Terms

1. K.] = hot key
2. O.] = difference/similarities with other programming languages
3. ] = knowledge

[BI. F] = built in func

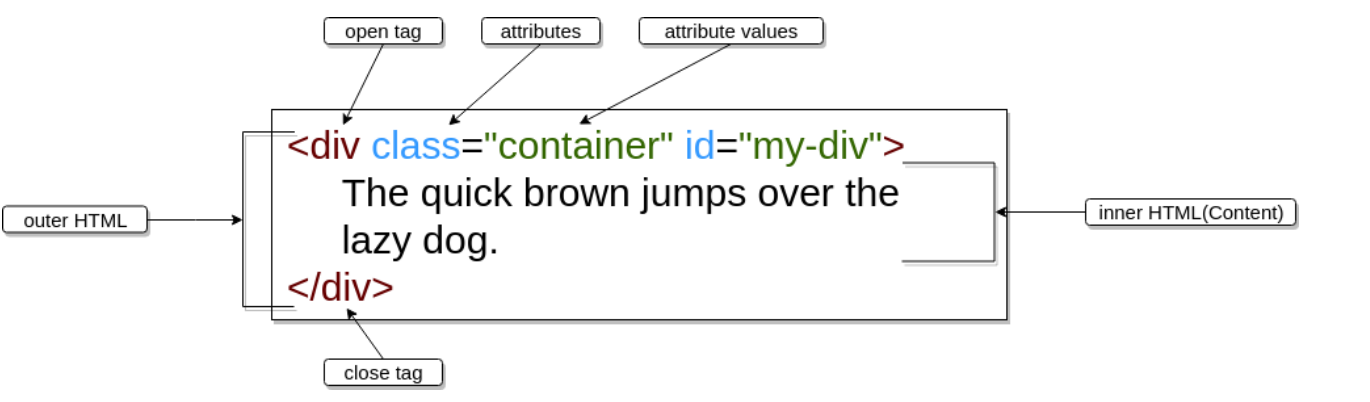
[T. ] = terms

green = keyword

red, blue = important hierachy

purple = my own observation

- html: mark up language



- run doc: drag vs file to chrome

- type 'html', select 'html:5' // automatically gives outer html format

- 'charset' -

- <meta name="viedwport" 'content'="width=width-device" ....>

//automatically scale smaller (w/ everything fits) for diff devices

-

- [IMP] all html elements <x> can add class within it, purpose: for same type to be seperated for css designs

- <> <words can also be here/>

- add lines

1. use <p> </p> for separate lines //has additional blank line between

2. <br> //='/n'

3. <hl> //horizontal breaking line

- words aspects

1. <strong> x </strong> //make x bold in paragraph

2. <em> x </em> //make x italic

3. <p> </p> can be put in <ol> or <ul>

- use utf-8 so that there'll be no 乱码 for diff languages mixed

- add style



within head:

<style>

p { color: rgba(255, 0, 0, 0.62);

background-color: #eeeeee;

font-size: x px;

font-family: x;

font-weight: bold;

text-align: right; //have left, middle

text-transform: uppercase; //all upper case

text-decoration: underline linethrough overline; //all effects added in once, overline: x

text-shadow: x px x px red; //shadow: x x y y, y y is optional

letter-spacing: 10px; //space between each char

line-height: x px; //if have font-background, its gap between them.

word-spacing: 20 px; //space between each word

}

.xx { //for class

}

#xy { //for id add hashtag

}

.box div{

}

</style>

=> p: paragraphs after style is selected for styling

=> color: font color, contains: x color, rbg (0, 0, 0), rgba (0, 0, 0, 0.-1), #ccyy00 //hex

=> background-color: background color for font

=> font-size: px, x em //x times bigger, x vw //viewport width -> x% of current divice, cm…

