SRC – Path to local files

Relative:

./ – Selects file in current folder

../ - Selects file in previous folder

Absolute:

/ - Selects root folder (not recommended)

Units

Absolute:

px/mm/cm – Fixed size, with no regard to the screen size. px recommended.

Relative:

vh & vw – View height & View width, units relative to screen size (viewport). used mainly for big containers like <body>.

% - percentage – Relative to father container.

em – For fonts. Relative to the font size of the closest ancestors.

rem – relative to the default font size of the root (of the main element - &lt;HTML&gt;)

Position

Static – default.

Relative – moves relative to original position. stays in the flow of the page.

Absolute – relative to any ancestor who's not in a static position. gets out of the flow of the page

Fixed – relative to viewport. meaning that it keeps its position even if the user scrolls up and down. Gets out of main flow.

Sticky – a combination of all the above. Stays in the flow. Remember to specify a location parameter (top/left ect...), otherwise he won't move if scrolled.

Additional parameter the goes along with all the above, to specify how we want to place the element:

top | bottom | left | right | z-index (front and back axis, in order to control who's on top )

Overflow

Deals with content that gets spilled out of its container.

Overflow: Visible; - default.

Overflow: Hidden; – wont show the parts that overflow.

Overflow: Scroll; – will force a scroll bar on all sides, regardless of the necessity of it.

Overflow: Auto; – will create a scroll only if the content get spilled in that direction.

Pseudo selectors

Pseudo selectors in MDN

The pseudo-selectors we've learned:

:hover – defines the style of an element when the mouse cursor is on top of it.

:active - defines the style of an element when it's clicked with the mouse.

:focus – for inputs. defines the style of the input field when it's selected.

:visited – defines the style of a link that was previously clicked.

:checked – defines what happens when a radio/checkbox is checked.

:nth-child(N/even/odd) - select the nth element of a sort

Pseudo elements

Pseudo-elements in MDN

The pseudo-elements we've learned:

::first-letter – targets the style of the first letter.

::first-line - targets the style of the first line.

::selection – changes the default background color of the selected text.

Animations

אלמנט שמוגדר עליו טרנזישן ברגע שבו השתנה לו אחד הפרופרטיז כתוצאה מפסודו אלמנט וכדו' הדפדפן יבצע את המעבר בצורה שהיא הדרגתית

Controls the transition between different states of the same element.

Hover this text and see the animation.

Structure:

property name | duration | easing function | delay

Example:

transition: margin-right 4s ease-in-out 1s;

ease-in/ease-out/ease-in-out – controls the rhythm of the animation

to perform on 2 properties, add ','.

Example:

width 4s ease-in, margin 2s ease-out;

All – performs on all properties.

Example:

All 4s ease-in;

Flex Container

Css tricks – articles on css. Also got guides including Flex

Properties for the parent (Flex container):

display: flex;

flex-direction: row (left to right) | row-reverse (right to left) | column (up to down) | column-reverse (down to up)

flex-wrap: nowrap | wrap | wrap-reverse;

flex-flow: column wrap; combination of flex-direction and flex-wrap.

justify-content: flex-start | flex-end | center | space-between | space-around | space-evenly

align-items: stretch | flex-start | flex-end | center

align-content: flex-start | flex-end | center | space-between | space-around | space-evenly | stretch

gap: 10px;

row-gap: 10px;

column-gap: 20px;

gap: 10px 20px; */\* row-gap column gap \*/*

Properties for the Children (Flex items):

order: 5; */\* default is 0 \*/*

flex-grow: 4; */\* default 0 \*/*

flex-shrink: 3; */\* default 1 \*/*

align-self: auto | flex-start | flex-end | center | baseline | stretch;

Grid display

[Css tricks grid guide.](https://css-tricks.com/snippets/css/complete-guide-grid/)

[Grid game](https://cssgridgarden.com/#he)

Grid generator.

In the container:

Display: grid;

Grid-template-row: Repeat (5, 1fr); - 5 equal cells of 1 fr.

Or:

Grid-template-row: 20px 30px 1fr; - for unequal cells

Grid-template-column: \*same as above\*

In the cell:

Grid-row: 3/5; - start at row 3 and finish on row 5 (2 cells total).

Grid-colums: 3/ span 2; - same result.

Short combination of the two:

Grid-template: 3/5 / 3/span 2; (rows / columns)

2nd way to grid:

On the cell:

Grid-area: hello-item (any name will work)

On the container

Grid-template-area:

". . . hello-item ." - the dots represents empty cell

". . . ."

". . . ."; - (note the';' in the end)

box-sizing: border-box;

repeat(5, 1fr)

.item$\*8 – short to create 8 Divs named Items with according number

Display: none; (to hide elements)

Media query

Deals with style if certains critirias are met. Like screen size.

@media only screen and (min-width:200px) and (max-width: 700px) {

.ex1 {

Width:100% ;

}