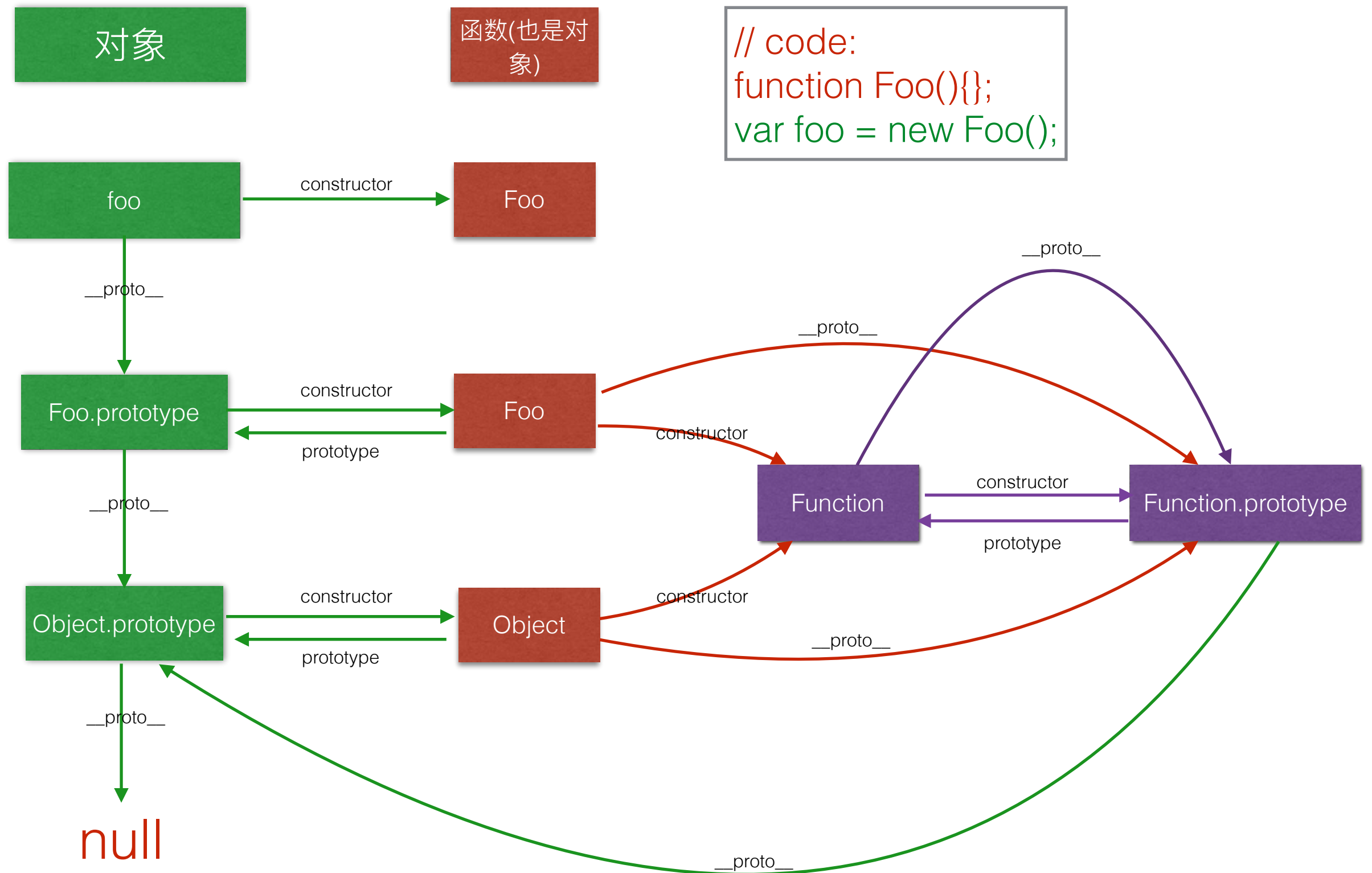


JavaScript继承

kangning1206@163.com

2018-12-08



javascript 对象、prototype、Function、Object

__proto__ 链路看继承

- 对象foo上的属性继承于Foo.prototype对象上的定义；
- Foo.prototype上的属性又继承于Object.prototype对象上的定义；
- Object.prototype 不继承任何原型；
- constructor、toString、hasOwnProperty都是顶级Object.prototype上定义的，提供给子对象使用；

constructor构造函数

- 对象foo的foo.constructor构造函数指向Foo;
- 原型Foo.prototype.constructor指向Foo;
- 结论:构造函数也是原型Foo.prototype上的方法;
- 所以 new Foo和new Foo.prototype.constructor都可以实例化相同类型对象

函数链路继承

- 函数`Foo.constructor == Function`, 说明`Foo`对象是`Function`的一个实例;
- `Foo.__proto__ == Function.prototype`, 函数`Foo`上的属性继承于`Function.prototype`对象上的定义; 比如`call`方法是定义在`Function.prototype.call`
- `Function.prototype`上的属性又继承于`Object.prototype`对象上的定义;
- 最终函数`Foo`继承于`Object`

创建对象经历了

- 实例化或声明一个构造函数Foo，它是工厂，具有制造对象的能力；
- 在Foo.prototype上定义属性和方法，作为模板将被实例继承；
- 使用new Foo()实例化对象；