**[JS中的call()和apply()方法](http://uule.iteye.com/blog/1158829)**

**1、方法定义**

**call方法:**   
语法：call([thisObj[,arg1[, arg2[,   [,.argN]]]]])   
定义：调用一个对象的一个方法，以另一个对象替换当前对象。   
说明：   
call 方法可以用来代替另一个对象调用一个方法。call 方法可将一个函数的对象上下文从初始的上下文改变为由 thisObj 指定的新对象。   
如果没有提供 thisObj 参数，那么 Global 对象被用作 thisObj。   
  
**apply方法：**   
语法：apply([thisObj[,argArray]])   
定义：应用某一对象的一个方法，用另一个对象替换当前对象。   
说明：   
如果 argArray 不是一个有效的数组或者不是 arguments 对象，那么将导致一个 TypeError。   
如果没有提供 argArray 和 thisObj 任何一个参数，那么 Global 对象将被用作 thisObj， 并且无法被传递任何参数。

**2、常用实例**

a、

**Java代码  [收藏代码](javascript:void())**

1. function add(a,b)
2. {
3. alert(a+b);
4. }
5. function sub(a,b)
6. {
7. alert(a-b);
8. }
10. add.call(sub,3,1);

 这个例子中的意思就是用 add 来替换 sub，add.call(sub,3,1) == add(3,1) ，所以运行结果为：alert(4); // 注意：js 中的函数其实是对象，函数名是对 Function 对象的引用。

b、

**Java代码  [收藏代码](javascript:void())**

1. function Animal(){
2. **this**.name = "Animal";
3. **this**.showName = function(){
4. alert(**this**.name);
5. }
6. }
8. function Cat(){
9. **this**.name = "Cat";
10. }
12. var animal = **new** Animal();
13. var cat = **new** Cat();
15. //通过call或apply方法，将原本属于Animal对象的showName()方法交给对象cat来使用了。
16. //输入结果为"Cat"
17. animal.showName.call(cat,",");
18. //animal.showName.apply(cat,[]);

 call 的意思是把 animal 的方法放到cat上执行，原来cat是没有showName() 方法，现在是把animal 的showName()方法放到 cat上来执行，所以this.name 应该是 Cat

c、实现继承

**Java代码  [收藏代码](javascript:void())**

1. function Animal(name){
2. **this**.name = name;
3. **this**.showName = function(){
4. alert(**this**.name);
5. }
6. }
8. function Cat(name){
9. Animal.call(**this**, name);
10. }
12. var cat = **new** Cat("Black Cat");
13. cat.showName();

 Animal.call(this) 的意思就是使用 Animal对象代替this对象，那么 Cat中不就有Animal的所有属性和方法了吗，Cat对象就能够直接调用Animal的方法以及属性了.

d、多重继承

**Java代码  [收藏代码](javascript:void())**

1. function Class10()
2. {
3. **this**.showSub = function(a,b)
4. {
5. alert(a-b);
6. }
7. }
9. function Class11()
10. {
11. **this**.showAdd = function(a,b)
12. {
13. alert(a+b);
14. }
15. }
17. function Class2()
18. {
19. Class10.call(**this**);
20. Class11.call(**this**);
21. }

 很简单，使用两个 call 就实现多重继承了  
当然，js的继承还有其他方法，例如使用原型链，这个不属于本文的范畴，只是在此说明call 的用法。说了call ，当然还有 apply，这两个方法基本上是一个意思，区别在于 call 的第二个参数可以是任意类型，而apply的第二个参数必须是数组，也可以是arguments  
还有 callee，caller..

例子来源：<http://xiaofeizm55333.iteye.com/blog/80913>

<http://www.iteye.com/topic/599108>