Java Script Day 10 Introduction to DOM and HTML basics

19 January 2024 14:42

Prefered way of linking Your Js Script to a Web document

Html → defer (best prefered) → start parsing of a some time & When Js executes completly parsing of HTML wont stop.

Normal linking → Parsing stop when Js execute completly

linking Your Src to end → Js execute at the end but it is slow

DOM

Document object Model

{ Collection of all the elements inside a web page in Key & value pair } → & adding it to the window object

getting element information through ID / Query Selector

document.getElementById("name of the id in Your HTML Document")

document.querySelector("name of class")
("# name of id")

For getting an element which is inside a particular div & have some class & within have an object you can get it by :

document.querySelector(div.class name h1) → any heading inside

since its returning the property You can update it by :-

Heading = document.querySelector(div.class_name h1)

Heading.styles.colour = 'Blue' → You can dynamically update the HTML document

.background colour = 'Blue'

Tip → You can just print the Heading.styles & see what other things You can update.

You can update the attribute of any html element :

```
const link = document.querySelector('a')
link.getAttribute('href')
```

→ if any attribute is present inside the anchor tag or any other tag you can access it with the help of .getAttribute.

(if you get output as a list, string or any complex result you can slice it by using .slice(1) → from which location you want answer.

document.getElementByClassName("class_name") → this will return collection of all elements having class name as "class_name".

Or you can use document.querySelectorAll(".classname") → return all the element having same class name but in the form of node list

Suppose you want to access those collections/node list :

- loops :

```
for (let element in (document.getElementByClassName("classname"))){
    element.style.colour = "#fff"
    element.style.font-weight = "bold"}
```

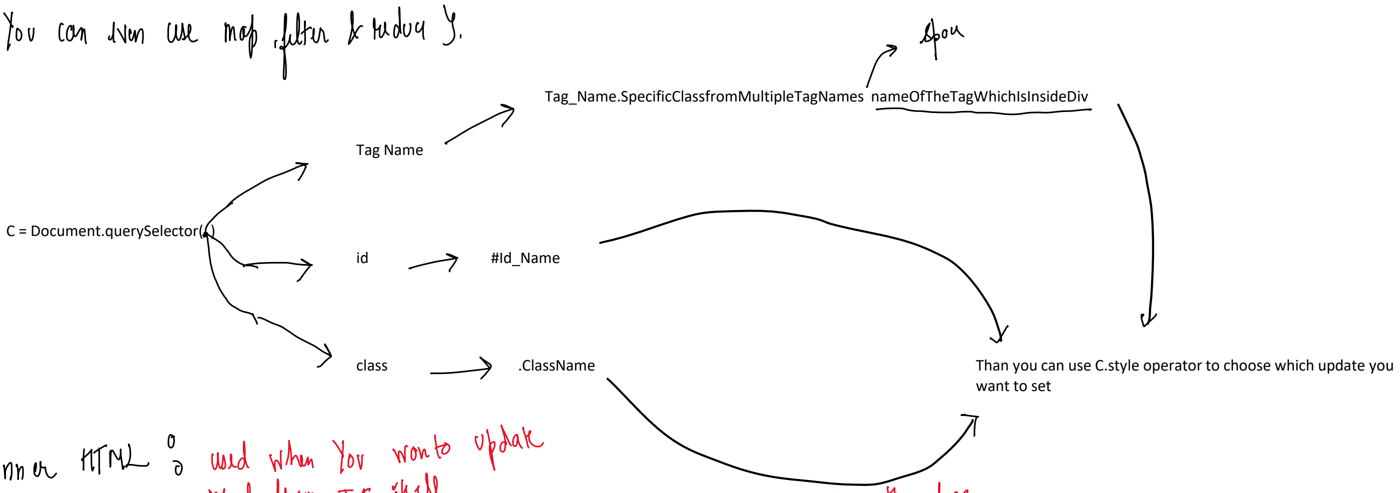
Or you can convert it into array & later you can use for each, map, filter, reduce

```
let array = Array.from(document.getElementByClassName("class_name"))
array.forEach((navItem) => {
    navItem.style.backgroundColour = 'Red'
    navItem.style.colour = 'green'}
```

& you can even use map, filter & reduce }.

↗ spoon

{You can even use map filter & reduce }.



Inner HTML : used when you want to update html from J/s itself

Inner Html

const element = Document.querySelector(.headline)
Element.innerHtml

every thing inside the class

Since Js returns key object pair you can update HTML from element . innerHTML

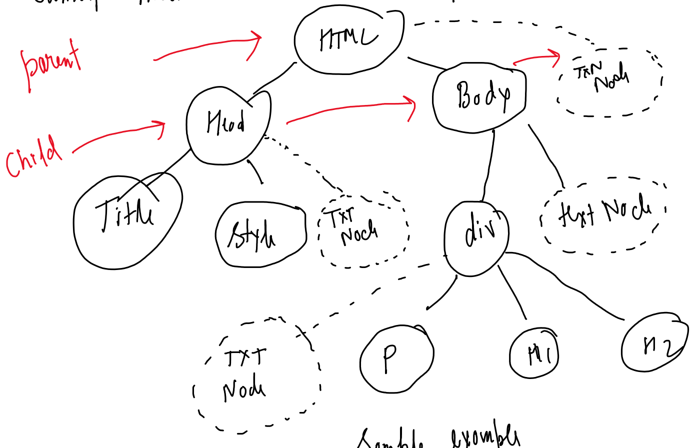
```
element.innerHTML = '<h1> Hello World <h2>'
element.innerHTML = element.innerHTML + '<button class="btn btn-headline">
Button </button>'

This will update the entire html page since element is treated as an object
is Js
```

Dom Tree

Dom Tree

Children , child node

Dom tree is the ordered hierchy of text node & element node inside the document HTML page.



Sample example

You can access these dom tree with following method :

root node = document. getrootnode ( )

root node → (I) parent Node, (II) Children, (III) Child Node, (IV) siblings.

(for getting respective member of the dom tree.

if you want only elements not text node.

(for getting respective values of the dom tree.

elements not first node.

You can later update the style. ✓

Update the inner html ✓

by using .style = or childnode.innerHtml = <DIV> // </DIV>

Example

root_node = document.getRootNode() → entire HTML code

root_node.childNodes → all child inside.

root_node.childNodes[0] → print the HTML code of first element

root_node.childNode[0].childNodes → print all the child node for the above HTML tag (root)

root_node.childNode[0].childNodes[2] → print the second child in this case HTML code for body tag.

root_node.childNode[0].childNode[2].childNodes → print all the children inside third child node of above step

- childNode
- [0] → first element
- childNodes → child nodes of first element
- [2] → third element
- childNodes → child node of the third element.