Internship Report – Frontend Dev

Week 5: JavaScript Advanced Topics

Name: Zainab
Father Name: Assad Qayyum
Date: 21st July, 2025
Internship Domain: Front-end Intern
Task: JS - Events (onclick, onchange, addEventListener), Objects & Object Methods

Task Overview: (Day1)

Today's task was to explore **JavaScript Advanced Topics**, specifically focusing on how JavaScript can make web pages interactive and data-driven using **Events, Objects,** and **Object Methods**.

Content Covered:

- JavaScript Events:
    - onclick
    - onchange
    - addEventListener
- JavaScript Objects
- Object Methods

1. JavaScript Events:

An event is something that happens in the browser — like a click, typing in a form, hovering, scrolling, loading a page, etc. JavaScript lets you respond to these events using event handlers.

Common Event Types:

a) <u>onclick – Click Event:</u>

It happens when a user clicks on an element

Syntax:

<button onclick="sayHello()">Click Me</button>

- **onclick:** This is an event attribute in HTML. It triggers when the button is clicked.
- **"sayHello()":** This is the JavaScript function to be called when the event occurs.

```
<button onclick="sayHello()">Click me</button>

<script>
  function sayHello() {
    alert("Hello!");
  }
</script>
```

b) <u>onchange – Change Event:</u>

Used when the value of an input (like a dropdown or text box) changes.

Example:

```
<select onchange="showChoice(this.value)">
  <option value="apple">Apple</option>
  <option value="banana">Banana</option>
</select>

<script>
  function showChoice(fruit) {
    alert("You selected: " + fruit);
  }
</script>
```

- **onchange:** Event triggered when the selected value is changed.
- **this.value:** Refers to the selected option's value.

c) <u>addEventListener():</u>

Instead of writing onclick or onchange in HTML, we use addEventListener() in JavaScript to keep **structure (HTML)** and **behavior (JS)** separate.

Example:

```
<button id="myBtn">Click Me</button>

<script>
  document.getElementById("myBtn").addEventListener("click", function() {
    alert("Button clicked using addEventListener!");
  });
</script>
```

- getElementById("myBtn"): Selects the button using its ID.
- .addEventListener("click", function): Adds a click event listener.
- function() {...}: The code inside runs when the button is clicked.

2. JavaScript Objects:

An **object** in JavaScript is a data structure that lets you **group related information** using key–value pairs.

Syntax:

```
let person = {
  name: "Zainab",
  age: 21,
  isStudent: true
};
```

Explanation:

- let person → Declares a variable named person.
- { ... } → This is the object literal. It holds the object data.
- name: "Zainab" → name is a property/key, "Zainab" is the value.
- age: 21 → number type value.
- isStudent: true → boolean type value.

Accessing Data

console.log(person.name); // "Zainab"

console.log(person["age"]); // 21

- object.key → Most common and readable.
- object["key"] → Useful when key is dynamic or contains spaces.

Updating Values

person.age = 22;

person["name"] = "Areeba";

3. Object Methods – Functions inside Objects:

A method is just a **function** that's defined **inside an object**. It describes what the object can do.

Example:

```
let person = {
  name: "Zainab",
  age: 21,
  greet: function() {
    console.log("Hi, I'm " + this.name);
  }
};
```

Explanation:

