CSS COURSE

**- ALWAYS SEARCH before asking-**

CSS 🡪 cascade style sheet (for presentation and visuals)

* For who 🡪 (web designer – UI developer – front end developer)
* Inspect 🡪 console to check errors
* Syntax:

Selector{

Propertyone value ;

Propertytwo: value ;

}

.selector 🡪 means that the selector is a class

* To comment🡪 ctrl slash (ظ)
* Styles calling:
* External style <link rel="stylesheet" href="css/first.css">
* Internal style <style> </style>
* Inline style (strongest but not professional)
* No identifier can begin with a number (class=”1test”) ❌

Best way to name in css is using hyphen between words (user-name)

* Color property:
* Rgb (red, green, blue / alpha)

Maximum degree is 255

All 0 = black

All max.= white

(255 0 0) means color is red

(0 255 0) means color is green

Alpha channel is responsible for transparence

* Hexadecimal color codes

|  |  |  |
| --- | --- | --- |
| **Color** | **Decimal (Red, Green, Blue)** | **Hexadecimal (#RRGGBB)** |
| **Red** | **(255, 0, 0)** | **#FF0000** |
| **Green** | **(0,255, 0)** | **#00FF00** |
| **Blue** | **(0, 0,255)** | **#0000FF** |

* Background image property:

 background-image: url(../images/download.jfif);

../ 🡪means back one step

* Background repeat property

Repeat: To repeat the image many times to fit in the space

Default value: repeat

Before the line of background repeat: attach the photo in CSS NOT in html

background-image: url(../../images/assignment-image.png);

  background-repeat: repeat;

Repeat x : تكرار عمودي

  background-repeat: repeat-x;

Repeat y: تكرار افقي

  background-repeat: repeat-y;

no repeat:

  background-repeat: no-repeat;

* Background attachment:

Default value is Scroll

Scroll (background is not fixed; disappear when scroll down)

background-attachment: scroll;

fixed (even if you scroll down; the background will appear)

background-attachment: fixed;

* Background position:

- Left/right/ center top/bottom/center

  background-position: left top;

* Using pixel or percentage (left up)

 background-position: 30px 70px;

* Background size:
* Cover

Cover the whole background BUT not responsive. As well it may crop parts of the picture to make it fit in.

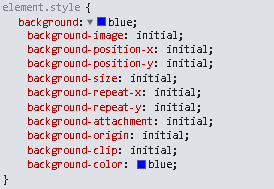
background-size: cover;

* Contain

The background is FULLY VISIBLE no matter you minimize or maximized the screen

background-size: contain;

* Percentage or pixels is to write background and it includes all other properties



* Short hand property for background
* Paddingهامش داخلي (top right bottom left )

padding: 30px; 🡪 all four directions will be padded with this dimensions

padding: 300px 400px 500px 700px; 🡪 here I specified each direction

padding: 300px 400px ; 🡪 here I didn’t specify the bottom and the left (so he will make the left like the right and the bottom like the top)

* Margin هامش خارجي

Like padding in pixel and percentage usage

Margin accept negative value

If I want to make the text in the middle of the page:

 width: 70%;

    margin-left: 15%;

    /\* don't need to write the margin right because he will generate it automatically

    from the left percentage \*/

    margin-right: 15%;

 width: 60%;

    margin-left: 20%;

easier make the margin automatic

width: 60%;

    margin-left: auto;

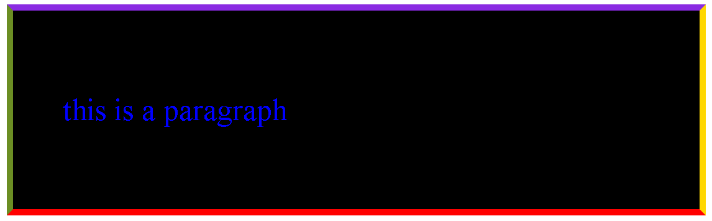
    margin-right: auto;

* Border الحافة

Three important properties (size – color - style) all with four directions (right left bottom top )

* Size (border width) right left bottom top
* Type (border style)
* border-width: 10px;
* border-color: blueviolet;
* border-style: solid;

