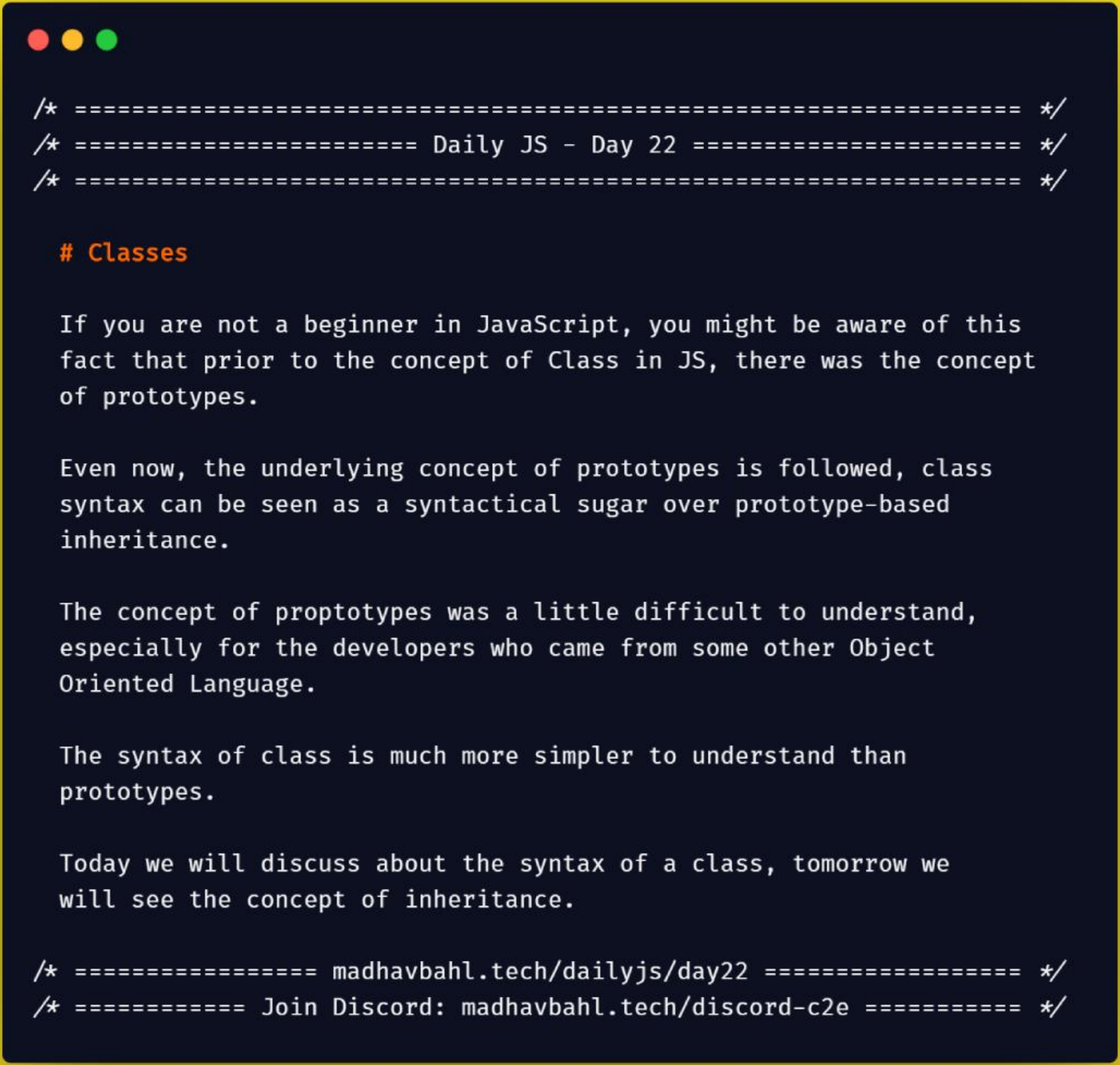
THE LEAN PROGRAMMER | Code To Express

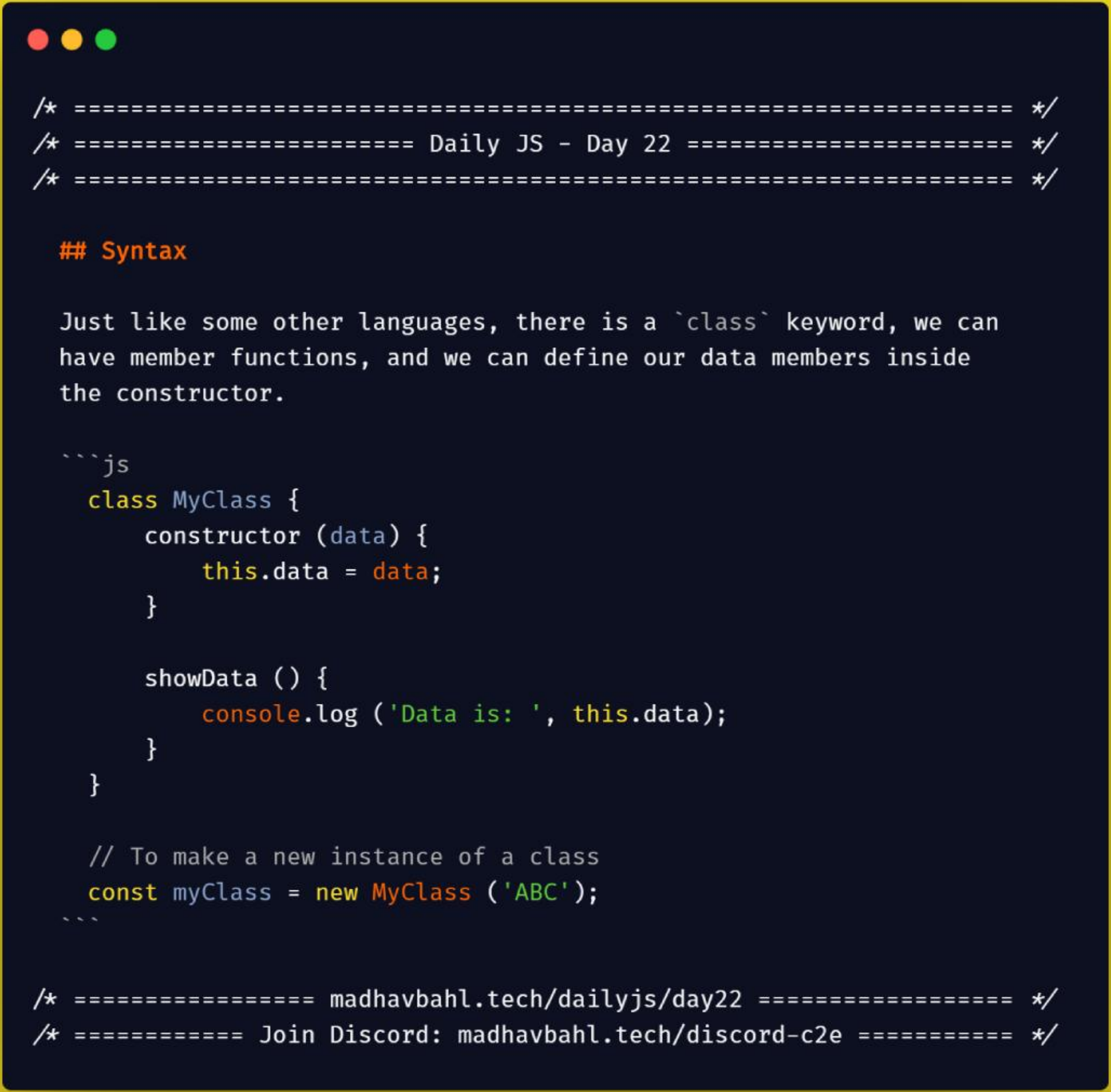
Day 22, Today's Topic

Classes Part 1

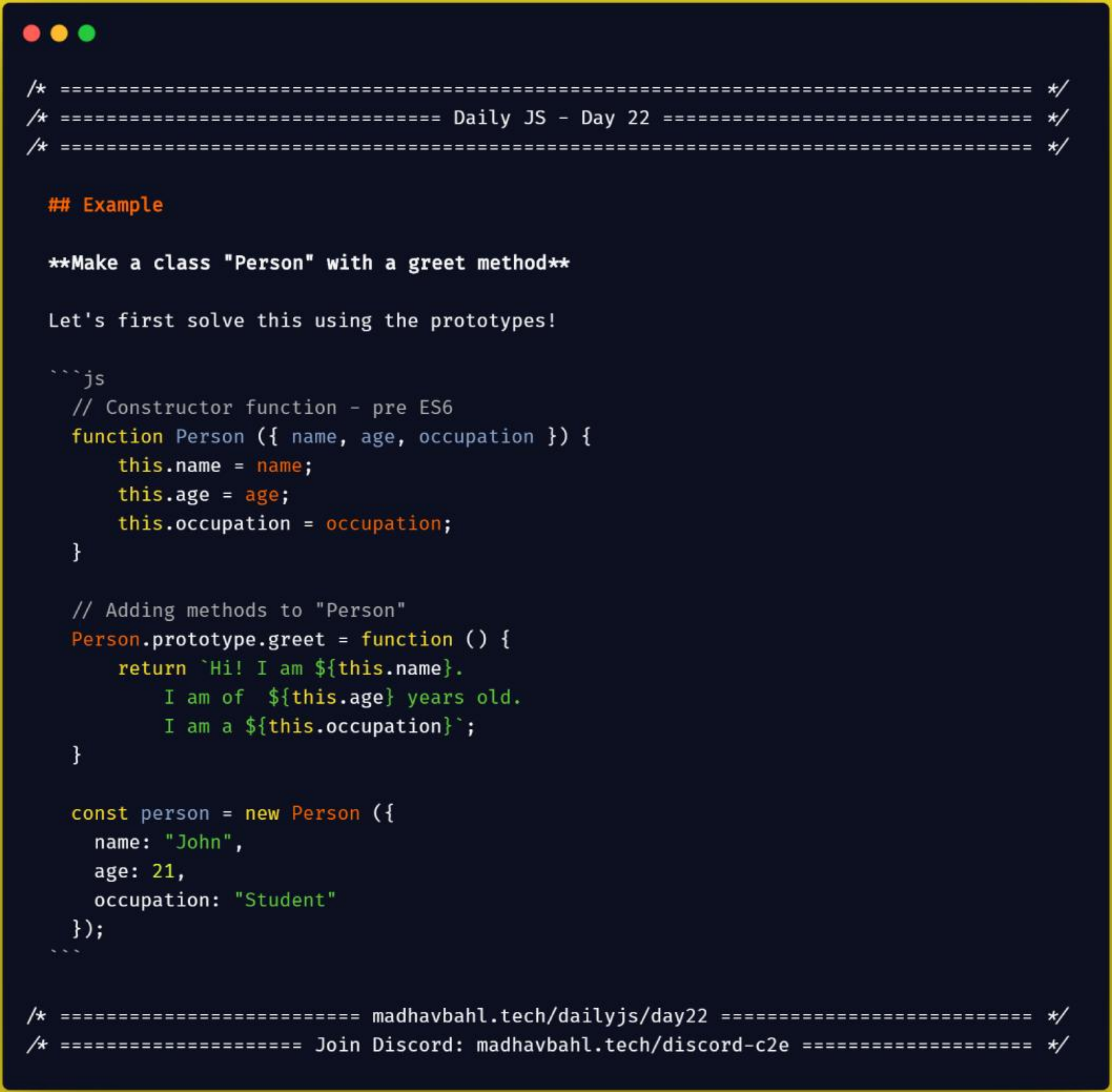
DAILY JS



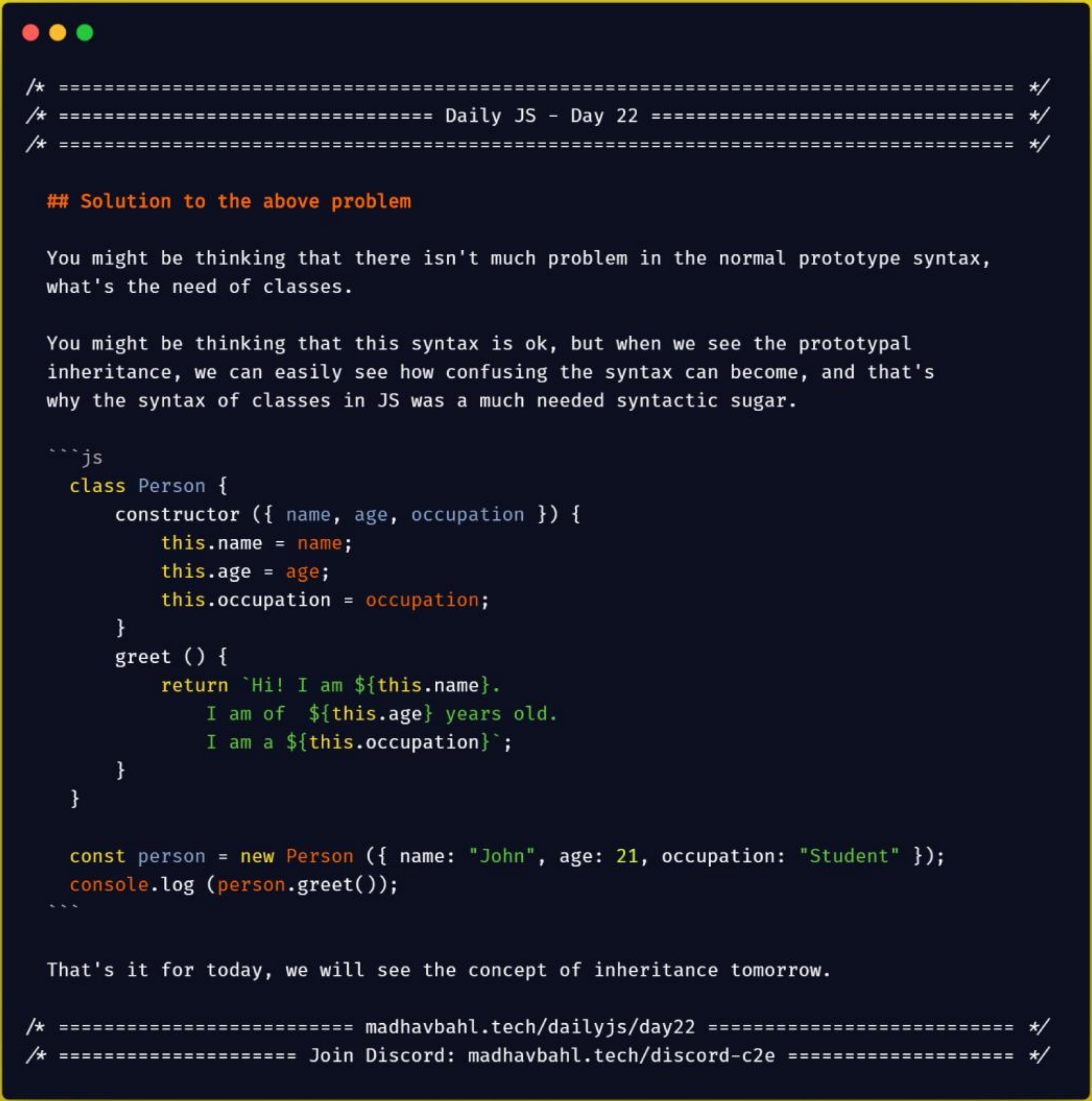
```
/* ================================================================== */
/* ======================== Daily JS - Day 22 ======================= */
/* ================================================================== */

  # Classes

  If you are not a beginner in JavaScript, you might be aware of this
  fact that prior to the concept of Class in JS, there was the concept
  of prototypes.

  Even now, the underlying concept of prototypes is followed, class
  syntax can be seen as a syntactical sugar over prototype-based
  inheritance.

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

  The syntax of class is much more simpler to understand than
  prototypes.

  Today we will discuss about the syntax of a class, tomorrow we
