**什么是继承？**

其实跟我们生活中的继承是差不多的，就像孩子继承父亲一样。

面向对象的继承：

在原有对象的基础上，略作修改，得到一个新的对象，并且不影响原有对象的功能

在这个里面，需要有两个对象，一个是原有的对象（可以看做是父亲），一个是新对象（可以看做是孩子），新对象继承原有的对象。这就是子类继承父类。

以系统弹窗做比喻，我们一个地方要方的按钮，一个地方要圆的按钮。这样就适合继承来做。

**继承 :** 子类不影响父类，子类可以继承父类的一些功能 ( 代码复用的一种形式 )

**Call**

**Apply**

**bind**

**一、拷贝继承**

原理：属性和方法分开继承

1、属性是通过调父类的构造函数，用call或apply来改this指向。来达到继承的。

2、方法是通过extend封装函数，来实现继承的。

//对象的复制：格式：extend( 新对象,原对象 )

function extend( obj1,obj2 ){

    for( var attr in obj2 ){

        obj1[attr] = obj2[attr];

    }

}

**二、类式继承？**

利用构造函数继承的方式。

Js中没有类的概念。我们可以把构造函数看做是类。所以说类式继承是通过构造函数来实现继承的一种方式。（是从后端演变过来的）

类式继承原理：父级的实例引用赋给子级的原型。

// 父类

function Aaa(){

this.name = '小明';

}

Aaa.prototype.showName = function (){

console.log(this.name);

}

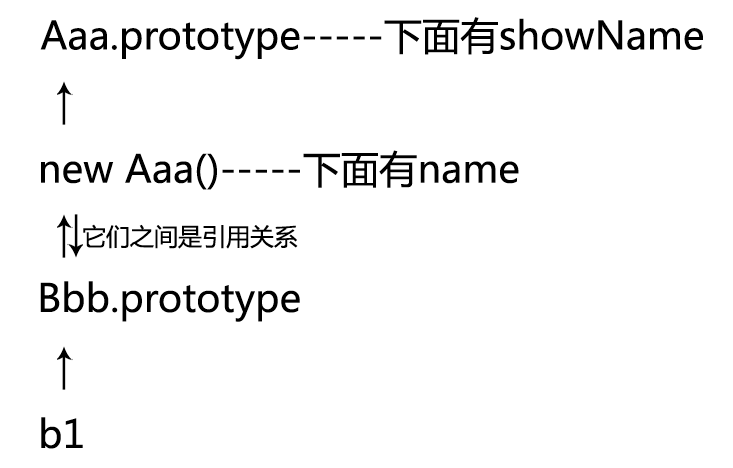
// 子类

function Bbb(){}

Bbb.prototype = new Aaa();

var b1 = new Bbb();

b1.showName();



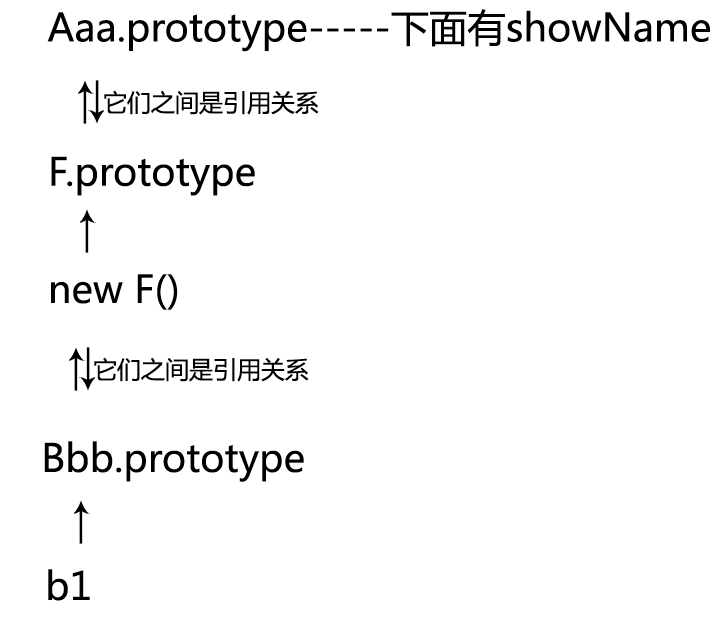
一般会有面试题，如何用一话来做到继承，其实他想要的就是“把父的实例给子的原型”这句话，但是这句话其实是不完善的，它有很多的坑。

坑一：打印子对象的constructor试试。竟然是Aaa。（这个需要再给子对象添加一句改constructor。如Bbb.prototype.constructor = Bbb;）

坑二：如果属性是引用类型，则父对象和子对象引用的是同一个引用类型。

**解决坑：**这种方式，就会过滤掉属性，仅仅只是把方法继承过去了，属性还是要用以前的方法来继承。比如下面，找b1.name，是找不到的，但是找b1.showName是能找到的。





function inherits(Child, Parent) {

var F = function () {};

F.prototype = Parent.prototype;

Child.prototype = new F();

Child.prototype.constructor = Child;

}

**三、原型继承？**

借助原型来实现对象继承对象，同书—原型式继承。（169页）

var a = {

name : '小明'

};

var b = cloneObj(a); //b继承自a

b.name = '小强';

alert( b.name ); //小强

alert( a.name ); //小明

//原型继承函数

function cloneObj(obj){

var F = function(){};

F.prototype = obj;

return new F();

}

归根结底，它只是一个浅复制而已。

ECMAScript 5 通过新增 Object.create() 方法规范化了原型式继承。

这个方法接收两个参数：一个用作新对象原型的对象和（可选的）一个为新对象定义额外属性的对象。在传入一个参数的情况下，Object.create() 与 object() 方法的行为相同。

**总结：**

拷贝继承: 通用型的 有new或无new的时候都可以

类式继承: 适合new构造函数的形式

原型继承: 适合无new的对象

平时使用时，可以根据需要选择，没有好与不好的问题，只有适合不适合的问题。