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!
```

Backend
Teamwork
-2-

Teamwork

Subject: OOP and Nodejs

Learning Goals

- Having knowledge about backend and nodejs.
- Understand how real-life entities/objects can be transferred to the computer environment.

Introduction

As developers, we should also be able to express what we know. This study was prepared to support this purpose. Enjoy your work.

Practice Using the `IDE` in Lesson

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

1. In the table below specifies the key differences between a front-end and back-end development. Indicate at the beginning of each item whether it is suitable for the front end or the back end.

.....is refers to the client-side of an application.

.....includes everything that attributes to the visual aspects of a web application.

.....refers to the server-side of an application.

..... technologies are HTML, CSS, Bootstrap, jQuery, JavaScript, AngularJS, and React.js.

.....generally includes a web server that communicates with the database to serve the users' requests.

..... some framework examples are AngularJS, React.js, jQuery, Sass, etc.

.....consists of everything that happens behind the scenes and users cannot see and interact with.

.....is the part of a web application where users can see and interact.

.....technologies are Java, PHP, Python, C++, Node.js, etc.

.....some framework examples are Express, Django, Rails, Laravel, Spring, etc.

Answer:

Frontend is refers to the client-side of an application.

Frontend includes everything that attributes to the visual aspects of a web application.

Backend refers to the server-side of an application.

Frontend technologies are HTML, CSS, Bootstrap, jQuery, JavaScript, AngularJS, and React.js.

Backend generally includes a web server that communicates with the database to serve the users' requests.

Frontend some framework examples are AngularJS, React.js, jQuery, Sass, etc.

Backend consists of everything that happens behind the scenes and users cannot see and interact with.

Frontend is the part of a web application where users can see and interact.

Backend technologies are Java, PHP, Python, C++, Node.js, etc.

Backend some framework examples are Express, Django, Rails, Laravel, Spring, etc.

2. Fill-in-the-Blank JavaScript Questions

JavaScript is primarily used for creating _____ behavior in web pages.

The _____ method is used to convert a JavaScript object into a JSON string.

In JavaScript, _____ is a function that calls another function after a specified number of milliseconds.

The _____ keyword is used to declare a variable that can be reassigned.

_____ is a JavaScript object used for handling dates and times.

A _____ is a function that takes another function as an argument and/or returns a function as a result.

The _____ method is used to add elements to the end of an array in JavaScript.

In JavaScript, _____ is the process of converting a string into a number.

The _____ keyword is used to create a new instance of an object in JavaScript.

JavaScript _____ is a runtime environment that allows executing JavaScript code server-side.

Answer:

1-dynamic 2-JSON.stringify 3-setTimeout 4-let 5-Date 6-higher-order function 7-push 8-parsing 9-new 10-Node.js

3. What is Nodejs? What can we do with Nodejs? Why use Nodejs?

Answer:

What; Node. js is a versatile runtime environment that allows developers to build fast and scalable network applications.

Why; The answer is simple – it's exceptional for data-intensive real-time applications, APIs, microservices, and streaming applications.

What can we do;

- Weal-time web applications
- Network applications
- Distributed systems
- General purpose applications

4. Lets try to write code axplanid above

- Define a class named "person".
- Let this class have a constructor function that takes a "name" parameter,
- I can learn the name of the object with the function called "sayName".
- Define a class named "BankAccount" derived from the Person class.
- Let it have a private property named "balance".
- Let's have a constructor that takes "name" and "balance" parameters.
- Money can be added to the acount with a method called "deposit".
- Money can be withdrawn from the account with a method called "withdraw".
- I can find out the total amount with the function called "balance".
  Answer:

```
class Animal:
    def __init__(self, species):
        self.species = species

    def get_species(self):
```

```
        return self.species


class Dog(Animal):
    def __init__(self, species, name):
        super().__init__(species)
        self.__name = name

    def get_name(self):
        return self.__name

    def bark(self):
        return "Woof!"



animal1 = Animal("Mammal")
print(animal1.get_species())  # Output: Mammal

dog1 = Dog("Canine", "Buddy")
print(dog1.get_species())  # Output: Canine
print(dog1.get_name())  # Output: Buddy
print(dog1.bark())  # Output: Woof!
```