  will see the concept of inheritance.

/* ================= madhavbahl.tech/dailyjs/day22 ================== */
/* ============ Join Discord: madhavbahl.tech/discord-c2e =========== */
```



~~~
/* =================================================================== */
/* ======================== Daily JS - Day 22 ======================== */
/* =================================================================== */

  ## Syntax

  Just like some other languages, there is a `class` keyword, we can
  have member functions, and we can define our data members inside
  the constructor.

  ```js
    class MyClass {
        constructor (data) {
            this.data = data;
        }

        showData () {
            console.log ('Data is: ', this.data);
        }
    }

    // To make a new instance of a class
    const myClass = new MyClass ('ABC');
  ```

/* ================= madhavbahl.tech/dailyjs/day22 ================== */
/* ============ Join Discord: madhavbahl.tech/discord-c2e =========== */
~~~



~~~
/* ==================================================================================== */
/* ================================= Daily JS - Day 22 ================================ */
/* ==================================================================================== */

  ## Example

  **Make a class "Person" with a greet method**

  Let's first solve this using the prototypes!

  ```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}`;
    }

    const person = new Person ({
      name: "John",
      age: 21,
      occupation: "Student"
    });
  ```

/* ========================== madhavbahl.tech/dailyjs/day22 =========================== */
/* ===================== Join Discord: madhavbahl.tech/discord-c2e ==================== */
~~~



```
/* ==================================================================================== */
/* ================================= Daily JS - Day 22 ================================ */
/* ==================================================================================== */
```

Solution to the above problem

You might be thinking that there isn't much problem in the normal prototype syntax, what's the need of classes.

You might be thinking that this syntax is ok, but when we see the prototypal inheritance, we can easily see how confusing the syntax can become, and that's why the syntax of classes in JS was a much needed syntactic sugar.

```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}`;
      }
  }

  const person = new Person ({ name: "John", age: 21, occupation: "Student" });
  console.log (person.greet());
```

That's it for today, we will see the concept of inheritance tomorrow.

```
/* ========================== madhavbahl.tech/dailyjs/day22 =========================== */
/* ===================== 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