=> font-family: can pick x, y, z; if no x, use y…

=> [IMP EXP] .xx refers to another <p> but assigned class="xx". since its .xx, it will have same

features as p, but can add additional properties to {} in .xx

- when >1 same type assigned id, use 'class' afterthen is fine

=> [IMP EXP] #xy refers to another <p>, same as above just "xy" is in id form

=> [IMP] EXP: > 1 of <div class=".xx box">

=> refered as '.box div{' .xx is previously mentioned, so only .box needed, div refers

to all div elements

- [H. K.] assign x lines with same tag

(windows) press ALT + shift, select lines, type tags

=>

=>

=>

=>

=>

=>

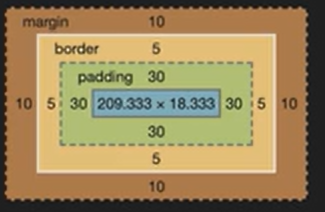
=>

=>

- block



space that 'item' (within element)(blue and orange) took

 - boss model: these elements will be shown in

-(a0) margin: x px;

-(a1) padding: y px;

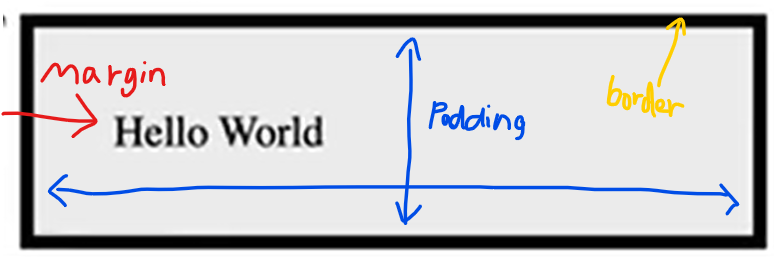
=> background color will be within padding

=> margin & padding can also assign padding-top ... so bot no extend size

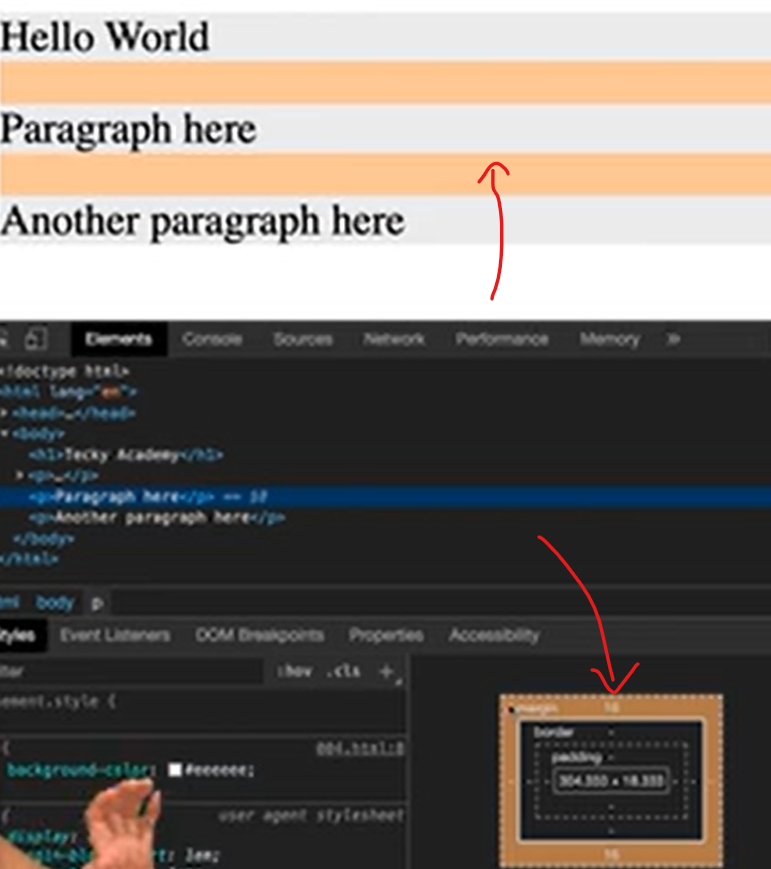
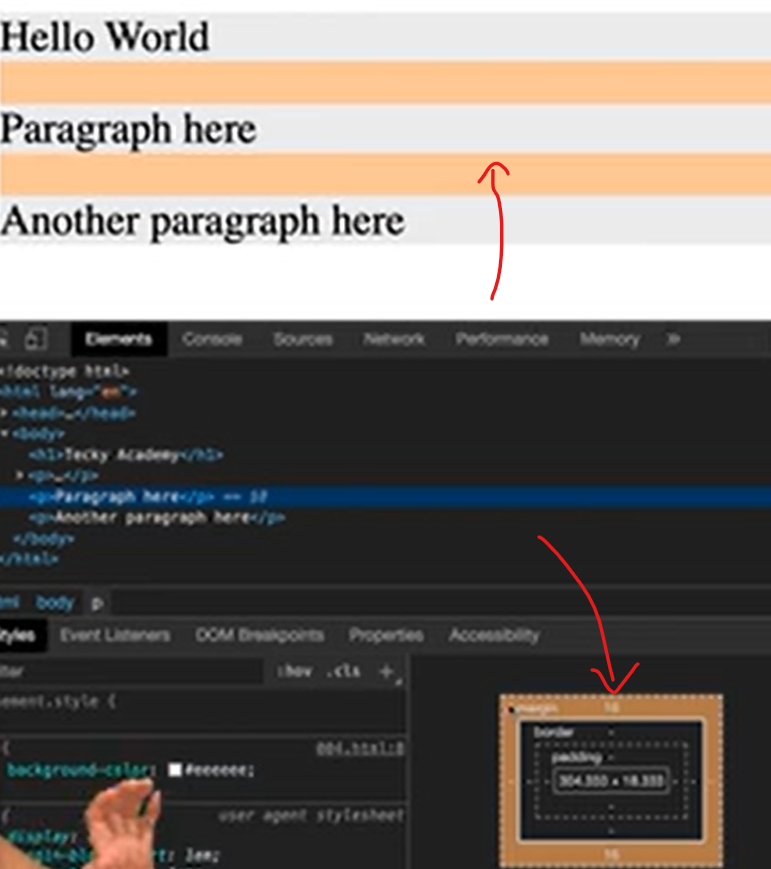
=> or (s=size): padding: (2) ups&downs left&right|| (3) ups downs left&rights

