**Object.create in JavaScript**

Object.create method is another method to create new object in JavaScript.

Other methods of creating object in JavaScript has been described in the [previous article](https://medium.com/@happymishra66/create-objects-in-javascript-10924cfa9fc7). Also, go through the [prototype](https://hackernoon.com/prototypes-in-javascript-5bba2990e04b)and [inheritance](https://medium.com/@happymishra66/inheritance-in-javascript-21d2b82ffa6f)article for better understanding.

**Basic syntax:**

Object.create(prototype\_object, propertiesObject)

Object.create methods accepts two arguments.

1. prototypeObject: Newly created objects prototype object. It has to be an object or null.
2. propertiesObject: Properties of the new object. This argument is optional

**Create object with Object.create with no prototype**

Consider the below example to create a new object in JavaScript

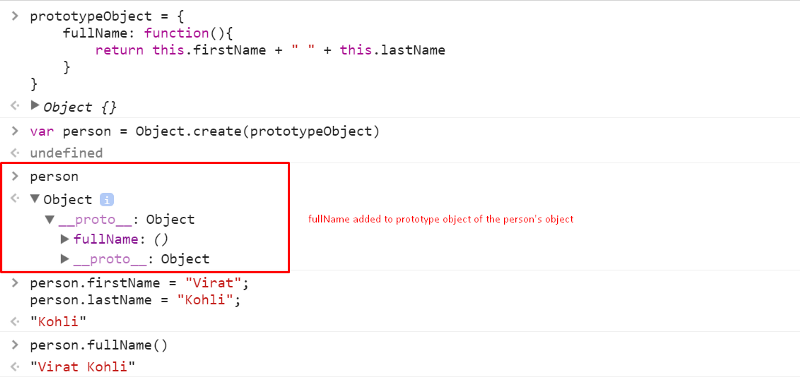


Here, we have created a new object *person* using *Object.create* method. As we have passed ***null***for the *prototypeObject*. *person*object does not have any prototype object.

Further, we have added *name*as new property to the person object.

Create object with prototype:

Console output:



In the above example, we have created a *prototypeObject*with *fullName*function. We created a *person* object with *prototypeObject*as prototype object of the *person’s* object using *Object.create()*. Further we added *firstName*and *lastName* properties to the *person*object. Here, we have added *firstName*and *lastName*properties after the object creation. It would have been great if we could add these properties while creating the object. To do that, we will use the 2nd argument of Object.create method.

**Object.create 2nd argument — propertiesObject**

*propertiesObject* is used to create properties on new object. It acts as a descriptor for the new properties to be defined. Descriptors can be data descriptor or access descriptors.

Data descriptors are

1. configurable
2. enumerable
3. value
4. writable

Access descriptors are

1. get
2. set

In detail, descriptors can be read [**here**](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Object/defineProperties)

Example:



In the above example we have created a new object *person*with prototype object as *prototypeObject*and properties as *firstName* and *lastName*.

Properties *firstName*and *lastName*have been added using the 2nd parameter of the *Object.create()*.

Inheritance using Object.create()

[Read Inheritance in JavaScript](https://medium.com/@happymishra66/inheritance-in-javascript-21d2b82ffa6f) before reading the below part.

Here we have copied the prototype of the SuperType to the SubType.prototype using Object.create method. Rest everything is same as the inheritance in JavaScript.

Other articles:

1. [Virtual DOM in ReactJS](https://medium.com/@happymishra66/virtual-dom-in-reactjs-43a3fdb1d130)
2. [this is JavaScript](https://hackernoon.com/this-in-javascript-8e8d4cd3930)
3. [Execution Context in JavaScript](https://medium.com/@happymishra66/execution-context-in-javascript-319dd72e8e2c)
4. [Prototypes in JavaScript](https://medium.com/@happymishra66/prototypes-in-javascript-5bba2990e04b)
5. [Inheritance in JavaScript](https://medium.com/@happymishra66/inheritance-in-javascript-21d2b82ffa6f)
6. [Create objects in JavaScript](https://medium.com/@happymishra66/create-objects-in-javascript-10924cfa9fc7)