Day-4

JavaScript Output Based Question

gitHub-rajeshjha2000

(16)
```
const sum = eval('10x10+5');
console.log(sum);
```

Ans:> The 'eval' function evaluates the string `'10x10+5'` as a JavaScript expression.

- The expression evaluates to `105`.

Therefore, the output will be —

- 105

(17)
```
const Obj = {1: "a", 2: "b", 3: "c"};
Obj.hasOwnproperty("1");
Obj.hasOwnProperty(1);
```

Ans:> 1. `Obj.hasOwnProperty("1");` →

- The 'hasOwnProperty' method checks if ~~has~~ the specified property as its own (not inherited) property.

- In JS, even though the keys are defined as numbers, they are internally stored as strings.

- Therefore, checking `obj.hasOwnProperty("1")` will return `true` because the key "1" exists in the Object.

2. `Obj.hasOwnProperty(1); ⟶
- Since keys are stored as strings, checking `Obj.hasOwnProperty(1)' is equivalent to checking `Obj.hasOwnProperty("1")'.

- JS automatically converts the numeric key to a string when performing check.

Therefore, the output will be ⟶
- true
- true

(18) const obj = {a: "one", b: "two", a: "three"};
console.log(obj);

A). In JS, when an Object is defined with duplicate keys, the last key-value pair will overwrite any previous one.

- Therefore, the second assignment of 'a: "three"' will overwrite the first assignment of `a: "one"'.

Therefore, the output will be ⟶
- {a: "three", b: "two"}

(19) for(let i=1; i<5; i++) {
    if (i===3) continue;
    console.log(i);

Ans:- • The 'for' loop runs from '1' to '4'.

- The 'continue' statement skips the iteration when 'i' is '3'.

- When 'continue' statement returns true, it skips only that portion of loop.
  ex → (i===3) evaluates to true, after printing 1 & 2.

Therefore, the output will be →

- 1
- 2
- 4

(20) const foo=()⇒console.log('First');
const bar=()⇒ setTimeOut(()⇒ console.log('second'));
const baz=()⇒console.log('Third');

A:- • 'foo()' logs 'first' immediately.

- 'bar()' schedules logging ''second'' to happen after the current execution stack is cleared.
- Sets a timeout to log ''second'' to the console. The default for 'setTimeOut' is 0 miliseconds, meaning the callback is added to the event queue will be executed after the current execution stack is cleared.

- 'baz()' logs ''third'' immediately after ''first'' logs on console.

Scanned with CamScanner

Therefore, the output will be -

- First
- Third
- Second

For more questions, Visit -

gitHub → rajeshjha2000