|| (4) ups rights downs lefts

-(aoptional) border: solid z px;



- 'fixate' mode (line, border, etc)



- box element size: height - (margin + border + padding + font) height, vise versa for width

=> can also assign width / height: x unit;

- [IMP] make border as circle

border-radius: x px; //for all

or: (4) upleft px upright px downleft px downright px

\*padding might effect the circle => rectangular like circle

- create word in assigned order within box

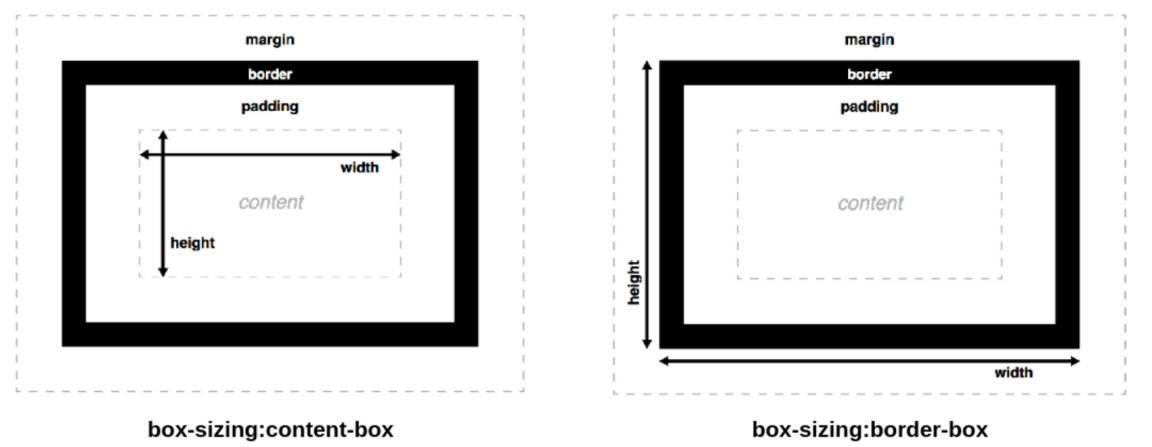
assign padding and margin smaller px for starting and overall text box, then add:

