

▲

赞同 274

➤

分享

## JS重点整理之JS原型链彻底搞清楚



咕噜咕噜

关注她

274 人赞同了该文章

修改：之前构造函数命名写的不严谨，现在首字母大写都已修正，图片中的部分就不做修正了。

### 对象

要清楚原型链,首先要弄清楚对象：

- 普通对象
  - 最普通的对象：有\_\_proto\_\_属性（指向其原型链），没有prototype属性。
  - 原型对象(Person.prototype 原型对象还有constructor属性（指向构造函数对象））
- 函数对象：
  - 凡是通过new Function()创建的都是函数对象。

拥有\_\_proto\_\_、prototype属性（指向原型对象）。

Function、Object、Array、Date、String、自定义函数

特例：Function.prototype(是原型对象，却是函数对象，下面会有解释)

```
// 函数对象
function F1(){};
var F2 = function(){};
var F3 = function("n1","n2","return n1+n2");

console.log(typeof F1); //function
console.log(typeof F2); //function
console.log(typeof F3); //function
console.log(typeof Object); //function
console.log(typeof Array); //function
console.log(typeof String); //function
console.log(typeof Date); //function
console.log(typeof Function); //function

Array
function Array() { [native code] }
Array.__proto__ === Function.prototype
```

```
// 普通对象
var o1 = new F1();
var o2 = {};
var o3 = new Object();

console.log(typeof o1); //Object
console.log(typeof o2); //Object
console.log(typeof o3); //Object
```

原型对象

每创建一个函数都会有一个prototype属性，这个属性是一个指针，指向一个对象（通过该构造函数创建实例对象的原型对象）。原型对象是包含特定类型的所有实例共享的属性和方法。原型对象的好处是，可以让所有实例对象共享它所包含的属性和方法。

第一块中有提到，原型对象属于普通对象。Function.prototype是个例外，它是原型对象，却又是函数对象，作为一个函数对象，它又没有prototype属性。

```
function Person(){};

console.log(typeof Person.prototype) //Object
console.log(typeof Object.prototype) // Object
console.log(typeof Function.prototype) // 特殊 Function
console.log(typeof Function.prototype.prototype) //undefined 函数对象却没有prototype属性
```

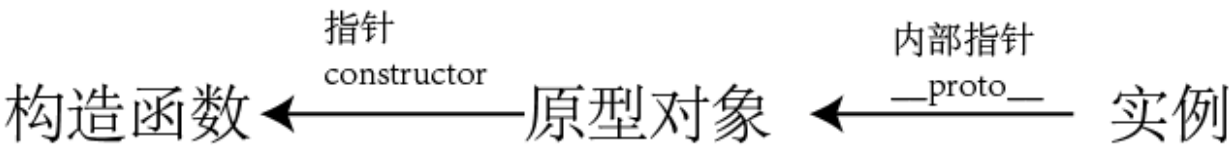
解释：

其实原型对象就是构造函数的一个实例对象。person.prototype就是person的一个实例对象。相当于在person创建的时候，自动创建了一个它的实例，并且把这个实例赋值给了prototype。

```
function Person(){};
var temp = new Person();
Person.prototype = temp;

function Function(){};
var temp = new Function();
Function.prototype = temp; //由new Function()产生的对象都是函数对象
```

