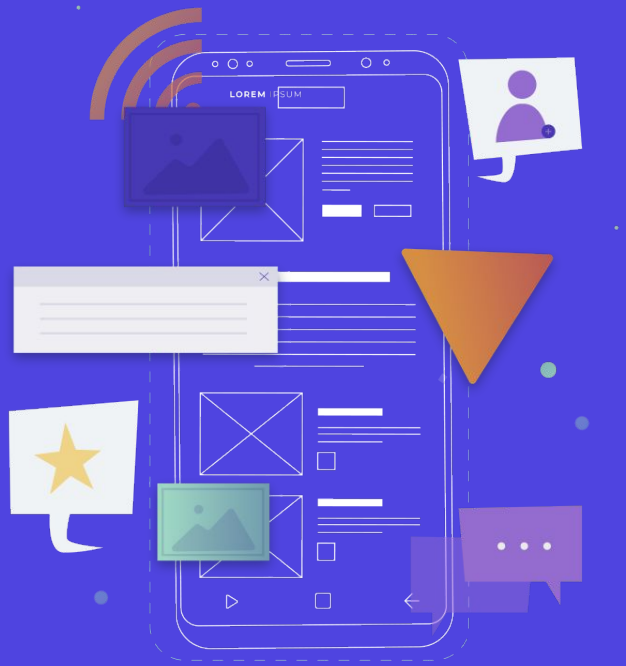
DOM



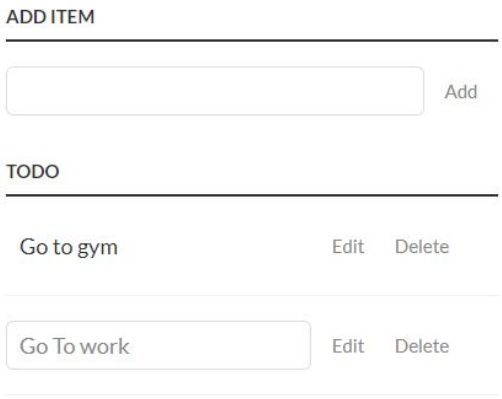
Relevel
by Unacademy

Topics Covered

- **App Feature**
  What we are going to lean and build today
- **Introduction**
  What is a DOM ?
- **Importance of DOM**
  Why we should know about the DOM
- **DOM Manipulation**
  Different ways of manipulating a DOM
- **Practice Questions**
  Assignments

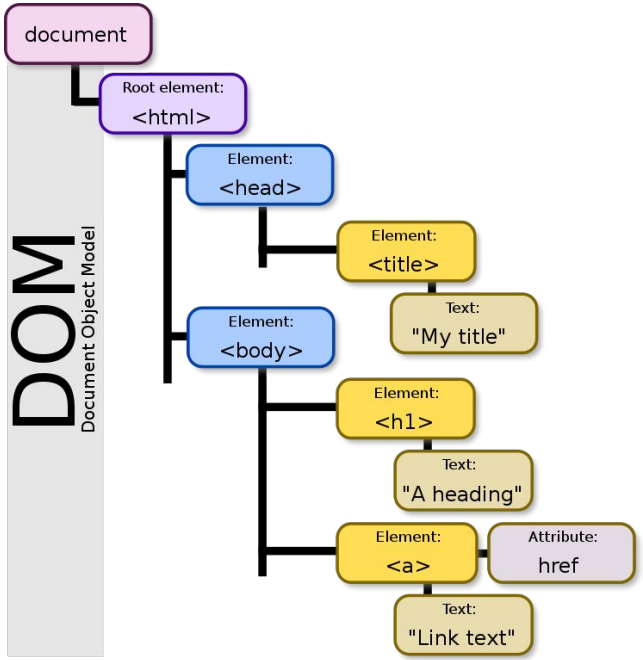
App Feature

- The app is a todo app with read, add, update, and delete functions.



What is a DOM

- DOM stands for Document Object Model
- It consists of a tree-like structure that encapsulates the content of a document on the web
- Assume DOM as a blueprint for the web pages that other programs

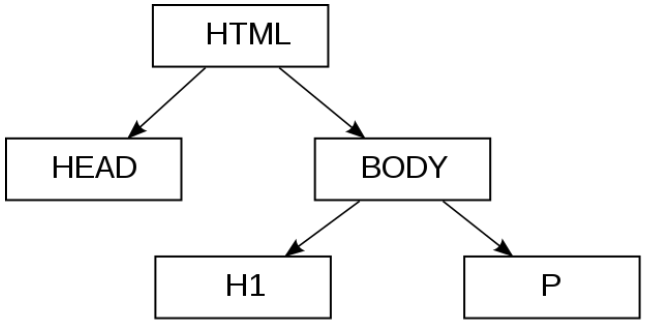


Importance of DOM

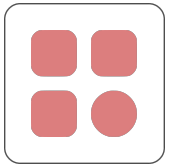
- Web page or a web application once deployed or hosted, allows less or no scope to the users to manipulate or change the content of the website/web page

- Using DOM, we can modify the content of the web page/web application even after it is deployed.