weight and height in designated size



- box-sizing: border-box;

padding and border are included in width & height



=> border-radius x px; //px size smaller and more round

Define for users (element types with no front-end changes)

Inline elemtns

- <p> this <span class="xx2"> is </span> paragraph </p>

-> does nth, for inline, just assign specific word to be assigned into class

that can be edited individually in <style> p {} .xx2 {assign properties}

 //IS is assigned xx2

- <span>: The most generic inline element that is usually used for styling purpose.

<a> : Link to external web page.

<b> : Bold text

<i> : italic text

\*\* <br> : Line breaker.

- <header> </header>

-> tells this is a header

- <div> </div>

- just like <span>, it does nth and can assign class/id => can stand alone

- [IMP] recommended to use for all, since no default settings (unlike <h3> that

has default settings with margin/etc)

- <article> </article>

- tells this is x content

- <section> </section>

- tells paragraphs

To check for all html elements, go to: mdn

- not recommended w3 -> for beginners

Document metadata

<link/>: Link external resource to html. commonly used to link stylesheet (css)

<meta>: Includes meta tag (Search keywords, utf-8...)

<style>: Tag to assign style(CSS)

<title>: title of the current page

Block elements

tags as <x> </x>

Embeded (can be)

image, audio, video

scripting

<script>,<noscript>: <script> - loading javascript to the webpage

<noscript> - showing explanation text when javascript is disabled.

= can load JS file or directly type JS code

<canvas>: a special element uses Canvas API or WEBGL to draw graphics and animations.

tables

<table>: The outermost tag of the table.

<thead>,<tbody>,<tfoot>: The header, body and the footer part of the table

<tr> : Row in table

<td>,<th> : Columns in tables while <th> is only for the header section.

The colspan allows you to have a column merged with the adjacent column.



forms

<form>: The outermost tag of the form.

<label>: The label tag for the inputs

<input>: Interactive controls that allow user to input values.

type="text": Normal Text input

type="password": Password Text input

type="email": Email input

type="date": Date input

type="number": Number input

type="radio": Radio button input

type="checkbox": Checkbox input

type="submit": Submit Button

type="reset": Reset Button

<select>: tag to select > 1 options

<option>: Options for selection

add linkage to CSS file

<head>

...

<link rel="stylesheet" href="style.css">

...

</head>

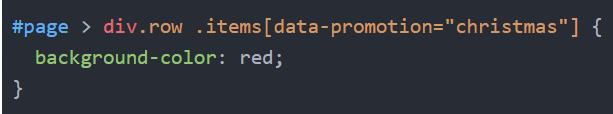
CSS

- put <style> in html's <head>

CSS selectors to apply tailored rules

| **-Selector** | **Description** |
| --- | --- |
| div | Select the element tag |
| .myClassName | Select element based on class name. Class name can be reused. One element may have more than one class. It would be written as class='class1 class2 class3' |
| #myId | Select element based on id. id must be unique in page. One element has only one id |
| [attr=value] | Select element by matching the value of the attribute |
| div.outer div.inner | Select element by parent and child elements |
| div.parent > div.child | Select the immediate children div.child under the div.parent |
| div:hover | Pseudo selector for mouse hovering the div element |

- [attr=value] EXP



CSS style box + text

| **Category** | **Usage** |
| --- | --- |
| ****Box**** | control the box-model properties of the elements. It includes the margin, padding and border properties. |
| ****Text**** | Tthe text in HTML elements. It includes the fonts,color, size and decorations of the text. |
| ****Display**** | control how the elements are displayed relative to the adjacent elements |
| ****Position**** | control the positions of the elements. |
| ****Animation**** | Most rules added in CSS3 to add the animation capability for CSS. |

