Contents

CSS

Control appearance of html elements on a webpage

Priority

- files can override in load order, even media queries
- 1000 inline CSS
- 100 id selector
- 10 class selector & attribute-selector & pseudoclass-selector
- 1 element selector
- `div#a` (101 = 100+1) > `#a` (100) –> values add up together

Selectors

```
#id_name { /* id selector: only one element per html file */ }
.class-name { /* class-selector: multiple element per html file */ }
div.class-name { /* all div elements with class_name class */ }
```

Attribute Selector

```
div[name] { /* all elements with the attribute "name" */ }
div[name="value"] { /* all div elements with name="value" */ }
div[name*="value"] { /* all div elements where "name" contains "value" */ }
div[name~="value"] {
    /* all div elements where "name" contains "value" as a word with spacing */
}
```

Pseudo-Class Selector

Pseudo-classes target states of an element (e.g., when a link is hovered or an input is focused)

```
a:link { /* never visited link*/ }
a:visited { /* link already visited */ }
a:active { /* selected link */ }
div:hover { /* mouse hover */ }
div:first-child {
    /* first child div element: in this case inside body since no parent was defined*/
}
div:last-child { /* last child element */ }
div:empty { /* elements without children */ }
div:nth-child(2n) { /* for every even child element */ }
input:focus { }
input:valid { /* vanilla -> required */ }
```

Pseudo-Element Selector

Pseudo-elements target specific parts of an element's content (e.g., first letter, before/after content)

```
p::first-line { /* first line text */ }
p::first-letter { /* first letter text */ }
p::selection { /* selection text */ }
p::before { /* before text */}
p::after { /* after text */  content: ", add some Text to the end" }
input::placeholder { }
```

Context Selector

```
div p { /* all child elements inside div, doesn't matter if inside another element */ }
div > p { /* all direct (not nested) child elements p inside div */ }
div + p {
    /* the next following direct neighbor ->
        matters if inside another element or elements are in between */
}
div ~ p {
    /* next following elements ->
        all elements on the same level, doesn't matter if elements are in between but
        matters if inside another element */
}
```

Attributes

Units

- SI units: `cm`, `mm`, `in`
- `px`, `pt`, `pc`: pixel, point, picas are fractions of 1 inch
- `em`: relative to parent element 2rem = 2*parent element
- `rem`: relative to root element

- `vw`, `vh`: relative to 1% width/height of viewport
- `%`: relative to parent element

Important Attributes

```
div {
    /* container props */
    background-color: red;
    border: 1px solid black; /* dashed, dotted */
    border-collapse: collapse; /* remove double borders by collapsing them good for tables *
    box-sizing: content-box; /* width = content-width + padding + border */
    box-sizing: border-box; /* width = width */
    display: none, inline, block; /* span = inline, div = block | height & width cannot be s
    display: inline-block; /* width & height can be specified, display elements inline but t
    float: right, left; /* align elements */
    overflow: visible, hidden, scroll, auto;
    overflow-x: visible; overflow-y: visible;
    width: 10%, 100px;
    height: 10px, 10%, 100vh;
    padding: 1px 1px 1px 1px; /* clockwise, spacing between containers */
    margin: 1px 1px 1px 1px;  /* clockwise, spacing inside containers */
    content: "some text";

    /* flexbox */
    display: flex;
    flex-direction: row, column; /* align items in a row (horizontal) or in a column (vertic
    justify-content: flex-start, center, flex-end; /* align items along the main axes row/co
    align-items: flex-start, center, flex-end; /* align items on the other axis

    /* center methods */
    margin: auto; /* center element */
    margin: auto; display: block; /* center image */

    /* typography */
    color: black; /* font color */
    font-family: Arial, Helvetica, sans-serif;
    font-size: 16px;
    font-style: normal, italic, oblique;
    font-weight: normal, bold;
    line-height: 10px;
    text-align: center left right;
    text-decoration: underline dotted red;
    text-decoration: overline wavy green;
    text-decoration: line-through;
    text-indent: 10px; /* Einrücken */
    box-shadow: 1px 1px 1px 1px red;
}

ul li {
    list-style-type: disc, circle, square, none;
}
```

Media Queries

- Conditional styling based on screen size or orientation

```
@media (max-width: 500px) {
    /* override or new selectors */
}

@media screen and (max-width: 500px) {
    /* for devices with a screen and max-width */
}
```