# DOM- Document Object Model

DOM is a standard object model that allows programs and scripts to dynamically access and update the content, structure, and style of a document

Document Object Model (DOM) connects web pages to scripts languages by representing the structure of a document

The DOM represents a document with a logical tree. Each branch of the tree ends in a node, and each node contains objects. DOM methods allow programmatic access to the tree. With them, you can change the document's structure, style, or content.

## Main Object:
Here's a breakdown of some key concepts related to the JavaScript DOM:

• Document: The top-level object in the DOM hierarchy, representing the entire HTML document. It serves as an entry point to access and manipulate the document's content.

```
console.log(document);
```

Logging document to the console in JavaScript will display the entire Document Object Model (DOM) of the current HTML page.

## Methods for Accessing Elements:(Get methods using dom)

**1)document.getElementById():** Retrieves an element by its unique ID

**Example:**
```
<div id="demo">Content1</div>
var a=document.getElementById("demo");
console.log(a);
```

**2)document.getElementsByClassName():** Retrieves elements by their class name.

**Example:**
```
 <p class="myClass">Paragraph 1</p>
 <p class="myClass">Paragraph 2</p>
 var a=document.getElementsByClassName("myClass");
 console.log(a);
```

```
 //here the point to note is classnames are always in collections
 You can get the element by their index numbers
 Var a= document.getElementsByClassName("myClass")[0]
```

3) **document.getElementsByTagName():** Retrieves elements by their tag name.

```
 <h1>Heading</h1>
 <p>Paragraph 1</p>
 <p>Paragraph 2</p>
 var a= document.getElementsByTagName("p");
 console.log(a);
```

```
//here the point to note is tagnames are always in collections
You can get the element by their index numbers
Var elementsByTagName= document.getElementByTagName("p")[0]
```

4)**Accessing Elements by CSS Selector:**
**Example:**
```
 <div class="container">
   <p class="para">Paragraph 1</p>
   <p class="para">Paragraph 2</p>
</div>
```

**querySelector()** method allows you to select the first element in the document
**Example:**
```
var elementBySelector = document.querySelector(".para");//selects by classname
var myDiv = document.querySelector("#myDiv");//select by id
var elselector = document.querySelector("div");//select by element name
```

**querySelectorAll** -iIt operates similarly to querySelector(), but instead of returning only the first matching element, it returns a list of all matching elements.
**Example:**
```
var paragraphs = document.querySelectorAll(".para");//select all elements by class names
var divs = document.querySelectorAll("div");//select all div elements in a collections
```

# Get content of the html
innerText and innerHTML are properties of DOM elements in JavaScript that deal with the content of HTML elements
## innerText:
• innerText is a property that represents the visible text content of an element.
• It retrieves the text content of the element, excluding any HTML tags.

## innerHTML:
• innerHTML is a property that represents the HTML content of an element.
• It retrieves or sets the HTML markup within the element, including any nested elements and tags.
• It can be used to dynamically change the structure and content of an element.
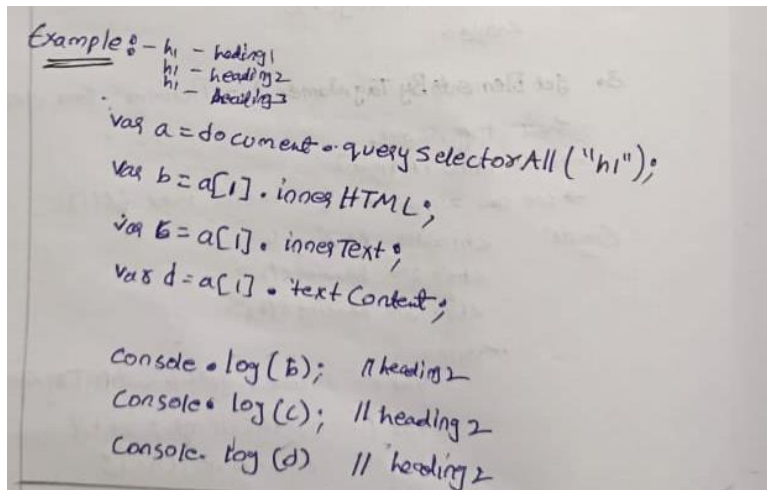


# How to modify existing content

```
   // Select the element by its ID
   var paragraph = document.getElementById("myParagraph");

   // Update the text content using innerText
   paragraph.innerText = "Updated text!";
```

# Modifing the Content :-

document.getElementById ("id"). innerHTML= "modified content".

