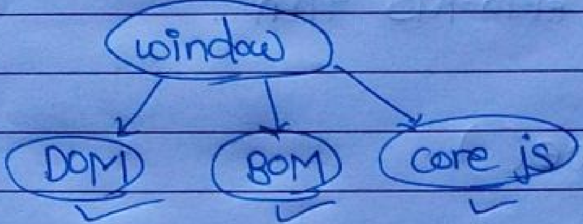
DOM : Document Object Model.

Window

↳ global object ↳ created by browser..
↳ browser ↳ top level entity.
↳ represents a browser window
↳ window contains everything under it.



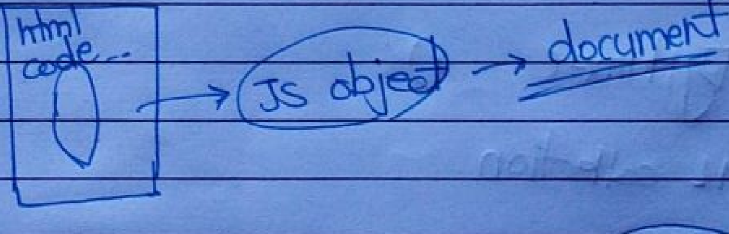
↳ you controls the browsers window.

BOM alert,
↳ allows you to interact with browser other than content of webpage

e.g. window. console.log("Hey");

DOM

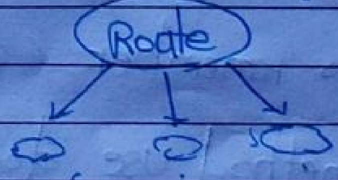
↳ our entire HTML file is converted to document



↳ #document ↳ gives you all html

it. document.body ⇓ returns body of html.

↳ tree like structure



characters→

```
<html>
    <head> -- </head>
    <body> -- </body>
</html>
```

character → tags → token → nodes → DOM

tokenizer

everything is converted to character→

get Element by ID

getElement by ID ('—');

↳ 1) it is called on document object

unique

2) it returns a single object.

null if no id matched -?

if you want multiple elements then

↓

getElementsby className()

↓ returns array of objects.

e.g. document. get ElementBy ClassName ('hidden');

↳ returns HTML Collection

getElementsby TagName('...');

↳ returns HTML collection.

Things to remember (class & Tag)

① both method use document object

② both returns multiple elements

③ The list returned is not an array (HTML Collection)

Trick

$0

↳ if you select (or) hover on something on a webpage, & then do '$0' on console then it will return all that object's body.

& also you can equate it to the variable name.

e.g. let para = $0;

Now, you can access it's properties like

para.className.
(or)
$0.className ..

∴ # Access elements..

↳ getElementByID(' ')
↳ getElementsByClassName(' ');
↳ getElementsByTagName(' ');

↳ $0

* more ways _

id = # ✓
class = . ✓
tagName ✓

} querySelector(' . ');
↓
returns single object
&
the first one -

document. querySelectorAll(' '); → returns all objects.

Update existing content

↳ .innerHTML → ① gets an elements / all it's descendent
→ ② set an element's HTML content

↳ .outerHTML

↳ .textContent

↳ .innerText

① .InnerHTML

e.g. let Hero = document.querySelector('#hero');

Hero.innerHTML → returns all HTML

Hero.innerHTML = ' '; → set all HTML to null.

② .outerHTML

↳ gives the outer tag of that HTML object..

③ .textContent

e.g. <a href="#" class="download"> Download </a>

let d = ... quer('.download');

d.innerHTML ⇒ 'Download'
d.text content ⇒ 'Download';

then whats the difference ??

if. <a> <b> Download </b> </a>

d.innerHTML
→ '<b> Download <bgt;

HTML rendered...

d.textContent
→ <b> Download </b>

not rendered...!

4. innerText

→ if something is hidden in it's class, then that part only is not included in this object..

Adding new element

· CreateElement();

↓ how to add?

content.appendChild(--);

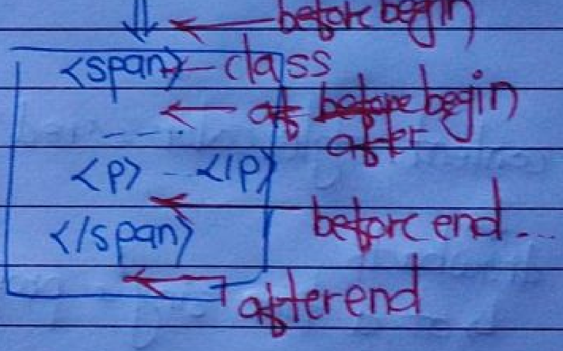
but, it is always added at the end

<span> --- </span>
↓
let para = document.querySelector('--

let newPara = createElement('p');

para.appendChild(newPara);

Now, if you want to add text to 'p' then, ←



newPara.textContent = 'I am para';

positioning the newly created element.

