



Live Cohort

Dom Manipulation



DOM Manipulation: Visual & Detailed Notes (JavaScript)

Topic: Selecting Elements in the DOM

Method

The first step in DOM manipulation is selecting HTML elements. JavaScript provides several methods to access and work with different parts of the DOM.

Code

(Examples):

Select Single Element

```
const heading = document.getElementById('main-heading');
const firstItem = document.querySelector('.item');
```

Select Multiple Elements

```
const buttons = document.getElementsByClassName('btn');
const allParagraphs = document.getElementsByTagName('p');
const allItems = document.querySelectorAll('.list-item');
```

Use Case:

- **Highlighting a header.**
- **Targeting all buttons for applying event listeners.**
- **Dynamically styling or updating a group of elements.**

Interview Q&A:

Q: What is the difference between getElementById() and querySelector()?

A:

- `getElementById()` selects by ID only.
- `querySelector()` is more flexible and allows CSS-style selectors (`.class`, `#id`, `tag`).

Q: What's the difference between querySelectorAll() and getElementsByClassName()?

A:

- `querySelectorAll()` returns a static `NodeList`.
- `getElementsByClassName()` returns a live `HTMLCollection` that updates with the DOM.

Topic: Changing HTML Content

Definition:

JavaScript allows dynamic content updates using `innerText` and `innerHTML`.

Code (Examples):

```
document.querySelector('h1').innerText = 'Welcome to Web Dev!';  
document.querySelector('.box').innerHTML = '<p>New content added</p>';
```

Use Case:

- **Displaying messages after form submission.**
- **Updating part of a page without reloading (e.g., AJAX results).**

Interview Q&A:

Q: When should you use innerText vs innerHTML?

A:

- Use innerText when you're dealing with plain text.
- Use innerHTML when you want to render HTML tags or elements inside an element.

Topic: Changing CSS with JavaScript

Definition:

You can dynamically update the appearance of elements using the .style property.

Code (Examples):

```
const card = document.querySelector('.card');
card.style.backgroundColor = 'lightblue';
card.style.padding = '20px';
card.style.borderRadius = '10px';
```

Use Case:

- Highlighting active menu items.
- Changing themes or styling during runtime.
- Visual feedback for user interaction.

Interview Q&A:

Q: How do you dynamically change the background color of an element using JS?

A:

Use the `.style` property:

```
element.style.backgroundColor = 'color';
```

Topic: Changing CSS with JavaScript

Definition:

You can dynamically update the appearance of elements using the `.style` property.

Code (Examples):

```
const btn = document.querySelector('#submitBtn');

btn.addEventListener('click', () => {
  alert('Form Submitted!');
});

btn.addEventListener('mouseover', () => {
  btn.style.backgroundColor = 'green';
});
```

Use Case:

- **Highlighting active menu items.**
- **Changing themes or styling during runtime.**
- **Visual feedback for user interaction.**

Interview Q&A:

Q: Can you attach multiple event listeners to a single element?

A: Yes. You can attach as many event listeners as needed to the same element for different or even the same events.

Q: Name some commonly used DOM events.

A: click, mouseover, mouseout, keydown, keyup, submit, scroll.

Topic: Project – Bulb Application

Definition:

A mini-project that toggles a bulb image on button click using DOM manipulation and events.

Code (Examples):

HTML:

```

<button id="toggleBtn">Turn On</button>
```

JavaScript:

```
const bulb = document.getElementById('bulb');
const button = document.getElementById('toggleBtn');
let isOn = false;

button.addEventListener('click', () => {
  if (isOn) {
    bulb.src = 'https://www.w3schools.com/js/pic_bulboff.gif';
    button.innerText = 'Turn On';
    isOn = false;
  } else {
    bulb.src = 'https://www.w3schools.com/js/pic_bulbon.gif';
    button.innerText = 'Turn Off';
    isOn = true;
  }
});
```

Use Case:

- Demonstrates how JavaScript updates DOM attributes and handles state logic.

Interview Q&A:

Q: How do you change the src attribute of an image dynamically?

A:

By setting the .src property:

```
imgElement.src = 'new-image-url';
```

Q: How do you track toggle state in such apps?

A:

By using a boolean flag (e.g., let isOn = false) and updating it on each interaction.