4-Document Object Model (DOM)

Chapter 4: JavaScript and the Web

JavaScript becomes truly powerful when combined with HTML and CSS to create interactive web applications. This chapter will introduce the DOM, show you how to manipulate HTML and CSS with JavaScript, and teach you how to handle user input and events.

4.1 What is the DOM?

The Document Object Model (DOM) is a programming interface that represents the structure of an HTML document as a tree of objects. JavaScript interacts with the DOM to access, modify, and control elements on a webpage.

Example of an HTML Document:

```
<!DOCTYPE html>
<html>
  <head>
    <title>Sample Page</title>
  </head>
  <body>
    <h1 id="title">Hello, World!</h1>
    <p class="description">Welcome to this sample page.</p>
  </body>
</html>
```

DOM Representation:
The HTML elements (`<h1>` , `<p>`) become nodes in the DOM tree. JavaScript can traverse this tree to manipulate or access these nodes.

4.2 Selecting HTML Elements

JavaScript provides methods to find and select elements on a webpage. Common methods include:

1. `getElementById`
   Selects an element by its `id` attribute.
   Example:

```
const title = document.getElementById("title");
console.log(title.textContent); // Output: "Hello, World!"
```

2. `getElementsByClassName`
   Selects elements by their `class` attribute.
   Example:

```
const descriptions = document.getElementsByClassName("description");
console.log(descriptions[0].textContent); // Output: "Welcome to this
sample page."
```

3. `querySelector`
   Selects the first element matching a CSS selector.
   Example:

```
const title = document.querySelector("#title");
console.log(title.textContent);
```

4. `querySelectorAll`
   Selects all elements matching a CSS selector, returning a NodeList.
   Example:

```
const paragraphs = document.querySelectorAll("p");
paragraphs.forEach((p) => console.log(p.textContent));
```

4.3 Manipulating HTML Elements

Once you've selected elements, you can modify their properties, styles, and content.

1. **Changing Text Content:**
   Example:

```
const title = document.getElementById("title");
title.textContent = "Hello, Rick!";
```

2. **Changing Attributes:**
   Example:

```
const image = document.querySelector("img");
image.src = "newImage.jpg";
```

3. **Changing CSS Styles:**
   Example:

```
const title = document.querySelector("#title");
title.style.color = "blue";
title.style.fontSize = "24px";
```

4. **Adding or Removing Classes:**
   Example:

```
const title = document.querySelector("#title");
title.classList.add("highlight"); // Adds the class
title.classList.remove("highlight"); // Removes the class
```

4.4 Creating and Appending Elements

You can dynamically create new HTML elements and add them to the DOM.

Example:

```
const newParagraph = document.createElement("p");
newParagraph.textContent = "This is a dynamically added paragraph.";
document.body.appendChild(newParagraph);
```

4.5 Handling User Input and Events

Events are actions like clicks, key presses, or mouse movements that occur on a webpage. JavaScript can listen for these events and respond accordingly.

4.5.1 Adding Event Listeners

Use the `addEventListener` method to attach an event handler to an element.

Example:

```
const button = document.querySelector("#myButton");
button.addEventListener("click", () => {
  console.log("Button clicked!");
});
```

4.5.2 Common Event Types

1. **Click Events:**

```
element.addEventListener("click", () => {
  console.log("Element clicked!");
});
```

2. **Key Events:**

```
document.addEventListener("keydown", (event) => {
  console.log(`Key pressed: ${event.key}`);
});
```

3. **Mouse Events:**

```
element.addEventListener("mouseover", () => {
  console.log("Mouse entered!");
});
```

4.5.3 Form Events

Listen to events like form submission, input, or change.

Example:

```
const form = document.querySelector("#myForm");
form.addEventListener("submit", (event) => {
  event.preventDefault(); // Prevent form from submitting
  console.log("Form submitted!");
```

```
});
```

4.6 Practical Examples

1. **Interactive Color Picker:**

```
const colorInput = document.querySelector("#colorPicker");
colorInput.addEventListener("input", () => {
  document.body.style.backgroundColor = colorInput.value;
});
```

2. **To-Do List:**

```
const input = document.querySelector("#taskInput");
const button = document.querySelector("#addTask");
const list = document.querySelector("#taskList");

button.addEventListener("click", () => {
  const task = input.value;
  if (task) {
    const newTask = document.createElement("li");
    newTask.textContent = task;
    list.appendChild(newTask);
    input.value = ""; // Clear the input
  }
});
```