Jax Code Academy

and ID), and hierarchy in markup.

For HTML element diagrams:

These three elements

could be in any order

& in any hierarchy.

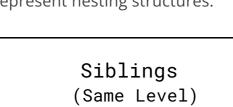
CSS Selectors CSS Selectors target elements based on html tags, attributes (including class

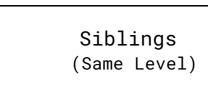
 Inline elements are siblings. Stacked & centered elements represent nesting structures.

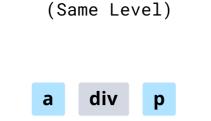
Stacked & left aligned elements are without a specified relationship



(No	relations	hip)	(Sa	ıme Lev	veI
	div				
	section		a	div	þ
	p				







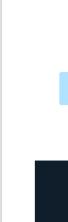






Stacked & centered elements	represent nesting structure
Stacked	Siblings
(No relationship)	(Same Level)





Basic

Select all elements with an asterisk. This selector excludes pseudo elements, so you

must specifically target *::before & *::after

UNIVERSAL

TAG by using the tag name alone (no special characters).

CLASS Select all elements with the same class by adding a period to the front of the class name. Multiple elements can have the same class.

DIRECT CHILD

Select elements that are direct children

of the parent (therefore excluding elements nested within the children)

Advanced

ADJACENT SIBLING Select an element based on the element directly above in the html. This example targets any paragraph which immediately follows a div.

comma.

Select multiple elements to apply the

same styles. Separate selectors with a

Select elements which have a specific tag

name and attribute by joining selectors. This example targets all divs with a class of .class

Select all elements which do not match the

paragraphs without a class of .class

selector inside :not(). This example targets all

MULTIPLE

NOT

WITH

HAS Select all elements with a specific

attribute.

FIRST OF TYPE Select elements which are the first of its type inside a container.

LAST OF TYPE

type inside a container.

Select elements which are the last of its

HAS (EXACT)

attribute and value.

Select all elements with a specific

ONLY CHILD Select elements which are the only child inside a container.

NTH CHILD

inside a container.

Select elements which are the nth child

Select elements which are odd OR even

inside a container based on a specified

Select elements which are the nth child

Style form inputs which have a required

Style the mouse hover state of interactive

Style the tab focus state of interactive

from the end, inside a container.

NTH LAST CHILD

REQUIRED

attribute.

based on position inside a container.

Select elements which are the nth child

pattern.

States

HOVER

elements.

TAB FOCUS

elements.

CHECKED Style form inputs which are checked.

• The anchoring element must have **relative positioning** • The pseudo element must have absolute positioning, content, and a distance relative to the boundary of the parent container, using either top, right, bottom, and/or left with a specified value

Creates an empty element before the

Creates an empty element after the

children of the parent element.

Jax Code Academy // CSS

children of the parent element.

BEFORE

Pseudo Elements

AFTER div {

> content: ''; top: 0; left: 0;

position: relative; div:before { position: absolute; content: '';

before

div

P

<a> <div></div>

 </div>

property: value;

div a

p

a.class

div#id

a

div div { Select all elements with the same html tag property: value; a div p.class

- .class {
 - property: value; #id {

div p {

 $div > p {$

div + p {

a, p.class {

property: value;

property: value;

property: value;

property: value;

- property: value;
 - - p

p

img

div a img

a

- section div section p
- div a p.class a.class
- div div.class p.class
- div.class { property: value; p p:not(.class) { p.class property: value; a.class div div title="info"
- div[title] { property: value; div title="more info" div div[title="info"] { div title="info" property: value; div title="more info" p:first-of-type {

}

property: value;

p:last-of-type {

a:only-child {

a:nth-child(3) {

property: value;

a:nth-child(odd) {

a:nth-child(2n) {

a:nth-last-child(2) {

property: value;

input:required {

button:hover {

button:focus {

input:checked {

}

property: value;

property: value;

property: value;

property: value;

}

property: value;

property: value;

property: value;

property: value;

a div a div

a

a

a

a

div

img

img

div

p

p

- a div a div a
- div a a input*

button

button

div

a

Pseudo elements are created in CSS. They are not in the flow of content, so will sit either on top of or behind other elements on the page. Pseudo elements can be created by attaching ::before or ::after to the selector for a parent element which exists in your HTML.

div {

top: 0;

left: 0; position: relative; div:after { position: absolute; div after

© Jax Code 2023