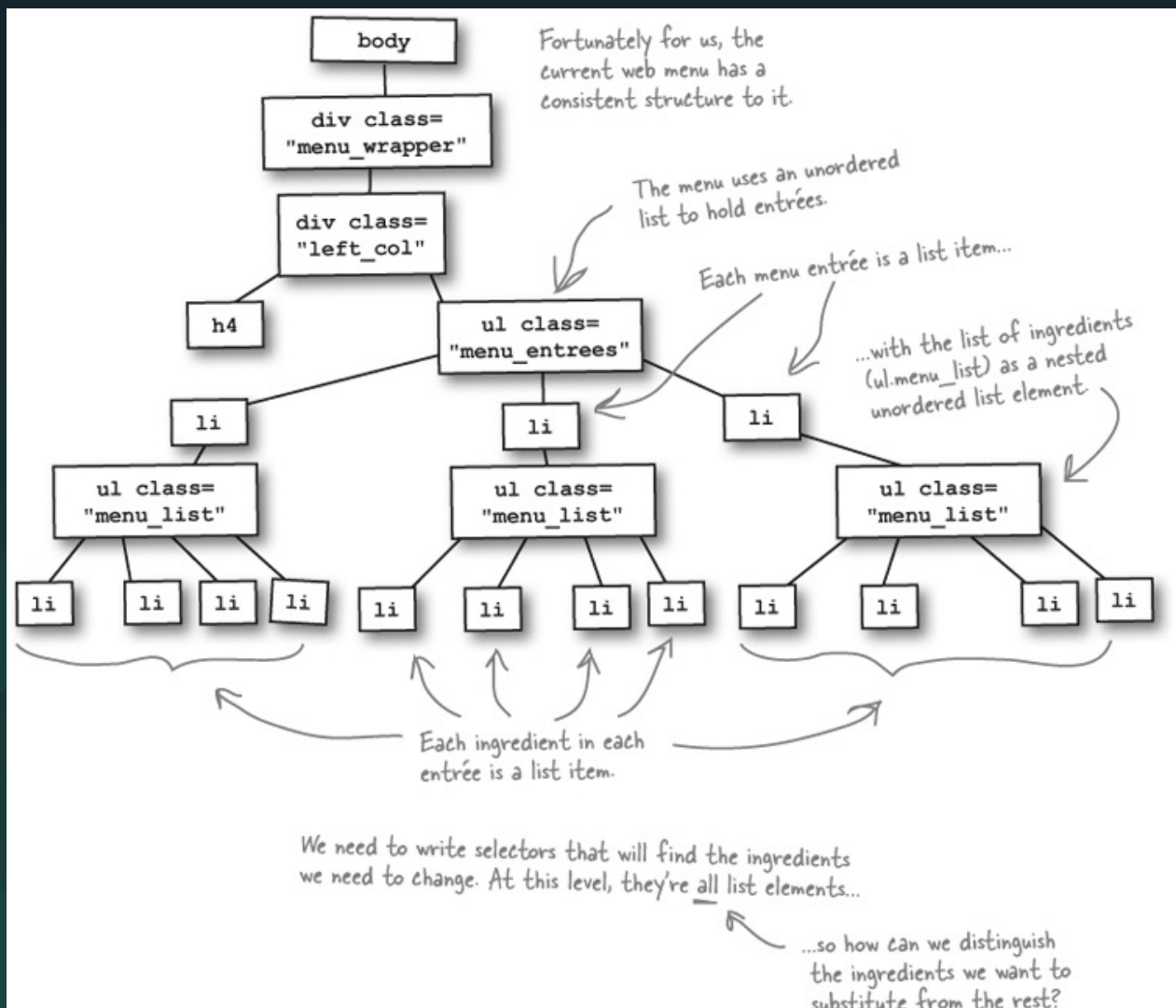




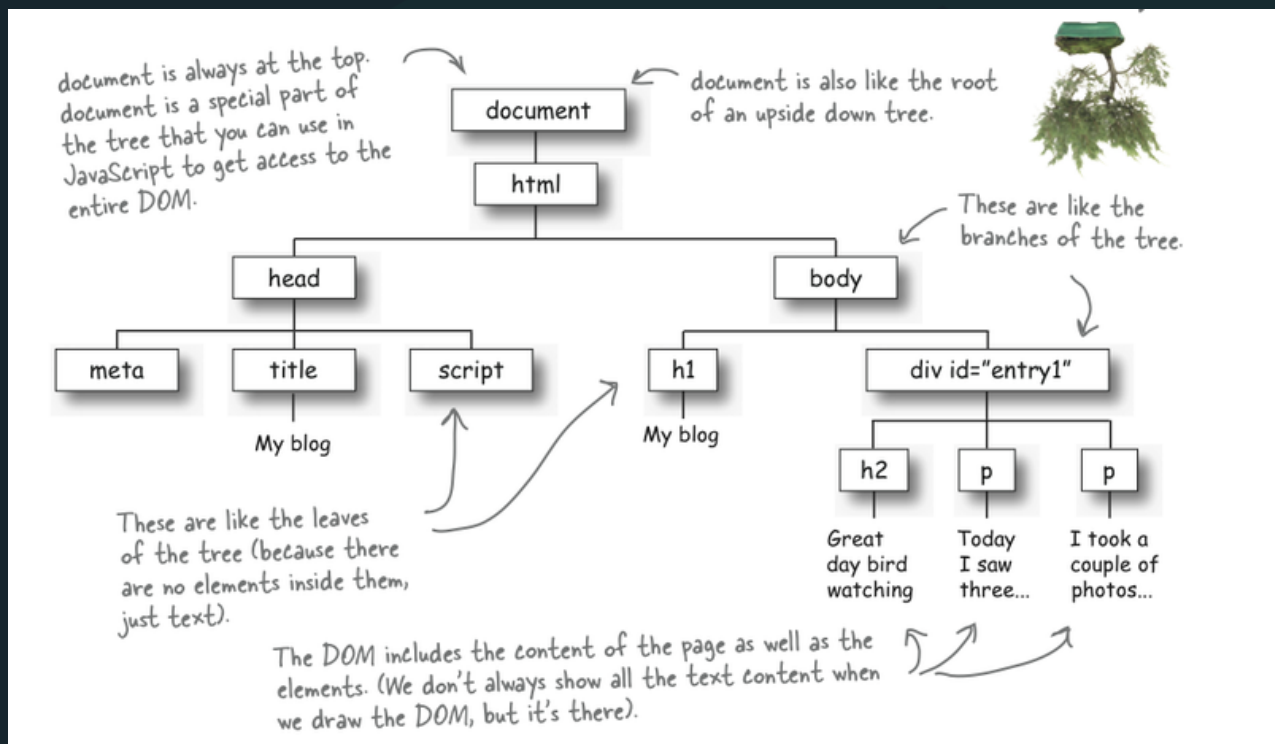
DOM





What is DOM?

DOM stands for Document Object Model. It is a programming interface for web documents and represents the structure of HTML and XML documents in a tree-like format. The DOM represents the web page so that programs can access and manipulate its content, structure, and style. It allows scripts (e.g., JavaScript) to dynamically interact with and modify the content and layout of a web page.

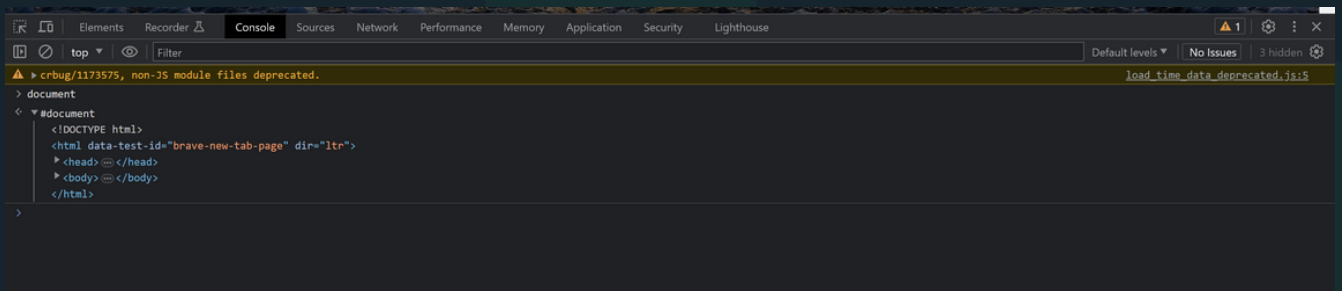




DOM In Console

To view the DOM in the console, you can open the developer tools of your browser (usually by pressing F12 or right-clicking and selecting "Inspect" or "Inspect Element"). In the developer tools, navigate to the "Console" tab and type `document` to view the root of the DOM tree.

To view the DOM in a tree view, you can expand the elements in the Elements (or similar) tab in the developer tools. This tree view visually represents the structure of the DOM.





Purpose of DOM

The purpose of the DOM is to provide a structured representation of a web page that can be easily accessed and manipulated by programs, such as JavaScript. It enables developers to:

- Access and modify individual elements and their attributes on a web page.
- Add, remove, or change HTML elements dynamically.
- Update the content and style of a web page in response to user interactions or events.
- Create interactive and dynamic web applications.



How to get the reference of any HTML element in JavaScript:

You can get references to HTML elements in JavaScript using various methods:

- By ID: Use `document.getElementById('elementId')` to get an element by its ID.
- By class: Use `document.getElementsByClassName('className')` to get elements by their class name.
- By tag name: Use `document.getElementsByTagName('tagName')` to get elements by their tag name.
- By CSS selector: Use `document.querySelector('selector')` to get the first matching element for a CSS selector.
- By CSS selector (multiple elements): Use `document.querySelectorAll('selector')` to get all matching elements for a CSS selector.




