CSS Glossary

v1.3

Table of Contents

Three Ways to Insert CSS

There are three ways of inserting a style sheet:

- External style sheet
- Internal style sheet
- Inline style

External Style Sheet

Each page must include a reference to the external style sheet file inside the `<link>` element inside the `<head>` section:

```
<head>
<link rel="stylesheet" type="text/css"
href="mystyle.css">
</head>
```

The external file should not contain any html tags and must be saved with a .css extension.

Internal Style Sheet

used if one single page has a unique style.

```
<head>
<style>
 h1 { color: maroon}
</style>
</head>
```

Internal styles are defined in the `<style>` element, inside the `<head>` section of an HTML page

Inline Styles

used to apply a unique style for a single element.

```
<h1 style="color:blue;margin-
left:30px;">This is a heading.</h1>
```

Multiple Style Sheets

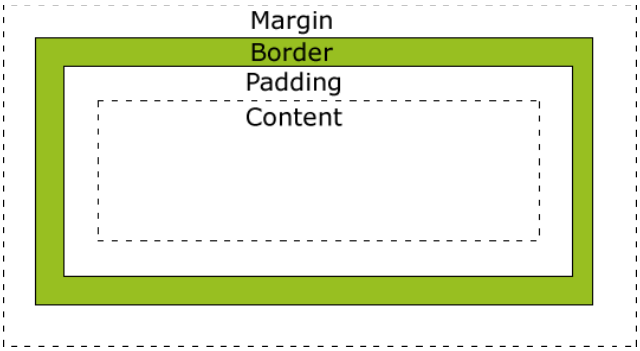
the value from the last style sheet will be used

Cascading Order

1. Inline style (inside an HTML element)
2. External and internal style sheets (in the `head`)
3. Browser default

So, an inline style (inside a specific HTML element) has the highest priority, which means that it will override a style defined inside the <head> tag, or in an external style sheet, or a browser default value.

BOX MODEL



Padding

The padding is the spacing between the content and the border (edge of the element.). We can adjust this value with CSS to move the border closer to or farther from the content. Here, the div with id 'box' will get 10px of padding all around it.

Example

```
#box {  padding: 10px;}
```

Margin

The margin is the space around the element. The larger the margin, the more space between our element and the elements around it. We can adjust the margin to move our HTML elements closer to or farther from each other. Here, the div with id 'box' will get 10px of margin above and below it, and 5px of margin to the left and right.

Example

```
#box {margin: 10px 5px 10px 5px;}
          the centering magic:
    width:50%; margin: 0 auto;
```

Comments

Comments in CSS are signified by a forward-slash and asterisk.

`/* single line */`	`/* This is a multi-line comment */`

properties

many properties

Each CSS rule can have as many properties as you like. Each of them applies to the elements that the selector applies to.

Example

```
h1 {
  font-size: 24px;
  font-weight: bold;
  border: 1px solid black;
  color: pink;}
```

This will make all <h1> headers big, bold, pink and inside of a thin black rectangle

display

```
body {display: inline;}
```

	Default value. Displays an element as an inline element (like `<span>`)
`inline`	makes the element sit on the same line as another element, but without formatting it like a block. It only takes up as much width as it needs The `inline` display value is better suited for HTML elements that are blocks by default, such as headers and paragraphs.
`block`	Displays an element as a block element (like `<p>`) It won't let anything sit next to it
`inline-block`	makes the element a block box, but will allow other elements to sit next to it on the same line.
`flex`	Displays an element as an block-level flex container. New in CSS3
`initial`	Sets this property to its default value.
`inherit`	Inherits this property from its parent element.

Position

`position: static;` the default one; not affected by the top, bottom, left, and right properties.
not positioned in any special way; it is always positioned according to the normal flow of the page. `{position: static;}`

`position: relative;` positioned relative to its normal position.
`{position: relative; left: 30px;}`

`position: fixed;` it always stays in the same place even if the page is scrolled.
The top, right, bottom, and left properties are used to position the element.
`{position: fixed; bottom: 0; right: 0;}`

