JS JavaScript TUTORIAL

Document Object Model in JavaScript

Summary: in this tutorial, you will learn about the Document Object Model in JavaScript.

Introduction to Document Object Model in JavaScript

Suppose you have an HTML file called `index.html` with the following code:

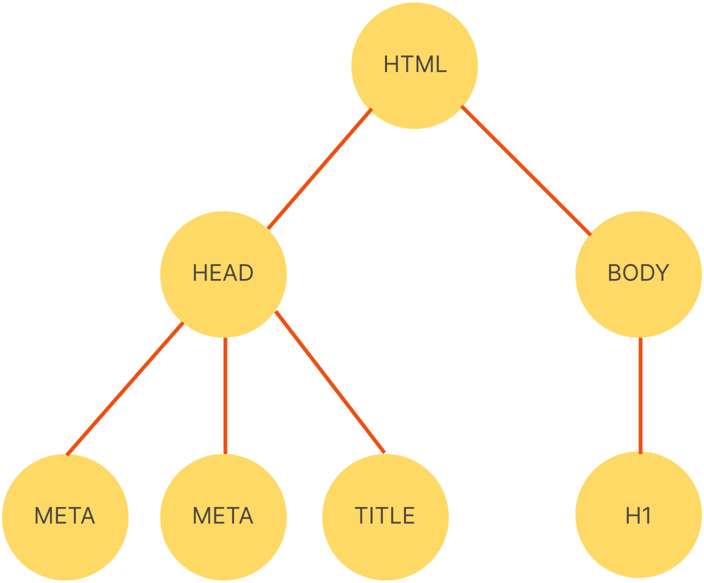
```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>JavaScript DOM</title>
  </head>
  <body>
    <h1>Hello DOM!</h1>
  </body>
</html>
```

If you open the `index.html` on your web browser, you'll see the following page.

Hello DOM!

The web browser displays the message `Hello DOM!` as specified by the `<h1>` tag on the `index.html` document.

Additionally, the web browser also creates something called a DOM tree internally:



DOM stands for document object model.

The web browser uses DOM to represent the HTML document internally. Additionally, it provides a set of functions and methods to modify the HTML document programmatically.

These functions and methods are often called DOM Application Programming Interfaces or DOM API.

For example, you can use JavaScript to select the h1 tag using the querySelector() method:

```
const h1 = document.querySelector('h1');
```

And you can get the text content of the `h1` element:

```
console.log(h1.textContent)
```

It'll show the text content of the `h1` tag in the console:

```
Hello DOM!
```

You can change the `h1` tag dynamically by setting the `textContent` to a new text:

```
h1.textContent = 'Hi JS';
```

Using DOM API in JavaScript, you can manipulate the HTML document effectively.

In the next tutorials, you'll learn various methods for selecting, traversing, and manipulating DOM.

Summary

- DOM stands for Document Object Model.
- DOM API provides a set of functions and methods to modify the HTML document dynamically via JavaScript.