**1. inner Text :-**

document. getElementById ("id"). innerText = "Modified content";

Ex:-

```
<h1> heading 1 </h1>
<h1> heading 2 </h1>
<h1> heading 3 </h1>

<style>
      var a = document.querySelectorAll(" h1");
      a[0].innerText = "hello world".
      console log (a);
</style>
```

output:- :hello world
         heading 2
         heading 3

**2. inner HTML :-**

Example:-

```
var a= document.querySelectorAll ("h1");
a[0]. innerText="<u>hello world"</u>";
console log (a);      // <u> Hello world </u>


a[0]. innerHTML = "<u> Hello world </u>";
a[1]. innerHTML = "<del> Hello world </u>";

console.log(a);
```

Output:-
    Hello world ;

    ~~Hello world;~~

    heading 2

**3. text content:-**
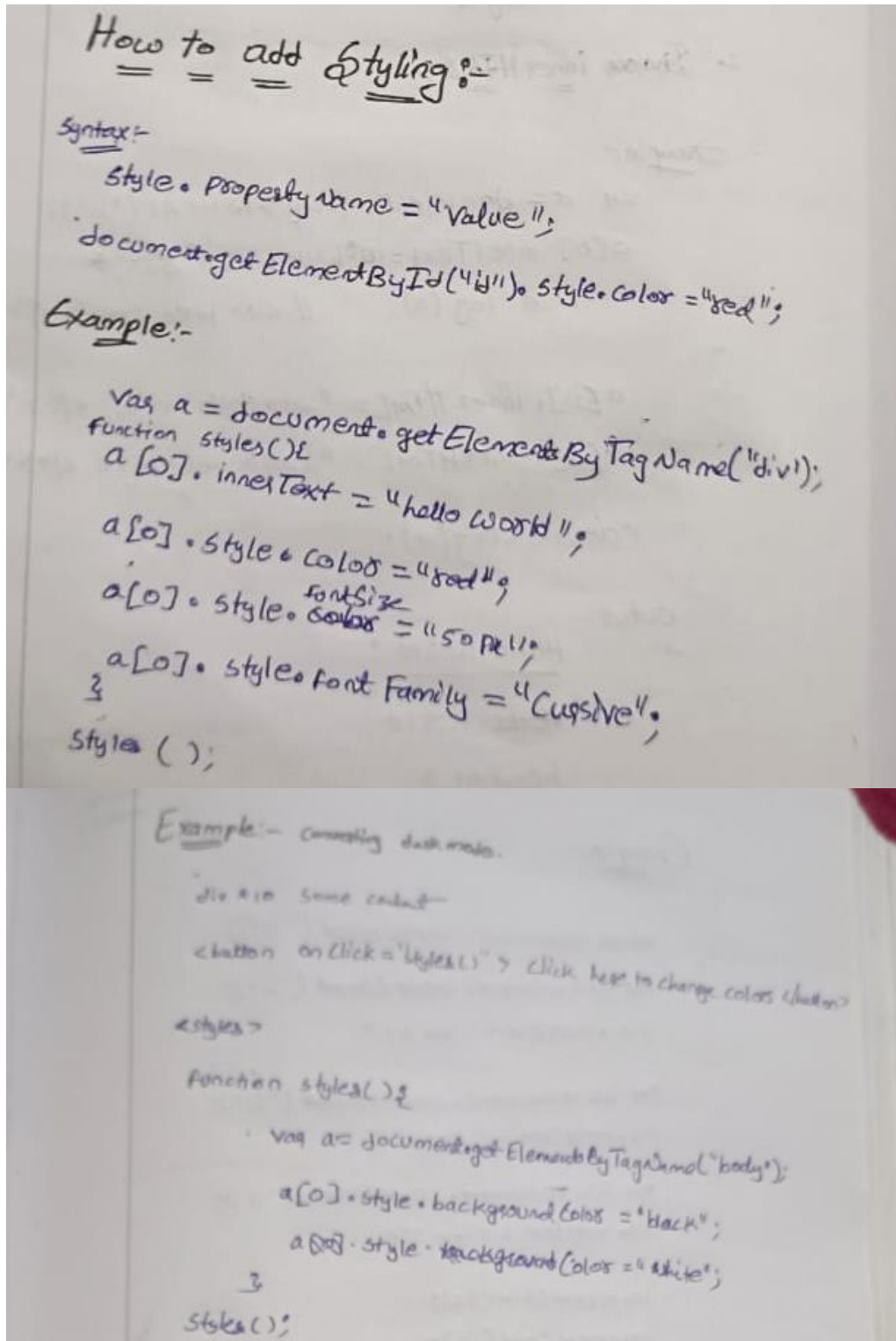    a[0].textContent = "<u> Hello world </u>";
    // <u> Hello world </u>

