

## 「每日一题」什么是 JS 原型链?

**方应杭**

四年打工，七年创业。

关注他

319 人赞同了该文章

深入解读:

JS 中 `__proto__` 和 `prototype` 存在的意义是什么?

824 赞同 · 45 评论 · 回答



我们知道 JS 有对象，比如

```
var obj = { name: 'obj' }
```

我们可以对 `obj` 进行一些操作，包括

- 「读」属性
- 「新增」属性
- 「更新」属性
- 「删除」属性

下面我们主要来看一下「读」和「新增」属性。

### 为什么有 `valueOf` / `toString` 属性呢?

在我们没有对 `obj` 进行任何其他操作之前，发现 `obj` 已经有几个属性（方法）了：

```
> obj.valueOf()
< ► Object {}
> obj.toString()
< "[object Object]"
> obj.constructor
< function Object() { [native code] }
```

那么问题来了：`valueOf` / `toString` / `constructor` 是怎么来的？我们并没有给 `obj.valueOf` 赋值呀。

要搞清楚 `valueOf` / `toString` / `constructor` 是怎么来的，就要用到 `console.dir` 了。

知乎

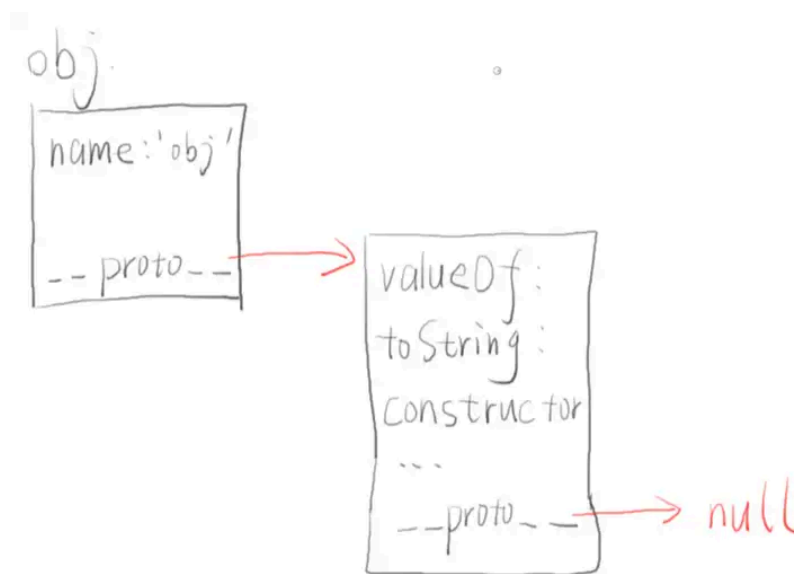
首发于  
前端

```

▼ __proto__: Object
  ▶ __defineGetter__: function __defineGetter__()
  ▶ __defineSetter__: function __defineSetter__()
  ▶ __lookupGetter__: function __lookupGetter__()
  ▶ __lookupSetter__: function __lookupSetter__()
  ▶ constructor: function Object()
  ▶ hasOwnProperty: function hasOwnProperty()
  ▶ isPrototypeOf: function isPrototypeOf()
  ▶ propertyIsEnumerable: function propertyIsEnumerable()
  ▶ toLocaleString: function toLocaleString()
  ▶ toString: function toString()
  ▶ valueOf: function valueOf()
  ▶ get __proto__: function __proto__()
  ▶ set __proto__: function __proto__()

```

上面这个图有点难懂，我手画一个示意图：



我们发现 `console.dir(obj)` 打出来的结果是：

1. obj 本身有一个属性 name（这是我们给它加的）
2. obj 还有一个属性叫做 `__proto__`（它是一个对象）
3. `obj.__proto__` 有很多属性，包括 `valueOf`、`toString`、`constructor` 等
4. `obj.__proto__` 其实也有一个叫做 `__proto__` 的属性（`console.log` 没有显示），值为 `null`

现在回到我们的问题：obj 为什么会拥有 `valueOf` / `toString` / `constructor` 这几个属性？

答案：

这跟 `__proto__` 有关。

当我们「读取」`obj.toString` 时，JS 引擎会做下面的事情：

1. 看看 obj 对象本身有没有 `toString` 属性。没有就走到下一步。
2. 看看 `obj.__proto__` 对象有没有 `toString` 属性，发现 `obj.__proto__` 有 `toString` 属性，于是找到了

所以 `obj.toString` 实际上就是第 2 步中找到的 `obj.__proto__.toString`

知乎

首发于  
前端3. 如果 `obj.__proto__` 没有, 那么浏览器会继续查看 `obj.__proto__.__proto__`4. 如果 `obj.__proto__.__proto__` 也没有, 那么浏览器会继续查看 `obj.__proto__.__proto__.proto__`5. 直到找到 `toString` 或者 `__proto__` 为 `null`。