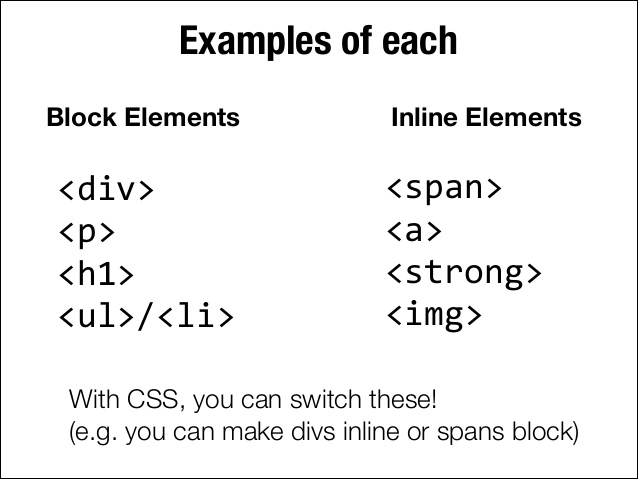
border-color: blueviolet gold red olivedrab;



* Short hand:

But here you cannot control directions

* border:10px solid green;
* Outline برا العنصر اللي انا شغالة عليه على عكس البوردر كان جوا العنصر
* No directions are controlled
* No control in radius
* Can be replaced with border
* Display



* Block

- take full width if no width

- add line break (each tag is in new line - no tag is beside the other)

- respect padding, margin, width, height

* Inline

- don’t respect width and height

- do not add line break (each tag is beside the other)

- respect padding and margin (only left and right )

* Inline-block

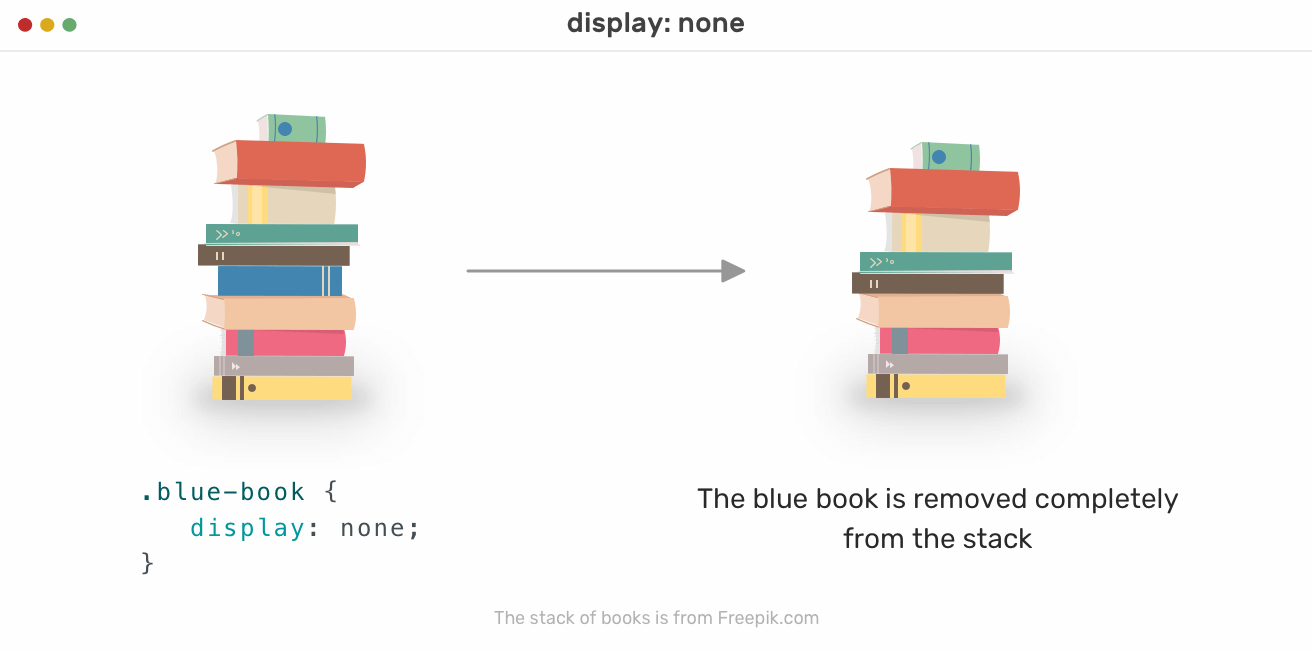
- respect padding, margin, width, height

  display: block;

