orange™ أكاديمية البرمجة
Coding Academy
powered by Simplon

JavaScript Objects

Duration: 4 Hour

Basic

1-JavaScript objects

1. **Create an object with properties name, age, and gender and print its values to the console.**

```
Example:
   Input: { name: "Adam", age: 25, gender: "male" };
   Output: Adam 25 male
```

2. **Create an object and add a new property to it using dot notation.**

```
Example:
      Input: { name: "Adam", age: 25 }, gender: "male";
      Output: { name: "Adam", age: 25, gender: "male" }
```

3. **Create an object and add a new property to it using bracket notation.**

```
Example:
      Input: { name: "Adam", age: 25 }, "gender": "male";
      Output: { name: "Adam", age: 25, gender: "male" }
```

4. **Access the value of a property in an object using dot notation.**

```
Example:
      Input: { name: "Adam", age: 25 }, "name";
      Output: "Adam"
```

5. **Access the value of a property in an object using bracket notation.**

```
Example:
      Input: { name: "Adam", age: 25 }, "name";
      Output: "Adam"
```

Mid

6. **Use a for-in loop to iterate through the properties of an object.**

```
      Example:
      Input: { name: "Adam", age: 25, gender: "male" };
      Output: name: Adam, age: 25, gender: male
```

7. **Use the Object.keys() method to get an array of an object's properties.**

```
Example:
      Input: { name: "Adam", age: 25, gender: "male" };
      Output: ["name", "age", "gender"]
```

8. **Use the Object.values() method to get an array of an object's values.**

```
   Example:
       Input: { name: "Adam", age: 25, gender: "male" };
       Output: ["Adam", 25, "male"]
```

9. **Use the Object.entries() method to get an array of key-value pairs for an object.**

```
Example:
        Input: { name: "Adam", age: 25, gender: "male" };
        Output: [["name", "Adam"], ["age", 25], ["gender", "male"]]
```

Advanced

10. Use the Object.assign() method to merge two objects.

```
Example:
        Input: { name: "Adam", age: 25 }, { gender: "male" };
        Output: { name: "Adam", age: 25, gender: "male" }
```

11. Use the Object.freeze() method to prevent changes to an object.

```
Example:
      Input: { name: "Adam", age: 25 };
      Output: { name: "Adam", age: 25 }
```

12. Use the Object.seal() method to prevent changes to an object's properties but allows changes to its values.

```
 Example:
     Input: { name: "Adam", age: 25 };
     Output: { name: "Adam", age: 25 }
```

13. create a simple To-Do List application with the following:

The HTML structure provides the layout for the To-Do List application. It includes:

- An input field for entering tasks.
- A button for adding tasks to the list.
- An unordered list for displaying tasks.

Create a Class **ToDo,** The class will handle the logic for adding, completing, and deleting tasks.

- Class Properties:
  - todoList: References the unordered list element where tasks will be displayed.
  - todoInput: References the input element for entering tasks.
  - addButton: References the "Add Task" button.
- Class Methods:
  1. addTask(): Adds a new task to the list with "**Complete**" and "**Delete**" buttons.
  2. toggleComplete(task): Toggles the "**completed**" state of a task (complete button make the task color gray with line through).
  3. deleteTask(task): Removes a task from the list.