

JavaScript ES6

Lesson: 14

Working with Classes



Lesson Objectives

At the end of this module you will be able to:

- Explain what are classes in ES6?
- Create objects using classes
- Implement inheritance in classes
- Use the features of classes





ES6 – Class Introduction

Although it's an object-oriented programming language; JavaScript never had the concept of classes.

Programmers from the other programming language background often found it difficult to understand JavaScript's object-oriented model and inheritance due to lack of classes.

ES6 introduced classes that provide a much simpler and clearer syntax to creating constructors and dealing with inheritance.



Using Classes

ES6 classes aim to provide a much simpler and clearer syntax for dealing with the constructors and inheritance.

Classes are just a new syntax for creating functions that are used as constructors. In fact, classes are functions.

There are two ways to define a class

- **Using the class declaration** : For defining a class using the class declaration required the class keyword, and a name for the class
- **Using the class expression** : A class expression has a similar syntax to a class declaration. However, with class expressions, class name can be omitted.

ES6 class doesn't pollute the global namespace. i.e. it won't get attached to window object and classes are not hoisted.

Demo



Using-Classes





The get and set methods

Class instance members / fields can be initialized inside the constructor using ***this*** keyword.

ES6 introduced the get and set prefixes for methods, which is used to encapsulate the instance members.

When get and set methods are used in a class body, they are added to the prototype property of the class.

Demo



get-set-methods





The static method

The methods that are added to the body of the class with the static prefix are called as static methods.

Static methods are the own methods of the class, i.e. they are not added to the prototype property of the class, rather they are added to the class itself.

Static methods are often used to create utility functions for an application.

Static methods should be access through its class name.

At present we can create only static methods in order to create static members of the class it need to attached to it's class name outside the class.

Demo



static-method





Implementing inheritance in classes

ES6 implement the inheritance hierarchy in the classes with the help of extends clause, and the super keyword.

By using the extends clause, a class can inherit static and non-static properties of another constructor(which may or may not be defined using a class).

The super keyword is used in two ways:

- It's used in a class constructor method to call the parent constructor;
- When used inside methods of a class, it references the static and non-static methods of the parent constructor
- super keyword can be used with object literal as well.

If constructor is defined in child class, then super() need to be compulsorily provided even though constructor is not defined in parent class.

Demo



inheritance





The "new.target" implicit parameter

ES6 adds a parameter named `new.target` to all the functions.

The default value of `new.target` is undefined, but when a function is invoked as a constructor, the value of the `new.target` parameter depends on the following conditions

- If a constructor is invoked using a `new` operator, then `new.target` points to this constructor
- If a constructor is invoked via `super` keyword, then the value of `new.target` in it is the same as the value of `new.target` of the constructor that is called `super`.

Using `new.target` static methods can be accessed in the initial constructor call even though it is a part of the prototype chain.

Demo



new.target





Summary

ES6 classes aim to provide a much simpler and clearer syntax for dealing with the constructors and inheritance.

Classes are not hoisted.

There can only be one constructor method in a class.

Using the extends clause, a class can inherit static and non-static properties of another constructor.

ES6 implement the inheritance hierarchy in the classes with the help of extends clause, and the super keyword
super keyword is used in a class constructor method to call the parent constructor.

