

js的有没有类这一概念啊?

JavaScript

js的有没有类这一概念啊?

学完c++和java以后直接学nodejs。。。表示对里面的写法很是奇怪啊? 例如下面这个User () 它到底是类吗? 还是函数? User.prototype...显示全部

关注问题

写回答

邀请回答

好问题 1

添加评论

分享

...

9 个回答

默认排序



江湖术士  
DAYZ is coming

+ 关注

8 人赞同了该回答

我觉得这是个哲学问题，基于类的是创建模型，基于原型的是经验模型，前者是现有马这个概念，才有胯下之马这个实体，后者是先有胯下之马这个实体，才有马这个概念。

所以到底是有马还是无马，这是个很深邃的问题。

函数原型和对象原型属性，这也是个哲学概念啊，你想，是一个实体的行为和特征唯一标识这个实体，还是因为这个实体存在，他的行为和特征才会存在。

所以施主你是存在还是不存在?

好吧，扯这些没用，也不是一朝一夕能够理解的，毕竟哲学。

在思考面向原型的时候，类的概念有是可以的，因为本质上对象的原型就是类。但是，类构造出对象的这个过程却要不得，面向原型里，是由具体对象推导出原型来的。

实现上，访问对象的属性或者方法时，会访问其\_\_proto\_\_ 以及 \_\_proto\_\_ 的 \_\_proto\_\_，直到找到这个属性或者方法为止，而函数的prototype属性执行后会返回执行结果的\_\_proto\_\_。

当然，最重要的是：（各单位注意，我要装逼了，也要毁你三观了！）

JS的this不是其他语言里的this指针，而是函数的调用记录。

重要的事情说三遍：

JS的this不是其他语言里的this指针，而是函数的调用记录。

JS的this不是其他语言里的this指针，而是函数的调用记录。

而new操作符，也不是其他语言的构造指令，而是创建一个新的对象，并将其this替换为函数的this，即js三种this绑定之一的new绑定（另外两个是软绑定和硬绑定）。

所以只要函数运行返回的对象this是函数的this，是不是new构造的有什么所谓?

就喜欢看你们这些其他技术栈的人折服于js强（la）大（ji）设计的样子，特别开心。

编辑于 2017-12-14 22:49



相关问题

js应该怎么样去学习，求推荐学习方法？  
1 个回答

我是个it小白，现在刚学完js的基础知识，求一些可以用js做的简单项目来熟悉应用一下，求各位大神帮忙？  
1 个回答

如何有效的去学习js？有没有大佬说明一下？  
6 个回答

js 技术的学习过程有哪些注意事项？  
0 个回答

js的一些基础问题？  
0 个回答



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  
下载知乎客户端 知乎专栏 知乎书店 知乎知识 知乎日报 知乎视频 知乎直播 知乎招聘 知乎公益 知乎百科 知乎问答 知乎搜索 知乎商城 知乎会员 知乎广告 知乎企业号 知乎直播 知乎招聘 知乎公益 知乎百科 知乎问答 知乎搜索 知乎商城 知乎会员 知乎广告 知乎企业号

## js的有没有类这一概念啊?

2 人赞同了该回答

说几点，帮助题主了解一下js

- 1.js在ES2015的新特性中添加了类的概念（可以使用class了），这样的写法更像主流的[面向对象语言](#)。
- 2.当前网上的资料，ES5版本仍多于ES2015版本，而ES5是没有类的。
- 3.ES5的构造函数就相当于类，可以使用 new 一个构造函数的方式创建类的实例。如果希望扩展类的方法，可以在给构造函数的prototype属性添加方法，这样也能面向对象，就是这个对象有点丑。
- 4.ES5的继承可以用[原型链](#)的方式实现，但是，这个继承丑得已经一塌糊涂了，要不然新版本也不会把class和extends加进来。
- 5.ES2015的class是一个[语法糖](#)，原理还是那样的原理，就是写得舒服了，建议题主用ES2015的语法来写node，这样相对于ES5更像Java。
- 6.如果题主感觉还是不够爽，那就用[typescript](#)，题主有java基础，学习typescript就是分分钟的事儿，而且写起来更清爽，妈妈再也不用担心我的数据类型了，谁用谁知道。

没有JavaScript基础的朋友，可以关注[微信公众号](#)：晓舟报告，从零开始讲解JavaScript，日更哟！

编辑于 2017-12-14 22:39

▲ 赞同 2 ▼

● 添加评论

🔗 分享

★ 收藏

♥ 喜欢

...