CSS flexbox

concept to design flexible responsive layout without using positioning

CSS div items

Given exp: <div class="previous">Before</div><div class="after">After</>\*/

div.previous {...}

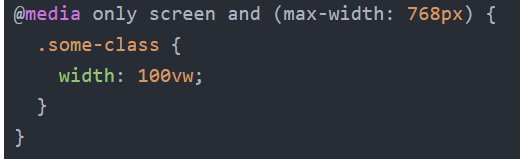
div.after{...}

div.overflow{...} //when content has longer width of element (x & y specify horizontal/not)

div.invisible{...} //cant be seen but takes space

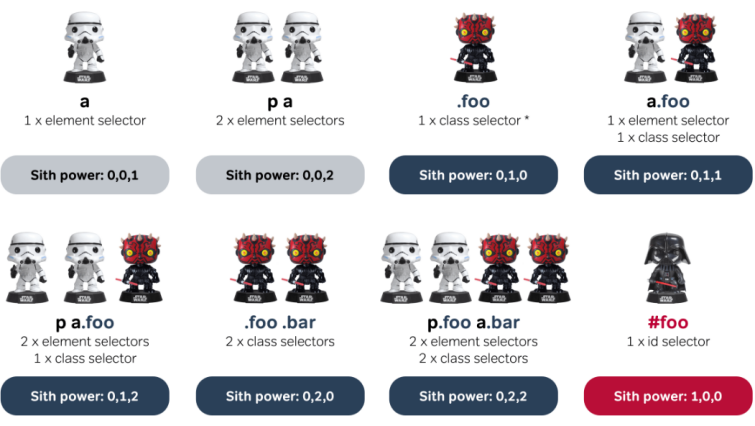
CSS media query

modify css according to device's type and resolution. EXP:



cascading order

specificity: ordering rules. EXP:



flexbox / flex layout (CSS)

.biggerdiv'sclass { display: flex; } //block is individually, each takes all line

all elements layed in x style

- [IMP] flex style:

every item exist individually (no overlap), row format

- block style:

column (up to down) format

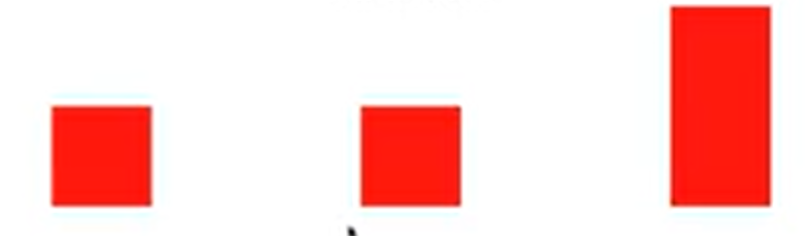
boxes position (within <style>)

(row -> left to right)

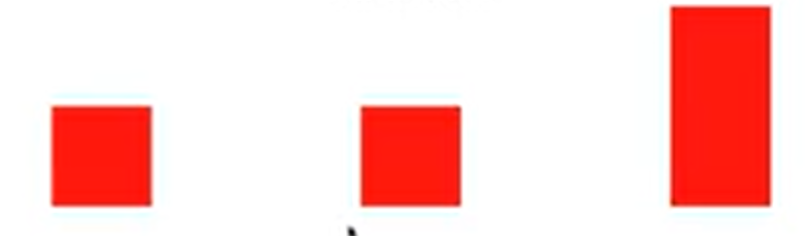
& x{ justify-content: center/...}

all moved to right: flex-end;