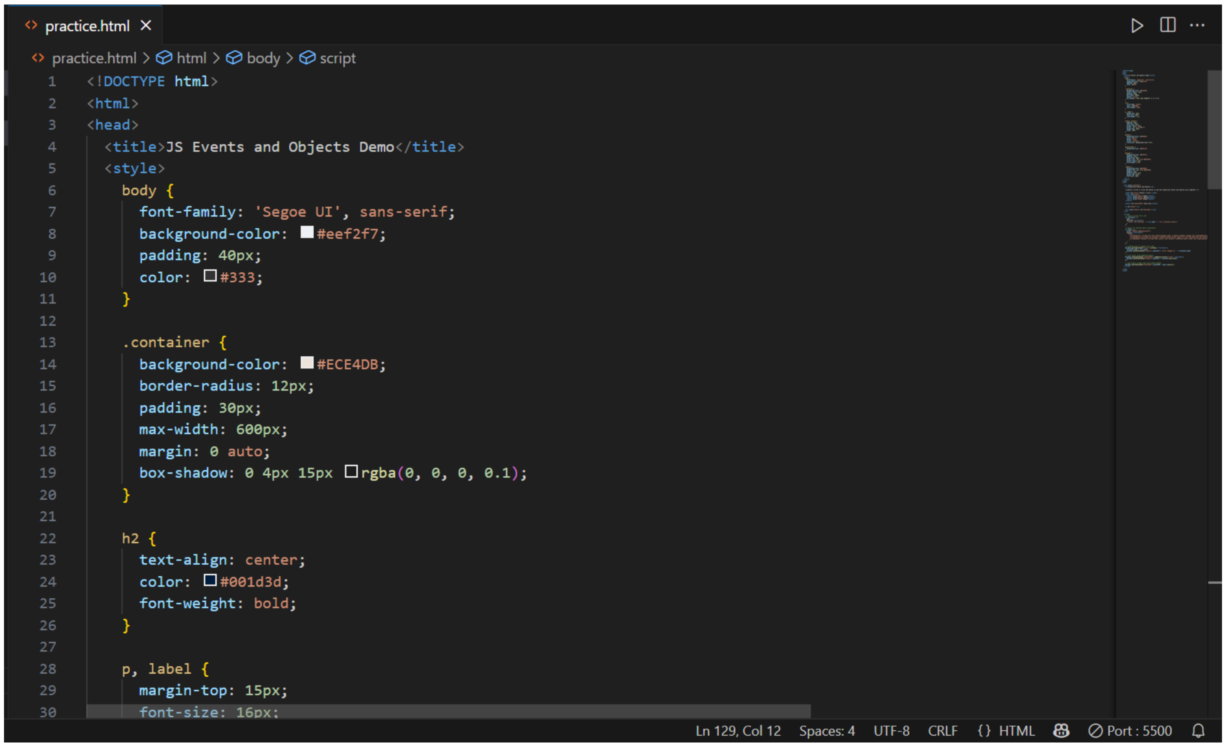
- greet is a method.
- function() {...} defines the method's behavior.
- this.name refers to the name property of this object

Calling the method:

person.greet();   // Output: Hi, I'm Zainab

Practice Code:

This code demonstrates how JavaScript handles **events (onclick, onchange, and addEventListener)** and how **objects and object methods** work. It lets the user select a fruit and displays a message using an object method. A second object is used to show a brief theory explanation on the webpage, making it a practical example of today's JavaScript concepts.

