<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <title>Dom learning </title>

    <style>

        .bg-black{

            background-color: #212121;

            color : rgb(255, 128, 0);

        }

    </style>

</head>

<body>

    <div class = "bg-black">

        <h1 id = "title" class="hello"  > DOM (document object model)learning on chai aur code

            <span style = "display: none;">test text</span>

        </h1>

        <h2> hello world </h2>

        <h2> hey computer  </h2>

        <h2> java script is life  </h2>

        <p>

            <P> dom :- document onject model means jo ye apka document bana hua hai

                wo kis model ka bana hua hai , kya uska structure hai , page kaisa dekta

            </P>

            <p>

                google -> inspect ->console -> console.log (window) or ->

                console .dir(document);

                html collection seems like an array but they are not a array

            </p>

            <ul>

                <li class = "list" >one</li>

                <li class = "list" >two</li>

                <li class = "list" >three</li>

                <li class = "list" >four</li>

            </ul>

            <input type = "password" name="" id="">

            <input type = "text" name="" id="">

            <input type = "submit" name="" id="">

    </div>

</body>

</html>

<!--

    INSPECT CONSOLE OPERATIONS .....

i/p = document.getElementById('title')

o/p = <h1 id=​"title" class>​ DOM (document object model)learning on chai aur code​</h1>​

i/p = document.getElementById('title').id

o/p = 'title'

i/p = document.getElementById('title').class

o/p = undefined

here we don't write class directly instead of that we write class name

i/p = document.getElementById('title').getAttribute

o/p = ƒ getAttribute() { [native code] }

i/p = document.getElementById('title').getAttribute('class')

o/p = 'heading'

i/p = document.getElementById('title').getAttribute('id')

o/p = 'title'

note = document.getElementById('title').setAttribute('class' , 'test') , yaha hello se test ho gaya

i/p = document.getElementById('title').setAttribute('class' , 'test')

o/p = undefined (output comes this but actually the value get change)

#storing in variables

i/p = const title = document.getElementById('title')

o/p = undefined

i/p = title.style.backgroundColor = 'green '

o/p = 'green '

i/p = title.style.padding = "15px"

i/p = title.innerText:- jo dek raha h ushi k hisaab se change karoonga

o/p = 'DOM (document object model)learning on chai aur code'

i/p = title.textContent:- sara text content par

0/p = ' DOM (document object model)learning on chai aur code\n test text\n'

Q)what is difference between inner text and content ?

in h1 we add span test text , then we stlye in sapan to display none

test text shown in text content where as it will not shown in innerText

means inner text shows only visible text where as textContent show all text

i/p = title.innerHTML  // gives whole value of html

o/p = ' DOM (document object model)learning on chai aur code\n<span style="display: none;">test text</span>\n

i/p = document.querySelector('h2')

o/p = <h2>​ hello world ​</h2>​

i/p = document.querySelector('#title')

i/p = document.querySelector('.hello')

o/p = <h1 id=​"title" class=​"hello">​…​</h1>​

i/p = document.querySelector('input[type="Password"]')

o/p = <input type=​"password" name id>​

const myul = document.querySelector('ul')

undefined

myul.querySelector('li')

<li>​…​</li>​

const turnGreen = myul.querySelector('li')

undefined

turnGreen.style.backgroundColor = "green"

'green'

turnGreen.innerText = "five"

'five'

document.querySelectorAll('li')

NodeList(3) [li, li, li]

0

:

li

1

:

li

2

:

li

length

:

3

[[Prototype]]

:

NodeList

const tempLiList = document.querySelectorAll('li')

undefined

tempLiList

NodeList(3) [li, li, li]

tempLiList.style.color='green'(iska color green ni kar skte bcoz ye ek node list h)

tempLiList[0].style.color='green'

'green'

node element me batana padta hai ki kon sa element with index\

const myH1 = document.querySelectorAll('h1')

undefined

myH1.style.color ='greeen'

VM311:1 Uncaught TypeError: Cannot set properties of undefined (setting 'color')

    at <anonymous>:1:18

(anonymous) @ VM311:1

tempLiList

NodeList(3) [li, li, li]

tempLiList.forEach(function(l){

    l.style.backgroundColor = "yellow"

})

undefined

node list ko array me be convert kr skte hai

but jada tar cases me node list me for each loop use kerte hai bcoz wo array ni h waha map ni laga skte hai

document.getElementsByClassName("list")

HTMLCollection []

length

:

0

[[Prototype]]

:

HTMLCollection

item

:

ƒ item()

length

:

(...)

namedItem

:

ƒ namedItem()

constructor

:

ƒ HTMLCollection()

Symbol(Symbol.iterator)

:

ƒ values()

Symbol(Symbol.toStringTag)

:

"HTMLCollection"

get length

:

ƒ length()

[[Prototype]]

:

Object

yaha proto typing me koi looping ka option he ni dek raha h

to yha agr for each use kiya to kaam he ni karega

const tempClassList = document.getElementsByClassName("list")

undefined

tempClassList

HTMLCollection []

tempClassList.forEach(function(li){

    console.log(li)