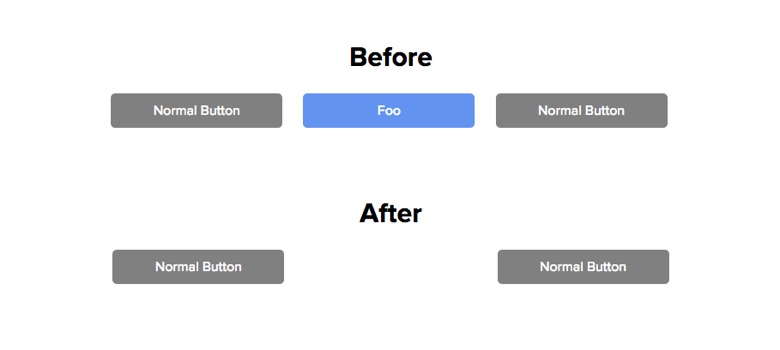
        display: inline;

        display: inline-block;

* Display none
* Used to show info and hide them using a button (example: FAQs in any website)

   display: none

* Visibility hidden



* Grouping multiple selectors

Syntax of grouping:

.first,

.second,

p

{

    font-size: 30px;

    padding:10px;

    border:3px solid green;

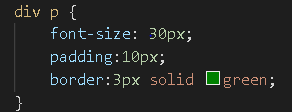
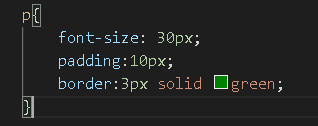
}

* We separate the selectors with comma except for the last one
* Nesting: اصطاد حاجة من جوا حاجة كبيرة

<div class="first"> <p> this is a paragraph </p>  </div>

<div class="second"> second </div>

<p>this is a paragraph without class</p>



Vs

* Dimensions (width – height)

Same apply height

* min-width: 400px;
* Means start from 400 to infinity (not less than 400 but can be more than)

من اول 400 الي ما شاء الله

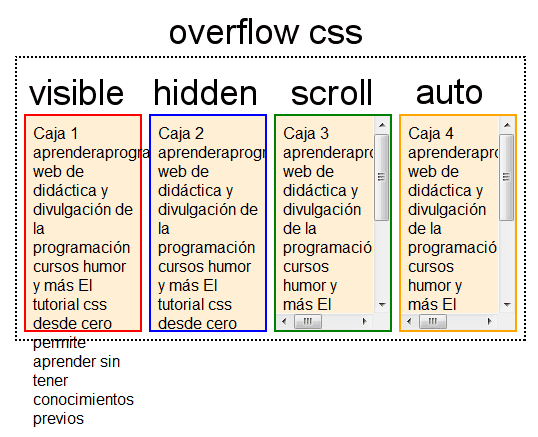
 max-width: 600px;

* The limit: not more than 600 (but can be less than this)
* Use both minimum and maximum to create a range
* To set a fixed width use width property only
* Fit content property:
* width:fit-content;
* overflow x and y:
* the default value of overflow is visible
* overflow:visible
* Overflow hidden: اي حاجة زايدة عن الطول والعرض اللي انا مديها له هيقطعها
* Overflow scroll:

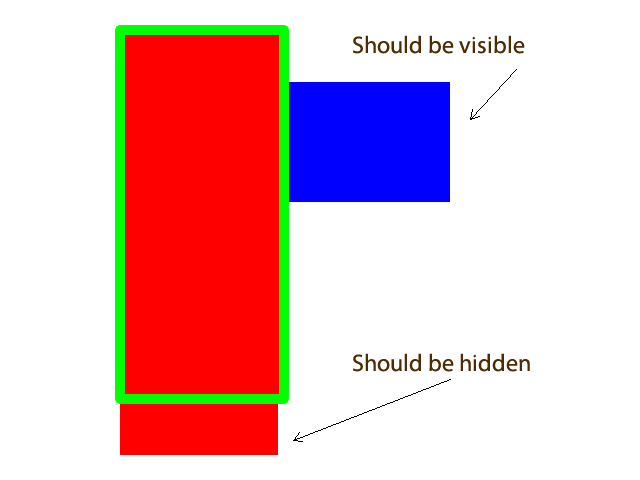
العيب هنا انه هيفضل حاطط علامة السكرول حتي لو كلام قل وما بقيناش محتاجين السكرول