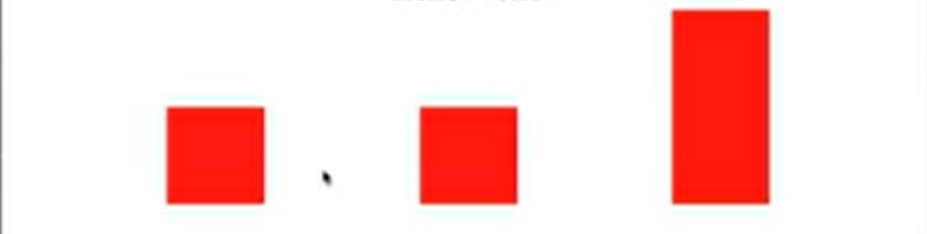
space-around: widely spread



space-between: use all space to spread



space-evenly: euqally distributed



(column -> up to down)

& x{ align-items: center/...}

stretch: all stick & start from top



- <div class='x'> without content can be used to create box. in <style> assign x height & width

-- within a box's {flex: x;}, this spread the box automatically according to x's gap (row format,

left to right). width -> overrided. According to browser size.

MUST NEED 'display:flex;' in bigger div

(me: flex takes all row's space, if there's more box's {flex: x;), then flex's total space can be

splitted equally: e.g. this flex box will be split half if both flex x is same



- {flex-wraps: wrap;}

if used displayed: flex but its squeezed ignored width in same row's line, this func

will make items 'go down' and use its size

-> shopping cart trick: Flexbox & Responsive 20:30

- phone size checking: @media screen

@media screen and (max-width: 480px) {

classx: {…}}

// if screen's max-width within 400px, do…

- {flex-direction: column;}

// column format, from up to down

- when input operators (> <) in html, type: &lt; (<) || &gt; (>) || &copy; …

- position validate

movement (slide down/up) of the object

all can assign: top, bottom, left, right: ..;

within <style>, xclass {position: x;}

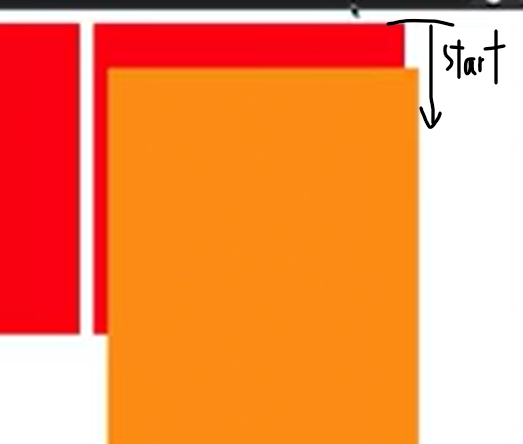
relative;

//still stays, but changes according to added(assigned) pos (top: xpx...)



//[IMP] if there's a box above assigned box (orange), the added pos will

make it starts from the box instead of screen. 对上最近的box开数



absolute;

// lost its pos, start from left top + added pos (top: xpx, left: xpx)



EXP: a shape that don't change (when scaling) its shape



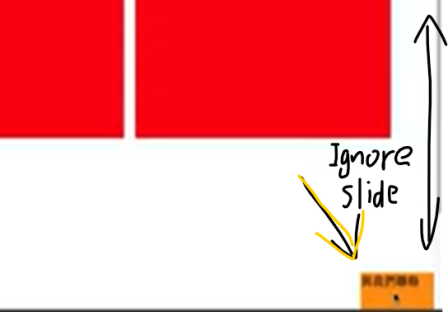
-> when sqeeze up & down, the box's

static;

-> default, like red box

fixed;

//fixed exact pos in the page, ignore slide and always same pos



sticky;

// stays in assigned pos. when drag down, it sticks to the screen's edge that

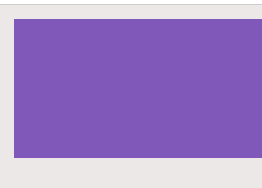
touches it



- [IMP][H. K.] select emoji: WIN + .

- to make header have no space indentation

in <style>, body {margin: 0;}



Select elements/tags

- \*{...} //all elements

- A \*//select all elements inside A

- Adjacent sibling selector: select all y tag elements followed by X tags' class

p + .intro {…} //all <p> under <div class="intro" >

or div + p {…}

- sibling selector: A ~ B //selects all B that follow a A.

