**继承模式**

1. 圣杯模式(好处：改变Son中的属性不影响Father属性，又可以继承Father中的属性方法)

Father.prototype.name=’chen’;

function Father(){}  
 function Son(){}  
 function Inherit(target,origin){  
 function F(){}  
 F.prototype=origin.prototype;  
 target.prototype=new F();

aa//不能换顺序，target继承的是没有改变指向的 F.prototype，无法继承origin中的属性，目的没有达到  
 target.prototype.constructor=target;  
 //target.prototype.uber=origin.prototype;便于信息保存  
 }  
 Inherit(Son,Father);  
 var son1=new Son();  
 son1.name; //’chen’，此时Son添加属性不再影响Father

1. 圣杯模式继承函数（雅虎工程师写法）

var inherit=(function (target,origin){  
 var F=function(){};  
 return function(target,origin){

F.prototype=origin.prototype;  
 target.prototype=new F();  
 target.prototype.constructor=target;  
 target.prototype.uber=origin.prototype;  
 }

}());

3、传统继承

Father.prototype.name=’chen’;

function Father(){}  
 function Son(){}  
 function Inherit(target,origin){  
 target.prototype=origin.prototype;  
 }  
 Inherit(Son,Father);  
 var son1=new Son();  
 var father1=new Father();  
 son1.sex=’male’;   
 console.log(father1.sex); //’male’

//缺点：在Son中添加方法属性也会添加到Father中