

# JS 中 \_\_proto\_\_ 和 prototype 存在的意义是什么?

关注问题

写回答

JavaScript    prototype

关注者264

被浏览167,489

## JS 中 \_\_proto\_\_ 和 prototype 存在的意义是什么?

初学js对这两个概念很模糊。个人理解: js中万物皆对象\_\_proto\_\_是每个对象都有的属性, 而prototype是函数对象特有的。\_\_proto\_\_...显示全部

关注问题

写回答

邀请回答

好问题 16

4 条评论

分享

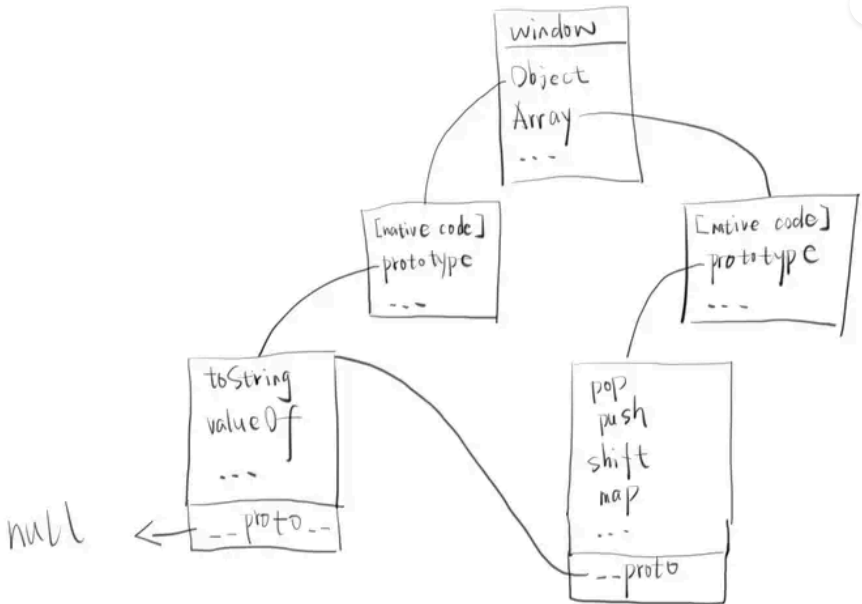
查看全部 14 个回答



**方应杭**  
四年打工, 七年创业。

+ 关注

824 人赞同了该回答



1. 你的 JS 代码还没运行的时候, JS 环境里已经有一个 window 对象了
2. window 对象有一个 Object 属性, window.Object 是一个函数对象
3. window.Object 这个函数对象有一个重要属性是 prototype, 干什么用的等会说
4. window.Object.prototype 里面有这么几个属性 toString (函数)、valueOf (函数)

好, 目前先知道这些就够了。

然后我们写一句代码

```
var obj = {}  
obj.toString()
```

这句代码做了啥? 为什么 obj 有 toString() 属性?