* Overflow auto:

نظام هتزود هازود هاتنقص هانقص :)



* Overflow-y



* Overflow-x
* Text shadow:
* Properties: H-shadow (horizontal)**,** V-shadow (vertical), Blur, Color
* text-shadow: 0px 0px 0px blue;

no commas between numbers don’t forget to write the unit (pixel)

* Text alignment
* Default value: left
* text-align: left;
* text-align: center;
* text-align: right;

text align بيتعامل مع التيكست علي انه جزء واحد

* Direction:

مفيد لو الكلام العربي مش مظبوط يعني مثلا النقطة حاططها في الاول مش في الاخر





* Vertical align
* Top:



* Bottom:



* Middle:



* Text decoration:

   text-decoration: line-through; 🡪 ~~menna~~

text-decoration: underline; 🡪 menna

    text-decoration: overline; 🡪 menna (المفروض بيكون خط فوق الكلام)

* Common usage:

To remove the underline of the links



* Text transform

text-transform: capitalize; 🡪 Menna Ahmed Mohsen

    text-transform: uppercase; 🡪 MENNA AHMED MOHSEN

    text-transform: lowercase; 🡪 menna ahmed mohsen

* Text spacing
* Letter spacing (accept negative value)
* letter-spacing: 7px; Menna 🡪 M e n n a (for example)
* Identation (the space before starting the text)
* text-indent: 15px;

example:

menna

menna

* Line height التباعد بين الاسطر
* line-height:1;

كدا بقوله يضاعف المساحة بين السطور مرة

Most fit use is 1.5 or 1.6

* Word spacing: (space between words)
* word-spacing: 7px; menna ahmed Mohsen 🡪 menna ahmed mohsen (for example)
* white space:

 - normal:

 - nowrap

* word break: (most used with links)

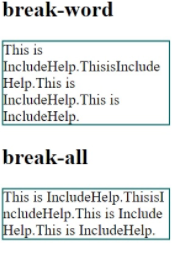
- break all:

word-break: break-all;

Break words at any character:

- break word:

word-break: break-word;



* Text overflow
* div {
* white-space: nowrap;
* overflow: hidden;
* text-overflow: ellipsis;
* }



يطلعلي النقط دي كدت لأن الكلام لسة ممتد

* Inherit
* Typography
* Font family:

Use web fonts so that they are visible to all users even the users who don’t have the font downloaded in there pc

Example: google fonts



* Font size
* CSS unites

1. Px 🡪 pixel
2. Em 🡪 time (1 em 🡪 one time – 2em🡪 two times)

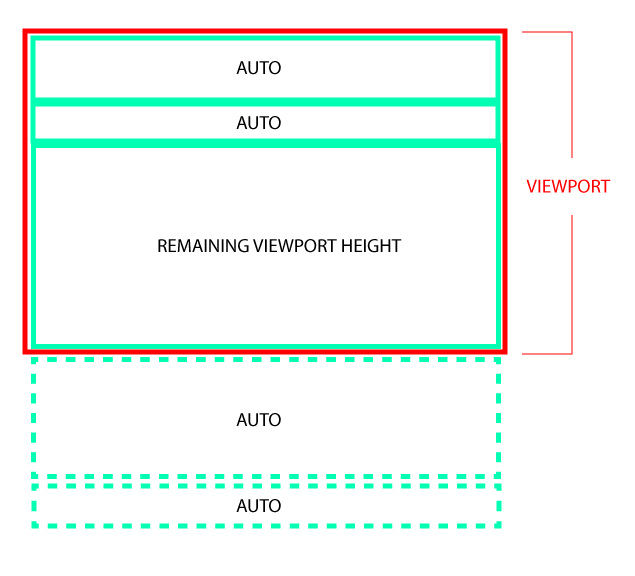
1em🡪 It will multiply by one time from the thing it inherits from

1. Rem 🡪 root time (will take from the root of the page which is the html)
2. Vw 🡪 view port width

The whole white space in the page

1vw width = 1% of screen pixels

**(RESPONSIVE)**



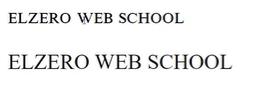
* Font style

Change style of word (italic – normal – oblique)