EXP: <A>

<B>

- child selector: A > B //selects all B that are a direct children A

EXP: <A>

<B> </A>

- Pseudo-selector (list like 'index' selector)

:first-child //selects all first child elements.

p:first-child //selects all first child within p elements.

div p:first-child //selects all first child p elements in a div.

- Only Child Pseudo-selector

span :only-child //selects the <span> elements that are only child in x elements

ul li :only-child //selects the only <li> in <ul>

- :last-child

:last-child.small // selects last child in div class="small"

- :nth-child(A)

- :nth-child(8) //selects all elements that's 8th child of x element

- div p:nth-child(2) //selects 2nd p in every div

- :nth-last-child(A)

//counts from back

EXP: bento:nth-last-child(3)

- when you want bento only but there's also other elements that

are selected, you type x :nth-last-child(3) (<x>)

- first of type selector

:first-of-type //select first element of that type

p:first-of-type //selects first p

- :nth-of-type(A)

selects element based on its type and order

exp: div:nth-of-type(2) selects the second instance of a div.

.example:nth-of-type(odd)

- :nth-of-type(An+B)

selects every nth element starting from B (including this)

EXP: span:nth-of-type(6n+2) //selects every 6th, starting from 2

- :only-of-type

selects the x element type (if its the only) within x element

EXP: p only-of-type //selects a <span> within any <p> if it is the only <span> in there.

- :last-of-type

EXP: .small:last-of-type //same with above but for 'last' & select class (span class="small")

selectors

- empty selector

- x:empty //Select x element/tag(s) without children

- Negation Pseudo-class

- :not(X) //Select all elements that don't match the negation selector

EXP: :not(.big, .medium) //selects all elements that dont have class="big" or class="medium".

- Attribute Selector

xtag[attribute-type] //select all elements with specific attribute

EXP: a[href] //all <a> with href='anything'

- attribute value selector

[attribute='value']

EXP: a[class="link"] //<a class="link"...>

- attribute value starts with specific character

[attribute^="value"]

- Attribute Ends With Selector

[attribute$="value"]

EXP: img[src$='...']

- attribute wildcard selector

[attribute\*="value"]

Select all elements with an attribute value that contains specific characters anywhere

EXP: img[src\*="/thumbnails/"] selects all image elements that show images from the "thumbnails" folder.

CSS animations (simple, without javascript)

@keyframes xclass{

from {margin-left: 100; width: 300%;}

to {margin-left: 0%; width: 100%;}}

/// more - check advanced CSS3.doc & Animate.css

|  |  |
| --- | --- |
| animation-name | The name of the animation defined with @keyframes |
| animation-duration | The duration of the animation |
| animation-timing-function | There are multiple-type of built-in timing functions like ease,ease-in and linear |
| animation-iteration-count | You can set how many times the animations are being run. In this case, we run the animation infinitely |

Transform can: rotate,scale, skew or translate elements.

EXP: div.rotate-square{transform: rotate(1turn);}

bootstrap

reusable code written in html, css, JS

install bootstrap

CDN - Content Delivery Network

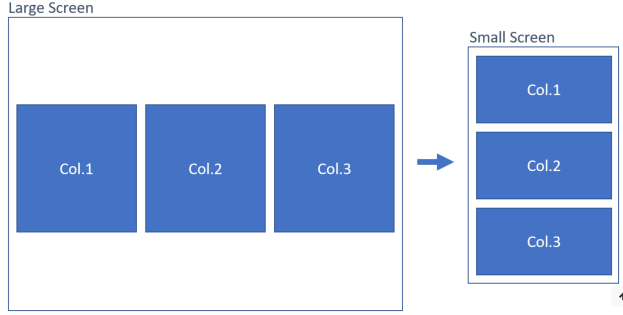
a globally network of proxy servers deployed in multiple data centers to deliver static content to the users in a high performant way.