- The DOM goes hand in hand with JavaScript, the language can be used to manipulate the DOM and without DOM JavaScript would lose its object notation and web manipulation properties for web pages.



DOM manipulations



- In this session we are going to learn different ways of manipulating the DOM and with it to manipulate and change the web dynamically.

- But before we proceed let just learn about the basic concept of DOM .

What is “document” Object :

example:

```
console.dir(document)
```

i> The ‘document’ keyword represents the DOM properties available to us via the browser.

ii> Use console.dir(document) or console.log(document) and see the log

Grabbing the elements

Fetching all of **DOM properties**:
example:

```
console.log(document.all)
```

This prints the contents of the head tag of the HTML document.

Fetching the contents of **body tag**:
example:

```
console.log(document.body)
```

Fetching the contents inside the body tag

Fetching the contents of **head tag**:
example:

```
console.log(document.head)
```

This prints the contents of the head tag of the HTML document.

Fetching elements by ID

- The **getElementById** DOM method returns an element based on its ID that it receives as a parameter.

- var headerTitle = document.getElementById('header-title')
  This return the HTML element whose ID is header-title and the node representing the element will be stored in the headerTitle,

- Changing the Text and HTML
  - i> content: headerTitle.innerHTML = "<h3>Hello</h3>"
  - ii> The innerHTML attribute allows us to modify the content of the HTML element.

- Example Code :
  https://jsfiddle.net/TanayTapanshu/7x1vdo9e/8/

Fetching elements by Class Name

- The **getElementsByClassName** DOM method returns an array of HTML elements based on the class name it receives as a parameter.

- This will fetch an array of elements containing the class name list-group-item and store it in items variable.

- Example Code : **https://jsfiddle.net/TanayTapanshu/7x1vdo9e/26/**

Fetching elements by Tag Name

- **getElementsByTagName()** DOM method returns an array of HTML elements matching the HTML tag it returns as a parameter.

- Example Code : https://jsfiddle.net/TanayTapanshu/7x1vdo9e/35/

Query Selectors:

- The **querySelector()** method returns the first element that matches a specified CSS selector(s) in the document.
- The **querySelectorAll()** method all elements in the document that matches a specified CSS selector(s), as a static NodeList object
- Code Example - **https://jsfiddle.net/TanayTapanshu/7x1vdo9e/40/**

Traversing the DOM

- A JavaScript developer should know how to traverse the DOM
- It’s the way of selecting an element from another element

Parent Node Property

- The parentNode property returns the parent node of the specified node, as a Node object.
- It’s the way of selecting an element from another element
  Example -
  var itemList = document.querySelector('#items');
  console.log(itemList.parentNode);
  itemList.innerHTML = “traversing”

  In the above example we are getting the element with the ID as items and storing it in the itemList variable.

Parent Element property

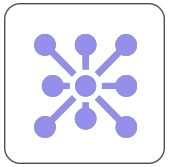
- The parentNode property returns the parent node of the specified node, as a Node object.

- The parentElement property returns the parent element of the specified element.

- Example -
  console.log(itemList.parentElement);
  This prints the parent element of the HTML element whose ID is items.

  itemList.parentElement.style.backgroundColor = "#f4f4f4"
  This will modify the background colour of the parent using the style.backgroundColor property.

Child Node property



- The childNode property returns the child node of the specified node, as a Node object.

- Example -
  console.log(itemList.childNodes)
  This prints the child nodes of itemList

Children Element property

- The childrenElement property returns the children element of the specified element.

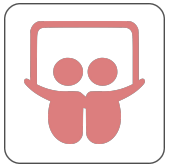
- Example -
  console.log(itemList.children)
  This print the child elements of itemList.

First & last Child:

- The firstChild & lastChild properties return the first and last node of respective HTML elements.

- Example -
  console.log(itemList.lastChild);
  console.log(itemList.firstChild);
  This prints first node and last node

Slideshare Application



- Let’s Build a Slideshare application using the concepts we have learned in this session till now.
- We will define the title of the document using the title tag inside head tags.
- We will add an img tag inside body to create images and the tag will have the name attribute set to slide, width set to 400 & height set to 200
- We will define a script tag to house our JavaScript function.
- Check the Code - https://jsfiddle.net/TanayTapanshu/7x1vdo9e/58/

Todo Application

- In today's session we will be also building the todo application
- We will be using the concepts which we learnt in this session.
- Check the Code

  Link: **https://codepen.io/tanaytapanshu/pen/bGLKpzL**

Code Explanation

- **CreateNewTaskElement, addTask, editTask, and deleteTask** are functions for adding a new task, amending a task, and removing a task.
- **CreateNewTaskElement** function is used to create rext label, edit Input, edit and update buttons are used to create task elements.
- Using the **CreateNewTaskElement** function, the **addTask function** adds a new task to the todo list.
- **editTask** function is used for editing a task
- At last **deleteTask** function for deleting a task

Code - **https://codepen.io/tanaytapanshu/pen/bGLKpzL**

Assignment

1. Create the login/ sign up form of ecommerce app.

Thank You!