{ CSS Training }

*#2*

Attribute selectors

\* [href], [data-type] { ... } Attribute exists

[class="complicación"] { Match

color: crimson;

}

[class\*="motivación"] { Contains

background-color: gold;

}

[href$=".com"] { Ends

background-color: green;

}

[class~="acción"] { One in the list (exactly value)

border: 1px solid black;

}

<element data-type="corrección-light" ....

[data-type|="corrección"] { Begins (and followed by hyphen -)

background-color: GreenYellow;

}

[class="GORDO" i] { Case insensitive

font-weight: bold;

}

<span class="decisión muy grande">

[class^="decisión m"] { Starts with

background-color: coral;

}

::before ::after

::before / ::after {

content: „...”

}

pseudo-element = keyword added to a selector that lets you style a specific part of the selected elements

<h1>Hola</h1>

h1::before {

content: "¡ "

}

h1::after {

content: " !"

}

Language selector

\*:lang()

pseudo-class = specifies a state of the selected elements

try to avoid if you want to localise your page: <html lang=”en”>

<h1 lang="es">Hola</h1>

<h1 lang="es-mx">Hola amigo</h1>

<h1 lang="es-bo">Buenos días a todos</h1>

<h1 lang="es-co">Hombre</h1>

h1:lang(es) { All is selected

color: crimson;

}

:lang(es-mx) { [lang|="es-mx"] {

background-color: gold;

}

Negation

:not() pseudo class – do not match a selector

h1:not(:lang(es-mx)) {

display: none;

}

:checked

<input type="radio" name="language" value="es" checked="checked" />

<label for="es">español</label>

input[name="language"]:checked + label {

background-color: lightgreen;

}

Tutorial

**HTML:**

<input type="radio" name="language" value="es" checked="checked" />

<label for="es">español</label>

<input type="radio" name="language" value="es-mx" />

<label for="es-mx">mexicano</label>

<input type="radio" name="language" value="es-bo" />

<label for="es-bo">boliviano</label>

<input type="radio" name="language" value="es-co" />

<label for="es-co">columbiano</label>

<h1 lang="es">Hola</h1>

<h1 lang="es-mx">Hola amigo</h1>

<h1 lang="es-bo">Buenos días a todos</h1>

<h1 lang="es-co">Hombre</h1>

**CSS:**

input[name="language"]:checked + label {

background-color: lightgreen;

}

.hidden {

display: none;

}

**JS:**

let radioButtons = document.getElementsByName('language');

[...radioButtons].map(radioButton => {

radioButton.addEventListener('change', (e) => {

let lang = e.target.value;

changeLang(lang);

});

changeLang('es');

});

function changeLang(lang) {

let elems = document.querySelectorAll('h1:not([lang="' + lang + '"])');

[...elems].map(elem => elem.classList.add('hidden'));

elems = document.querySelectorAll('[lang="' + lang + '"]');

[...elems].map(elem => elem.classList.remove('hidden'));

}

Focusing on a target

:target() - URL target http://example.com/#p1

<p><a href="#p1">Lorem ipsum</a></p>

<section id="p1" ...

section:target {

background-color: yellow;

font-size: 1.5em;

}

section:not(:target) {

color: grey;

font-size: 0.8em;

}

Structural selectors

first child = represents the first element in a group of sibling elements

first of type = with same type

li:first-child { background-color: coral; }

li:last-child { background-color: coral; }

li:nth-child(3n) { color: green; }

li:nth-child(3n+1) { color: red; }

li:nth-child(odd) { color: green; } odd | even

li:nth-last-child(2) { background-color: skyblue; } 2nd element from back

div[class="task"]:first-of-type { background-color: coral; }

E:first-child

E:last-child

E:nth-child(n)

E:nth-last-child(n)

E:first-of-type

E:last-of-type

E:nth-of-type(n)

E:nth-of-last-type(n)

E:only-child = :first-child:last-child

E:only-of-type = :first-of-type:last-of-type

(This won’t work if your parent element contains more than one child with an identical tag.)

h1:only-child { background-color: blue; }

= This selects every H1 which is an only child.

:root highest level element or document tree