In JavaScript, the DOM (Document Object Model) properties refer to the attributes or characteristics of the elements in a web page that can be accessed and manipulated using JavaScript. These properties provide a way to interact with the elements and obtain information or change their behavior dynamically.

Here are some examples of DOM manipulation in the browser console:

- Changing the content of an element:


javascript

 Copy code

```
document.getElementById('elementId').textContent = 'New text content';
```

- Changing the style of an element:


javascript

 Copy code

```
document.querySelector('selector').style.color = 'red';
```

- Creating a new element and appending it to the document:

javascript

 Copy code

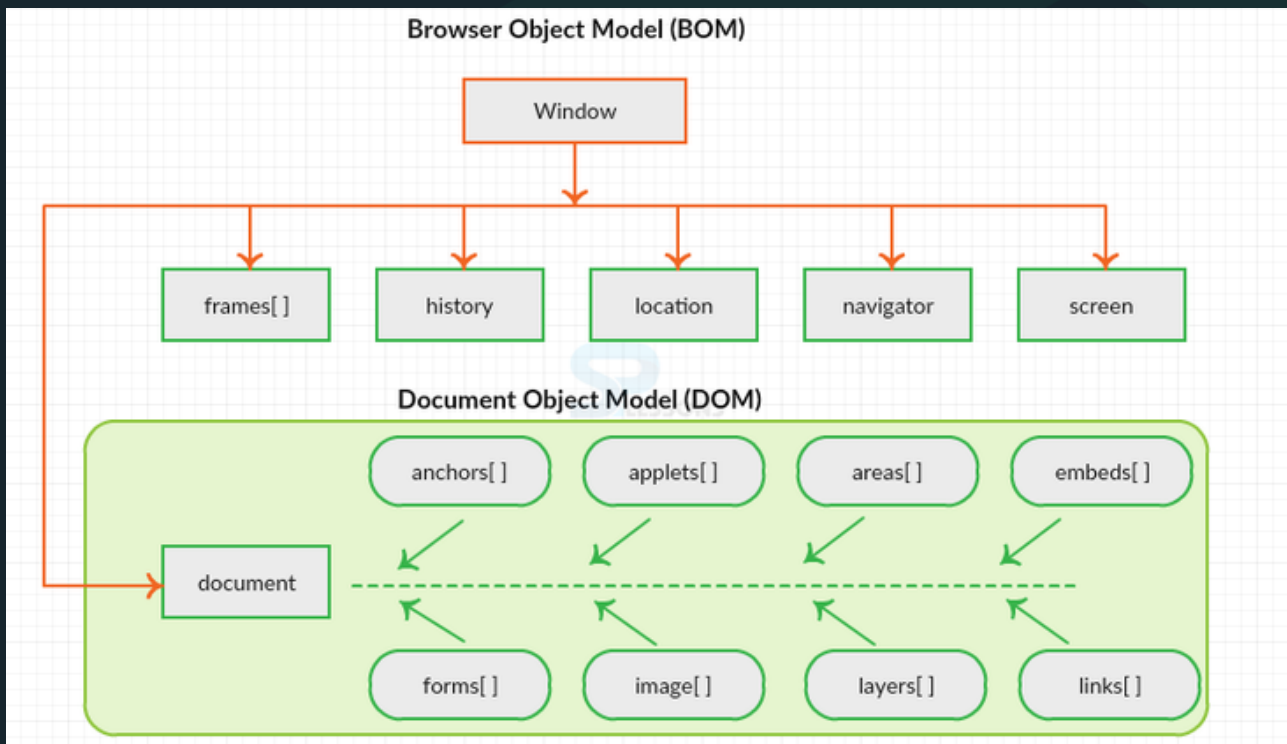
```
const newElement = document.createElement('div');  
newElement.textContent = 'This is a new element';  
document.body.appendChild(newElement);
```



Windows v/s DOM Object

The window object in JavaScript represents the browser window and is the top-level object that contains the DOM, among other properties and methods. It acts as the global object for a web page and provides various functionalities such as opening new windows, setting timeouts, and accessing the document.

On the other hand, the DOM object is a hierarchical representation of the HTML document and provides methods and properties to interact with the elements on the web page.





Task 1: Change the amount of any product on Amazon:

- Open the Amazon website.
- Inspect the element representing the product's amount (e.g., the price).
- Use JavaScript to select the element and change its content.

Task 2: Show Extract Data as well using DOM:

- Identify the data you want to extract from a web page.
- Use DOM methods to select the elements containing the desired data.
- Retrieve the data from the selected elements and display it in the console.