Web pages are built with HTML, which specifies the content of a page. CSS (Cascading Style Sheets) is a separate language which specifies a page's **appearance**.

CSS code is made of static *rules*. Each rule takes one or more *selectors* and gives specific *values* to a number of visual *properties*. Those properties are then applied to the page elements indicated by the selectors.

This guide has been written with CSS 2 in mind, which is extended by the new features of CSS 3.

NOTE: Because CSS produces visual results, in order to learn it, you need to try everything in a CSS playground like dabblet. The main focus of this article is on the syntax and some general tips.

Syntax

```
/* comments appear inside slash-asterisk, just like this line!
   there are no "one-line comments"; this is the only comment style */

/* ####################
   ## SELECTORS
   #################### */

/* the selector is used to target an element on a page. */
selector { property: value; /* more properties...*/ }

/*
Here is an example element:

<div class='class1 class2' id='anID' attr='value' otherAttr='en-us foo bar' />
*/

/* You can target it using one of its CSS classes */
.class1 { }

/* or both classes! */
.class1.class2 { }

/* or its name */
div { }

/* or its id */
#anID { }

/* or using the fact that it has an attribute! */
[attr] { font-size:smaller; }

/* or that the attribute has a specific value */
[attr='value'] { font-size:smaller; }

/* starts with a value (CSS 3) */
[attr^='val'] { font-size:smaller; }

/* or ends with a value (CSS 3) */
[attr$='ue'] { font-size:smaller; }

/* or contains a value in a space-separated list */
[otherAttr~='foo'] { }
```

```
[otherAttr~='bar'] { }

/* or contains a value in a dash-separated list, ie, "-" (U+002D) */
[otherAttr|='en'] { font-size:smaller; }


/* You can combine different selectors to create a more focused selector. Don't
   put spaces between them. */
div.some-class[attr$='ue'] { }

/* You can select an element which is a child of another element */
div.some-parent > .class-name { }

/* or a descendant of another element. Children are the direct descendants of
   their parent element, only one level down the tree. Descendants can be any
   level down the tree. */
div.some-parent .class-name { }

/* Warning: the same selector without a space has another meaning.
   Can you guess what? */
div.some-parent.class-name { }

/* You may also select an element based on its adjacent sibling */
.i-am-just-before + .this-element { }

/* or any sibling preceding it */
.i-am-any-element-before ~ .this-element { }

/* There are some selectors called pseudo classes that can be used to select an
   element only when it is in a particular state */

/* for example, when the cursor hovers over an element */
selector:hover { }

/* or a link has been visited */
selector:visited { }

/* or hasn't been visited */
selected:link { }

/* or an element is in focus */
selected:focus { }

/* any element that is the first child of its parent */
selector:first-child {}

/* any element that is the last child of its parent */
selector:last-child {}

/* Just like pseudo classes, pseudo elements allow you to style certain parts of a document  */

/* matches a virtual first child of the selected element */
selector::before {}
```

```css
/* matches a virtual last child of the selected element */
selector::after {}

/* At appropriate places, an asterisk may be used as a wildcard to select every
   element */
* { } /* all elements */
.parent * { } /* all descendants */
.parent > * { } /* all children */

/* ####################
   ## PROPERTIES
   #################### */

selector {

    /* Units of length can be absolute or relative. */

    /* Relative units */
    width: 50%;       /* percentage of parent element width */
    font-size: 2em;   /* multiples of element's original font-size */
    font-size: 2rem;  /* or the root element's font-size */
    font-size: 2vw;   /* multiples of 1% of the viewport's width (CSS 3) */
    font-size: 2vh;   /* or its height */
    font-size: 2vmin; /* whichever of a vh or a vw is smaller */
    font-size: 2vmax; /* or greater */

    /* Absolute units */
    width: 200px;     /* pixels */
    font-size: 20pt;  /* points */
    width: 5cm;       /* centimeters */
    min-width: 50mm;  /* millimeters */
    max-width: 5in;   /* inches */

    /* Colors */
    color: #F6E;                /* short hex format */
    color: #FF66EE;             /* long hex format */
    color: tomato;              /* a named color */
    color: rgb(255, 255, 255);  /* as rgb values */
    color: rgb(10%, 20%, 50%);  /* as rgb percentages */
    color: rgba(255, 0, 0, 0.3); /* as rgba values (CSS 3) Note: 0 <= a <= 1 */
    color: transparent;         /* equivalent to setting the alpha to 0 */
    color: hsl(0, 100%, 50%);   /* as hsl percentages (CSS 3) */
    color: hsla(0, 100%, 50%, 0.3); /* as hsl percentages with alpha */

    /* Images as backgrounds of elements */
    background-image: url(/img-path/img.jpg); /* quotes inside url() optional */

    /* Fonts */
    font-family: Arial;
    /* if the font family name has a space, it must be quoted */
    font-family: "Courier New";
    /* if the first one is not found, the browser uses the next, and so on */
    font-family: "Courier New", Trebuchet, Arial, sans-serif;
}
```

Usage

Save a CSS stylesheet with the extension `.css`.

```
<!-- You need to include the css file in your page's <head>. This is the
     recommended method. Refer to http://stackoverflow.com/questions/8284365 -->
<link rel='stylesheet' type='text/css' href='path/to/style.css' />

<!-- You can also include some CSS inline in your markup. -->
<style>
   a { color: purple; }
</style>

<!-- Or directly set CSS properties on the element. -->
<div style="border: 1px solid red;">
</div>
```

Precedence or Cascade

An element may be targeted by multiple selectors and may have a property set on it in more than once. In these cases, one of the rules takes precedence over others. Rules with a more specific selector take precedence over a less specific one, and a rule occuring later in the stylesheet overwrites a previous one.

This process is called cascading, hence the name Cascading Style Sheets.

Given the following CSS:

```
/* A */
p.class1[attr='value']

/* B */
p.class1 { }

/* C */
p.class2 { }

/* D */
p { }

/* E */
p { property: value !important; }
```

and the following markup:

```
<p style='/*F*/ property:value;' class='class1 class2' attr='value' />
```

The precedence of style is as follows. Remember, the precedence is for each **property**, not for the entire block.

- `E` has the highest precedence because of the keyword `!important`. It is recommended that you avoid its usage.
- `F` is next, because it is an inline style.

- `A` is next, because it is more "specific" than anything else. It has 3 specifiers: The name of the element `p`, its class `class1`, an attribute `attr='value'`.
- `C` is next, even though it has the same specificity as `B`. This is because it appears after `B`.
- `B` is next.
- `D` is the last one.

Compatibility

Most of the features in CSS 2 (and many in CSS 3) are available across all browsers and devices. But it's always good practice to check before using a new feature.

Resources

- CanIUse (Detailed compatibility info)
- Dabblet (CSS playground)
- Mozilla Developer Network's CSS documentation (Tutorials and reference)
- Codrops' CSS Reference (Reference)

Further Reading

- Understanding Style Precedence in CSS: Specificity, Inheritance, and the Cascade
- Selecting elements using attributes
- QuirksMode CSS
- Z-Index - The stacking context
- SASS and LESS for CSS pre-processing
- CSS-Tricks