**深入理解javascript原型和闭包（6）——继承**

<http://www.cnblogs.com/wangfupeng1988/p/3979985.html>

function Foo(){}

var f1 = new Foo();

f1.a = 10;

Foo.prototype.a = 100;

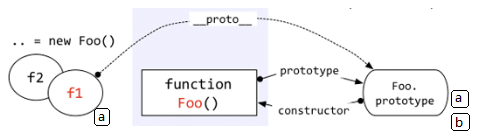
Foo.prototype.b = 200;

console.log(f1.a); //10

console.log(f1.b); //200

以上代码中，f1是Foo函数new出来的对象，f1.a是f1对象的基本属性，f1.b是怎么来的呢？——从Foo.prototype得来，因为f1.\_\_proto\_\_指向的是Foo.prototype

访问一个对象的属性时，先在基本属性中查找，如果没有，再沿着\_\_proto\_\_这条链向上找，这就是原型链。



图中，访问f1.b时，f1的基本属性中没有b，于是沿着\_\_proto\_\_找到了Foo.prototype.b。

在实际应用中如何区分一个属性到底是基本的还是从原型中找到的呢？大家可能都知道答案了——hasOwnProperty，特别是在for…in…循环中，一定要注意。