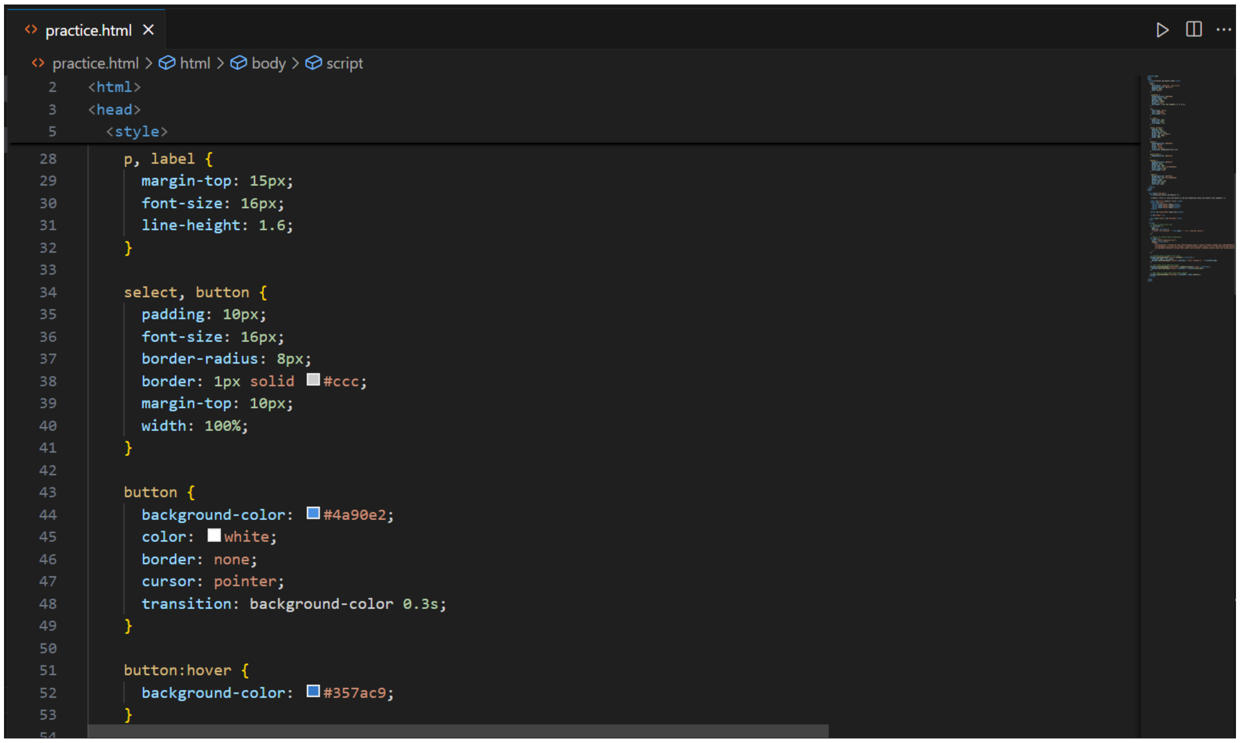


practice.html

practice.html > html > body > script

```
<!DOCTYPE html>
<html>
<head>
  <title>JS Events and Objects Demo</title>
  <style>
    body {
      font-family: 'Segoe UI', sans-serif;
      background-color: #eef2f7;
      padding: 40px;
      color: #333;
    }

    .container {
      background-color: #ECE4DB;
      border-radius: 12px;
      padding: 30px;
      max-width: 600px;
      margin: 0 auto;
      box-shadow: 0 4px 15px rgba(0, 0, 0, 0.1);
    }

    h2 {
      text-align: center;
      color: #001d3d;
      font-weight: bold;
    }

    p, label {
      margin-top: 15px;
      font-size: 16px;
```

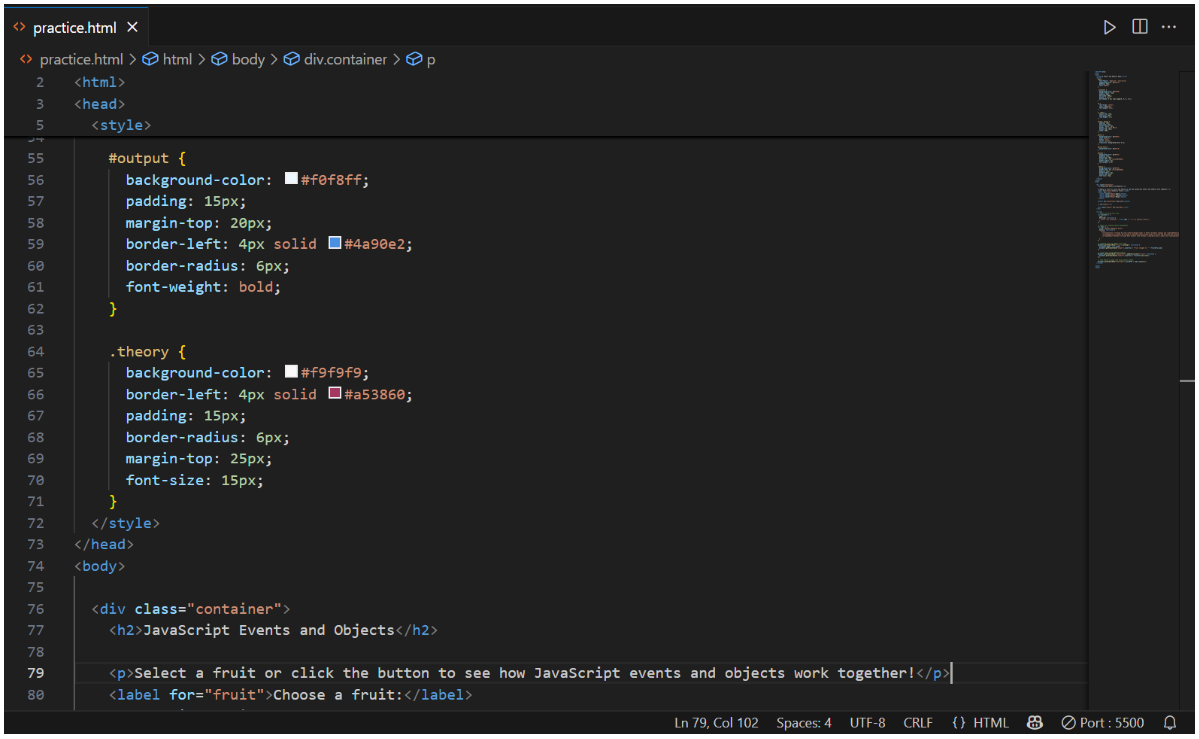
Ln 129, Col 12   Spaces: 4   UTF-8   CRLF   {} HTML   Port : 5500



