# TypeScript Class

**Summary**: in this tutorial, you will learn about the TypeScript Class and how to use classes create to objects.

## Introduction to the TypeScript Class

JavaScript does not have a concept of class like other programming languages such as Java and C#. In ES5, you can use a constructor function and prototype inheritance to create a “class”.

For example, to create a `Person` class that has three properties ssn, first name, and last name, you use the following constructor function:

```
function Person(ssn, firstName, lastName) {
    this.ssn = ssn;
    this.firstName = firstName;
    this.lastName = lastName;
}
```

Next, you can define a prototype method to get the full name of the person by concatenating first name and last name like this:

```
Person.prototype.getFullName = function () {
    return `${this.firstName} ${this.lastName}`;
}
```

Then, you can use the Person “class” by creating a new object:

```
let person = new Person('171-28-0926','John','Doe');
console.log(person.getFullName());
```

It would output the following to the console:

```
John Doe
```

ES6 allows you to define a class, which is simply syntactic sugar for creating constructor functions and prototypal inheritance:

```
class Person {
    ssn;
    firstName;
    lastName;

    constructor(ssn, firstName, lastName) {
        this.ssn = ssn;
        this.firstName = firstName;
        this.lastName = lastName;
    }
}
```

In the class syntax, the constructor is clearly defined and placed inside the class. The following adds `getFullName()` method to the class:

```
class Person {
    ssn;
    firstName;
    lastName;

    constructor(ssn, firstName, lastName) {
        this.ssn = ssn;
        this.firstName = firstName;
        this.lastName = lastName;
    }

    getFullName() {
        return `${this.firstName} ${this.lastName}`;
    }
}
```

Using the `Person` class is the same as the `Person` constructor function:

```
let person = new Person('171-28-0926','John','Doe');
console.log(person.getFullName());
```

TypeScript class adds type annotations to the properties and methods of the class. The following shows the `Person` class in TypeScript:

```
class Person {
    ssn: string;
    firstName: string;
    lastName: string;

    constructor(ssn: string, firstName: string, lastName: string) {
        this.ssn = ssn;
        this.firstName = firstName;
        this.lastName = lastName;
    }

    getFullName(): string {
        return `${this.firstName} ${this.lastName}`;
    }
}
```

When you annotate types to properties, constructors, and methods, the TypeScript compiler will carry the corresponding type checks.

For example, you cannot initialize the `ssn` with a `number`. The following code will result in an error:

```
let person = new Person(171280926, 'John', 'Doe');
```

## Summary

- Use `class` keyword to define a class in TypeScript.

- TypeScript leverages the ES6 class syntax and adds type annotations to make the class more robust.