从一张图看懂原型对象、构造函数、实例对象之间的关系



```
function Dog(){};

Dog.prototype.name = "小黄";
Dog.prototype.age = 13;
Dog.prototype.getAge = function(){
    return this.age;
}

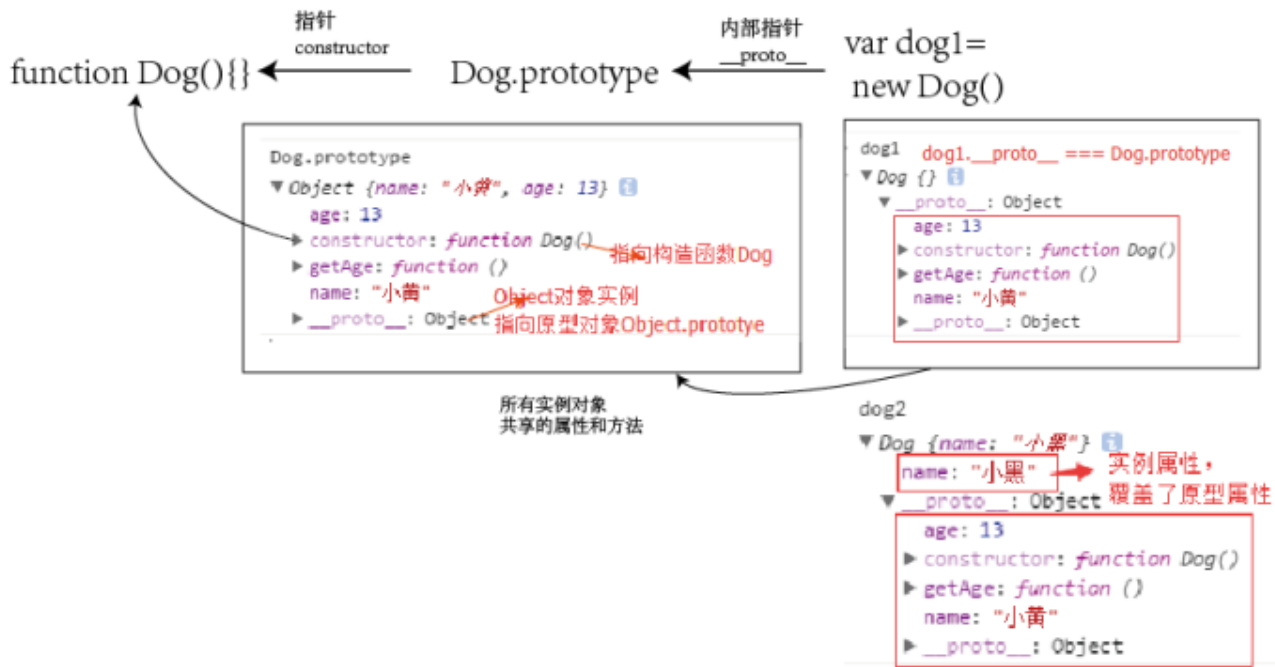
var dog1 = new Dog();
var dog2 = new Dog();
```

▲

赞同 274

➤

分享



赞同 274

分享

```
//图中的一些关系
dog1.__proto__ === Dog.prototype

Dog.prototype.__proto__ === Object.prototype //继承Object 下面原型链说

dog1.__proto__.__proto__ === Object.prototype

Dog.prototype.constructor === Dog

Dog.prototype.isPrototypeOf(dog1)

//获取对象的原型
dog1.__proto__ //不推荐
Object.getPrototypeOf(dog1) === Dog.prototype //推荐
```