practice.html

practice.html > html > body > script

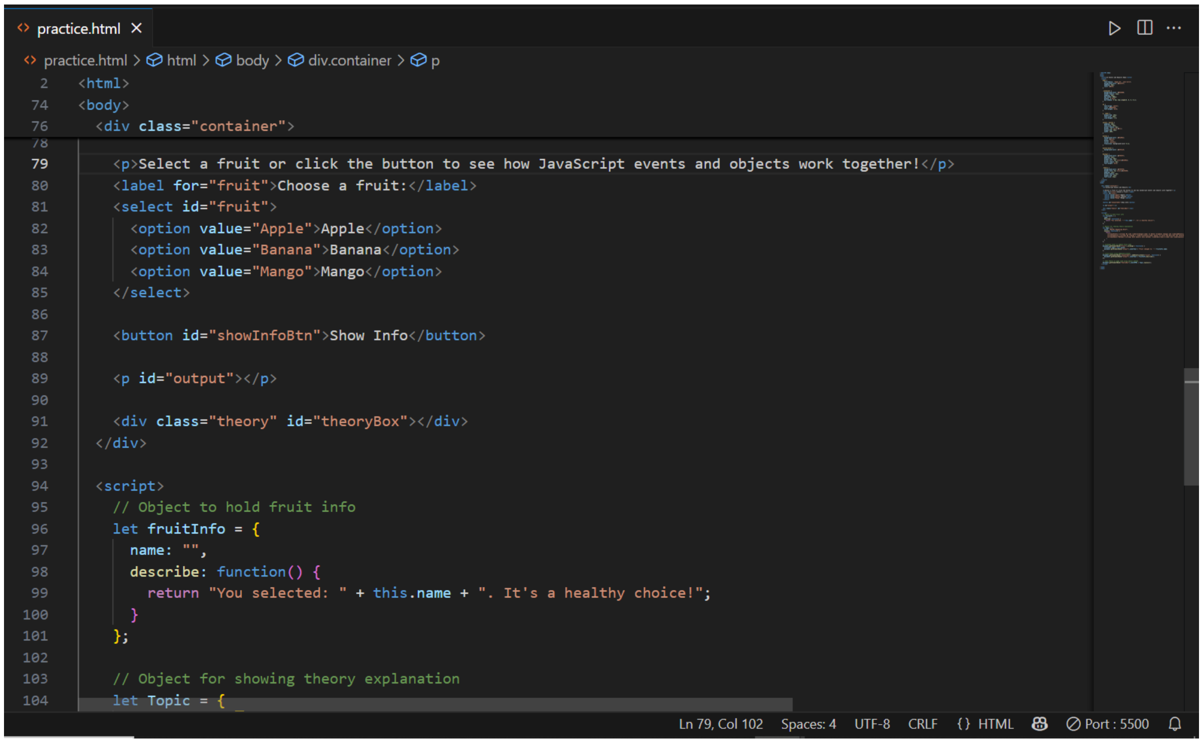
```
<html>
<head>
  <style>
    p, label {
      margin-top: 15px;
      font-size: 16px;
      line-height: 1.6;
    }

    select, button {
      padding: 10px;
      font-size: 16px;
      border-radius: 8px;
      border: 1px solid #ccc;
      margin-top: 10px;
      width: 100%;
    }

    button {
      background-color: #4a90e2;
      color: white;
      border: none;
      cursor: pointer;
      transition: background-color 0.3s;
    }

    button:hover {
      background-color: #357ac9;
    }
```