- proxy network: provides safe gateway between user and internet

benefits: includes plugins like: HTML5Shiv & Respond.js

reduce css codes to write

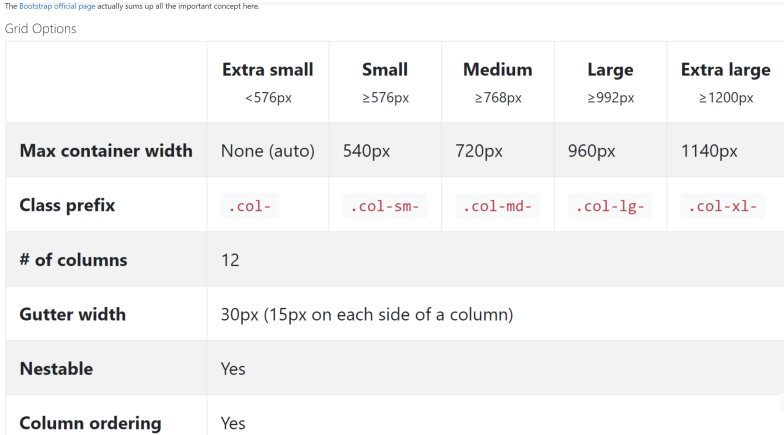
includes responsive frameworks: grid systems (or boxes) & flexibal images



built-in provided:

col-\*, col-sm-\*, col-md-\*, col-lg-\*, col-xl-\* and row.

//visual detailed EXP given in doc



[argument names]

- fixed-top: fix navbar to the top of the screen

- fixed-bottom: fix navbar to the bottom of the screen

- justify-content-between: distribute nav items evenly in navbar. a tag is on the left,

and whole ui is on the right

web type JS (on html for responsive purpose)

- [IMP] import JS into html

right before html file's </body>: <script wrc='localfile.js'> </script>

-> if JS's print not shown on website: inspect -> console

DOM - document object model

see html elements as object individually accordingly

Basic DOM [JS implemented in html]

- find/print x element/tag

let count = document.querySelector('span')

>>> <span>0</span>

document.getElementsByClassName('className');

>>> select class

- change variable within tag (e.g. 0 in <span>)

count.innerHTML(count)

>>> (<span> = 0 originally) 1

// this prints all value/variable between all <>

count.innerText(count)

// prints everything between count's outer assigned tag(all in span)

- change font color

count.style.color = #hex

- add/remove class list

count.classList.add('classname')

count.classList.remove('classname')

exp: //as count is <span>, so 'classname' is added/removed from <span>

original: <span> || after: <span class="classname">

==> if added JS sequence, the web will show final result only (e.g. for loop of 4 >>> 4; not 0-1- 2-3)

=> JS is needed for loop, set timing, if-else, assignment of update, etc

DOM events (JS implemented in html)

EXP (as lesson): arrow game [left, right, up, down]

- implemented link

<body>

<div class="current1">

<div class="x2">, add inner tags of <i class="fas3 y4"></i>

go to fontawesome.com, select [left, right, up, down], paste accordingly to y

in <head>, create <link rel="stylesheet" href="x link">

go to 'cdnjs font awesome', select website given in css and copied into "x link"

In <script>

let current = document.querrySelector('.current1')

current.innerHTML = "<i class='…up'></i>" //optional - this prints arrow

let lst = document.querySelectorAll('.x2 .fas3') //becomes a list

- when mouse clicked (multiple)

for (const l in lst) {

l.addEventListener('click', function (){

if (l.classList.contains('y4')) {do…} …}})

- when mouse clicked on specific element

let x = document.querySelector('.classname')

x.addEventListener('click', function (){do…})

- when keydown

document.body.addEventLisener('keydown', function(event) {

if event.keycode == num} {do…}};

- to check keydowncode

document.body.addEventListener('keydown', function(event1){

console.log(event1.keyCode) })

// this prints keycode every click

=concept: all keys are represented as number code (e.g. space = 13)