# ***Javascript Notes***

## ***Output***

Console.log(“Hello World !”)

## ***Variables***

1. ***Let:-***  The variable cannot be redeclared but can be updated. A block scope variable.
2. ***Var:***- The variable can be redeclared and can be updated.
3. ***Const:***- The variable cannot be redeclared and cannot be updated.

# ***DOM Manipulation***

The DOM (Document Object Model) is a programming interface for web documents. It represents the structure of an HTML or XML document as a tree of objects, where each node corresponds to part of the document (e.g., elements, attributes, text). The DOM provides a way for programming languages (like JavaScript) to interact with and manipulate the structure, style, and content of a web page in real-time.

## ***DOM Manipulation***

DOM Manipulation refers to the process of dynamically altering the content, structure, and style of a web page through the DOM. By using JavaScript, developers can add, remove, modify, or even animate elements on a web page without needing to reload the page.

## ***Key Functions of DOM Manipulation***

### ***Accessing Elements:***

You can access elements of the DOM by their id, class, tag, or any CSS selector. The following methods or functions can be used to achieve this functionality:-

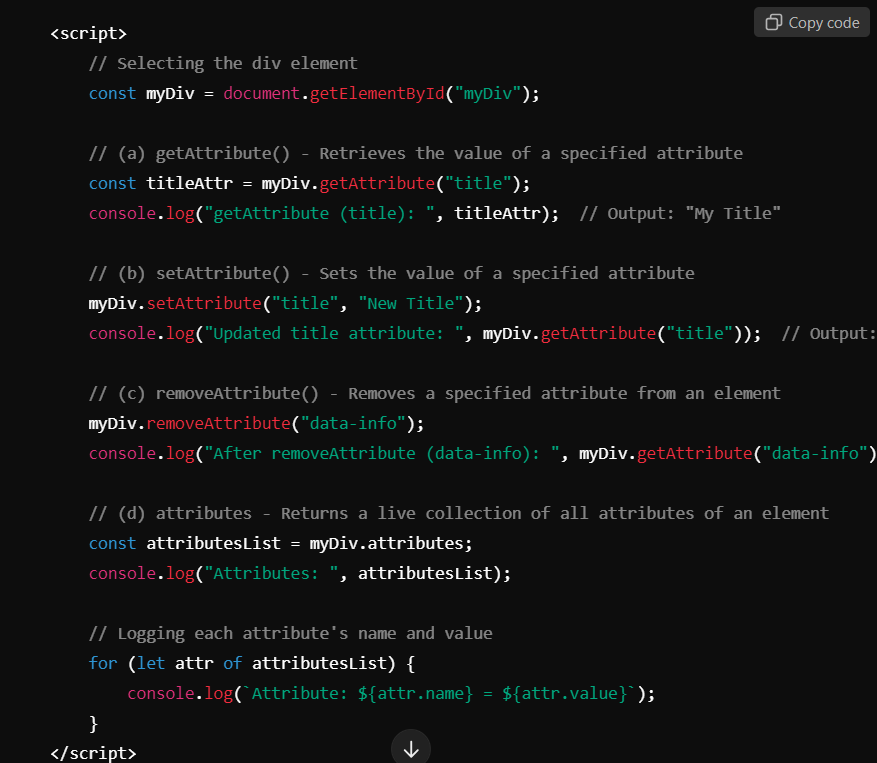
* + - * 1. **getElementById()** :- getElementById() selects the element with id="main".
        2. **getElementsByClassName()**:- returns all elements with the class "item" as an HTML Collection.
        3. **getElementsByTagName()** :- retrieves all <li> elements.
        4. **querySelector()**:- returns the first element with the class "item".
        5. **querySelectorAll()**:- returns a Node List of all elements with the class "item"

A screen shot of a computer program.


### ***Modifying Content:***

You can change the inner content of an element using some special methods. The following are the methods which can be used to manipulate the text.

* + - * 1. **getAttribute() :-** Retrieves the value of a specified attribute.
        2. **setAttribute() :-** Sets the value of a specified attribute.
        3. **removeAttribute() :-** Removes a specified attribute from an element.
        4. **Attributes :-** Returns a live collection of all attributes of an element.



### ***Creating and Removing Elements:***

Dynamic creation and removal of elements enhance interactivity and user experience. The following methods can be used for this purpose.

1. **createElement() :-** Creates a new element with the specified tag

name.

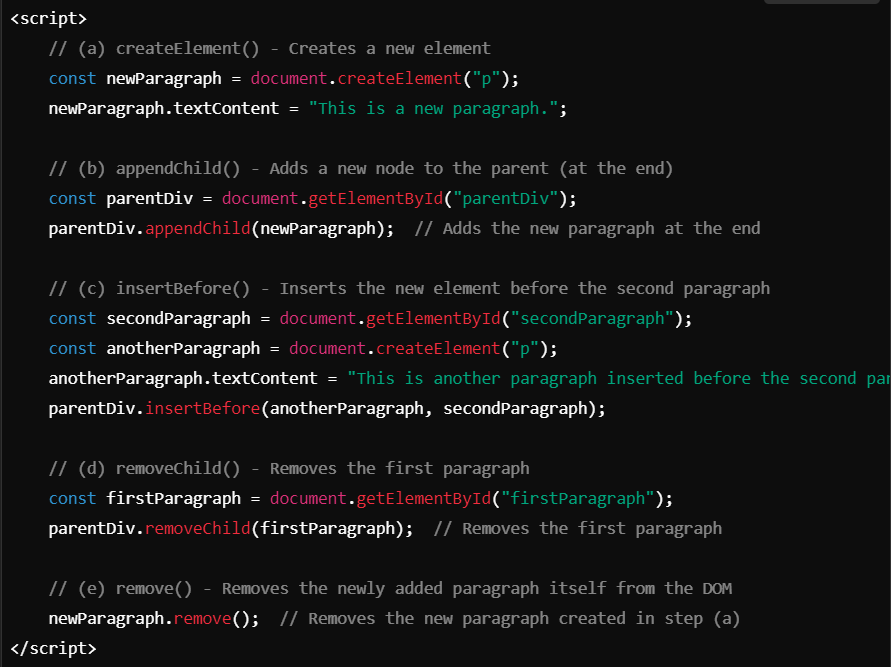
1. **appendChild() :-** Adds a node to the end of the list of children of a

specified parent node

1. **insertBefore():-** Inserts a node before a reference node as a child of a

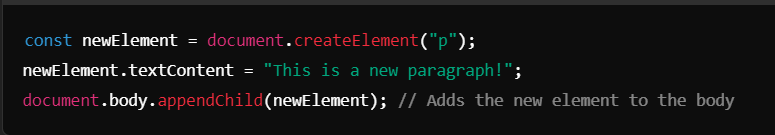
specified parent node.

1. **removeChild():-** Removes a specified child node from the DOM.
2. **remove():-** Removes the element itself from the DOM.



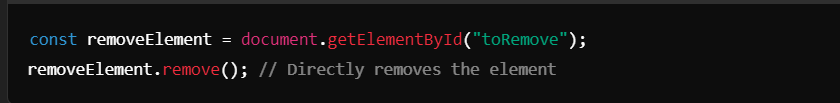
### ***Creating and Appending Elements:***

You can create new elements using document.createElement() and append them to the DOM using appendChild().



### ***Removing Elements:***

Elements can be removed from the DOM using removeChild() or remove().



### ***Event Handling:***

DOM allows you to assign events to elements (e.g., click, hover). This makes pages interactive.