### 原型链

原型链是实现继承的主要方法。

先说一下继承，许多OO语言都支持两张继承方式：接口继承、实现继承。

- 接口继承：只继承方法签名

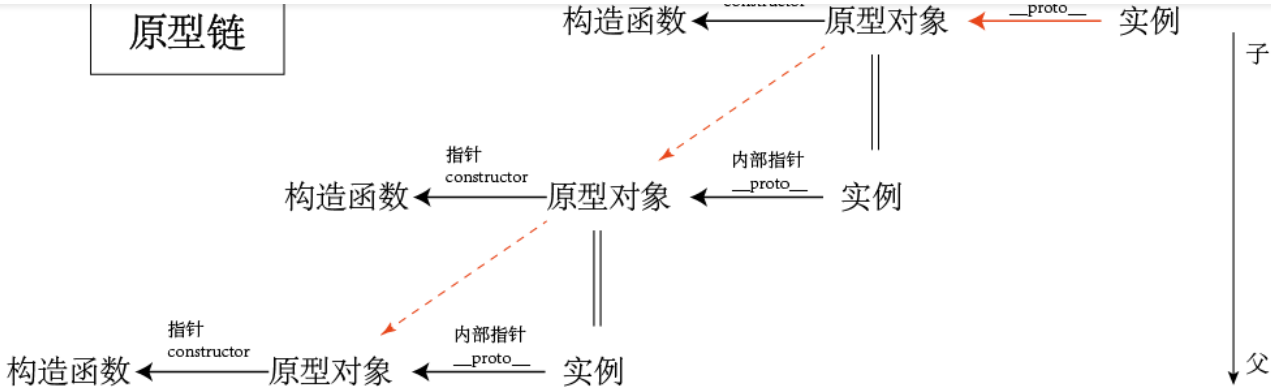
- 实现继承：继承实际的方法

由于函数没有签名，在ECMAScript中无法实现接口继承，只支持实现继承，而实现继承主要是依靠原型链来实现。

原型链基本思路：

利用原型让一个引用类型继承另一个引用类型的属性和方法。

每个构造函数都有一个原型对象，原型对象都包含一个指向构造函数指针(constructor)，而实例对象都包含一个指向原型对象的内部指针(\_\_proto\_\_)。如果让原型对象等于另一个类型的实例，此时的原型对象将包含一个指向另一个原型的指针(\_\_proto\_\_)，另一个原型也包含着一个指向另一个构造函数的指针(constructor)。假如另一个原型又是另一个类型的实例.....这就构成了实例与原型的链条。



继承：通过原型链来实现。  
本质：**重写**原型对象，代之以一个新类型的实例。  
思想：利用原型让一个引用类型继承另一个引用类型的属性和方法。  
@guifangzhang

举例说明：

```
function Animal(){
    this.type = "animal";
}
Animal.prototype.getType = function(){
    return this.type;
}

function Dog(){
    this.name = "dog";
}
Dog.prototype = new Animal();

Dog.prototype.getName = function(){
    return this.name;
}

var xiaohuang = new Dog();

// 原型链关系
xiaohuang.__proto__ === Dog.prototype
Dog.prototype.__proto__ === Animal.prototype
Animal.prototype.__proto__ === Object.prototype
Object.prototype.__proto__ === null
```

图解:

详细图

▲

赞同 274

▼

分享

从xiaohuang这个实例，看出整个链条

总结：

Xiaohuang这个Dog的实例对象继承了Animal，Animal继承了Object。

编辑于 2017-05-07

JavaScript

编程学习

前端开发

推荐阅读



前端三元同学  
扫码关注不迷路

前端每日一问，提升技术认知

**(建议收藏)原生JS灵魂之问, 请问你能接得住几个? (上)**

神三元

**都0202年了，你还不知道 javascript有几种继承方式？**

文章首发于：公众号【前端充电营】都0202年了，你还不知道 javascript有几种继承方式？前言当面试官问你：你了解js哪些继承方式？es6的class继承是如何实现的？你心中有很清晰的答案吗？ ...

欧小白      发表于前端充电营

**「每日一题」链？**

深入解读：JS prototype 存们知道 JS 有对 { name: &#3以对 obj 进行「读」属性

方应杭

37 条评论

切换为时间排序

写下你的评论...



陈氏太急

2016-10-07

不错啊妹子，这应该是你的笔记吧

赞



咕噜咕噜 (作者) 回复 陈氏太急

2016-10-07

是的，顺便就分享一下了

1



布客飞龙

2016-10-07

赞一个~

赞



一笑

2016-10-07

有一点没太懂，不是说js里所有对象都是由Object继承过来的吗，那为什么特殊Function的prototype是undefined？不应该是Object么？

赞



咕噜咕噜 (作者) 回复 一笑

2016-10-07

一般的原型对象是普通对象，但是Function.prototype(原型对象)是一个函数对象，一般的函数对象会有prototype属性，但是Function.prototype没有prototype属性，即

赞同 274

37 条评论

分享

喜欢

收藏

申请转载

...

