DOM (Document Object Model)

Document Object Model is like a map of Website.

Just like how map shows you where all streets and buildings are in a city, the DOM shows you where everything is on website.

The DOM helps your computer understand the different parts of website and how they are put together.

" the data representation of the objects that comprise the structure and content of the document on web".

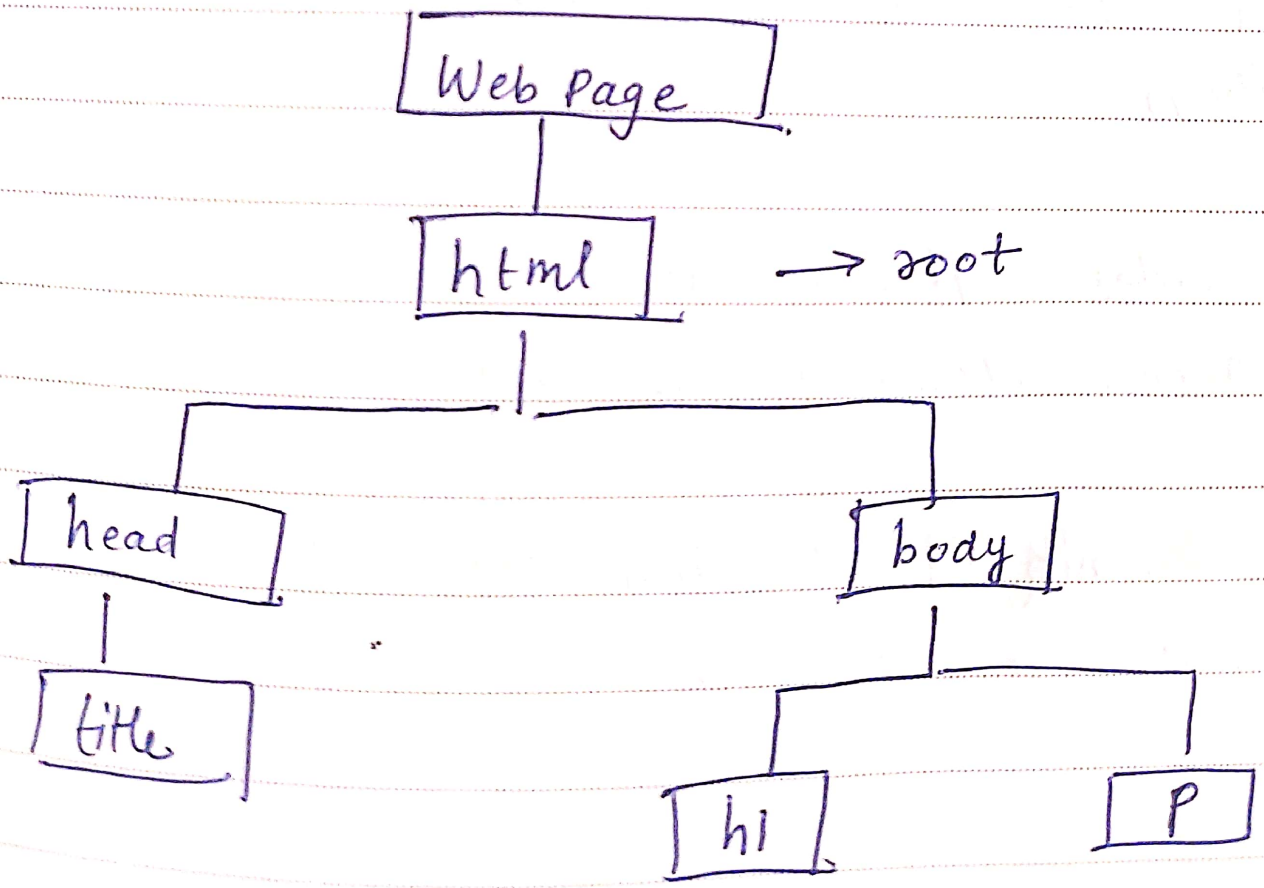
DOM Tree

Imagine a website is like a big book,

each page in the book represents a different part of website. The Dom Tree is like table of contents of that book.

Each part of website is called an "element" and these elements are arranged in Tree-like structure.

The Top of the tree is called root and It represents entire website.



DOM Tree represents the structure of website in a way that computers can understand. Developers can use it to access and manipulate different elements in that structure to create dynamic web pages.

How to Access the DOM?

Accessing element means finding specific parts of website and changing or manipulating them.

Js provides different methods to access element in DOM.

1- getElementById → find element based on its Id.

2. getElementsByTagName → Here you can get elements with help of tag name

3. querySelector → It returns the first element within the document that matches specified selector

4. querySelectorAll() → returns a static nodelist representing a list of document's element that match specified group of selectors.

For eg →

```
< div id = "student-list" >
    <div id = "student-1" class = "student" >
        Deepa </div>
</div>
```

CSS →

```
.student {
    padding : 40px ;
}

.student : hover {
    background-color : #F1F1F1 ;
}
```

Js Code

```
let student1 = document.getElementById(
                    "student-1");
Student1.addEventListener("click",
        () => {
            student1.style.backgroundColor="blue";
        });
```

In this eg Js is using getElementById method to select element with id "student-1" and it changes it background Color to blue when you click on it.

How to Add, Remove and Modify DOM Elements.

Adding, Removing and modifying elements in DOM refers to adding new elements to webpage

removing existing elements and changing properties of existing element.

For eg →

```
<div id="wrapper" class="btn-wrapper">
  <button id="create-btn" class="btn" >
        Create new Button </button>
</div>
```

Js Code

```
let createButton = document.getElementById
                    ("create-btn");
let wrapper = document.getElementById ("wrapper");

createButton.addEventListener ("click", () =>
{
    let newButton = document.createElement ("button");
  newButton.innerHTML = "Click me";
  wrapper.appendChild (newButton);
});
```