* Font variant (rarely used)

Make fonts in capital letters but smaller than the normal caps

* font-variant: small-caps;



* Font weight (from 100 🡪 900)

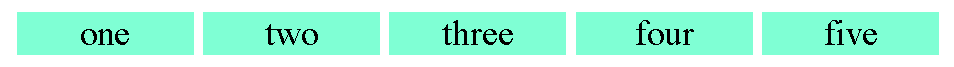
Note: if default value of something is bold you can make it normal by:

 font-weight: normal;

* Mouse curser

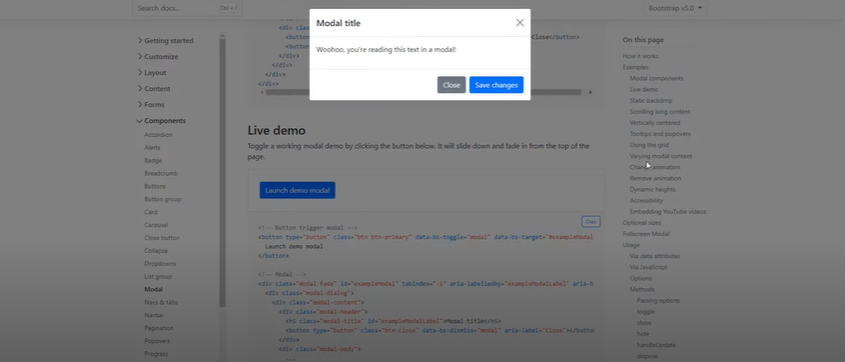
Guide the user to the behavior he needs to do (enhance user experience)

     cursor: pointer;

* Float and clear
* The CSS calculations:
* Content width is 100%
* Determine the margin you want and then put the width
* margin-left: 1%;
* width:calc(94%/5)

* Opacity: الشفافية للمجتوى كاملًا

Max is 1 (100%)

 Application:

* Position
* Default value of position:
* position: static;
* Position relative (بيتحرك بناءً على نفسه)

When I say left:30px 🡪 means the left of the div its self

* position: relative;
* top: 30px; (you can write a negative value)

top

Left right

bottom

* Position absolute (هيتعامل كأن العنصر مش موجود في الصفحة. هيبدأ يحرك من اول الصفحة)