})

VM890:1 Uncaught TypeError: tempClassList.forEach is not a function

    at <anonymous>:1:15

(anonymous) @ VM890:1

yaha temp list me loop karne k liye usee phele convert kerna padega

or ham yaha isko array me convert karenge

-->

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>DOM</title>

</head>

<body style="background-color: #212121; color: #fff;">

    <!--step 1st-->

    <div class="parent">

        <div class="day">monday</div>

        <div class="day">tuesday</div>

        <div class="day">wednesday</div>

        <div class="day">thursday</div>

    </div>

</body>

<script>

    const parent = document.querySelector('.parent');//class ko dot se access kerte h

    console.log(parent);

    console.log(parent.children); // here we get html collection which is array like property

    console.log(parent.children[1]);

    console.log(parent.children[1].innerHTML);

    // CLASSIC FOR LOOP TO CHAL JAYEGA AGER ARRAY JAISA YA HTML JAISE PER means we don't need to convert it into array

    // we can't use on parent bcoz parent child gives the HTML collection, ye direct array par na chal k parent par chalega

    console.log("-----------------------------------------------");

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

        console.log(parent.children[i].innerHTML);

    }

// parent.children se he access ker skte hai element koo

    parent.children[1].style.color = "orange";

    console.log(parent.firstElementChild); // monday

    console.log(parent.lastElementChild); // friday

console.log("------------------------------------")

    const dayOne = document.querySelector('.day');

    console.log(dayOne);

    console.log(dayOne.parentElement);

    console.log(dayOne.nextElementSibling);

    console.log("NODES: ", parent.childrenNodes);

</script>

</html>jj

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Dom</title>

</head>

<body style="background-color: #212121; color: #fff;">

</body>

<script>

    document.createElement('div');

    connsole.log(div);

    div.className = "main";// ye koi div hai

    div.id = Math.random( Math.random()\*10 + 1)// by default function , +1 done for no zero

    // baki or custom attribute set kerna hai then

    div.setAttribute('title' , 'generated title');

    div.style.backgroundColor = "green";

    div.style.padding = "12px";

// above we gives all the input but it will not display on page

    //bcoz the created div is in the memory

   // div.innerText = "chai aur code";//it over write , yeah us merory ka refrence le ker ayega then overwrite karega

    //here writing in text node

    const addText = document.createTextNode("chai aur code")

   div.appendChild(addText);

   /\*

   why we use div.appendChild(addText);<- this method if by using div.innerText

   we can easily add the text? which is optize in between them ?

   \*/

   // now attaching on document , ans document le then uski body le then append

   document.body.appendChild(div);

</script>

</html>

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Dom</title>

</head>

<body style="background-color: #212121; color: #fff;">

</body>

<script>

    // Corrected the typo in "console.log"

    var div = document.createElement('div');

    console.log(div);

    div.className = "main"; // This is a div element

    div.id = Math.random() \* 10 + 1; // Using Math.random() function, +1 is added to avoid zero

    // Now, other custom attributes can be set

    div.setAttribute('title', 'generated title');

    div.style.backgroundColor = "green";

    div.style.padding = "12px";

    // Adding text content to the div using text node

    const addText = document.createTextNode("chai aur code");

    div.appendChild(addText);

   /\*

   The reason we use div.appendChild(addText) instead of div.innerText is because

   appendChild() method is more versatile. It allows us to append various types of nodes

   like elements, text nodes, etc. In this case, we're appending a text node. If we were

   to use innerText, it would overwrite any existing content within the div, whereas

   appendChild() appends the text node after any existing content.

   \*/

   // Attaching the div to the document body

   document.body.appendChild(div);

</script>

</html>

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Document</title>

</head>

<body>

    <ul class = "language">

        <li> Javascript </li>

    </ul>

</body>

<script>

    function addLanguage(langName){

        const li = document.createElement('li');

        li.innerHTML = `${langName}`;

        document.querySelector('.language').appendChild(li)

//append kerne ki wajha se document ka pura tree traverse kerna padta hai , optimization issue ajata hai like koi bada project h to

    }

    addLanguage("typescript");

    addLanguage("python");

    addLanguage("java");

    function addOptiLangage(langName){

        const li = document.createElement('li');

        li.appendChild(document.createTextNode(langName))

        document.querySelector('.language').appendChild(li)

    }

    addOptiLangage('go lang');

    /\*

    above approch is optimize bcoz yaha per pura tree traverse ni

    kerna pad raha hai but abe function innerHtml use hota hai to

    pure ki pure traversing hote hai

    \*/

   //Edit value

   const secondLang =document.querySelector("li : nth-child(2)")

   secondLang.innerHTML = "Mojo"

   // 2nd approch

   const newli = document.createElement('li');//li selected

   newli.textContent = "Mojo"

   secondLang.replaceWith(newli)

</script>

</html>