Selectors & Specificity

Selectors

Selectors are used to target the HTML elements on our web pages that we want to style.

It is a pattern of elements and other terms that tell the browser which HTML elements should be selected to have the CSS property values inside the rule applied to them.

Selector Lists

If more than one element uses the same CSS then the individual selectors can be combined into a selector list so that the rule is applied to all of the individual selectors.

Read more

Selector	Example
Type selector	`h1 {  }`
Universal selector	`* {  }`
Class selector	`.box {  }`
id selector	`#unique { }`
Attribute selector	`a[title] {  }`
Pseudo-class selectors	`p:first-child { }`
Pseudo-element selectors	`p::first-line { }`
Descendant combinator	`article p`
Child combinator	`article > p`
Adjacent sibling combinator	`h1 + p`
General sibling combinator	`h1 ~ p`

Pseudo Classes & Elements

- Pseudo Elements
- ::after
- ::before
- ::first-line

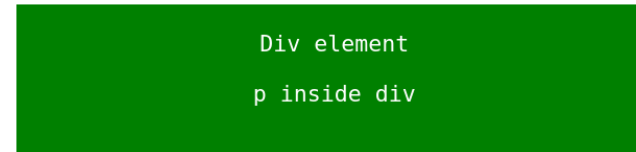- Pseudo Classes
- :nth-child(n)
- :hover

Inheritance

Inheritance also needs to be understood in this context — some CSS property values set on parent elements are inherited by their child elements, and some aren't.

Some styles, like font family, text-alignment etc., are automatically inherited by child elements from their parent element.
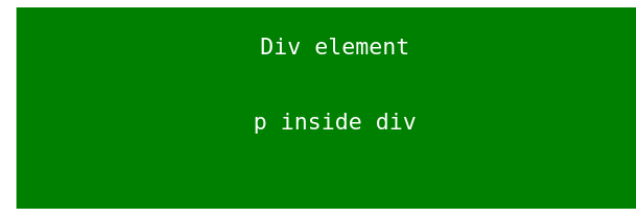
Others are not automatically inherited (margin, padding).

```css
div{
    padding:20px;
    font-family:monospace;
    text-align:center;
    color:white;
    background-color:green;
    font-size:1.5em;
}
```

Add the **padding to p** and notice the difference!

```css
p{
    padding:20px;
}
```


Div element

p inside div

Div element

p inside div

Cascade

- **Stylesheets cascade** — at a very simple level, this means that the order of CSS rules matter; when two rules apply that have equal specificity the one that comes last in the CSS is the one that will be used.

```css
/* select multiple elements separated by commas */
p, h1, h2 {
    color: green;
    background-color: grey;
}
/* when two styles set conflicting values for the same
property, the latter style takes precedence */
h2 {
    background-color: blue;
}
```

h2 background is lightblue

Paragraph background is grey.

Specificity

- **Specificity** is how the browser decides which rule applies if multiple rules have different selectors, but could still apply to the same element. It is basically a measure of how specific a selector's selection will be:

- An **element selector is less specific** — it will select all elements of that type that appear on a page — so will get a lower score.

- A **class selector is more specific** — it will select only the elements on a page that have a specific class attribute value — so will get a higher score.

```css
.special{
    background-color: grey;
}
p {
    color:white;
    background-color: lightblue;
}
```

p background is lightblue

p with class special

!important

- This is a special piece of CSS that you can use to overrule all of the above calculations, however, you should **be very careful** with using it**.**

```css
div{
    font-weight:bold !important;
    border:2px solid blue !important;
    height:20px;
}

p,.myDiv{
    padding-top:10px;
    font-weight:lighter;
    border:2px solid green;
}
```

Div element

p inside div

- **important!** Took over cascade, specificity and inheritance!!

Readings

- [CSS selectors - Learn web development | MDN (mozilla.org)](https://developer.mozilla.org)
- https://www.w3schools.com/css/css_specificity.asp
- https://www.w3schools.com/css/css_important.asp
- https://www.w3schools.com/css/css_combinators.asp
- https://www.w3schools.com/css/css_pseudo_classes.asp
- https://www.w3schools.com/css/css_attribute_selectors.asp