Work as the dimensions of the page

* Position fixed يفضل ثابت معاك في الشاشة
* Position sticky

Must determine where it will stick

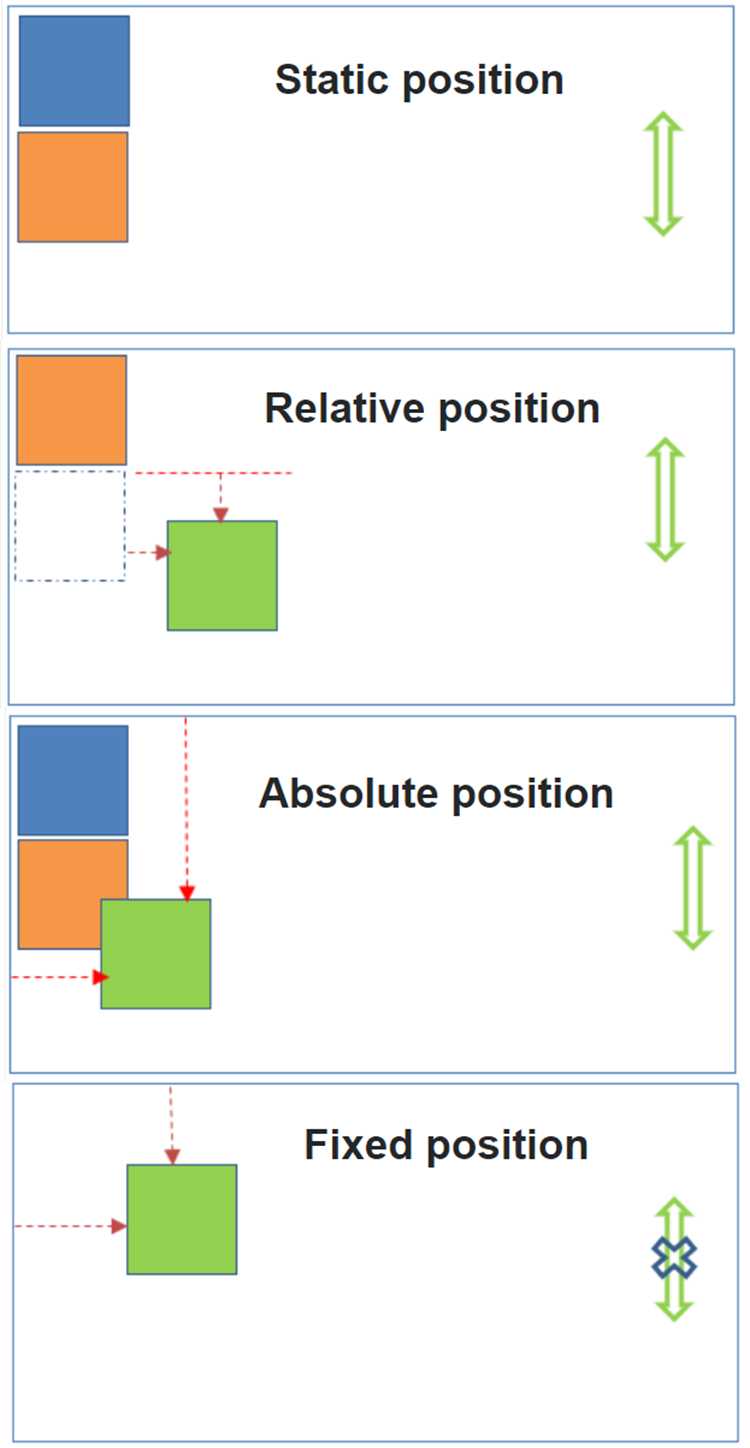
Application: navigation bar

Example:

position:sticky;

top:0px;

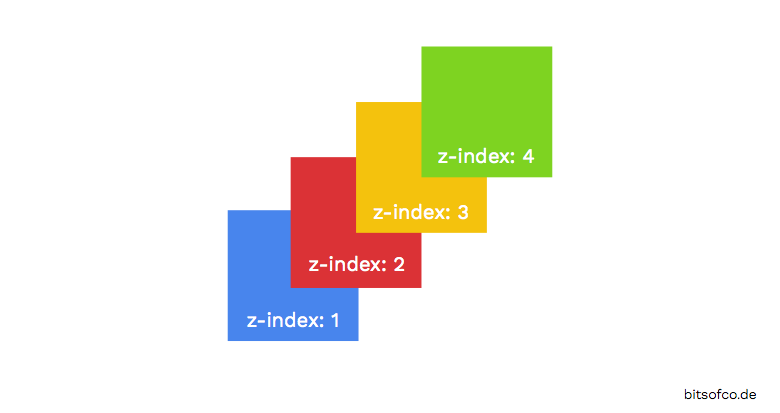
قأول لما يوصل لتوب 0 هيفضل ثابت



* Z index (layers)

I must use a position property in order for this property to work

Application: used with pop-ups



* Lists styling

    list-style: type(bullet wla numbers..) position(inside wla outside..) image(url);

* Table styling
* Border spacing 🡪 the thickness of the lines between the cells
* Pseudo classes (مزيف)

A pseudo-class is used to define a special state of an element.

For example, it can be used to:

1. Style an element when a user mouses over it
2. Style visited and unvisited links differently
3. Style an element when it gets focus

Syntax:

selector:pseudo-class {

    property: value;

  }

* Pseudo elements

A CSS pseudo-element is used to style specified parts of an element.