祖玛传奇单机版

### 关于作者



**方应杭**  
四年打工, 七年创业。

回答3,137

文章306

关注者69,417

关注他

发私信

### 被收藏 649 次

- 犀利时刻

只有我活在终端 创建

170 人关注
- 方老师文集

Charlie 创建

3 人关注
- 前端

张磊 创建

2 人关注
- 好东西

王爱平 创建

2 人关注
- 前端

jinyan yu 创建

2 人关注

### 相关问题

赞同 824

45 条评论

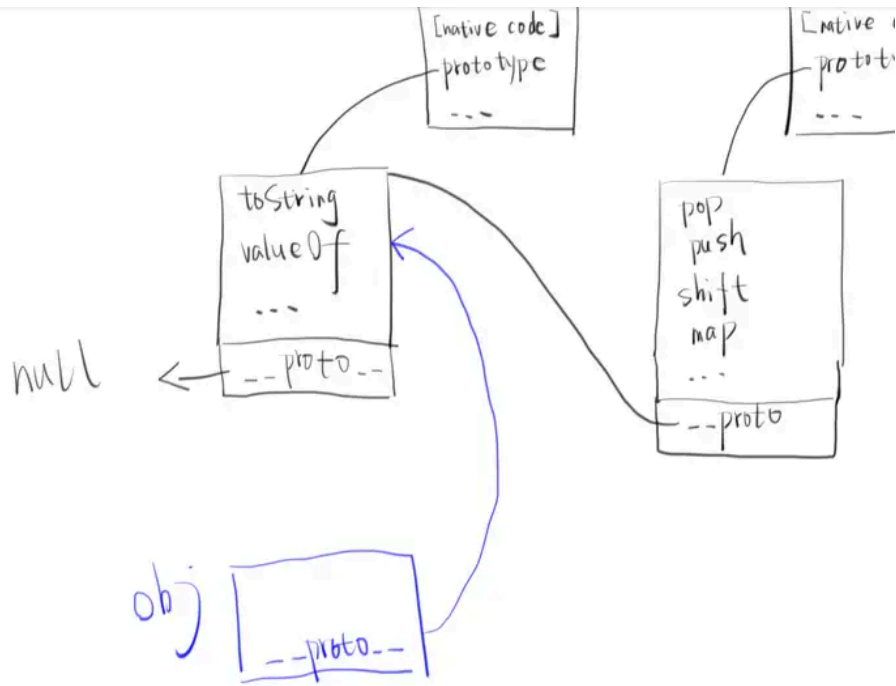
分享

收藏

喜欢

收起

JS 中 \_\_proto\_\_ 和 prototype 存在的意义是什么？



这句话大概是让 **obj 变量** 指向一个空对象，这个空对象有个 `__proto__` 属性指向 `window.Object.prototype`。

这样你在调用 `obj.toString()` 的时候，`obj` 本身没有 `toString`，就去 `obj.__proto__` 上面去找 `toString`。

所以你调用 `obj.toString` 的时候，实际上调用的是 `window.Object.prototype.toString`

那么 `window.Object.prototype.toString` 是怎么获取 `obj` 的内容的呢？

那是因为 `obj.toString()` 等价于 `obj.toString.call(obj)`

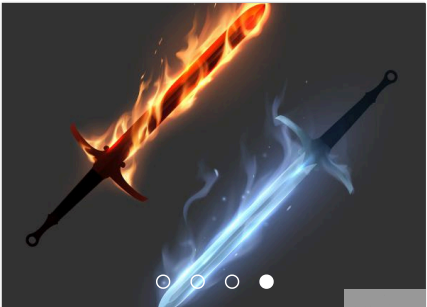
同时 `obj.toString.call(obj)` 等价于 `window.Object.prototype.toString.call(obj)`

这句话把 `obj` 传给 `toString` 了。

再看复杂一点的

回到第一幅图

- Prototype.js, 对象的prototype属性, 以及\_\_proto\_\_, 这三者之间有关系吗? 4 个回答
- Js 中的 prototype 是怎样的? 0 个回答
- js 中使用原型对象 prototype 需要注意的地方有哪些? 0 个回答



传奇网页版

- 帮助中心
- 知乎隐私保护指引 申请开通机构号 联系我们
- 举报中心
- 涉未成年举报 网络谣言举报 涉企侵权举报 更多
- 关于知乎
- 下载知乎 知乎招聘 知乎指南 知乎协议 更多

京 ICP 证 110745 号 · 京 ICP 备 13052560 号 - 1 ·  
京公网安备 11010802020088 号 · 京网文  
[2022]2674-081 号 · 药品医疗器械网络信息服务备  
案 (京) 网药械信息备字 (2022) 第00334号 · 广  
播电视节目制作经营许可证: (京) 字第06591号 ·  
服务热线: 400-919-0001 · Investor Relations ·  
© 2024 知乎 北京智者天下科技有限公司版权所有 ·  
违法和不良信息举报: 010-82716601 · 举报邮箱:  
jubao@zhihu.com



## JS 中 \_\_proto\_\_ 和 prototype 存在的意义是什么?

我们写一句代码

```
var arr = []  
arr.push(1) // [1]
```

请问这两句话做了什么?

看红色部分, `var arr = []` 大概会让 `arr` 指向一个空对象, 然后 `arr.__proto__` 指向 `window.Array.prototype`。(其实 `arr` 有一个 `length:0`, 不过这里就忽略吧)

这样你在调用 `arr.push` 的时候, `arr` 自身没有 `push` 属性, 就去 `arr.__proto__` 上找 `push`

因此 `arr.push` 实际上是 `window.Array.prototype.push`



## JS 中 \_\_proto\_\_ 和 prototype 存在的意义是什么?

`arr.push.call(arr,1)` 等价于 `window.Array.prototype.push.call(arr, 1)`

### 再再复杂一点

`arr.valueOf()` 做了什么?

`arr` 自身没有 `valueOf`, 于是去 `arr.__proto__` 上找

`arr.__proto__` 只有 `pop`、`push` 也没有 `valueOf`, 于是去 `arr.__proto__.__proto__` 上找

`arr.__proto__.__proto__` 就是 `window.Object.prototype`

所以 `arr.valueOf` 其实就是 `window.Object.prototype.valueOf`

`arr.valueOf()` 等价于 `arr.valueOf.call(arr)`

`arr.valueOf.call(arr)` 等价于 `window.Object.prototype.valueOf.call(arr)`

看, JavaScript 其实很优美很简单。

只是你想复杂了而已:

`prototype` 指向一块内存, 这个内存里面有共用属性

`__proto__` 指向同一块内存

`prototype` 和 `__proto__` 的不同点在于

`prototype` 是构造函数的属性, 而 `__proto__` 是对象的属性

难点在于.....构造函数也是对象!