`position: absolute;` is positioned relative to the nearest positioned ancestor (instead of positioned relative to the viewport, like fixed).
`{position: relative;}`

text-align

`h1 { text-align: center; }`
values: `left, center, right, justify`

font-family

`{font-family: "Times New Roman", Serif;}`

Serif	Serif fonts have small lines at the
Sans-Serif	"Sans" means without - this font
Monospace	`all char have the same width`
Cursive	Glyphs in cursive fonts generally
Fantasy	`Fantasy fonts are primarily decorative fonts`

Selectors

Selectors are used in CSS to select the parts of the HTML that are being styled. You can use several different methods for selecting an element

```
selector {rules;rules;rules;}
```

Class selectors

select HTML elements by their Class name. Unlike ID selectors, Class selectors select all elements with a matching class.

```
a.link {font-size: 12px;}
```

```
/* HTML Selected: <a
href="http://google.com"
class="link">,
<a href="http://codecademy.com"
class="link jumbo"> */
```

```
.jumbo {text-size: 1000px;}
```

```
/* HTML Selected: <a
href="http://codecademy.com"
class="link jumbo">, <span
class="jumbo"> */
```

Element selectors

You are able to select HTML elements first by simply using the name of the element.

```
body {background-color: #333;}
```

```
h1 {  color: blue;}
```

```
a {text-underline: none;}
```

ID selectors

select only a single item on a page. Like the term ("identification") indicates, ID selectors will ONLY select the first element with a matching ID

```
#thatThingINeededToStyle {
  color: blue;
  font-size: 24px;}
```

```
/* HTML Selected: <span
id="thatThingINeededToStyle"> */
```

```
a#codecademy {color: purple;}
```

```
/* HTML Selected: <a
href="http://codecademy.com"
id="codecademy"> */
```

Attribute selectors

to select HTML by their attributes.

```
a[href="http://codecademy.com"] {
  color: purple;}

/* HTML Selected: <a
href="http://codecademy.com"> */
```

```
input[type="text"] {width: 100px;}

/* HTML Selected: <input type="text"> */
```

```
input[required]{border: 1px red
solid;}

/* HTML Selected: <input type="text"
required> */
```

Child selectors

multiple selectors to get the exact elements you want, by using parental nesting.
By using >, you can select only the direct children of an element, going down only one lvl

```
ul > li {display: inline-block}
```

Selects only the first-level list items in all unord.lists in the HTML

```
ul a {text-underline: none;}
```

Selects all anchors which have an unordered list as their ancestry

```
ul + span {display: inline;}
```

Selects only spans that directly follow an unordered list

```
a ~ h1 {color: blue;}
```

Selects all h1 elements that are in the general vicinity of an anchor

Universal selector

to select all the elements in a particular range.
the universal selector is the most performance taxing selector, and should be used sparingly.

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

//Selects ALL HTML elements in the page

```
body * {color: red;}
```

/* Selects ALL children of the body */

```
div > * {color: red;}
```

/* Selects ALL first-level children of all divs on the page */

Pseudo class selectors

Pseudo Selectors can be used to narrow down a selection with certain rules.

```
li:first-child {color: red;}

/* This selects only <li> elements
that have no elements before them
    <ul>
      <li>Selected; will be red</li>
      <li>Not selected</li>
      <li>Not selected</li>
    </ul> */

li:last-child {color: red;}

/* This does the opposite; only the
last <li> will be red. */

ol li:nth-child(4)

/*This selects the 4th child of the
ordered list inside the unordered list

<ul><li><ol>
   <li>Start</li>
   <li>Inputs go between ()</li>
   <li>Actions go between {}</li>
   <li>jQuery is for chumps!</li>
</ol></li>
   <li>Inputs separated by commas</li>
   <li>Inputs blabla</li></ul>*/
```

```
a:hover {text-decoration: underline;}

/* Will underline all links when the
user puts their mouse over them */

a:active {font-weight: bold;}

/* Will make all links bold while the
user is clicking on them. */
```

Red glow:

```
box-shadow: 0 0 128px red;
```

syntax:

```
/* offset-x | offset-y | color */
box-shadow: 60px -16px teal;

/* offset-x | offset-y | blur-radius |
spread-radius | color */
box-shadow: 2px 2px 2px 1px rgba(0, 0,
0, 0.2);
```

/* Any number of shadows, separated by ,*/

```
box-shadow: 3px 3px red, -1em 0 0.4em
olive;
```