1) WINDOWS VS DOCUMENT IN JAVASCRIPT

a) Window is the main container or we can say global object and any operation related to the entire browser window can be a part of the window object.	a) DOM is the child of the window object.
b) All the members like objects, methods or properties. If they are the part of the window object then we do not refer to the window object.	b) In DOM we refer to the document, if we want to use the document object, method or properties.
c) Windows has methods, properties and objects. Ex: setTimeout() or setInterval() are the methods, whereas Document is the object of the window and it also has a screen object with properties describing the physical display.	c) Document is just the object of the global object that is window, which deals with the document, the HTML elements themselves.

2) USE DIAGRAMS TO EXPLAIN WINDOW GLOBAL OBJECTS

WINDOW		
DOM	**BOM**	**JAVASCRIPT**
DOCUMENT	NAVIGATOR	OBJECT
<HTML>.....</HTML>	SCREEN	ARRAY
	LOCATION	FUNCTION
	FRAME	
	HISTORY	
	XML HTTP REQUEST	

DOM: Document Object Model, which deals with the document, the HTML elements themselves
eg: document and all traversal you would do in IT events etc
eg:change the background colour to Red. Document.body.style.background=”red”;

BOM: is the **Browser Object Model**, which deals with browser components aside from the document like history, location, navigator and screen (as well as some others that may vary by browsers). OR
In simple meaning all all the window operation which comes under BOM are performed using BOM

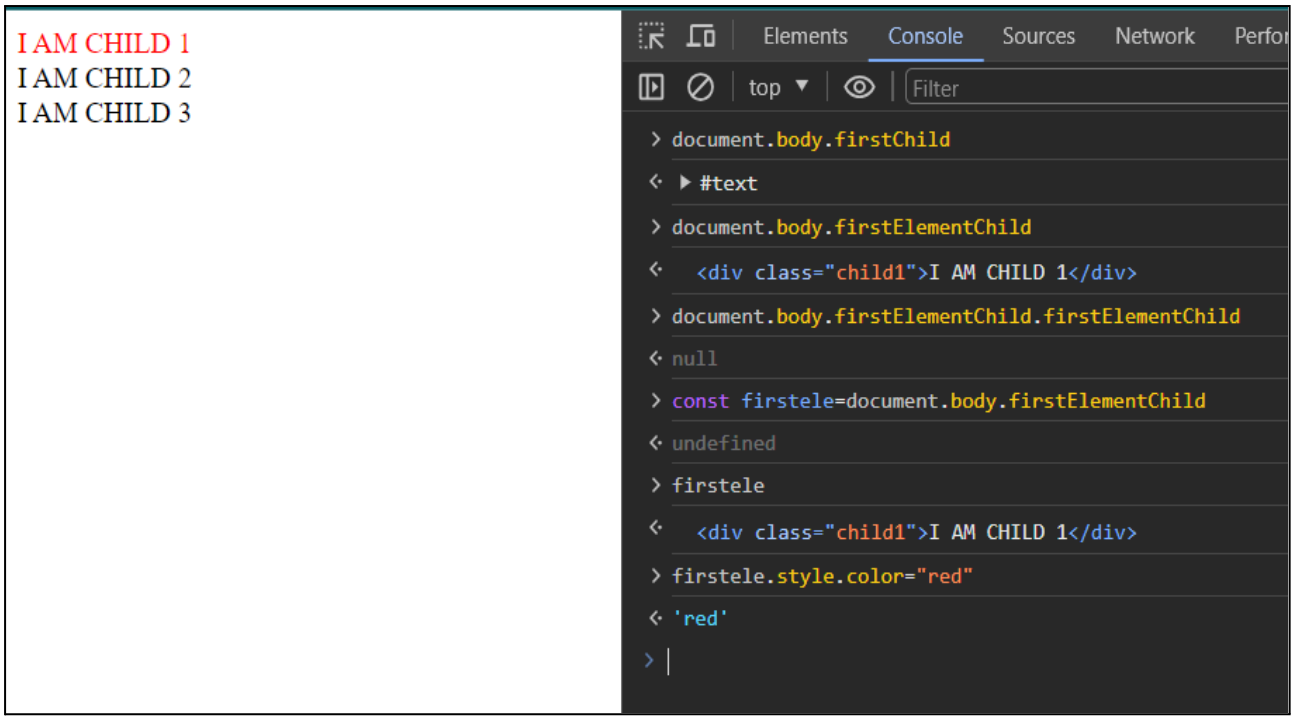
CODE

```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>
    <div class="child1">I AM CHILD 1</div>
    <div class="child2">I AM CHILD 2</div>
    <div class="child3">I AM CHILD 3</div>
    <script>



    </script>
</body>
</html>
```

