Constructor Function in JavaScript

Constructor functions are functions that are used to create new objects. The [new operator](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Operators/new) is used to create new instances based off a constructor function.

Here is an example showing how we can create constructor function and create an object using new operator.

function Employee (name, dep,age){

this.name = name;

this.dep = dep;

this.age = age;

}

var Brad = new Employee('Brad','QA',26);

var Mack = new Employee('Mack','Admin',32);

var aliza = new Employee('Aliza','Software',30);

# We have built in Constructor in JavaScript

Using built in constructor we can create new objects of Array (), Date (), Boolean () etc.

Ex;-

Var nums = new Array ([2, 4, 1, 0, 9, 5,] ;

Var today = new Date ();

Var status = new Boolean (true);

Var number = new Number (55);

Prototype in JavaScript

The properties and methods of an object can be shared through generalized objects that have the ability to be extended is called prototypical inheritance.

We can check the prototype of an object using “getprototypeOf ()” function.

Ex: - var arr = [1, 2, 5];

Object.getPrototypeOf (arr);

We can add our own property or function to an Object using prototype.

Below is an example that shows how we can create our own prototype properties and add to the existing object. We are considering the above Employee constructor.

**Adding a new function to existing Employee constructor.**

Employee.prototype.getDep = function ()

{return 'My Department is '+ this.dep ;}

console.log (aliza.getDep ());

**Copying the property of another constructor**

In below example we are copying the property of another constructor into newly created constructor using call () method.

function salaryBased (name, dep,age, salary)

{

Employee. call (this, name, dep,age);

this.salary = salary;

}

var devlop = new salaryBased('Abrahem','software',35,10524556);

console.log (devlop);

Prototype properties and methods are not automatically linked when you use call() to chain constructors. We have to use Object.create () to link the prototypes, making sure to put it before any additional methods are created and added to the prototype.

salaryBased.prototype = Object.create (Employee.prototype);

Now we can access the property of Employee objects through salaryBased objects because its get linked with Employee object

console.log (devlop.getDep ());

Without linking the objects if we call the function of another object will Couse an error

Uncaught TypeError: devlop.getDep is not a function