如果没有 `prototype`, 那么共用属性就没有立足之地

如果没有 `__proto__`, 那么一个对象就不知道自己的共用属性有哪些。

### 反证法<sup>Q</sup>

假设我们把 `__proto__` 去掉, 那么

```
var obj = {}  
obj.toString() // 报错, 没有 toString 方法
```

所以你能这样声明一个对象咯:

```
var obj = {  
  toString: window.Object.prototype.toStringQ,  
  valueOf: window.Object.prototype.valueOf  
}
```

## JS 中 \_\_proto\_\_ 和 prototype 存在的意义是什么?

知道 \_\_proto\_\_ 帮你省多少代码了吗?

假设我们删掉 prototype, 包括 window.Object.prototype<sup>Q</sup> 和 window.Array.prototype。

那么 window.Object.prototype.toString 也一并被删除了。

然后我们基本就没法写代码了.....

```
var objQ = {}  
obj.toString() // toString 不存在, 因为 toString 没有定义过啊
```

prototype 的意义就是把共有属性预先定义好, 给之后的对象用。

自己想想吧~

新人搞不懂原型大抵是因为

1. 不懂内存、引用
2. 不懂链表、树等数据结构<sup>Q</sup>
3. 不知道函数是一种对象
4. 被 Java 的 class 关键字毒害了
5. 还有一种可能是因为没遇到我

以上是「方三篇」新人一定要看哦。想看视频版本可以购买我的网课（收费）：

这幅图我还可以继续讲, 把 JS 所有基础知识都能串起来。

比如很多人不懂什么是伪数组, 很简单:

1. 如果一个数组的 \_\_proto\_\_ 直接或间接是真数组



## JS 中 \_\_proto\_\_ 和 prototype 存在的意义是什么?

```
var realArr = {0: 'a', 1: 'b', length: 2}
realArr.__proto__ = Array.prototype
// 这就是真数组🔗 (并不完全是)
// 基本等价于 realArr = ['a', 'b']
realArr.push !== undefined // true

var fakeArr = {0: 'a', 1: 'b', length: 2}
// 这就是伪数组
realArr.push === undefined🔗 // true
```

完。

编辑于 2020-10-10 12:18

更多回答



flyingsoul

首先, \_\_proto\_\_ 是非标准用法, 不是所有的JS引擎都支持它, 它作为 Object.prototype 上的一个存取器属性主要用来获取或者设置某个对象的 [[prototype]] internal-slot。JS中所有的对象都拥有 [[prototype]] internal-slot, 它形成对象的原型链, 用于对象的属性访问 (当然设置属性的时候也会访问原型链, 比如原型上存在同名的 [[writable]] attribute 为false的属性, 此时设置属性的值会失败; 又或者原型上存在同名的具有 [[set]] attribute 的存取器属性, 此时会调用该存取器来设置值。

- [[prototype]] 用于原型链的访问, 每个JS对象都有; (纠正一下: ES6的proxy exotic object不存在 [[prototype]] internal slot)
- prototype 往往存在于构造函数, 在生成实例对象的时候会隐式的将实例对象的 [[prototype]] internal-slot 指向 prototype 属性; (当然, prototype 也可能存储于 非构造函数 当中, 只要该函数可以用于生成新的对象, 并且生成的新对象需要一个区域存放公共属性(例子: proxy对象直接代理到target对象, 新生成的proxy对象没有 [[prototype]] 需要初始化, 因此 Proxy constructor 就不需要 prototype 属性用于初始化新生成的proxy对象); 那么理论上就应该设置该函数的 prototype 属性, 并在其中为生成的实例对象添加一些公共属性, 最直接的例子是 生成器函数, 有兴趣的可以查看 [为什么Math下的函数没有prototype? \(Math.abs.prototype\)?](#)下的回答。

▲ 赞同 3 ▼ ● 1 条评论 ↗ 分享 ★ 收藏 ❤ 喜欢 ...



smallpdf  
.cn (收集高质量网站资源)

### 一、prototype 是什么?

prototype 是每个函数(不包括箭头函数)都默认具有的属性, 默认值为: 指向函数本身的 constructor 对象。

#### 1、查看 prototype

## JS 中 \_\_proto\_\_ 和 prototype 存在的意义是什么?

```
alert(Rabbit.prototype.constructor == Rabbit)
```

▲ 赞同 1 ▼

● 添加评论

🚩 分享

★ 收藏

♥ 喜欢

...

查看全部 14 个回答