

Object Oriented Programming System in Javascript

Object Oriented Programming is a programming paradigm based on the concept of objects.

We use objects to describe real world abstract features. Object may contain data (properties) and code (methods). Objects are building blocks and interact with one another (using public interface).

A class is a type of blueprint that you can use to make objects.

A concrete 'thing' that you constructed using a certain class is an object, which is an instance of a class. So, while the terms 'object' and 'instance' are interchangeable, the term 'instance' refers to an object's relationship to its class

So how to create objects from classes, there are multiple ways and 4 core principles of OOPs.

Abstraction, Encapsulation, Inheritance and Polymorphism (you can checkout basic definitions).

OOPs in Javascript works a bit differently as compared to other languages. In javascript we have something called prototypal inheritance. Objects in javascript are linked to a prototype object.

Prototypal Inheritance : The prototype contains methods (behavior) that are accessible to all objects linked to that prototype. Basically objects inherit properties from prototype (it's different from the inheritance in OOPs).

How do we actually create prototypes.

Constructor Functions

Technique to create objects from a function This is how built-in objects like arrays, maps or sets are actually implemented.

ES6 Classes

Modern alternative to constructor function syntax. But it's just a syntactic sugar. It does not work like classes in other languages.

Object.create()

Easiest and most straightforward way of linking an object to prototype object.