JAVASCRIPT

ES6 Classes

Sreekanth M. E.

Freelance Trainer & Consultant

http://www.SreekanthME.com

JS classes are syntactical sugar over its existing prototype-based inheritance.

It provides a simpler and clearer syntax to create objects and deal with inheritance.

A class is nothing but a function constructor.

```
> typeof MyClass
function
```

But classes cannot be invoked via a function call. They have to be instantiated through new keyword only.

There are two ways to define a class:

- Class expressions
- Class declarations

Class expressions can be anonymous.

Function declarations are hoisted but class declarations are not.

Class has to be first declared before it can be instantiated. Else, code will throw error

Methods defined inside the class should not have a function keyword at the beginning.

A class can have three types of methods:

- A single Constructor method
- Prototype methods
- Static methods

The constructor method is used for creating and initializing objects created from the class.

There can be only one constructor in a class.

A constructor can use the super keyword to call the constructor of the parent class (if it has one).

Prototype methods are just normal methods.

The static keyword is used to define a static method.

Static methods can be called without instantiating the class. But cannot be called through an object of the class.

Static methods are often used to create utility functions.

The extends clause is used for sub-classing.

Class may also extend from a function constructor.

Classes cannot extend from objects.

Superclass is the prototype of a subclass.

If there is a constructor present in subclass, it needs to first call super() method before using this.

The super keyword is used to call static and prototype methods of the superclass.

super.getParentMethod()

If a constructor is not specified for a non-inheriting class, the following default constructor is used:

```
constructor() {}
```

For subclasses, the following default constructor is used:

```
constructor(...args) {
    super(...args);
}
```

In ES6, it is possible to subclass all built-in constructors (like Array, String etc.).

A class body can contain methods, but not data properties.

Properties must be implemented as setter and getter methods instead.

SETTER & GETTER SYNTAX

```
class MyClass {
    get prop() {
        return this.x;
    }
    set prop(value) {
        this.x=value;
    }
}
```

```
const obj = new MyClass();
obj.prop = 123;
const x = obj.prop;
```

JS does not support multiple inheritance.

END OF CHAPTER

APPENDIX