∴ insertAdjacentHTML() → has to be called with 2 args
↳ location -- where?
↳ HTML text -- what

- beforebegin ✓
- afterbegin ✓
- beforeend ✓
- afterend ✓

Remove

.remove Child();
    ↳opposite of append Child();
↳ must know parent
↳ must know child element to delete

i.e. parent. removeChild(childele);

styling using js

↳ .style
↳ . cssText
↳ . setAttribute
↳ . className
↳ . classList

specificity
↓
external
inline → high++
internal

1) content. style. color = 'red';
↓
drawback
↳ at a time single property is changed/applied -

2) content. style. cssText = 'color: red; background-color: geen'; --

3) content. setAttributes ("style", "color: orange; background: -;");

↓
you can set the id, class as well
→ content. setAttribute ("id", "HeroSection");

4) className

if there are multiple classes available in a tag, then it list downs all.

but, they will be string
e.g. 'one two three';

∴ you need to convert it to array, then,

content.className.split(' ');

↳ ['one', 'two', 'three'], etc

5) classlist

↳ returns a list of classes

Methods
- ↳ add
- ↳ remove

Browser events

run if you want to see the events.

↳ events

↳ → Invisible world

monitorEvents(documents);

↳ to see events occured on webpage -

if you want some particular actions on your browser whenever you click, double click or scroll ... loading of HTML. etc.

Agenda

- → events
- → respond to event
- → data stored in event
- → stop an event
- ↳ lifecycle of -11-

unmonitorEvents(); ← to not to see

event listener's

3 methods - →

- ↳ add Event listener() ✓
- → removeEventListener() ✓
- ↳ dispatch Event() ✓

Event Target ← Node ← element

↓

interface, blueprint

receives events ← if you click on button - → event

& may have listener's for them.

↓

defines what action is taken ..!! ← listnere

tag -

<p> --- </p>

↑

element

char → tags → token → node → DOM

element have all the properties of node -

node has all -11- event target

event target is top level entity has no parent

1) Add Event Listener

↳ listen to event

↳ respond to event

↳ hook into "

Pseudocode:

~~i~~ event target
- ↳ document.
- ↳ p
- ↳ Article
- ↳ video

} all elements ..

first need to add event target. .

window is global object
event_target is an interface

i.e.

```
<event_target>.addEventListener(<event-to-listen-for>,
                    <Funⁿ_to_run_when_event_happen>);
```

always a document, p, elements ←

① event target → ~~component~~ or → always a top level entity ..

② event type → (click, double click, scroll, ..)

③ function ⇒ what to do ?

e.g.
```
document.addEventListener('click', function() {
        console.log("I clicked on document");});
```

↳ it's amazing yaar .. ☺

e.g.
```
let content = document.querySelector('.heading');
  content.addEventListener('click', function () {
        content.style.color = 'red';
     });
```

✱ Imp!

↳ event listeners are mush powerful as they can store all your activities which you perform on web. such as mouse position, how much time you spent on a particular part etc. ☺

2) Remove event listener

↳ we need concept of strict equality

∴ it needs exact same function which you use while adding event listener.

```
document.addEventListener('click', function(){
                                     console.log('Hi');
                                   });

document.removeEventListener('click', function(){
                                        console.log('Hi');});
```

→ this won't remove as these 2 aren't same cause, everything is object in js.

↓ sol^n:

```
function print(){
  console.log('Hi');
}
document.addEventListener('click', print);
document.removeEventlistener('click', print));
```

yes ✓

*

event listeners in console window

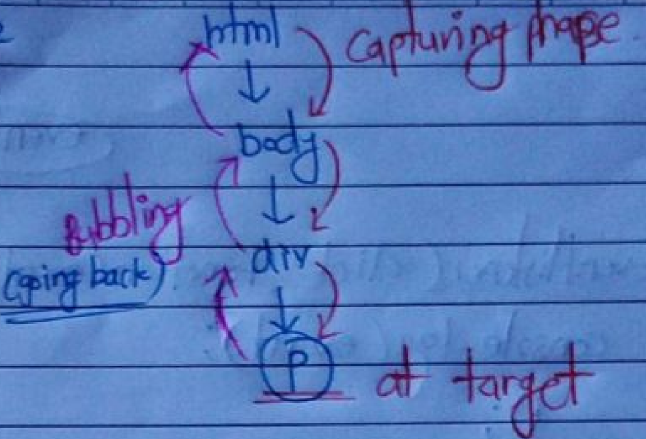
phases of Events

↳ capturing phase

↳ at target phase

↳ bubbling -||-

let say you wanna find 'p' with id then,



```
eg. <html>
     <body>
        <div ...>
           <p id='#...') ... </p>
        </div>
     </body>
    </html>
```

useCapture → determines at what phase you wanna a capture event..

(by default → bubbling).

↓ how to change..

```
              type  listener  useCapture
.addEvent('click', print, true)
                            ↓
```

at capturing phase enabled..!

The event object

when an event occurs, addEvent listener function gets an object..

↓

event object

↘ a lot of info. ☺

```
content.addEventlistener('click', function(event){
                console.log(event);
            });
```

The default action

a tag → link opens..

↓

.preventDefault() = prevents the action.

fatebook1   facebook2   facebook3

↓

need to prevent..

```
∴ let links = document.queySelectorAll('a');
  let thirdlink = links[2];
  thirdlink.addEventListener('click', function(event){
                  event.preventDefault();
                  console.log("maxa agja"); });
```

DOMContentloaded ..

```
<head>
 <script>
   document.querySelector(.wrapper). style.bg = 'red';
 </script>
</head>
<body>
  ,
</body>
```

↑ this is illogical, as when your entire html is not even loaded but then also you are trying to change bg color.

else, what you can do..?

```
 <script>

   document.querySelector('.footer').style.bg)=(
          ↓
   document.addEventListener('DOMContentLoaded', function(){

          });
```

you can find this on Network tab on console.