# DOM (Document Object Model)

* D = Document in HTML form
* O = Every Element / object in HTML
* M = hierarchy / objects in a structural way

What we can do with DOM??

1. Get Elements / objects
2. Element / object Styling
3. Creating Element / object
4. Modifying Element’s Text
5. Modify Attributes and Classes
6. Traversing of DOM
7. Event Listeners
8. Event Propagation
9. Event Delegation
10. **Get Elements / Objects**

* **querySelector()** Select Match tag, class, id
* **querySelectorAll()** Select matches tag, class, id from whole HTML but can’t style all for

it we use loop

// grab element and style it by using style

let outer = document.querySelectorAll(".list\_item");

for(let i=0; i<outer.length; i++){

    outer[i].style.color ="brown";

}

console.log(outer)

1. **Element / object Styling**
2. After .Style always use camelCase convention and direct equal( = )

*Document.querySelector(“”).style.fontSize = 2rem;*

1. CSS colon & semi colon are not applying here like

*Font-Size:2rem;*

1. **Create Elements**

let ul = document.querySelector("ul");

let li = document.createElement("h5");

ul.append(li)

console.log(ul.append(li));

1. **Modifying the Text & Tag**

* ul.innerHTML return tags + text inside tags
* ul.innerText return text only

**5.Modifying Attribute / Classes**

i. setAttribute (“Attribute”, “Attribute name”) required 2 parameter

ii. removeAttribute (“Attribute”, “Attribute name”) required 1 parameter

returns <ul place = “gujranwala”><ul/>

1. classList.add(“class1”, “class2”, “class3”, “class4”,…………..)
2. classList.remove(“class1”, “class2”, “class3”, “class4”,…………..)
3. classList.toggle(“class”) click to add class and click to remove class
4. classList.remove(“x”,”y”) if HTML has class x onclick it remove to given class
5. classList.contain(“x”) if HTML has class x then it return true value
6. ***Traversing of DOM***

*These are used to target and style Elements*

* .ParentElement child per laga kr parents ki details a jati hai
* .Children(index) parents per laga kr all children a jaty hn
* .FirstElement parent per lagaya jata hai us k first child ko target krny k liye
* .lastElementChild parent per lagaya jata hai us k last child ko target krny k liye
* .nextElementSibling child per lagta hai us se agla child element bta deta hai
* .previousElementSibling child per lagta hai us se pechla child element bta deta hai

Now,

These are for extra details like comments, enter, spaces in DOM & can’t be used for styling

* parentNode
* childeNode
* firstChild
* lastChild
* nextSibling
* previousSibling

***6.Event Listener***

1. user interface
2. focus and blur events
3. mouse events
4. keyboard events
5. form events
6. mutation events and observers
7. HTML 5 Events
8. CSS Events
9. ***User Interface***

* .load (when page load)
* .unload (when page close)
* .scroll (when user scroll)

1. ***Focus and blur events (skip)***
2. ***Mouse events***

* .click (when user click)
* .Dbclick (when user double click)
* .mousedown (when user click and drag curser down)
* .Mouseup (when user click and drag curser up)
* .mouseover (when user hover curser on element)
* .mouseout (when curser leaves a part)
* .mousemove (on mouse movements)

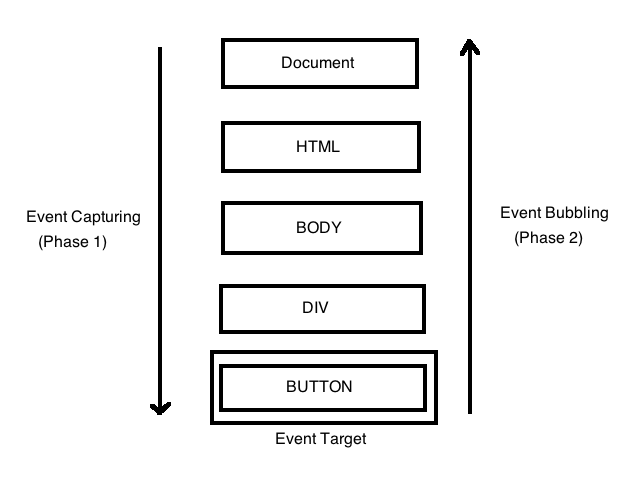
1. **Keyboard Events**

* Input
* Keydown
* Keyup
* Keypress

1. form events (Skiped)
2. mutation events and observers (Skiped)
3. HTML 5 Events (Skiped)
4. ***CSS Events***

* .transitionend
* Animationstart
* Animationitration
* .animationend

***7. Event propagation***



* Child event to parent event approach is called bubbling (default)
* Parents events to child events approach is called capture phase
* addEventListener(argument1, argument2, argument3)
* First argument is “click”
* Second argument is function
* Third argument is boolian
* True means capture approach
* False means bubbling approach
* Event hold all the details of clicked area

let outd = document.querySelector(".outer\_div");

let innerd = document.querySelector(".inner\_div");

outd.addEventListener("click",(event)=>{

    alert(`outer div`)

    event.stopPropagation()

},false)

innerd.addEventListener("click",(event)=>{

    alert(`inner div`)

    event.stopPropagation()

},false)

8. ***Event Delegation***

* Event in function possessed multiple powers
* Event delegation is that in infinite scrolling there are infinite children of a parent, in this case do you really think that you put event listener to ever child if you do than its not productive way
* So to avoid this we use event delegation and put event listener to parent/ grandparent id / tag

let grandd = document.querySelector(".grandd");

let ul = document.querySelector("ul");

grandd.addEventListener("click",(event)=>{

    alert(`grand div`)

    let target = event.target

    if(target.matches("li")){

        target.style.fontSize = "5rem"

    }

},false)

Example

let ul = document.querySelector("ul");

let btn = document.querySelector("#btn")

btn.addEventListener("click",()=>{

    let li = document.createElement("li");

    li.textContent = "hi! i am new here"

    ul.appendChild(li)

})

* It will give you
* <li>1<li/>
* <li>1<li/>
* <li>1<li/>
* <li>hi! I am new here<li/>