

Advanced Javascript

Day 4 - Js Prototypes, Inheritance

Our goal today is, understanding what prototype means in a real world and then connecting that knowledge with how Js works.

What is a Prototype?

An original model on which something is patterned.



Lets create a simple object and see what a prototype is in Js.

```
var objectA = {}
```

Lets create a simple object using another object and see what a prototype is in Js.

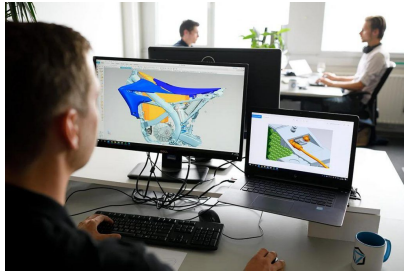
```
var objectB = Object.create(objectA)
```

__proto__?

Points to original model (prototype) from which it is built from.

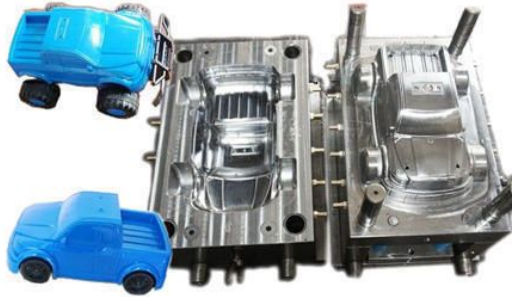


In Js, a **prototype** can also have a prototype hence creating a prototype chain.



In Js, it's better to borrow, than own a property or method.

Using Prototype we can create a blueprint from which we can borrow.



What is an Inheritance?

One object gets access to properties and methods of other object.

Js has Prototypal Inheritance.

We can use Js Prototypal inheritance in two ways:

1. Function Constructors

```
Person.prototype.showName=function () {  
    console.log( this.name )  
}
```

2. Object.create() method

```
var a = Object.create(b)
```

Let's build an iPhone 11.

hasOwnProperty()

The `hasOwnProperty()` method in JavaScript is used to check whether the object has the specified property as its own property.

This is useful for checking if the object has inherited the property rather than being its own.

object.hasOwnProperty(prop)

Everything in Js is an Object!

What that means is, Js has an object as its base prototype. From that, it creates other data types and methods that we can use on them.

It's just how Js was decided to be built.

Objects

```
var o = {a: 1};
```

1. The newly created object o has Object.prototype as its Prototype
2. So o inherits hasOwnProperty from Object.prototype
3. Object.prototype has null as its prototype.
- 4. o ---> Object.prototype ---> null**

functions

```
function f() {  
  return 2;  
}
```

1. Functions inherit from `Function.prototype`
2. (which has methods `call`, `bind`, etc.)
3. **`f` ---> `Function.prototype` ---> `Object.prototype` ---> `null`**

Array

```
var cities = ['Pune', 'Mumbai', 'Delhi'];
```

1. Arrays inherit from `Array.prototype`
2. (which has methods `indexOf`, `length`, `push` etc.)
3. The prototype chain looks like:
4. **b ---> `Array.prototype` ---> `Object.prototype` ---> `null`**

Prototype -



Inheritance