## How to apply styles using dom
Step 1: Access the element where you want to append the text node
Step 2: Apply styles



## How to change attribute values by using setAttribute

```
<div id="myDiv">This is some text and have id myDiv but it will changed to demo</div>
 var a=document.getElementById("myDiv").setAttribute("id","demo");
console.log(document)//can inspect and check weather it was changed or not
```
we can change the attribute by using .setAttribute.("attribute name","attribute value") //output

## Creating element

- A new paragraph element is created using document.createElement("p").
- The innerText property of the newly created paragraph element is set to "This is a dynamically created paragraph."
- The paragraph element is appended to the document body using **document.body.appendChild(newParagraph).**

## Appendchild and Append

Append and appendChild methods are used in JavaScript to add nodes to the DOM, but they have some differences in terms of usage, accepted parameters, and behavior:

## appendChild syntax:

parentNode.appendChild(newChild);

## Parameters:

newChild: A single node (an element, text node, or any other node) that will be appended as the last child of parentNode

## Behavior:

If the newChild is already in the DOM, it will be removed from its current position and moved to the new position.
**Only accepts a single node**.

## Append syntax:

parentNode.append(node1, node2, node3);

## Parameters:

nodes: One or more nodes or strings that will be appended as the last children of parentNode.

## Behavior:

Can append multiple nodes and/or strings at once.
If a string is provided, it will be added as a text node.
Allows appending a combination of nodes and text.

# How to create textNode

```
 // Step 1: Access the element where you want to append the text node
 var myDiv = document.getElementById("myDiv");

// Step 2: Create a text node
var textNode = document.createTextNode("This is a dynamically created text node.");

// Step 3: Append the text node to the element
myDiv.appendChild(textNode);
```

- Access the element where you want to append the text node.
- Create a text node using document.createTextNode().
- Append the text node to the desired element.

## Example:

```
   // Create a new paragraph element
   var newParagraph= document.createElement("p");

   // Set innertext or other properties if needed
   newParagraph.innerText = "This is a dynamically created paragraph.";

   // Append the paragraph to the document body
   document.body.appendChild(newParagraph);
```

**Example :-**

```
var ul = document.CreateElement("ul");
var li1 = document.createElement("li");
li1.innerText = "item 1;"

var li2 = document.createElement("li");
li2.innerText = "Item-2";

var li3 = document.createElement("li");
li3.innerText = "item-3";

ul.appendChild(li1);
ul.appendChild(li2);
ul.appendChild(li3);

document.body.appendChild();

console.log(ul);
```

<mark>Tasks:</mark>
1.  <mark>Color Genrator:</mark> By clicking button change multiple colors (use count and arrays)