收起 ^



Kell

Hello world

C++ 和 Java中的类 跟js中的类不是一回事 它们的继承，封装，[多态](#)在JS的类中是不存在的。js中的类可以说是 函数 + prototype 机制 叠加上去的，只是徒有其表而已。JS中并不推崇使用类去建模，抽象，多态，Js中更喜欢让你用 List (js中是Array),Object 这两个"类"去承包所有的数据，如现在[大行其道](#)的JSON，。js中的函数可以没着没躁的传递到任何位置，so用函数来进行抽象。

js也可以写出美的代码 so [看着办吧](#)

编辑于 2017-12-14 23:35

▲ 赞同 ▼

● 添加评论

🔗 分享

★ 收藏

♥ 喜欢

...

收起 ^

智谱清言

广告 ▼

### 来智谱清言“AI搜索”一下吧!

智谱清言GLMs智能体的高级联网功能，不仅可以精准识别用户意图，还能快速准确回答问题，无需自主分辨信息，智谱清言直接给出答案~ [查看详情](#)



Albert

(ᄇᄇᄇ)hiahiahia

嗯.....这个问题嘛。

js是有类这个概念滴.我推荐你多了解一下js的原型链.还有慕课网的js深入浅出

相信你学完就会更加懂得不少东西。面向对象是一种[程序设计范型](#)。就连C都可以用这种方式去开发。只不过是java 跟 C# 本身就是按照这个设计的。所以语法都是面向对象的。因为他们设计之初就是用这个标准。而js 跟 php 这种[弱类型语言](#)。刚开始并不是使用的面向对象。所以php 跟 js都是既支持 [面向过程](#) 又 支持面向对象 这种混开的模式。其中js 的[原型链](#)就相当于java的一个继承链。java的顶级类是Object，js的[原型链](#)顶级也是 Object

发布于 2017-12-13 20:32

[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



适老化  
无障碍服务

## js的有没有类这一概念啊?



joker

是有class，但是原理不一样，所以不能完全套用其他语言的做法。

发布于 2017-12-14 23:32

赞同



添加评论

分享

收藏

喜欢



收起



知乎用户joLI8h

1 人赞同了该回答

去看JavaScript[高级程序设计](#)，书里面把所有继承都讲了。

发布于 2017-12-13 23:29

赞同 1



添加评论

分享

收藏

喜欢



收起



蓝色微笑

这不是知乎蓝，它比知乎蓝明亮26.77%

Javascript[高级程序设计](#)第六章，

发布于 2017-12-14 04:32

赞同



添加评论

分享

收藏

喜欢



收起



贪婪的陈

混口饭吃的码畜

并没有，都是基于[原型链](#)和对对象委托实现的类似于类的效果

发布于 2017-12-14 22:19

赞同



添加评论

分享

收藏

喜欢



收起



为霜

琥珀色的黄昏是什么样子

1 人赞同了该回答

js的有没有类这一概念啊?

ES6之后就有类的概念了，但是也只是个[语法糖](#)(特殊的函数吧)，和传统的语言如java, c++里面的类还是不同的。A class declaration is syntactic sugar over **prototypal inheritance** with additional enhancements.

User.prototype () 是属于User () 里面的一个函数吗?

User.prototype 并不是一个函数，而是一个对象。当定义一个函数之后被创建出来的空对象。

我也是先接触的cpp, java, 最近才接触的js, 但也不打算再搞js了, 弱类型, 动态...写着真的不舒服。

你可能会疑惑为什么把匿名函数赋值给数呢，答案就是为了高效,reuse. 如果在函

## js的有没有类这一概念啊?

```
const obj = new User();
```

new做了几件事,其中有件事就是把对象obj的 `__proto__` 属性赋值为User的 `prototype`, 等价于:  
`obj.__proto__ = User.prototype`. 这其实就是说让对象obj的父亲为User, 这样当obj调用一个函数, 如果发现没有,就会到User里去找([原型链](#)).

想要理解还得说语言本身 js是prototypal继承而不是classical继承(java,cpp). 所以它不是class-based language, 一些行为就很不同. classical继承里面, 都是子类继承父类, 而在js里, 是对象 “继承” 另外一个对象, 怎么实现的呢, 就是通过 `__proto__`, 原型链(prototype chain). 上面已经解释. 哦, js是object-based语言, 所以你懂了吧.

另外注意 [obj.prototype](#) 和 `obj.__proto__` 并不是一个东西,

编辑于 2022-10-11 12:27

赞同 1



1 条评论

分享

收藏

喜欢



收起

[写回答](#)