Day-10

JavaScript Output Based Question

github → rajeshjha2000

(47) `const set = new Set([1,2,3,4]);`
`console.log(set);`

Ans) • A 'set' is a collection of unique values. It does not allow duplicate elements.

• When we create a 'set' with the array [1,2,3,4] it adds each element of the array to the 'set'.

• Since all the elements in [1,2,3,4] are unique, they will all be added to the 'set'.

Therefore, the output will be →

• Set(4) {1, 2, 3, 4}

ex → `const set = new Set([1,1,2,3,4,4]);`

console → Set(4) {1,2,3,4}

(48) `let data = {name: "Rajesh"};`
`console.log(delete data.name);`

Ans) 1. 'delete' Operator →

• The 'delete' operator in JS is used to remove a property from an object.

• In this case, 'delete data.name' attempts to delete the 'name' property from the 'data' object.

2. Behavior of 'delete' :→

- If the property exists and is successfully deleted, the 'delete' operator return 'true'.
- After deleting the 'name' property no longer exists on the 'data' Object.

Therefore the output will be →

- true

(49) const data = {name: "Rajesh"};
console.log(delete data);

Ans:- 1. 'delete' Operator →

- The 'delete' operator is used to remove properties from objects, not variable.
- We can use 'delete' to remove a property like 'data.name' but trying to delete the entire 'data' object (or any variable) won't work as intended.

2. Behavior of 'delete' on Variables. →

- In strict mode, attempting to delete a variable (like 'data') will throw an error.
- In non-strict mode, 'delete data' will simply return 'false' because variables declared with 'var', 'let', or 'const' can't be deleted.
- The 'data' variable remains unchanged after the 'delete' operation.

Therefore, the output will be →

- false

(50) `const data = ["Rajesh", "Kumar", "Jha"];`
`const [y] = data;`
`console.log(y);`

Ans:- 1. Array Destructuring →

- Array Destructuring allows us to unpack values from arrays into distinct variables.
- In the line 'const [y] = data'; we are destructuring the 'data' array and assigning the first elements of the array to the variable 'y'.

2. Value Assignment →

- The 'data' array contains '["Rajesh", "Kumar", "Jha"]'.
- When we destructure 'data' as 'const [y] = data;' the first element "Rajesh" is assigned to 'y'.

3. Why only the first value →

- The square bracket '[]' in the destructuring pattern correspond to the position in the array.
- Since we are only specifying a single variable 'y', only the first element of the array is captured. The other elements are ignored.

Therfore, the output will be →

- Rajesh

Note → If we wanted to Capture more elements, we could extend the destructuring pattern.

```
const [x, y, z] = data;
console.log(x); // "Rajesh"
console.log(y); // "Kumar"
console.log(z); // "Jha"
```

Note → How to get 2nd value without getting first value?

```
const [, y] = data
```

↓

```
const [ , y] = data
console.log(y); → // "Kumar"
```

For more questions, visit →

gitHub → rajeshjha2000