socrative

Answer Key
October 25, 2023

JavaScript Functions

1.

```
function divide(numerator, denominator) {
    return numerator / denominator;
}
```

Given the function defined in the diagram, what would be logged to the console with the following code?

let answer = divide(10,2);
console.log(answer);

- A 0.2
- ✓ B 5
- C 8
- D 20
- E NaN

i In this case the division being performed is 10/2 since the numbers are evenly divisible, the result is a whole number, 5.

2.

```
function divide(numerator, denominator) {
    return numerator / denominator;
}
```

Given the function defined in the diagram, what would be logged to the console with the following code?

let answer = divide(2,0);
console.log(answer);

- A 0.2
- B 5
- ✓ C Infinity
- D 20
- E NaN
- F A division by 0 error message

i JavaScript does not perform strict integer division. Division by 0 is only illegal when using strict integer division. In this scenario, JavaScript performs floating point division. The answer of dividing by 0 in floating point math is **infinity**.

3. Parameters in JavaScript functions are always optional.

✓ **T** True

F False

i In JavaScript you can declare parameters, but they ARE NOT required. If a parameter is not passed to the function, the variable is **undefined**.

4.

```
1
2  function add(first, second) {
3      return first + second;
4  }
5
6  let answer = add(15);
7  console.log(answer);
8
```

Given the code pictured, what value would be displayed on the console?

NaN
nan
NAN

i Since we did not provide an input for the **second** parameter, it is undefined.

15 + undefined = NaN

5.

```
1
2  function divide(numerator, denominator) {
3      let answer = numerator / denominator;
4  }
5
6  divide(15, 3);
7  console.log(answer);
8
```

Given the code pictured, which line of code will cause an error?

A 2

B 3

C 6

✓ **D** 7

E NONE - the code will run without errors

i The variable **answer** is declared inside of the **divide()** function, and therefore is only *in scope* within that function.

Line 7 will throw an exception, because **answer** is not *in scope*, and does not exist at that point in the code.

6.

```
1
2  function add(first = 0, second = 0) {
3      return first + second;
4  }
5
6  let answer = add(15);
7  console.log(answer);
8
```

Given the code pictured, what value would be displayed on the console?

15

i Since we did not provide an input for the **second** parameter, it is assigned a default value of **0**.

15 + 0 = 15

7.

```
1
2  let answer;
3
4  function multiply(first = 1, second = 1) {
5      answer = first * second;
6  }
7
8  multiply(15);
9  console.log(answer);
10
```

Given the code pictured, which line of code will cause an error?

A 2

B 5

C 8

D 9

✓ E NONE - the code will run without errors

i The variable **answer** is declared as a **global** variable - this means that it is always *in scope*.

8. Comments should:

A Explain what is happening, not why

✓ B Explain the purpose of the code, not what it does

C Contain swear words, and lots of 'em

D Explain all the decisions that led up to how this code was created

9.

```
1  "use strict"
2  const answer = 0;
3
4  function multiply(first = 1, second = 1) {
5      answer = first * second;
6  }
7
8  multiply(15);
9  console.log(answer);
10
```

Given the code pictured, which line of code will cause an error?

A 2

✓ B 5

C 8

D 9

E NONE - the code will run without errors

i The variable **answer** is declared as a **constant**. Line 5 will throw an exception because the value of the constant **answer** cannot be changed.