DOM Manipulation: Complete Theory with Examples

# What is the DOM?

DOM stands for Document Object Model. It is a programming interface for HTML and XML documents. When a webpage is loaded, the browser creates a live tree structure representing the HTML content. This structure allows developers to access and manipulate elements, attributes, and content dynamically using JavaScript.

# Why DOM Manipulation?

DOM manipulation is crucial for:  
- Making interactive web pages.  
- Dynamically updating content without reloading the page.  
- Handling events like clicks, keypresses, and mouse movements.  
- Creating dynamic UIs, modals, tabs, sliders, forms, etc.

Methods for DOM Manipulation

## 1. Selecting Elements

document.getElementById("title"); // By ID  
document.getElementsByClassName("description"); // By Class  
document.getElementsByTagName("p"); // By Tag  
document.querySelector("h1"); // First match (CSS-style)  
document.querySelectorAll("p"); // All matches

## 2. Changing Content

document.getElementById("title").innerText = "Welcome!";  
document.querySelector(".description").innerHTML = "<strong>Updated</strong>";

## 3. Changing Styles

document.getElementById("title").style.color = "blue";  
document.body.style.backgroundColor = "#f0f0f0";

## 4. Changing Attributes

let button = document.querySelector("button");  
button.setAttribute("disabled", true);  
button.removeAttribute("disabled");

## 5. Creating and Appending Elements

let newDiv = document.createElement("div");  
newDiv.innerText = "This is a new div!";  
document.body.appendChild(newDiv);

## 6. Removing Elements

let para = document.querySelector("p");  
para.remove();

## 7. Event Handling

document.querySelector("button").addEventListener("click", function() {  
 alert("Button was clicked!");  
});

# Real-Time DOM Manipulation Example: Live Clock

<div id="clock"></div>  
<script>  
 setInterval(() => {  
 const now = new Date();  
 document.getElementById("clock").innerText = now.toLocaleTimeString();  
 }, 1000);  
</script>

# Summary

Operation - Method  
Select element - getElementById, querySelector, etc.  
Change content - .innerText, .innerHTML  
Change style - .style.property  
Add/remove element - createElement, appendChild, remove()  
Handle events - addEventListener("click", callback)

# Final Thoughts

DOM manipulation is a core part of front-end development. While libraries like React and Vue abstract much of this work, understanding raw DOM operations makes you a better developer, helps debug browser-based issues, and is essential for small projects or scripts.