For example, it can be used to:

1. Style the first letter, or line, of an element
2. Insert content before, or after, the content of an element

selector::pseudo-element {

    property: value;

  }

* Don’t forget the content “” line
* Before and after content
* Vendor prefixes

If the browser is not supporting a feature in the css we write the prefix for each browser

-webkit- 🡪 chrome, safari, new opera versions

-moz- 🡪 firefox

-ms- 🡪microsoft edge – ie

-o- 🡪old versions of opera

Website to check on the feature: <https://caniuse.com/>

Also use VS code addon to add the prefixes automatically

Use task runner

* Border radius بتتحكم في حواف العنصر

Circle is formed when the border radius is half the width/ height (50%)

height:70px;

    width:70px;

    border-radius: 50%;

    border-bottom-left-radius: ;

    border-top-left-radius: ;

    border-top-right-radius: ;

    border-bottom-right-radius: ;

* Box shadow

H shadow | V shadow | blur | spread | color | inset

Can make more than one shadow

* box sizing
* Transition (VERY IMPORTANT)
* Transition duration
* Transition delay
* Transition property
* transition-duration: 1s;
* transition-delay: 2s;
* transition-property: 1s;
* Important declaration

The !important rule in CSS is used to add more importance to a property/value than normal.

In fact, if you use the !important rule, it will override ALL previous styling rules for that specific property on that element!

#myid {

    background-color: blue;

  }

  .myclass {

    background-color: gray;

  }

  p {

    background-color: red !important;

  }

* Only vertical margins collapse (they don’t add up)
* Bigger vertical margins win
* Nesting doesn’t prevent collapse
* CSS variables

Global variable 🡪 visible to all system

Local variable 🡪 visible to one element

* Flex box PROPERTIES FOR PARENT
* Identify the flex container
* Display:flex;
* Flex direction default value is row
* When we change direction of the page to rtl the flow will automatically change
* The flex container properties are:
* flex-direction
* flex-wrap
* flex-flow
* justify-content

- The flex-start value aligns the flex items at the beginning of the container (this is default)

- The flex-end value aligns the flex items at the end of the container

- The space-between value displays the flex items with space between the lines

- The space-around value displays the flex items with space before, between, and after the lines

- The space-evenly value make space between and around together

- HORIZONTALLY

* align-items

-stretch 🡪 The stretch value stretches the flex items to fill the container (this is default)

- VERTICALLY

* align-content

- align content not items

* align-self:

This property accepts the same values as align-items and its value for the specific item.

* flexbox PROPERTIES FOR CHILD
* flex grow 🡪 Make the one flex item grow (eight) times faster than the other flex item

default value (0)

* flexshrink🡪specifies how much a flex item will shrink relative to the rest of the flex items.

Default value (1)

* order 🡪specifies the order of the flex items.
* Flex-basis 🡪 property specifies the initial length of a flex item
* Flex 🡪 property is a shorthand property for the flex-grow, flex-shrink, and flex-basis properties.
* display inline-flex
* align self
* games for practice: <https://flexboxfroggy.com/>
* filters
* not supported by all browsers
* filter:grayscale(100%)
* filter:blur(100%)
* filter:invert(100%)
* search for different filters
* gradients
* linear gradient (direction angle color stop1 colorstop2 colorstop3)
* background-image: linear-gradient(to right, red , yellow);
* You can specify the space covered by a certain color 100px … 50%...
* background-image: linear-gradient(to right, red 50% , yellow);
* Pointer events
* Grid template columns: <https://www.youtube.com/watch?v=wYSAjaB3mL8&list=PLDoPjvoNmBAzjsz06gkzlSrlev53MGIKe&index=57>

1. Display grid
2. Grid-template-columns property 🡪 you can use px, percents…..
3. Grid-template-row 🡪 same as the above property
4. Row-gap and column-gap

* Grid for child:

grid-column: ;

grid-row: ;

* Transform (2d)
* Scale x بيغير العرض
* Scale y بيغير الطول
* Scale -ve value 🡪 flip
* Rotate

- deg 🡪 degree

- rad🡪 radians

- grad🡪 gradients

- turn 🡪 turn

* Translate (best for moving)

- +ve move to right / up

- -ve move to left / down

* Skew (انحراف)
* Matrix

The matrix() method combines all the 2D transform methods into one.

matrix(scaleX(),skewY(),skewX(),scaleY(),translateX(),translateY())

* Transform origin

بيلف حاولين ايه

Default value: 50% 50% (center center)

* 3D transform
* Back face visibility
* Animation
* Selector reference

\* 🡪 all elements in the page

Div p 🡪 target all paragraphs found in divs

.parent .child 🡪 search for class named child in class named parent

.class-name < .child 🡪direct child

Element + element 🡪 Selects the first <p> element that is placed immediately after <div> elements