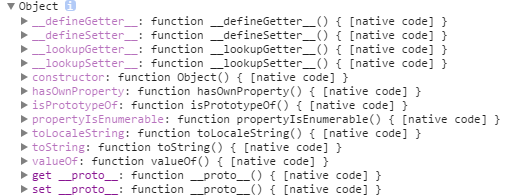
**深入理解javascript原型和闭包（4）——隐式原型**

<http://www.cnblogs.com/wangfupeng1988/p/3979290.html>

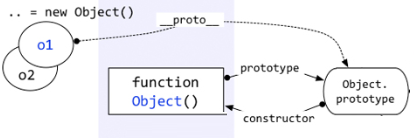
var obj= {};

console.log(obj.\_\_proto\_\_);

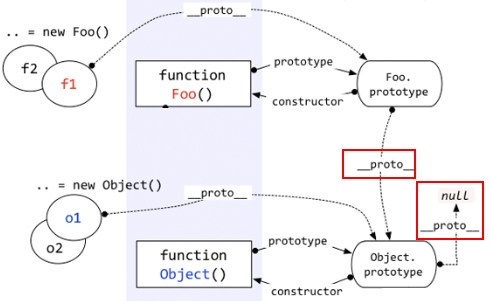


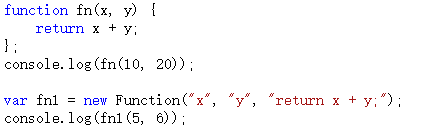
obj.\_\_proto\_\_和Object.prototype的属性一样

obj本质上是被Object函数创建的，因此Obj.\_\_proto\_\_ === Object.prototype

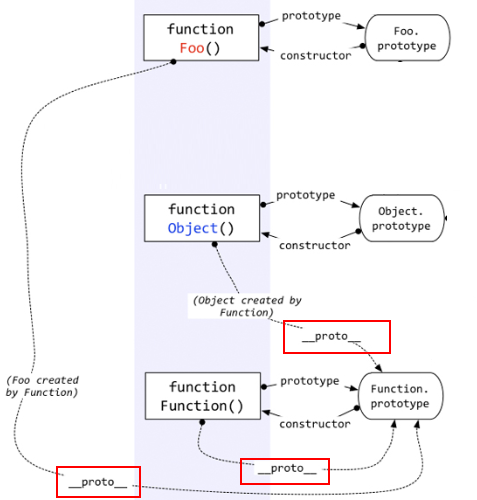


Object.prototype确实一个特例—它的\_\_proto\_\_指向的是null





对象的\_\_proto\_\_指向的是创建它的函数的prototype,就会出现：Object.\_\_proto\_\_ === Function.prototype



上图中，很明显的标出了：自定义函数Foo.\_\_proto\_\_指向Function.prototype，Object.\_\_proto\_\_指向Function.prototype。是一个环形结构。Function也是一个函数，函数是一种对象，也有\_\_proto\_\_属性。既然是函数，那么它一定是被Function是被创建。所以——Function是被自身创建的。所以它的\_\_proto\_\_指向了自身的prototype。

Function.prototype指向的对象，它的\_\_proto\_\_也指向Object.prototype。因为Function.prototype指向的对象也是一个普通的被Object创建的对象。

