Clarusway

Backend Workshop -2-

Workshop

Subject:

- JS Recap

Learning Goals

- Reviewing and remembering our basic JavaScript knowledge.

Introduction

- Let's refresh our knowledge on Js.

Prerequisites

We will use the VSCode you are familiar with. At the same time, we need to install Nodejs on our computer.

Lets start

1. What is the output of the following code block ?

```
console.log(0.1 + 0.2);
console.log(0.1 + 0.2 == 0.3);
```

Answer:

```
0.30000000000000004
False
```

An educated answer to this question would simply be: “You can’t be sure. it might print out 0.3 and true, or it might not. Numbers in JavaScript are all treated with floating point precision, and as such, may not always yield the expected results.”

2. What is the output of the following code block ?

```
console.log(1 < 2 < 3);
console.log(3 > 2 > 1);
```

Answer:

```
True
False
```

The first statement returns true which is as expected. The second returns false because of how the engine works regarding operator associativity for < and >. It compares left to right, so 3 > 2 > 1 JavaScript translates to true > 1. true has value 1, so it then compares 1 > 1, which is false.

3. Write program to find the sum of positive numbers. But if the user enters a negative numbers, the loop ends, if the negative number entered is not added to sum

Answer:

```
const prompt = require('prompt-sync')();
let sum = 0;
let number = parseInt(prompt('Enter a number: '));
while(number >= 0) {
        sum += number;
        number = parseInt(prompt('Enter a number: '));
```

```
}
console.log(`The sum is ${sum}.`);
```

4. What is the output of the following code block ?

```
null == undefined
null === undefined
isNaN(2 +  null)
isNaN(2 + undefined)
null ? console.log("true") : console.log("false")
```

Answer:

```
true
false
false
true
false
//Null is also referred as false.
```

5. What is the output of the following code block ?

```
var hash = "";
var count = 1;
var n = 3;
for (var x = 1; x <= 7; x++) {
  while (hash.length != count)
    hash += "#";
  hash += "\n";
  count += n;
  n++;
}
console.log(hash);
```

Answer:

```
#
##
###
####
#####
######
#######
```

6. What is the output of the following code block ?

```
let firstName = null
let lastName = null
let nickName = "coderBond"
console.log(firstName ?? lastName ?? nickName ?? "Anonymous")
```

Answer:

```
// shows the first defined value:
coderBond
```

7. What is the output of the following code block ?

```
function onZoom(x){
    console.log("Zoom active for", x)
}

function startClass(x,y,z){
    console.log(" Class starts at", x);
    y(z);
}
startClass("20:00",onZoom,"FS");
```

Answer:

```
Class starts at 20:00
Zoom active for FS
```

8. What is the output of the following code block ?

```
console.log
((function f(n){return ((n > 1) ? n * f(n-1) : n)})(5));
```

Answer:

```
720
```

The code will output the value of 6 factorial (i.e., 6!, or720).
f(1): returns n, which is 1
f(2): returns 2 * f(1), which is 2
f(3): returns 3 * f(2), which is 6
f(4): returns 4 * f(3), which is 24
f(5): returns 5 * f(4), which is 120
f(6): returns 6 * f(5), which is 720

The named function f() calls itself recursively, until it gets down to calling f(1) which simply returns 1. Here, therefore, is what this does:

9. What is the output of the following code block ?

```
(function () {
    try {
        throw new Error();
    } catch (x) {
        var x = 1, y = 2;
        console.log(x);
    }
    console.log(x);
    console.log(y);
})();
```

Answer:

```
1
undefined
2
```

10. What is the output of the following code block ?

```
let a = [10, 20, 30];
a[10] = 100;
console.log(a[6]);
let b = [undefined];
b[2] = 1;
console.log(b);
console.log(b.map(e => 99));
```

Answer:

```
undefined
[undefined, <1 empty item>, 1]
[99,  <1 empty item>, 99]
```

11. What is the output of the following code block ?

```
function orderPizza(type, ingredients, callback) {
    console.log('Pizza ordered...');
    console.log('Pizza is for preparation');
    setTimeout(function () {
        let msg = `Your ${type} ${ingredients} Pizza is ready! The total bill is
$10`;
        callback(msg);
    }, 3000);
}
orderPizza('Vegeterian', 'Cheese', function(message){
  console.log(message);
});
```

Answer:

```
Pizza ordered...
Pizza is for preparation
Your Vegeterian Cheese Pizza is ready! The total bill is $10
```

The setTimeout function demonstrates that the pizza preparation takes some time. We log a message in the console after the pizza is ready. The message gets logged but no clue about it. We need to notify people saying the pizza is ready. We need to introduce a callback function now to let people know about the status of the pizza. Let's change the orderPizza function to pass a callback function as an argument. Also notice that we are calling the callback function with the message when the pizza is ready

12. What is the output of the following code block ?

```
class Employee{
    constructor(id,name){
        this.id=id;
        this.name=name;
    }
    detail(){
        console.log(this.id+" "+this.name)
    }
  }
let e1=new Employee(10,"Qadir Adamson");
let e2=new Employee("Victor Hug");
let e3=new Employee(12)
e1.detail();
```

```
e2.detail();
e3.detail();
```

Answer:

```
10 Qadir Adamson
Victor Hug undefined
12 undefined
```

13. What is the output of the following code block ?

```
class Animal {
    constructor(name, weight) {
        this.name = name;
        this.weight = weight;
    }
    eat() {
        return `${this.name} is eating`;
    }
    sound(){
        return `${this.name} is says`;
    }
}
class Cat extends Animal {
    constructor(name, weight) {
        super(name, weight);
    }
    sound(){
        return `${super.sound()}  Meow!`;
    }
}

let felix=new Cat("felix",5)
console.log(felix.sound())
```

Answer:

```
felix is says  Meow!
```

☺ **Thanks for Attending** ✍

Clarusway