Day 3

JavaScript Output Based Question

gitHub→rajeshjha2000

(11)
```
let name;
nmae = { }; // Typo
console.log(name);
```

Ans 1. let name →

- The Variable 'name' is declared but not initialized. It's initial Value is 'undefined'.

2. Typo Error : nmae = { } →

- Due to typo, 'nmae' is treated as a new, undeclared variable.
- JS implicitly creates a global variable 'nmae' (assuming the code is running in non-strict mode) and assigns an empty object.

Therefore, the output will be →

- Undefined

(12)
```
function fruit() {
  console.log ("Woof!");
  }
fruit . name = "apple";
console. warn(fruit());
```

Ans 1. Function Dec →

- The 'fruit' function is defined and logs "Woof!" to the console when called.

2. Adding property (fruit.name = "apple") →

- Function in JS are objects, so we can add properties to them.

Scanned with CamScanner

- The property assignment does not affect the function's behavior when it is called.

3. Calling the function: console.warn(fruit()); →

- When 'fruit()' is called, it logs ' "Woof!" ' to the console.
- The function does not have a return statement, so it returns 'undefined'.

→ • The call to 'fruit()' will log "Woof!".
  - The 'console.warn(fruit())' will log 'undefined'.

Therefore, the output will be →

- Woof!
- undefined

Q> What if we replace 'console.warn(fruit())' with 'fruit()'?

A> • It logs "Woof!" to the console.
  - Since the function does not have return statement, it implicitly return 'undefined', but this return value is not logged.

Therefore, the output will be →

- Woof !

(13)
```
function sum(a, b) {
    return a+b;
}
console.warn(sum(1, "2"));
```

1. Function Def →

- The 'sum' function takes two parameters, 'a' and 'b', and return their sum.

Scanned with CamScanner

2. Function call : console.warn(sum(1, "2")); ⟶

- The 'sum' function is called with the arguement '1' (a number) and '"2"' (a string).
- In JS, the '+' operator is used for both numeric addition and string concatenation.
- When one operand is a string, JS converts the other operand to a string and performs concatenation.
- The expression 'a + b' results in '"1" + "2"' due to type coercion.

Therefore, the output will be ⟶

- 12 → ("12")

(14)
```
let number = 0;
console.log(++number);
console.log(number++);
console.log(number);
```

Ans:-
- '++number' increments 'number' and returns the new value.
- 'number++' returns the current value of 'number' and then increments it.
- The final value of 'number' is '2' after both increments.

Therefore, the output will be ⟶

- 1
- 1
- 2

Scanned with CamScanner

(15)
```
function getAge(...args) {
  console.log(typeOf args);
}
getAge(21);
```

Ans:- • The 'getAge' function uses the rest parameter syntax ('...args').

• The 'args' parameter collects all arguement passed to the function into an array.

• The rest parameter 'args' collects the arguement '21' into an array $\longrightarrow$ [21]

• Since 'args' is an array, the 'typeOf' operator will return "Object".

Therefore, the output will be $\longrightarrow$

• Object

gitHub $\rightarrow$ rajeshjha2000

Scanned with CamScanner