Day-9 | JavaScript Output Based Question

gitHub → rajeshjha2000

(42) let a = [];
let b = a;
console.log(a == b);
console.log(a === b);

A> 1. let b = a :→

- This line assign the reference of the array 'a' to the variable 'b'. This means that both 'a' and 'b' are pointing to the same array in memory.

2. 'a == b' :→

- The '==' operator checks for equality of values, but when dealing with objects (including arrays), it checks whether the two variables point to the same object in memory.

3. 'a === b' :→

- The '===' operator checks for both value equality and type equality (strict equality)

Therefore, the output will be →

- true
- true

(43) let a = [20];
let b = [20];
console.log(a[0] == b[0]);
console.log(a[0] === b[0]);

A>1. Dec. of 'a' and 'b' →

- 'let a = [20];' creates an array 'a' with one element, the number '20'.
- same goes for b.
- Both 'a' and 'b' are separate arrays, each containing the value '20' as their first element.

2. 'a[0] == b[0]' : →

- It compares the value '20' from 'a[0]' with the value '20' from 'b[0]'.

3. 'a[0] === b[0]' : →

- They are both of the same ('number').

Therefore, the output will be →

- true
- true

(44) let z = [1, 2, 3, 4];
let a = {name: "Rajesh"};
console.log(...z);

A> Spread Operator ⟶ (...)

- The spread operator takes the elements of the array and "spreads" them out as individual arguments.

Therefore, the output will be ⟶

- 1 2 3 4

(45) console.log(typeof NaN);

A> 'NaN' is a special value in JS that represents an invalid or unrepresentable number, such as the result of a division by zero or an operation that doesn't produce a valid number.

- NaN stands for "Not-a-Number", the 'typeof' operator identifies it as a "number" in JS.
  This is because "NaN" is consider a special value of number type.

Therefore, the output will be ⟶

- "number"

(46) let data = 10 - -10;
console.log(data);

A> - The expression '10 - -10' involves subtracting '-10' from '10'.

- Since we have '- -10', the double negation cancels out, effectively turning it into '10 + 10'.

Therefore, the output will be ⟶

- 20

For more questions, visit ⟶
gitHub → rajeshjha2000