practice.html ×

practice.html > html > body > div.container > p

```
<html>
<head>
  <style>
    #output {
      background-color: #f0f8ff;
      padding: 15px;
      margin-top: 20px;
      border-left: 4px solid #4a90e2;
      border-radius: 6px;
      font-weight: bold;
    }

    .theory {
      background-color: #f9f9f9;
      border-left: 4px solid #a53860;
      padding: 15px;
      border-radius: 6px;
      margin-top: 25px;
      font-size: 15px;
    }
  </style>
</head>
<body>

  <div class="container">
    <h2>JavaScript Events and Objects</h2>

    <p>Select a fruit or click the button to see how JavaScript events and objects work together!</p>
    <label for="fruit">Choose a fruit:</label>
```

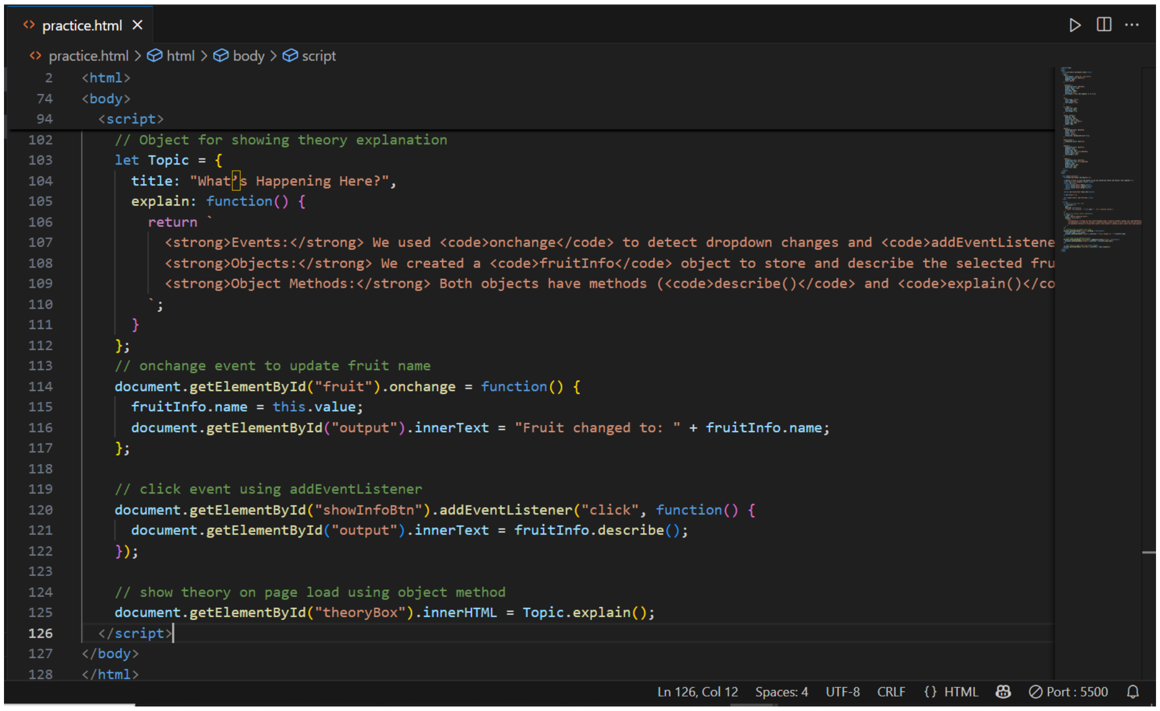
Ln 79, Col 102 Spaces: 4 UTF-8 CRLF {} HTML Port : 5500


