THE LEAN PROGRAMMER | CodeTo Express

Day 23, Today's Topic

Classes Part 2

DAILY JS

```
/* ====================================================================== */
/* ======================== Daily JS - Day 23 ========================= */
/* ====================================================================== */

  # Day 23: Classes in JS - Inheritance

  Yesterday we started discussing about the concept of
  classes in JS, and we also discussed a little about
  prototypes.

  Today's topic is inheritance. As I told you yesterday,
  a `class` in JS is just a syntactic sugar over the
  prototype-based inheritance.

  Also, The concept of proptotypes was a little difficult to
  understand, especially for the developers who came from some
  other Object Oriented Language.

  And today you will see that the syntax of `class` made it much
  more easier for developers coming from different languages to
  work in JS.

  Let's see the syntax first and then we will look at an example..

/* ================= madhavbahl.tech/dailyjs/day23 ================== */
/* ============ Join Discord: madhavbahl.tech/discord-c2e =========== */
```

```
/* ================================================================================================ */
/* ================================== Daily JS - Day 23 ================================== */
/* ================================================================================================ */

  ## Syntax

  We use the `extends` keyword for inheritance

  ```js
    class MySuperClass {
        constructor ({ parentData }) {
            this.parentData = parentData;
        }

        parentFunction () {
            console.log ("This is from super class!");
        }
    }

    class MySubClass {
        constructor ({ childData }) {
            this.childData = childData;
        }

        childFunction () {
            console.log ("This is from child class");
        }
    }

    const subInstant = new MySubClass ({ parentData: "Super", childData: "Sub" });
    subInstant.childData ();
    subInstant.parentData ();
  ```

/* ============================== madhavbahl.tech/dailyjs/day23 ============================== */
/* ======================== Join Discord: madhavbahl.tech/discord-c2e ======================== */
```

```
/* ========================================================== */
/* ================== Daily JS - Day 23 ================== */
/* ========================================================== */

  ## Example

  Let's extend the yesterday's example too see
  how we can implement inheritance in JS Classes.

  **Make a class "Person" with a greet method,
  Make another class "Student" that extends the
  "Person" class**

  First we will see the solution using the old
  school prototype method.

  Then we will see the solution using Class Syntax.

/* ============ madhavbahl.tech/dailyjs/day23 ============== */
/* ======= Join Discord: madhavbahl.tech/discord-c2e ====== */
```

```
/* ================================================================================================== */
/* ========================================== Daily JS - Day 23 ========================================== */
/* ================================================================================================== */

  ## Solution using prototype

  ```js
    // Constructor function - pre ES6
    function Person ({ name, age, occupation }) {
        this.name = name;
        this.age = age;
        this.occupation = occupation;
    }

    // Adding methods to "Person"
    Person.prototype.greet = function () {
        return `Hi! I am ${this.name}.
            I am of  ${this.age} years old.
            I am a ${this.occupation}`;
    }

    function Student (options) {
        Person.call (this, options);
        this.marks = options.marks;
    }

    Student.prototype = Object.create (Person.prototype);
    Student.prototype.constructor = Student;

    Student.prototype.grades = function () {
        if (this.marks > 90)    return 'A';
        else if (this.marks > 80) return 'B';
        else if (this.marks > 70) return 'C';
        else if (this.marks > 60) return 'D';
        else if (this.marks > 50) return 'E';
        else    return 'F';
    }

    const john = new Student ({
        name: 'John',
        age: 20,
        occupation: 'Student',
        marks: 95
    });

    console.log (john);
    console.log (john.greet ());
    console.log ("Grade - ", john.grades ());
  ```

/* ================================ madhavbahl.tech/dailyjs/day23 ================================ */
/* ========================== Join Discord: madhavbahl.tech/discord-c2e ========================== */
```

```
/* ============================================================================================== */
/* ========================================== Daily JS - Day 23 ========================================== */
/* ============================================================================================== */

  ## Solution using Class Syntax

  Now let's see the solution using the class syntax. Now you might understand that how easy it is to
  read the code usinig the `class` syntax (especially if you are new to JS and coming from some
  other object oriented language)

  ```js
    class Person {
        constructor ({ name, age, occupation }) {
            this.name = name;
            this.age = age;
            this.occupation = occupation;
        }

        greet () {
            return `Hi! I am ${this.name}.
                I am of  ${this.age} years old.
                I am a ${this.occupation}`;
        }
    }

    class Student extends Person {
        constructor (options) {
            super (options);
            this.marks = options.marks;
        }

        grades () {
            if (this.marks > 90)    return 'A';
            else if (this.marks > 80) return 'B';
            else if (this.marks > 70) return 'C';
            else if (this.marks > 60) return 'D';
            else if (this.marks > 50) return 'E';
            else    return 'F';
        }
    }

    const john = new Student ({
        name: 'John',
        age: 20,
        occupation: 'Student',
        marks: 95
    });

    console.log (john);
    console.log (john.greet ());
    console.log ("Grade - ", john.grades ());
  ```

  That's it for today, see you tomorrow.

/* ================================ madhavbahl.tech/dailyjs/day23 ================================ */
/* ============================ Join Discord: madhavbahl.tech/discord-c2e ============================ */
```

THE LEAN PROGRAMMER | CodeTo Express

Thank you!

Feel free to reach out...

Email: theleanprogrammer@gmail.com
Web: madhavbahl.tech/
Github: github.com/MadhavBahlMD
LinkedIn: linkedin.com/in/madhavbahl/
Insta: instagram.com/theleanprogrammer/

DAILY JS