JS Objects - Practice Questions

1. Create a Basic Object

Question: Create an object representing a person with properties: `name`, `age`, and `city`.

Expected Output:

```
{ name: "John", age: 25, city: "New York" }
```

2. **Access Object Properties (Read Operation)**

Question: Retrieve the `age` from the given object.

Input:

```
const student = { name: "Alice", age: 20, course: "Math" };
```

Expected Output:

```
20
```

3. **Add a New Property (Insert Operation)**

Question: Add a new property `country: "USA"` to the given object.

Input:

```
const user = { name: "Bob", age: 30 };
```

Expected Output:

```
{ name: "Bob", age: 30, country: "USA" }
```

4. Update an Object Property

Question: Change the `city` of the given object from `"Delhi"` to `"Mumbai"`.

Input:

```
const employee = { name: "Ravi", city: "Delhi" };
```

Expected Output:

```
{ name: "Ravi", city: "Mumbai" }
```

5. **Delete a Property (Delete Operation)**

Question: Remove the `course` property from the given object.

Input:

```
const student = { name: "Emily", age: 22, course: "Physics" };
```

Expected Output:

```
{ name: "Emily", age: 22 }
```

6. Loop through an Object's Keys and Values

Question: Print all keys and values of the given object using a loop.

Input:

```
const product = { name: "Laptop", price: 1000, brand: "Dell" };
```

Expected Output:

```
name: Laptop
price: 1000
brand: Dell
```

7. Merge Two Objects using `Object.assign()`

Question: Merge two given objects into a new object.

Input:

```
const obj1 = { a: 1, b: 2 };
const obj2 = { c: 3, d: 4 };
```

Expected Output:

```
{ a: 1, b: 2, c: 3, d: 4 }
```

8. Seal an Object using `Object.seal()`

Question: Seal an object and try adding a new property.

Input:

```
const car = { brand: "Toyota", model: "Camry" };
```

Expected Output:

```
{ brand: "Toyota", model: "Corolla" }
```

9. Extract All Keys using `Object.keys()`

Question: Retrieve all keys of a given object.

Input:

```
const car = {
    brand: "Tesla",
    model: "Model S",
    year: 2023,
    color: "Red"
};
```

Expected Output:

```
[ "brand", "model", "year", "color" ]
```

10. Store Multiple Objects in an Array

Question: Create an array of objects representing multiple products.

Expected Output:

```
[
    { id: 1, name: "Laptop", price: 1200 },
    { id: 2, name: "Smartphone", price: 800 },
    { id: 3, name: "Tablet", price: 600 }
]
```

JS Objects - Assignment Questions

1. Access Deeply Nested Properties

Question: Retrieve the `city` where the company `TechCorp` is located.

```
const companies = [
  {
    id: 1,
    name: 'TechCorp',
    details: {
      address: {
        street: '123 Elm St',
        city: 'New York',
      },
    },
  },
  {
    id: 2,
    name: 'SoftSystems',
    details: {
      address: {
        street: '456 Pine St',
        city: 'San Francisco',
      },
    },
  },
];
```

```
Expected Output: 'New York'
```

2. Update a Nested Property

Question: Update the `salary` of the employee `Bob` to `75000`.

```
const employees = [
  {
    id: 1,
    name: 'Alice',
    details: {
      salary: 50000,
      role: 'Developer',
    },
  },
  {
    id: 2,
    name: 'Bob',
    details: {
      salary: 60000,
      role: 'Manager',
    },
  },
];

Expected Output:

[
  { id: 1, name: 'Alice', details: { salary: 50000, role: 'Developer' } },
  { id: 2, name: 'Bob', details: { salary: 75000, role: 'Manager' } },
]
```

3. Add a New Item to a Nested Array

Question: Add `'Node.js'` to the `skills` array of the developer `Alice`.

```
const team = [
  {
    name: 'Alice',
    skills: ['HTML', 'CSS', 'JavaScript'],
```

```
  },
  {
    name: 'Bob',
    skills: ['Java', 'Spring Boot'],
  },
];
```

```
Expected Output:
[
  { name: 'Alice', skills: ['HTML', 'CSS', 'JavaScript', 'Node.js'] },
  { name: 'Bob', skills: ['Java', 'Spring Boot'] },
]
```

4. Filter by Nested Property

Question: Retrieve all tasks assigned to the employee `Alice`.

```
const projects = [
  {
    project: 'Website',
    tasks: [
      { task: 'Design', assignedTo: 'Alice' },
      { task: 'Code', assignedTo: 'Bob' },
    ],
  },
  {
    project: 'App',
    tasks: [
      { task: 'Develop', assignedTo: 'Alice' },
      { task: 'Test', assignedTo: 'Charlie' },
    ],
  },
];

Expected Output:
[
  { task: 'Design', assignedTo: 'Alice' },
  { task: 'Develop', assignedTo: 'Alice' },
]
```

5. Access Dynamic Properties

Question: Access the `role` of `Charlie` dynamically using the `key` variable.

```
const team = {
  Alice: { role: 'Developer', age: 25 },
  Bob: { role: 'Manager', age: 30 },
  Charlie: { role: 'Tester', age: 28 },
};
const key = 'Charlie';

Expected Output: 'Tester'
```