

What is DOM

- 1.DOM is a Document Object Model in JS. It is used to interact and control the web page's function. Without JS we can't do any interactions with the web page.
2. When the web page is loaded with HTML structure and we use DOM in java script it tells what is happen when the user clicks or doing interactions in the web page. Without DOM we can only use the static web page.
3. The DOM acts as a bridge between HTML and JavaScript. It transforms static HTML into a dynamic and interactive structure, allowing developers to create modern, responsive, and user-friendly web applications.
- 4 . Using DOM we can add, change or delete and add styles in the web page.
5. Everything in DOM is considered as an object.
6. We can select elements by using objects.

We can select the elements by using

- o `document.getElementById("id")`
- o `document.getElementsByClassName("class")`
- o `document.querySelector("selector")`

DOM Events

The DOM supports **events** that make web pages interactive. Examples: click, mouseover, keydown, scroll, etc.

Tables in html

Tables in html is used to represent the data in rows and columns. It helps to organise the information in a structured format.

`<table>` tag is used to create a table. `<thead>` is used to group the columns. `<th>` is used to define column headers. `<tbody>` is used to define the contents in the table. Inside body we have `<tr>` tag to define the rows and `<td>` tag to define the data cell. There are same number of `<td>` tags equal to the `<th>` tags.

We can use attributes like rowspan, colspan, table border, cell spacing and cell padding to style the table.

Array Methods

Reduce()

The `reduce()` method is used to reduce an array to a single value by running a function on each element (from left to right).

It is commonly used for calculations like sum, average, product, or combining data.

Syntax

```
array.reduce(function(accumulator, currentValue) {  
  // logic  
})
```

Where accumulator stores the value of previous operations. Current value represents the current element to be processed.

map()

The map() method is used to create a new array by applying a function to each element of an existing array. It does not change the original array, but returns a new array.

Syntax

```
array.map((currentValue, index, array) => {  
});
```

Filter()

The filter() method creates a **new array** that contains **only the elements which pass a specific test (condition)**.

It doesn't modify the original array ,it just filters out the elements that meet the condition.

Syntax

```
array.filter(currentValue => condition);
```

forEach()

The forEach() method is used to execute a function once for each element in an array. It's mainly used for looping through arrays to perform an action (like printing, updating, or calculating something).

It does not return a new array, it simply runs the function for each item.

```
array.forEach((currentValue, index, array) => {  
});
```

Fetch()

The fetch() function is used to get data from a server (API) or send data to a server. It returns a Promise, meaning it works asynchronously , it doesn't block other code while waiting for a response.

Syntax

```
fetch(url, options)
  .then(response => {
    // handle response
  })
  .catch(error => {
    // handle error
  });
});
```