IT School
FROM ZERO TO HERO!

CSS Selectors - Part#2

Sesiunea 5 - 06/05/2021

IT School
FROM ZERO TO HERO!

/menti.com\

IT School
FROM ZERO TO HERO!

/Solution: CSS types exercise\

IT School
FROM ZERO TO HERO!

/Recap\

CSS selectors: universal, type, class, id

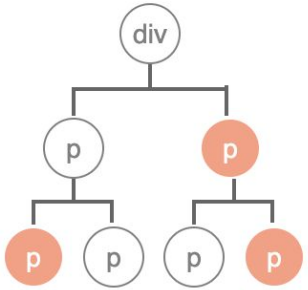
- Universal selector: * {}
- Type selector: div {}, h1 {}, p {}, html {}, body {}
- Class selector: .box {}, .fruits-list {}
- Id selector: #unique {}

CSS: selectors naming&grouping

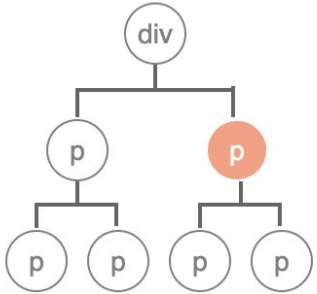
- Selectors should be meaningful (semantic)
- Selectors: only numbers won't work (Unicode)
- Selectors: containing a number will work (not necessarily recommended, but it can be used for generated class names)
- Class selector: dashes, underscores are allowed
- Class selector (HTML): space means more classes
- Class selector (CSS): space means descending in hierarchy
- Class selector: "," allows to **group** more selectors to a certain set of rules
- All selector types can be combined and used together

CSS: selectors&combinators

- Descendant combinator: div p {} (2nd selector matches if the 1st is an ancestor of any degree)



- Child combinator: div > p {} (2nd selector matches a direct child of the 1st)



CSS: selectors&combinators

- Adjacent sibling combinator: div + p {} (siblings: 2nd selector matches if it **immediately** follows the 1st)

- General sibling combinator: div ~ p {} (siblings: 2nd selector matches if it follows the 1st)

CSS: priority scheme

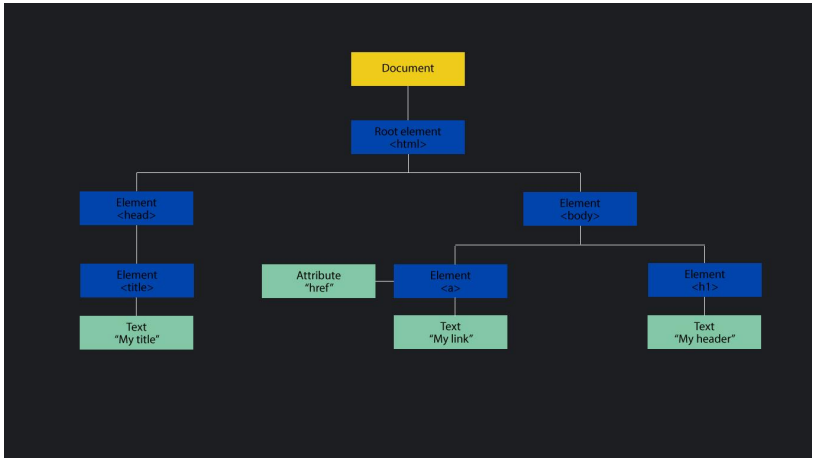
- !important (to be avoided as much as possible)
- bottom > top
- inline > internal > external (bottom > top)
- **Selectors** (bottom > top)
  - id selector > class selector > type selector > universal selector
  - 2 class selectors > 1 class selector

IT School
FROM ZERO TO HERO!

/DOM - Tree representation\

DOM (Document Object Model): tree representation

```html
<!DOCTYPE html>
    <html>
        <head>
            <title>My title</title>
        </head>
        <body>
            <a href="https://www.google.com/">My link</a>
            <h1>My header</h1>
        </body>
    </html>
</html>
```



IT School
FROM ZERO TO HERO!

/Attribute Selector\

CSS: attribute selector

```css
/* Selects all img elements with an alt attribute */
img[alt] {
    border: 1px solid #999999;
}

/* Selects all img elements with alt="Icon Smile" */
img[alt="Icon Smile"] {
    border: 1px solid blue;
}

/* Selects all img elements with an alt containing
a whitespace-separated list of words, one of which is exactly the word "Sad" */
img[alt~="Sad"] {
    border: 1px solid hotpink;
}

/* Selects all img elements with an alt value identically with "Icon" or with "Icon-" */
img[alt|="Icon"] {
    border: 1px solid yellow;
}

/* Selects all img elements with an alt value starting with "Icon" */
img[alt^="Icon"] {
    border: 1px solid orange;
}

/* Selects all img elements with an alt value ending with "Sad" */
img[alt$="Sad"] {
    border: 1px solid green;
}

/* Selects all img elements with an alt value contains at least one occurence
of the substring "Neutral" */
img[alt*="Neutral"] {
    border: 1px solid black;
}
```


Note: pay attention to bad/incomplete documentation.

IT School
FROM ZERO TO HERO!

/code\

IT School
FROM ZERO TO HERO!

/Attribute Selector: Hands On\

IT School
FROM ZERO TO HERO!

/more selectors\

CSS selectors: pseudo-class, pseudo-element

- Pseudo-element selector (selects/adds a specific part of an element):
  - p::first-line {} (selects the first line of a paragraph)
  - p::first-letter {} (selects the first letter of an element)
  - p::after {} (add cosmetic content to an element with content property)
  - p::before {} (add cosmetic content to an element with content property)
- Pseudo-class selector (reference to an existing available state or index):
  - a:hover {}, a:active {}, a:link {}, a:visited {}
  - p:first-child {}, p:last-child {}, p:nth-child {}
  - p:first-of-type {}, p:last-of-type {}, p:nth-of-type {}

CSS selectors: pseudo-class, pseudo-element

- HTML is not being crowded
- There can be implemented logic only on CSS side
- If children nodes are of the same type nth-child = nth-of-type
- Logical parameter:
  - If 2n - index starts from 1 (even indexes)
  - If 2n+1 - index starts from 0 (odd indexes)

IT School
FROM ZERO TO HERO!

/Exercise: CSS selectors\

CSS: selectors practice (using pseudo-class selector)

```html
<!DOCTYPE html>
<html>
    <head>
        <style>
            span {
                display: inline-block;
                width: 100px;
                height: 100px;
                margin-right: 15px;
                background-color: black;
            }
            span:last-child{
                margin-right: 0;
            }
            div {
                margin-bottom: 15px;
            }
        </style>
    </head>
    <body>
        <div>
            <div>
                <span></span>
                <span></span>
                <span></span>
                <span></span>
            </div>
            <div>
                <span></span>
                <span></span>
                <span></span>
                <span></span>
            </div>
            <div>
                <span></span>
                <span></span>
                <span></span>
                <span></span>
            </div>
            <div>
                <span></span>
                <span></span>
                <span></span>
                <span></span>
            </div>
        </div>
    </body>
</html>
```

IT School
FROM ZERO TO HERO!

/Q&A\

Resources

- MDN Web docs: https://developer.mozilla.org/en-US/
- W3Schools: https://www.w3schools.com/
- Please don’t take for granted everything you read on stackoverflow (read, learn, try, repeat)

IT School
FROM ZERO TO HERO!

Thank you

Next: CSS Basic Rules - Part#1