

adding or removing classes from a java script

const obj = document.querySelector("Class-home")
 obj.classList → give all the classes in key & value pair
 obj.classList.add() → adding any specific class
 obj.classList.remove() → removing any specific class.

document. → query selector
 getElementById
 getElementsByClassName

Main purpose is to give you elements / tags of your webpage.

let header = document.querySelector('div')

Now you manually update that particular tag in the
 header.style. font size ✓ → use em/rem
 font color ✓ → you can use any style on this tag.
 Background color ✓

"all classes in the tag"

header.classList.add("className")
 header.classList.remove("className")
 header.textContent = "New Heading"

Before, after, append, prepend ↓

new element = document.createElement('div')
 document.getElementById('NewText')

new element = document.createElement('div')

oldElement = document.getElementById('VaseText')

oldElement.appendChild(newElement)

before [" "]
after [" "]
append [" "]

dynamically

↓
getElementsByClass
When you print
the node list you will
see updated output

static

↓
querySelector

↓
You cannot see any
updated output

happens when you print the variable oldElement.

event listeners ?

listens everything that happens in a web page like clicking of a button, scrolling etc. You can program it based on your liking using Event Listeners.

Syntax → const button = document.querySelector('.bt-class')

button.forEach(bt => {
 bt.addEventListener('click', function() {
 console.log('Event listened')
 })
})

event object :

Whenever an event occurs itself the Browser API will also pass the object as a parameter by which you can access the button object

example `bt . add Event listener ("Click", event => {
 console.log(event.currentTarget.textContent)
 }
 }`

event loop call stack & callback queue.

↓
first the java script execute
itself completely then the event
loop will move the flow to execute listners
that were stored in the callback queue.