**Code:**
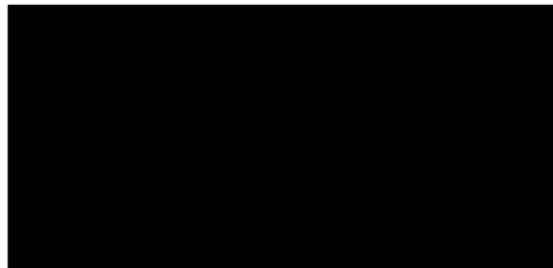
```
<script>
    var body=document.getElementsByTagName("body")[0];
    body.style.display="flex";
    body.style.flexDirection="column";
    body.style.width="98vw";
    body.style.height="98vh";
    body.style.justifyContent="center";
    body.style.alignItems="center";
    body.style.gap="10px";

    var head1=document.createElement("h1");
    head1.innerText="Color Genrator";
    head1.style.fontSize="28px";
    document.body.appendChild(head1);

    var area=document.createElement("div");
    area.style.width="50%";
    area.style.height="50vh";
    area.style.border="3px solid black";
    area.style.color="white";
    document.body.appendChild(area);

    var btn=document.createElement("button");
    btn.innerText="Click Here to Change Color";
    btn.style.fontSize="15px";
    btn.addEventListener("click", cg);
    document.body.appendChild(btn);

    var count=0;
    var arr=['black','skyblue','lightgreen','grey','blue'];
    function cg(){
        area.style.backgroundColor=arr[count];
        count++;
        if(count>=arr.length){
            count=0;
        }
    }
</script>
```
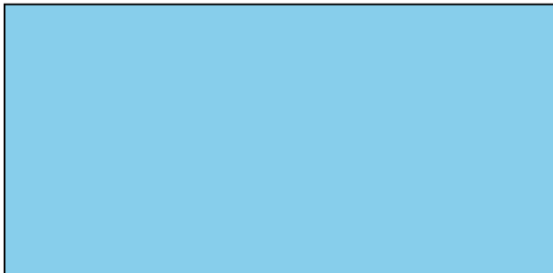
2. **Resume:** create a Resume without using HTML

```
var body=document.getElementsByTagName("body")[0];
body.style.width="48vw";
body.style.height="auto";
body.style.border="3px solid black";

// Name
var head1=document.createElement("h1");
head1.innerText="ABHINAV SAI ";
head1.style.textAlign="center";
// head1.style.paddingLeft="25px";
head1.style.color="green";
head1.style.fontSize="40px";
document.body.appendChild(head1);

// CAREER OBJECTIVE
var sh1=document.createElement("h3");
sh1.innerText="CAREER OBJECTIVE";
sh1.style.paddingLeft="25px";
sh1.style.color="red";
sh1.style.fontSize="26px";
document.body.appendChild(sh1);

// CAREER OBJECTIVE statement
var co=document.createElement("p");
```

```javascript
co.innerText="Responsible and motivated student ready to apply education in the workplace. Offers excellent technical abilities with software
and applications, ability to handle challenging work, and excellent time management skills.";
co.style.paddingLeft="25px";
document.body.appendChild(co);

// Technical Skills
var sh2=document.createElement("h3");
sh2.innerText="Technical Skills";
sh2.style.paddingLeft="25px";
sh2.style.color="red";
sh2.style.textAlign="left";
sh2.style.fontSize="26px";
document.body.appendChild(sh2);
var ts=document.createElement("ul");
var li1=document.createElement("li");
var li2=document.createElement("li");
var li3=document.createElement("li");
var li4=document.createElement("li");
var li5=document.createElement("li");
ts.style.listStyleType="none";
li1.innerText="Java";
li2.innerText="Python";
li3.innerText="HTML";
li4.innerText="CSS";
li5.innerText="Java Script";
ts.appendChild(li1);
ts.appendChild(li2);
ts.appendChild(li3);
ts.appendChild(li4);
ts.appendChild(li5);
document.body.appendChild(ts);


// Soft Skills
var sh3=document.createElement("h3");
sh3.innerText="Soft Skills";
sh3.style.paddingLeft="25px";
sh3.style.color="red";
sh3.style.textAlign="left";
sh3.style.fontSize="26px";
document.body.appendChild(sh3);
var ts=document.createElement("ul");
var li1=document.createElement("li");
var li2=document.createElement("li");
var li3=document.createElement("li");
var li4=document.createElement("li");
ts.style.listStyleType="none";
li1.innerText="Adaptapility";
li2.innerText="Self Motivation";
li3.innerText="Self Confidence";
li4.innerText="Team Work";
ts.appendChild(li1);
ts.appendChild(li2);
ts.appendChild(li3);
ts.appendChild(li4);
document.body.appendChild(ts);

//Hobbies.
var sh4=document.createElement("h3");
sh4.innerText="Soft Skills";
sh4.style.paddingLeft="25px";
sh4.style.color="red";
sh4.style.textAlign="left";
sh4.style.fontSize="26px";
document.body.appendChild(sh4);

var hob=document.createElement("ul");
hob.style.listStyleType="none";
hobbies=["playing Cricket","Listening Music","watching Movies"];
for(var i=0;i<hobbies.length;i++){
    var li=document.createElement("li");
    li.innerText=hobbies[i];
    li.style.color="green";
    li.style.fontSize="16px";
    hob.appendChild(li);
}
```

```
document.body.appendChild(hob);

// Declaration
var sh5=document.createElement("h3");
sh5.innerText="Declaration";
sh5.style.paddingLeft="25px";
sh5.style.color="red";
sh5.style.textAlign="left";
sh5.style.fontSize="26px";
document.body.appendChild(sh5);

// Declaration Statement
var ds=document.createElement("p");
ds.innerText="I hereby declare that the information furnished above is genuine to the best of my belief and I hold the responsibility of their
authenticity and correctness.";
ds.style.paddingLeft="25px";
document.body.appendChild(ds);

// sign
var sign=document.createElement("p");
sign.innerText="-Abhinav Sai";
sign.style.paddingLeft="600px";
document.body.appendChild(sign);
```

# ABHINAV SAI

## CAREER OBJECTIVE

Responsible and motivated student ready to apply education in the workplace. Offers excellent technical abilities with software and applications, ability to handle challenging work, and excellent time management skills.

## Technical Skills

Java
Python
HTML
CSS
Java Script

## Soft Skills

Adaptapility
Self Motivation
Self Confidence
Team Work

## Soft Skills

playing Cricket
Listening Music
watching Movies

## Declaration

I hereby declare that the information furnished above is genuine to the best of my belief and I hold the responsibility of their authenticity and correctness.

-Abhinav Sai