Lesson Content

OT Reading Material

VI Video Course (Lessons 8 - 9)

JavaScript: Objects, Document Object Model (DOM)

1. Introduction to Objects

- **Explain:** Objects in JavaScript are collections of key-value pairs. Each key (also known as a property) has a value, which can be any valid JavaScript type, including functions.

- **Demo:** Create an object representing a person with properties for `name`, `age`, and a method for greeting.

```
const person = {
  name: 'John',
  age: 30,
  greet: function() { console.log('Hello!'); }
};
```

- **Explain:** The object `person` has properties `name` and `age` as well as a method `greet`.

2. Accessing and Modifying Object Properties

- **Explain:** We can access properties of an object using dot notation or bracket notation.

- **Demo:** Access object properties using dot notation:

```
console.log(person.name);  // Output: John
console.log(person.age);   // Output: 30
```

  - **Explain:** Dot notation is used when the property name is a valid JavaScript identifier.

- **Demo:** Access object properties using bracket notation:

```
console.log(person['name']);  // Output: John
console.log(person['age']);   // Output: 30
```

  - **Explain:** Bracket notation is useful when the property name is dynamic (stored in a variable) or not a valid identifier.

- **Modifying Properties:**
  Change the `name` and `age` properties of the `person` object:

```
person.name = 'Alice';
person['age'] = 25;
console.log(person);
```

3. Object Methods

- **Explain:** Methods are functions that are part of an object. They can be used to perform actions related to the object.

- **Demo:** Call the `greet` method of the `person` object:

```
person.greet();  // Output: Hello!
```

- **Modify the greet method:**
  Modify the `greet` method to accept a parameter and print a personalized greeting.

```
person.greet = function(name) {
  console.log(`Hello, ${name}!`);
};
person.greet('Bob');  // Output: Hello, Bob!
```

4. Introduction to the DOM (Document Object Model)

- **Explain:** The DOM represents the structure of an HTML document as an object, allowing JavaScript to interact with and manipulate HTML elements.

- **Demo:** Show a simple HTML structure:

```
<div id="myDiv">Hello World!</div>
```

- **Explain:** We can manipulate this `div` element using JavaScript by interacting with the DOM.

5. Selecting Elements with JavaScript

- **Explain:** The `document` object in JavaScript provides methods to interact with the HTML document.

- **Demo:** Use `document.getElementById` to select an element by its ID:

```
const element = document.getElementById('myDiv');
console.log(element);  // Logs the div element
```

- **Demo:** Use `document.querySelector` to select the first matching element:

```
const button = document.querySelector('.myButton'
console.log(button);  // Logs the first matching
```

6. Modifying Element Content and Style

- **Explain:** You can change the content or style of an element selected from the DOM.

- **Demo:** Modify the content of an element:

```
element.textContent = 'Hello, Universe!';
```

- **Demo:** Modify the element's style:

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

7. Adding Event Listeners

- **Explain:** Event listeners allow JavaScript to respond to user interactions, such as clicks or key presses.

- **Demo:** Create a button in HTML and add a click event listener that changes the text of an element:

```
<button id="myButton">Click Me!</button>
<p id="message"></p>
```

```
const button = document.getElementById('myButton'
button.addEventListener('click', function() {
  const message = document.getElementById('messag
  message.textContent = 'Button clicked!';
});
```

8. Dynamic DOM Manipulation

- **Explain:** JavaScript can be used to add or remove elements dynamically from the DOM.

- **Demo:** Create a new list item and add it to an existing list when a button is clicked:

```
<ul id="itemList">
  <li>Item 1</li>
  <li>Item 2</li>
</ul>
<button id="addItem">Add Item</button>
```

```
const button = document.getElementById('addItem')
button.addEventListener('click', function() {
  const list = document.getElementById('itemList'
  const newItem = document.createElement('li');
  newItem.textContent = 'New Item';
  list.appendChild(newItem);
});
```

Exercises

Instructions for Students:

I.

Exercise 1:

Create an object representing a car with properties for `make`, `model`, `year`, and a method `getInfo` that logs the car's details.

II.

Exercise 2:

Create a `student` object with properties for `name`, `age`, and `grades` (array). Write a method that calculates the average grade.

III.

Exercise 3:

Using the DOM, select an element by its ID and change its background color when a button is clicked.

IV.

Exercise 4:

Add a click event listener to a button that changes the text of a paragraph when clicked.

V.

Exercise 5:

Create an HTML page with a list of items. Write a JavaScript function that adds a new item to the list when a button is clicked.

Mark Lesson As Complete

Cape - Mark Complete