

CSS

(Cascading Style Sheets)

Used to describe the presentation of HTML document

Import CSS

To add styles to the document could be used of the following methods:

Import external file with styles

```
<link rel="stylesheet" href="style.css" />
```

Add inline styles to specific element

```
<div style="color: red">text</div>
```

Add styles directly to the document

```
<style>  
  .class {  
    color: red;  
  }  
</style>
```

Which one to use depends on the situation. Usually most of the styles stores in the external file, but some the most critical styles could be added to the document or as inline styles

CSS Structure

CSS file built from the list of CSS Rules

CSS Rule has 3 parts:

Selector - define the elements to which the rule applies

Property - one of the style property

Value - one of the possible value for the given property

```
[selector] {  
    [property]: [value];  
}
```

Universal selector

* - Selects all the elements on the page

Better to avoid using universal selector unless it's necessary

```
<h1>Hello world</h1>  
<div>This is my <span>Article</span></div>
```

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

Type selector

tagName - Selects all the elements with the given node type

```
<span>one</span>  
<span>two</span>  
<span>three</span>
```

```
span {  
  color: blue;  
}
```

Class selector

.className - Selects all the elements with the given class

Page could contains any number of elements with the same class name

```
<p class="text">The first block of text</p>  
<p class="text">The second block of text</p>
```

```
.text {  
  margin-top: 10px;  
}
```

ID selector

#id - Selects the element with the given id

Page should contains only one element for each ID

```
<div id="error">The password is incorrect</div>
```

```
#error {  
  color: red;  
}
```

Attribute selector

selector[attr=value] - Selects all the elements with the given attribute pair - name/value

```
<input type="radio" name="vote" value="one" />  
<input type="radio" name="vote" value="two" />
```

```
input[type="radio"] {  
  margin: 0 10px;  
}
```


Combinators

Combinators establish a relationship between selectors

```
<div class="A">
  <span class="B">one</span>
  <span class="C">two</span>
  <span class="B">three</span>
  <span class="B">four</span>
</div>
  <span class="B">five</span>
</div>
<span class="B">six</span>
```

```
/* Descendant: all .B elements inside .A */
/* one, three, four, five */
.A .B { color: red; }
```

```
/* Child: all .B which are direct childs of .A */
/* one, three, four */
.A > .B { color: red; }
```

```
/* Adjacent sibling: .B that follows immediately after .C */
/* three */
.C + .B { color: red; }
```

```
/* General sibling: all sibling .B that follows after .C */
/* three, four */
.C ~ .B { color: red; }
```

Pseudo-classes

selector:pseudo-class - Selectes the elements with a special state

```
/* The button which is the first/last element in its container */
button:first-child { background-color: blue; }
button:last-child { background-color: blue; }

/* The button over which the user's pointer is hovering */
button:hover { background-color: blue; }

/* The input which received focus */
input:focus { color: red; }

/* Disabled input */
input:disabled { background-color: #ccc; }

/* Links that a user has already visited */
a:visited { color: red; }
```

Pseudo-elements

selector::pseudo-element - Used to create cosmetic content for the element or allows to style a specific part of the element

```
/**
 * Only the first letter in the .text elements
 * will have a red color
 */
.text::first-letter {
  color: red;
}

/**
 * After the .text elements will shown "*"
 * with the given styles
 */
.text::after {
  content: '*';
  color: red;
}
```

Cascade

In CSS rules applied to the element in the order which they are written in the document

That means the value of the property from the last rule will override values of the same property for all other rules which are target this element

```
.title {  
  /* Will be overridden by the next rule */  
  font-size: 36px;  
}
```

```
.title {  
  /* Will be used for .title */  
  font-size: 40px;  
}
```

Inheritance

Many properties are inheriting their values from the parents of the element

So you don't need to write a rule for every element

And in opposite need to be careful and **reset** the styles from the parents if they are not expected for this particular element

```
<article>  
  My first <span>article</span>  
</article>
```

```
article {  
  color: #444;  
}  
  
span {  
  /**  
   * Will also have "color: #444" from its parent  
   * even it's not defined here  
   */  
}
```

Specificity

If an element has multiple rules with the same property, to decide what value should be used, beside cascade browser will look on the specificity of each selector and choose the highest one

Specificity calculated by the number of each selector type:

X-0-0-0 : inline styles have more specificity than any selectors

0-N-0-0 : number of id selectors

0-0-N-0 : number of class selectors

0-0-0-N : number of type selectors

<https://specificity.com>

```
<div id="id" class="class1 class2">text</div>
```

```
/* Specificity: 0121 */
div#id.class1.class2 {
  /* Will be used as has more specificity */
  color: red;
}
/* Specificity: 0110 */
#id.class1 {
  color: blue;
}
```

!important

Overrides any other declarations of the property applied to the element (even inline styles)

Better to avoid using **!important** rule as it hard to debug and maintain the code when it contains it

```
<div  
  id="id"  
  class="class1 class2"  
  style="color: red;"  
>text</div>
```

```
.class1 {  
  /* Will overrides any other declarations */  
  color: green!important;  
}  
  
div#id.class1.class2 {  
  color: blue;  
}
```