



Cascading Style Sheets (CSS)

```
171 #content .article img.left.border {
172     padding: 0 9px 9px 0;
173     border-right: 1px dotted #999;
174     border-bottom: 1px dotted #999; }
175 #content .article blockquote {
176     margin-left: 10px;
177     padding-left: 10px;
178     border-left: 3px solid #252525; }
179 #content .article ul {
180     padding-left: 1em;
181     list-style-type: circle; }
```

Style Sheets Syntax

- ◆ Selectors are separated by commas
- ◆ Declarations are separated by semicolons
- ◆ Properties and values are separated by colons

```
h1,h2,h3 { color: green; }
```

```
body{background-color : #ff0000;}
```

<https://htmlcolorcodes.com/> to get HTML color codes

Linking HTML and CSS

- ◆ HTML (content) and CSS (presentation) can be linked in three ways:
 - ◆ Inline: the CSS rules in the `style` attribute
 - ◆ Embedded: in the `<head>` in a `<style>` tag
 - ◆ External: CSS rules in separate file (best)
 - ◆ Usually, a file with `.css` extension
 - ◆ Linked via `<link rel="stylesheet" href=...>` tag

- ◆ Three primary kinds of selectors:

- By tag (type selector):

```
h1 { color: red; }
```

- By element id:

```
#myId { color: #ff0000; }
```

- By element class name (only for HTML):

```
.myClass {color: #ff0000; }
```

- ◆ Selectors can be combined with commas:

```
h1, .myClass, #myId {color: gray; }
```

This will match `<h1>` tags, elements with class link, and element with id top-link

- ◆ Pseudo-classes define state

- ◆ :hover

```
a:hover { color: red; }
```

- ◆ :before, :after

```
.title:before { content: "»"; }  
.title:after { content: "«"; }
```

- ◆ Match relative to element placement:

```
p a {color: red; }
```

This will match all <a> tags that are inside of <p>

Text-related CSS Properties

- ◆ **color** - specifies the color of the text
- ◆ **font-size** - size of font: **xx-small, x-small, small, medium, large, x-large, xx-large, smaller, larger or numeric value**
- ◆ **font-family** - comma separated font names
 - ◆ Example: **verdana, sans-serif, etc.**
 - ◆ The following fonts are the best web safe fonts for HTML and CSS:

Arial, Verdana, Tahoma, Trebuchet MS, Times New Roman, Georgia, Garamond, Courier New, Brush Script MT

CSS Rules for Fonts (2)

- ◆ **font-style** – styles the font
 - ◆ Values: **normal**, **italic**
- ◆ **font-weight**:
 - ◆ Values: **normal**, **bold**, **lighter** or **numeric value**
- ◆ **text-decoration** – decorates the text
 - ◆ Values: **none**, **underline**, **line-through**, **overline**

Shorthand Font Property

- ◆ font

- ◆ Shorthand rule for setting multiple font properties at the same time

```
font:italic bold 12px verdana
```

is equal to writing this:

```
font-style: italic;  
font-weight: bold;  
font-size: 12px;  
font-family: verdana;
```

Backgrounds

- ◆ **background-image**

- ◆ URL of image to be used as background, e.g.:

```
background-image:url("back.jpg");
```

- ◆ **background-color**

```
background-color: red;
```

- ◆ **border-width:** thin, medium, thick or numerical value (e.g. 10px)
- ◆ **border-color:** color alias or RGB value
- ◆ **border-style:** none, hidden, dotted, dashed, solid, double, groove.. .

Border Shorthand Property

- ◆ **border:** shorthand rule for setting border properties at once:

```
border: 1px solid red
```

is equal to writing:

```
border-width:1px;  
border-color:red;  
border-style:solid;
```

Width and Height

- ◆ **width** – defines numerical value for the width of element, e.g. 200px
- ◆ **height** – defines numerical value for the height of element, e.g. 100px

```
div {  
  
    height: 50px;  
    width: 100%;  
    border: 7px solid red;  
  
}
```

Margin and Padding

Margin

Border

Padding

Content

- ◆ margin and padding define the spacing around the element
 - ◆ margin is the spacing outside of the border

```
.div1 {  
  
    margin: 70px;  
    border: 1px solid red;  
}
```

Margin and Padding: Short Rules

- ◆ **margin: 5px;**
 - ◆ Sets all four sides to have margin of 5 px;
- ◆ **margin: 10px 20px;**
 - ◆ top and bottom to 10px, left and right to 20px;
- ◆ **margin: 5px 3px 8px;**
 - ◆ top 5px, left/right 3px, bottom 8px
- ◆ **margin: 1px 3px 5px 7px;**
 - ◆ top, right, bottom, left (clockwise from top)
- ◆ Same for padding

- ◆ margin and padding define the spacing around the element
 - ◆ padding is the spacing between the border and the content

```
.div2 {  
    padding: 50px;  
    border: 1px solid #4CAF50;  
}
```

- ◆ **position:** defines the positioning of the element in the page content flow
- ◆ The value is one of:
 - ◆ **static** or **absolute**

```
.div1 {  
    position: static;  
    border: 3px solid #73AD21;  
}  
.div2 {  
    position: absolute;  
    border: 3px solid #73AD21;  
}
```

- ◆ **float:** the element “floats” to one side
 - ◆ **left:** places the element on the left and following content on the right
 - ◆ **right:** places the element on the right and following content on the left

```
.img1 {  
    float: right;  
}  
  
.img2 {  
    float: left;  
}
```

- ◆ **opacity:** specifies the opacity of the element

Floating point number from 0 to 1

```
img {  
    opacity: 0.5;  
}  
  
img:hover {  
    opacity: 0.7;  
}
```

◆ visibility

- Determines whether the element is visible
- **hidden**: element is not rendered, but still occupies place on the page
- **visible**: element is rendered normally

```
h2.a {  
  visibility: visible;  
}
```

```
h2.b {  
  visibility: hidden;  
}
```

- ◆ **display:** controls the display of the element
 - ◆ **inline:** no breaks are placed before and after
 - ◆ **block:** breaks are placed before AND after
 - ◆ **none:** used to hide element

```
.div1 {  
    display: inline;}  
  
.div2 {  
    display: block;}  
  
.div3 {  
    display: none;  
}
```

- ◆ **overflow:** defines the behavior of element when content needs more space than you have specified by the size properties or for other reasons. Values:
 - **visible** – content spills out of the element
 - **scroll** – always show scrollbars

```
#test {  
  
    width: 50%;  
    height: 100px;  
    overflow: scroll;  
    border: 1px solid #ccc;  
  
}
```