Day-6

JavaScript Output Based -Question

gitHub→rajeshjha2000

(26)
```
const numbers = [1, 2, 3];
numbers[3] = numbers;
console.log(numbers);
```

A> Setting an Element at Index 3 →

- The array 'numbers' is updated by setting the element at index '3' to reference the array itself.
- This creates a circular reference in the array.
- The array 'numbers' now looks like:

  [1, 2, 3, [1, 2, 3, [1, 2, 3, [...]]]]
- The fourth element is a reference to the array 'numbers' itself, creating a nested structure that reference itself.

Therefore, the output will be →

- [1, 2, 3, [circular]]
- The '"[circular]"' notion indicates that the array reference itself, creating a circular structure.

(27)
```
console.log(!!null);
console.log(!!"");
console.log(!!1);
```

A) 1. Double Negation Of 'null'; →

- 'null' is a falsy value in JS.
- The first negation ('!null') converts 'null' to 'true'.
- The second negation ('! true') converts 'true' to 'false'.
- Therefore, '!!null' evaluates to 'false'.

2. Double Negation Of an Empty string; →

- An empty string '""' is a falsy value in JS.
- The first negation ('!""') converts the empty string to 'true'.
- The second negation ('! true') converts 'true' to 'false'.
- Therefore, '!!""' evaluates to 'false'.

3. Double Negation Of '1'; →

- The number '1' is a truthy value in JS.
- The first negation ('! 1') converts 'false' to 'true'.
- Therefore, '!!1' evaluates to 'true'.

Therefore, the output will be →

- false
- false
- true

(28) console.log([..."anil"]);

A> • The spread syntax `...` is used to split the string `"anil"` into individual characters.

- The characters are placed into an array using the array literal syntax.

Therefore, the output will be →

- ['a', 'n', 'i', 'l']

(29) let data = 3 + 4 + '5';
console.log (typeof data);

A> • The expression `3+4+'5'` is evaluated as `7+'5'`, resulting in the string `'75'`.

- The variable `data` is assigned the string `'75'`.
- The `typeof` operator is used to determine the type of `data`.

Therefore, the output will be →

- String

(30) console.log (typeOf 3+4+'5');

A> 1. First Operation : `typeOf 3` →

- The 'typeof' operator is used to determine the type of the value `3`.
- Since `3` is a number, `typeof 3` evaluates to the string `"number"`.

concatenation Sequence:

The expression now become:

- "number" + 4 + '5'

2. Second Operation: `"number" +4`:

◦ When a string is concatenated with a number, the number is converted to a string.

- `"number" +4` results in the string `"number4"`.

3. Third operation: `"number4" + '5'`:

- When two strings are concatenated, they are combined into a single string.

- `"number4" + '5'` results in the string `"number45"`.

Therefore, the output will be ⟶

- number45

For More Questions, Visit ⟶

gitHub ⟶ rajeshjha2000