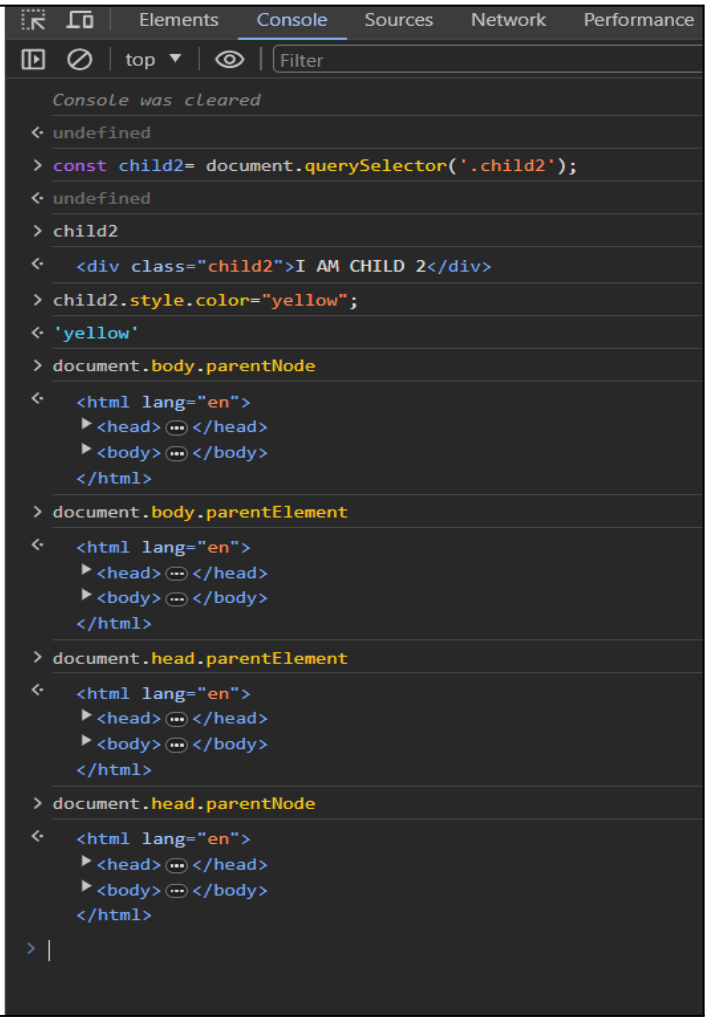
- **document.documentElement:** returns the root element
- **document.head:** returns the head section
- **document.body:** returns body section
- **document.body.childNodes:** [div.child1, text, div.child2, text, div.child3, text, script, text]
- **document.body.children:** [div.child1, div.child2, div.child3, script]
- **document.body.children.length:** 4

ELEMENT CHILD NODES

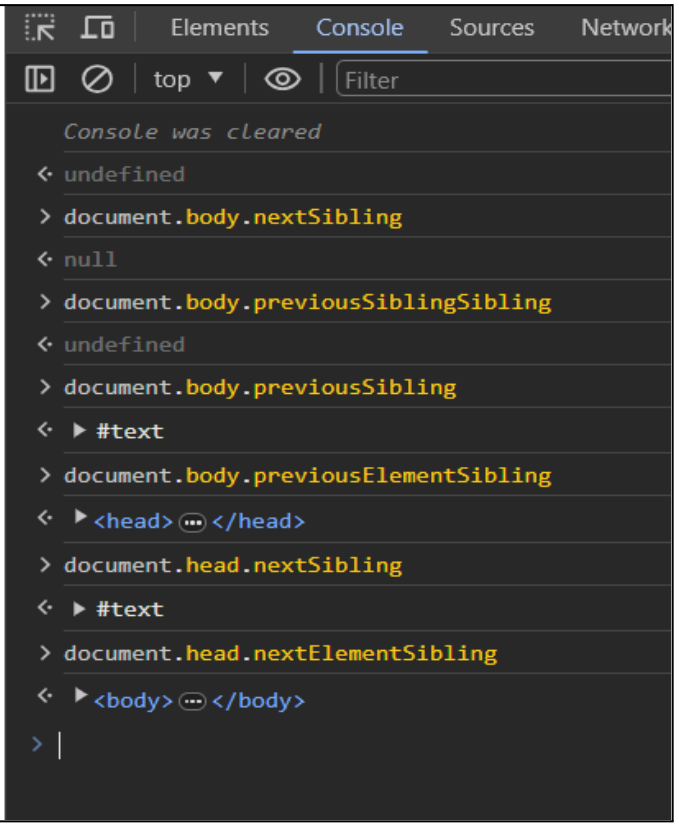
- **document.body.hasChildNodes():** true



I AM CHILD 1
I AM CHILD 2
I AM CHILD 3



I AM CHILD 1
I AM CHILD 2
I AM CHILD 3



//EXAMPLE OF DOM

```
<h1 id="heading">hiiiiiiiiiiiii</h1>
<button onclick="changecontent()">CLICK ME</button>
<script>
   const changecontent= () =>{
      document.getElementById('heading').innerHTML="welcome";
   }

   const headingchange=document.getElementById('heading');
   headingchange.innerHTML="welcome";

   console.log(document.getElementByClassName('para'));
   document.getElementByTagName('p');

   console.log(document.getElementByName('gender'));   //used in radio gender list
</script>
```

//QUERY SELECTOR

```
document.querySelector('.para').innerHTML="i changed again";
console.log(document.querySelectorAll(#heading));
```

- //querySelector returns the first matching value
- //querySelectorAll returns all the elements

//DIFFERENCE BETWEEN getElementById AND querySelector

- **getElementById**

  SYNTAX:= document.getElementById(id);

Returns a reference to the element by its ID. If the element with the specified ID is not in the document it will return NULL

- **querySelector**

  SYNTAX:= document.querySelector(SELECTORS);

Returns the first element within the document that matches the specified group of selectors, or NULL if no matches are found.