Objects

Question 1: What is an object in JavaScript? How are objects different from arrays?

Ans.

In JavaScript, an object is a dynamic data structure that stores collections of data as key-value pairs. Each key, also known as a property name, uniquely identifies its corresponding value. These values can be of any data type, including primitives (strings, numbers, booleans, etc.), other objects, or even functions (which are then referred to as methods when part of an object). Objects are used to represent entities with various characteristics and behaviors.

For example, a person object might have properties like firstName, lastName, and age, along with a method like greet().

```
const person = {
 firstName: "John",
lastName: "Doe",
 age: 30,
greet: function() {
console.log("Hello, my name is " + this.firstName);
}
 };
```

P.T.O

✓ Difference Between JavaScript Arrays and Objects :

Feature	JavaScript Arrays	JavaScript Objects
Index Type	Numeric indexes (0, 1, 2, ...)	Named keys (strings or symbols
Order	Ordered collection	Unordered collection
Use Case	Storing lists, sequences, ordered data	Storing data with key-value pairs, attributes
Accessing Elements	Accessed by index (e.g., arr[0])	Accessed by key (e.g., obj["key"])
Iteration	Typically iterated using loops like for or forEach	Iterated using for...in, Object.keys(), or Object.entries()
Size Flexibility	Dynamic, can grow or shrink in size	Dynamic, can add or remove keyvalue pairs

---

Question 2: Explain how to access and update object properties using dot notation and bracket notation.

Ans.

In JavaScript, object properties can be accessed and updated using two primary methods: dot notation and bracket notation.

❖Dot Notation :

Dot notation is generally preferred for its readability and simplicity when property names are known and are valid JavaScript identifiers (e.g., they don't contain spaces, hyphens, or start with numbers). Accessing a property.

```
const person = {
name: "Alice",
```

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

Updating a property.

```
const person = {
name: "Alice",
age: 30
};
person.age = 31;
console.log(person.age); // Output: 31
```

❖Bracket Notation :

Bracket notation offers greater flexibility, allowing access to properties with names that are not valid JavaScript identifiers (e.g., containing spaces or hyphens) or when the property name is determined dynamically at runtime (e.g., stored in a variable). Accessing a property.

```
const car = {
"model name": "Mustang",
year: 1969
};
car["model name"] = "Explorer";
console.log(car["model name"]); // Output: Mustang
```

Dynamic property access :

```
const user = {
 firstName: "Bob",
 lastName: "Smith"
};
 const propertyName = "firstName";
console.log(user[propertyName]); // Output: Bob
```

Differences:

• Dot notation: Use when property names are static and valid JavaScript identifiers.

• Bracket notation: Use when property names contain special characters, spaces, or need to be dynamically determined using variables or expressions.