上面的过程, 就是「读」属性的「搜索过程」。

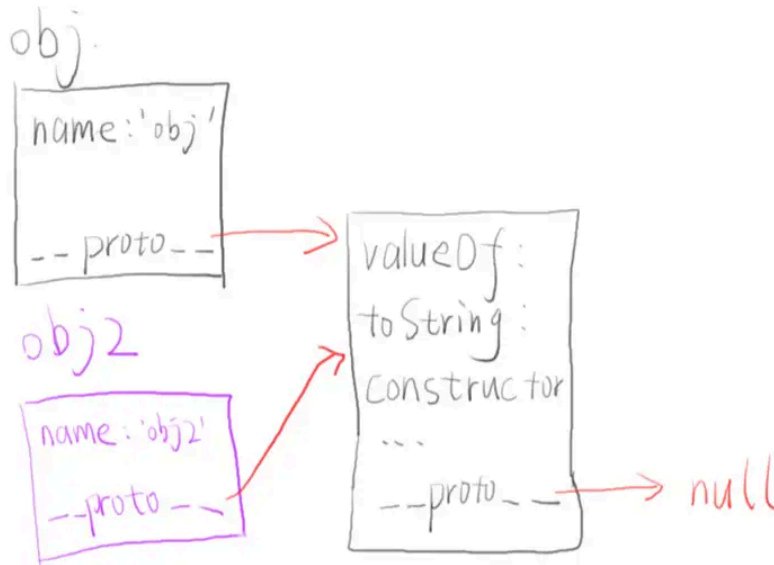
而这个「搜索过程」, 是连着由 `__proto__` 组成的链子一直走的。这个链子, 就叫做「原型链」。

### 共享原型链

现在我们有另一个对象

```
var obj2 = { name: 'obj2' }
```

如图:

那么 `obj.toString` 和 `obj2.toString` 其实是同一个东西, 也就是 `obj2.__proto__.toString`。

这有什么意义呢?

如果我们改写 `obj2.__proto__.toString`, 那么 `obj.toString` 其实也会变!这样 `obj` 和 `obj2` 就是具有某些相同行为的对象, 这就是意义所在。

### 差异化

如果我们想让 `obj.toString` 和 `obj2.toString` 的行为不同怎么做呢?

直接赋值就好了:

```
obj.toString = function(){ return '新的 toString 方法' }
```

总结：

- 「读」属性时会沿着原型链搜索。
- 「新增」属性时不会去看原型链（但是如果你给这个属性加了一些配置，则不一样，以后讲）。

欢迎进群与我探讨技术，暗号：前端/Java交流（附二维码）[qr.jirengu.com/api/task...](https://qr.jirengu.com/api/task...)

编辑于 2019-07-16 23:09

[前端开发](#) [前端入门](#) [面试问题](#)



欢迎参与讨论

41 条评论

默认 最新

**giscafer**

链啊链啊链\_\_proto\_\_.\_\_proto\_\_.\_\_proto\_\_——> null (Object), 最后一场空

2016-10-20

回复 18

**寒江水**

每个对象都有原型，当访问这个变量的时候，这个变量不存在就访问他的原型，就这样一直循环下去就产生了原型链！

2016-10-21

回复 14

**manxisuo**

刚写了篇文章，正好应景：[zhuanlan.zhihu.com/p/22...](https://zhuanlan.zhihu.com/p/22...)

2016-10-20

回复 7

**PrimaryK**

每日一题, 看到不懂的自己百度一下也能收穫很多, 多謝分享

2016-10-20

回复 4

**钢铁般的意志**

谢谢老师答疑解惑，小白受教了

2017-04-21

回复 2

**乔德**

文章里面怎么不介绍下很容易混淆的.prototype

2017-03-31

回复 1

2017-03-31

回复

1

**Koala**

大师解释的东西通俗易懂，没有一个有一个专业难懂的名词。

2017-02-21

回复

1

**Root**

谢谢方老师为我们分享~昨天看高程这里不懂，看了教程一下子就懂了。

2016-11-07

回复

1

**Wytthehuang**

mark

2016-10-20

回复

1

**士多啤梨苹果橙**

嗯，终于看到了

2016-10-20

回复

1

**海中月是天上月**

这个ID很熟悉呀

2021-03-10

回复

喜欢

点击查看全部评论 >

文章被以下专栏收录



前端

记录前端开发的一切干货

推荐阅读



想自学JS吗？想提升JS底层原理吗？76张脑图带你彻底搞...

圈圈儿

想自学JS吗？想提升JS底层原理吗？76张脑图带你彻底搞...

&#34;不畏惧，不讲究，未来的日子好好努力&#34;——大家好！我是小芝麻 在默默的更新了一段时间的文章后，小芝麻决定励志成为一个标题党 别走，别走，别走啊 .....搞错了，下面我们正式开...  
落幕 发表于web前端

【JS 口袋书】第 4 章：JS 引擎底层的工作原理

作者： valentinogagliardi 译者：前端小智 来源： github 阿里云最近在做活动，低至2折，有兴趣可以看看：  
https://promotion.aliyun.com/nt/userCode=pxuujn3r 为...  
前端小智 发表于小智译前端

浅拷贝



深拷贝



JS深浅拷贝探究

埋桑