h1></code> element, this rule would apply to both |

Selectors Reference

- w3schools [selectors page](#)

Ways to Add Style

Methods of Adding Style

1. Inline style
2. Internal Stylesheets
3. External Stylesheets



Inline Style

```
<h1 style="background-color: crimson; color: white;">Inline styling. Gross.</h1>
```



Internal Stylesheets

```
<head>
  <meta charset="UTF-8">
  <title>Internal</title>

  <style>
    h1 {
      background-color: crimson;
      color: white;
    }
  </style>

</head>
```



External Stylesheets

index.html:

```
<head>
  <meta charset="UTF-8">
  <title>External</title>
  <link rel="stylesheet" href="style.css">
</head>
```

style.css:

```
h1 {
  background-color: crimson;
  color: white;
}
```



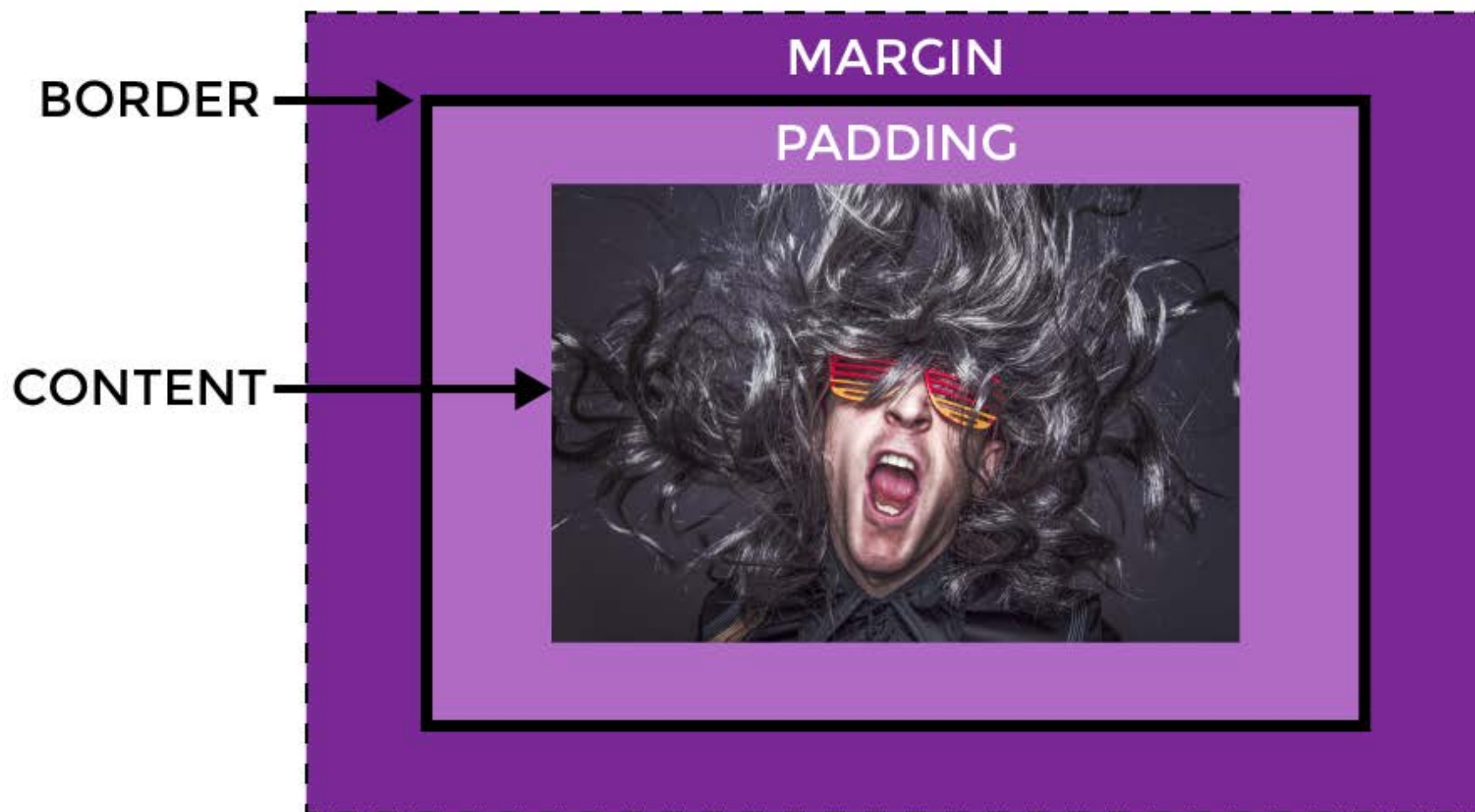

Cascading Order

1. Browser defaults
2. Internal & external stylesheets
3. Inline style

Box Model

Everything is a Box

```
* {  
  border: 1px solid red !important;  
}
```



Display

- Controls what type of box is used
- Common values:

```
display: inline;  
display: block;  
display: inline-block;  
display: flex;  
display: none;
```

- More info: [MDN](#)

Width & Height

- Width and height of box
- Depends on the display property
 - Block and inline-block work as expected
 - Inline is [quirky](#)

```
width: 200px;  
height: 200px;
```

```
width: 25%; /* % of parent's width */  
height: 25%; /* % of parent's height */
```

```
width: 200px;  
height: auto; /* Expand to fit */
```

Margins

- Space between elements
- Again, inline is “quirky”

```
margin-top: 20px;  
margin-bottom: 20px;  
margin-left: 10px;  
margin-right: 10px;
```

Margin Shorthand

`margin: 0px;`

ALL MARGINS

LEFT & RIGHT

`margin: 20px 10px;`

TOP & BOTTOM

RIGHT

LEFT

`margin: 10px 20px 30px 40px;`

TOP

BOTTOM

Padding

- Space between content and border
- Again, inline is “quirky”
- Same shorthand properties as margin

```
padding-top: 20px;  
padding-bottom: 20px;  
padding-left: 10px;  
padding-right: 10px;
```

Margin vs Padding

- Padding is part of the element, whereas margin is not. If a link has padding, the padded area is clickable.
- Margins [collapse](#), whereas padding does not.

Border

- More on borders: [w3schools](https://www.w3schools.com/css/css_borders.asp)

```
/* All borders at once */  
border: 1px solid black;
```

```
/* Each border individually */  
border-top: 1px solid black;  
border-bottom: 2px dashed crimson;  
border-left: 3px dotted deepskyblue;  
border-right: none;
```