	哈哈，我关注了你哦，猜猜我是谁？！ <div>👍 赞</div>	
	<div><div></div> 咕噜咕噜 (作者) 回复 胡歌 小彤子~(〃▽〃~) <div>👍 赞</div></div>	2016-10-07
	<div><div></div> 识食物者 写的很好! <div>👍 1</div></div>	2016-10-08
	<div><div></div> 卡夫卡 虽然看的有点晕，但是能感觉到整理的很有条理，思路很清晰，收藏以后在看。 <div>👍 赞</div></div>	2016-10-08
	<div><div></div> Driven 提个建议,构造函数 首字母大写, 比如 Dog 类 和 Animal 类 <div>👍 赞</div></div>	2016-10-12
	<div><div></div> 咕噜咕噜 (作者) 回复 Driven 谢谢提醒 <div>👍 赞</div></div>	2016-10-22
	<div><div></div> 好困大师 我今天刚搞懂这个，对着prototype想了一天。你总结的真挺好的 <div>👍 赞</div></div>	2017-03-06
	<div><div></div> 芦苇 妹子！你可以啊！！厉害👍 <div>👍 赞</div></div>	2017-05-05
	<div><div></div> 芦苇 有个地方有个小错误，xiaohuang.__proto__ === Dog.prototype <div>👍 赞</div></div>	2017-05-06
	<div><div></div> 咕噜咕噜 (作者) 回复 芦苇 谢谢，是图片上那个注释吧，已经修正了 <div>👍 赞</div></div>	2017-05-07
	<div><div></div> 小小仙的笔记 写的真好 <div>👍 赞</div></div>	2017-05-07
	<div><div></div> 老子 虽然一知半解 你能写出这样深层东西，我佩服~ <div>👍 赞</div></div>	2017-06-30
	<div><div></div> 林生生 非常感谢 终于明白一些了 <div>👍 赞</div></div>	2017-11-16
	<div><div></div> 「已注销」 这个简直太赞了 <div>👍 赞</div></div>	2017-12-22
	<div><div></div> 伞三 思路很清晰</div>	2017-12-27

- 1.实例复制构造函数内的属性，同时复制一个\_proto\_属性，
- 2.构造函数可以指定prototype属性，new的时候就变成实例的\_proto\_，如果没做prototype赋值，那实例的\_proto\_就是{};
- 3.一个对象的属性先从自己私有的来调，如果没有就去\_proto\_属性的值那里找，拿来即用;
- 4.\_proto\_的值是动态的，用的时候去查，对象私有的值是死的
- 5.作为\_proto\_值的那个对象，如果它身上没有属性，就继续去查作为它的\_proto\_的的值的那个对象有没有属性

👍 1

 giser 回复 1023ang2017-07-17


实例复制构造函数内的属性，同时复制一个\_proto\_属性

👍 赞

 giser 回复 1023ang2017-07-17

改动构造函数，却没有修正

👍 赞

 Joash2018-03-18

是不是应该在Dog.prototype=new Animal()后面加一句 Dog.prototype.constructor=Dog

👍 2

 镜中花 回复 Joash2018-06-06

我也这么觉得

👍 赞

 qwerty2018-08-22

不错

👍 赞

 梦绾2018-09-05

1有函数对象这个概念吗？

```
2function F1(){};
var F2 = function(){};
var F3 = function("n1","n2","return n1+n2");
```

你这里直接就报错了哪有这样写的？

👍 1

 梦绾2018-09-05

还有用new实例化的对象没有prototype吧，难道是我理解错了？

👍 赞

 镜中花 回复 梦绾2018-06-03

？ 实例本来就没prototype吧，构造函数才有，实例只有一个\_proto\_ 吧，连 constructor属性也是从原型对象继承来的吧

👍 赞