practice.html ×

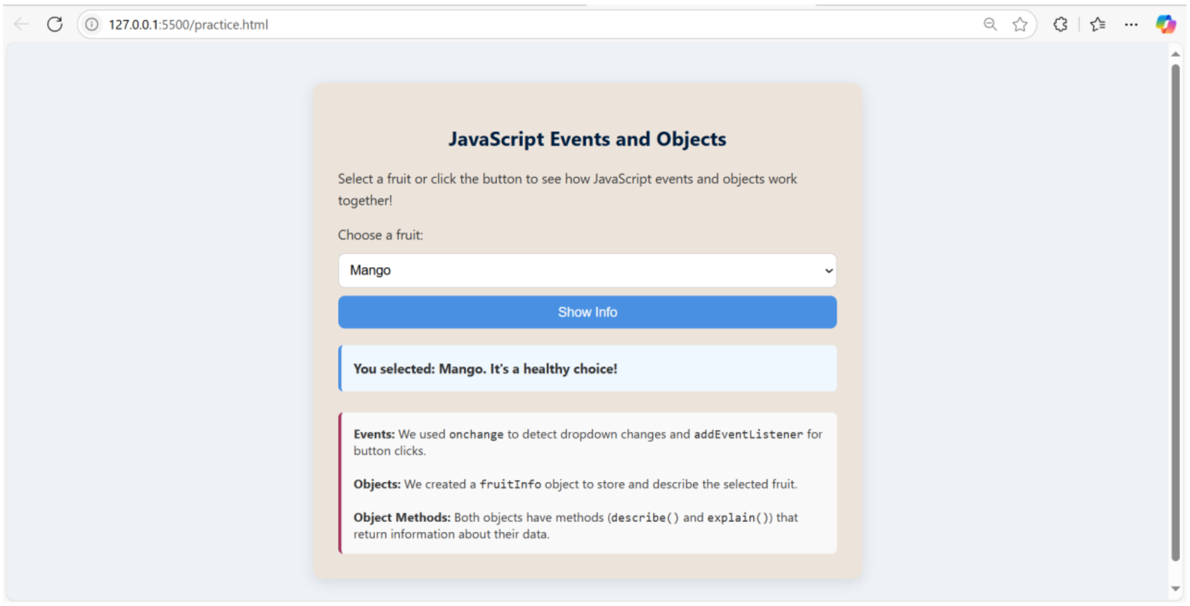
practice.html > html > body > div.container > p

```
<html>
<body>
  <div class="container">

    <p>Select a fruit or click the button to see how JavaScript events and objects work together!</p>
    <label for="fruit">Choose a fruit:</label>
    <select id="fruit">
      <option value="Apple">Apple</option>
      <option value="Banana">Banana</option>
      <option value="Mango">Mango</option>
    </select>

    <button id="showInfoBtn">Show Info</button>

    <p id="output"></p>

    <div class="theory" id="theoryBox"></div>
  </div>

  <script>
    // Object to hold fruit info
    let fruitInfo = {
      name: "",
      describe: function() {
        return "You selected: " + this.name + ". It's a healthy choice!";
      }
    };

    // Object for showing theory explanation
    let Topic = {
```

Ln 79, Col 102 Spaces: 4 UTF-8 CRLF {} HTML Port : 5500



```
<html>
<body>
  <script>
    // Object for showing theory explanation
    let Topic = {
      title: "What's Happening Here?",
      explain: function() {
        return `
          <strong>Events:</strong> We used <code>onchange</code> to detect dropdown changes and <code>addEventListene
          <strong>Objects:</strong> We created a <code>fruitInfo</code> object to store and describe the selected fru
          <strong>Object Methods:</strong> Both objects have methods (<code>describe()</code> and <code>explain()</co
        `;
      }
    };
    // onchange event to update fruit name
    document.getElementById("fruit").onchange = function() {
      fruitInfo.name = this.value;
      document.getElementById("output").innerText = "Fruit changed to: " + fruitInfo.name;
    };

    // click event using addEventListener
    document.getElementById("showInfoBtn").addEventListener("click", function() {
      document.getElementById("output").innerText = fruitInfo.describe();
    });

    // show theory on page load using object method
    document.getElementById("theoryBox").innerHTML = Topic.explain();
  </script>
</body>
</html>
```



JavaScript Events and Objects

Select a fruit or click the button to see how JavaScript events and objects work together!

Choose a fruit:

Mango

Show Info

You selected: Mango. It's a healthy choice!

Events: We used `onchange` to detect dropdown changes and `addEventListener` for button clicks.

Objects: We created a `fruitInfo` object to store and describe the selected fruit.

Object Methods: Both objects have methods (`describe()` and `explain()`) that return information about their data.

Conclusion:

Today's learning helped me understand how JavaScript brings interactivity to web pages through **event handling** and how **objects** organize and encapsulate data and functionality. These concepts are foundational for building responsive front-end applications.