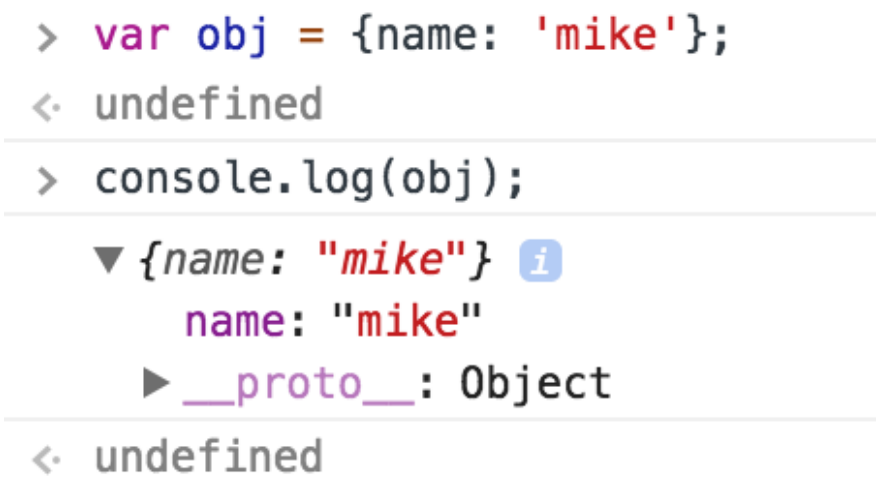
Understanding Prototype Chain

Ryuta Udo Follow
Oct 26, 2017 · 2 min read

Object is one of most scary feature of Javascript for me. Every time I log objects to console in browser, It shows unknown property named __proto__.
Then a question hit me. “How does it work ?” In order to understand it, I decided to write this topic. This post is for those who know experienced creating object and know about class and instance.

What is __proto__ ?

When object instance is created, mysterious __proto__ property is created at the same time.



What is this __proto__ ? It is the same as Object.prototype.

```
obj.__proto__ === Object.prototype // --> true
```

Then what is Object.prototype ? It looks like object with its property like;

```
obj.name
```

Yes, “Object” is a variable name of object and prototype is a property of Object. Then what is “Object” ? It is defined in window which is a global object.

```
obj.__proto__ === window.Object.prototype; // true
```

Thus __proto__ of instance object is equal to prototype of Object which is global variable.

Let’s play around with __proto__

When you look at Object.prototype in MDN reference, it is found that there is a method called toString. Let’s use it with instance object

```
console.log(obj.__proto__.toString);

  // --> ƒ toString() { [native code] }
```

OK, it seems to work. How about this ?

```
console.log(obj.toString); // --> ƒ toString() { [native
code] }
```

We could call toString method without “__proto__” property. Why does it work without __proto__ ? Now Prototype Chain comes up.

Prototype Chain

Prototype chain is the way Javascript looks for a property of an object. The flow is like this:

1. Check the object itself for the existence of the property.
2. If not found, it'll go to the object's prototype and check that object.
3. If not found, it'll go to the prototype's prototype.
4. keep going until it finds an object with a `__proto__` property equal to `null`
5. if null, it returns undefined.

Now __proto__ of Object is null. So If Javascript track object __proto__, gets to Object.prototype and cannot find the target property, it return undefined.

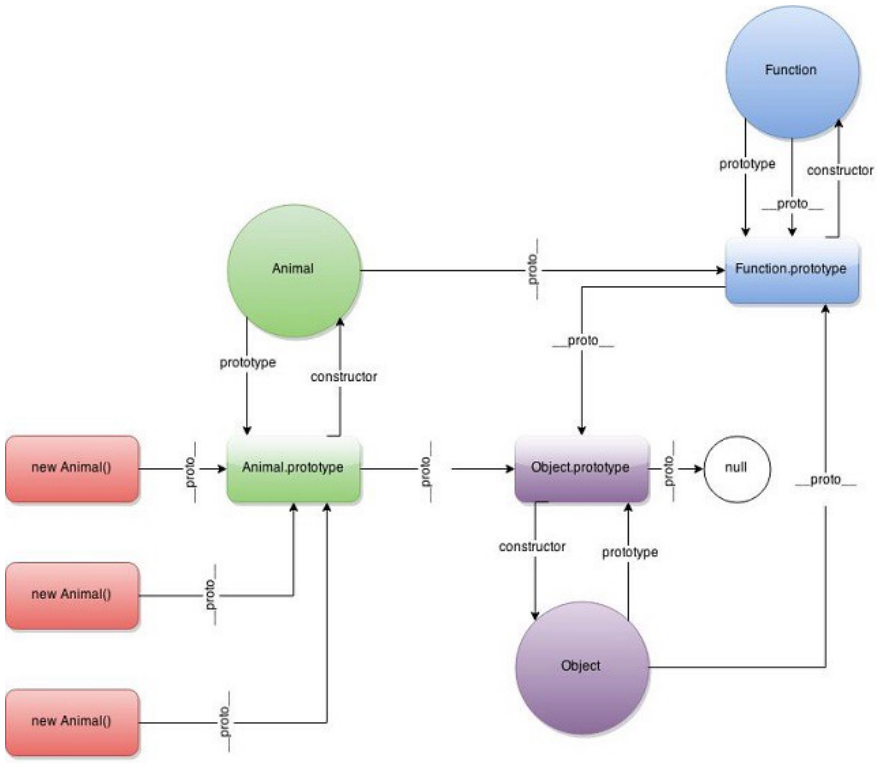
```
obj.__proto__.__proto__ === null; // true
Object.prototype.__proto__; // null
```

Now let's look back the previous example.

```
console.log(obj.__proto__.toString);
// --> ƒ toString() { [native code] }

console.log(obj.toString);
// --> ƒ toString() { [native code] }
```

Why obj.toString worked as obj.__proto__.toString ? That is because Javascript traverses up the chain of prototypes looking for the property and be able to find toString. This is how __proto__ works.



www.codeproject.com

References

Master JavaScript Prototypes & Inheritance
https://codeburst.io/master-javascript-prototypes-inheritance-d0a9a5a75c4e
MDN Object.prototype https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Object
Code Project
https://www.codeproject.com/Articles/887551/Prototypal-Inheritance-in-JavaScript