Clarusway

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. 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

3. 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 Person {
    constructor(name) {
      this.name = name;
    }

    introduceSelf() {
      console.log(`Hello! I'm ${this.name}`);
    }
  }
class BankAccount extends Person {
    _balance=0
    constructor(name, balance) {
      super(name)
      this._balance = balance;
    }
    deposit(amount) {
      this._balance += amount;
    }
    withdraw(amount) {
      if (amount > this._balance) {
        console.log("Funds are insufficient");
      } else {
        this._balance-= amount;
      }
    }
    get balance() {
      return this.balance;
    }
}

const account = new BankAccount('tom','1800');
console.log(account._balance);
```

```
account.withdraw(500);
console.log(account._balance);
account.withdraw(1000);
console.log(account._balance);
```