apply和call

apply:方法能劫持另外一个对象的方法，继承另外一个对象的属性.

```
Function.apply(obj,args)方法能接收两个参数
obj：这个对象将代替Function类里this对象（为空时，默认调用全局对象。）
args：这个是数组，它将作为参数传给Function（args-->arguments）
```

注解：this指的是，调用函数的那个对象

都是因为object没有某个方法，但是别的对象有，可以借助apply或call像别的对象借方法来操作。

> *猫吃鱼，狗吃肉，奥特曼打小怪兽。*
> *有天狗想吃鱼了*
> ***猫.吃鱼.call(狗，鱼)***
> *狗就吃到鱼了*
> *猫成精了，想打怪兽*
> ***奥特曼.打小怪兽.call(猫，小怪兽)***

call需要把参数按顺序传递进去，而apply则是把参数放在数组里。

> - *明确知道参数数量时，用call；*
> - *call需要将传递给函数的参数明确写出来，是多少参数就需要写多少参数*
> - *而不确定的时候，用apply把参数放在数组里传递进去。*

call和apply其实是同一种东西，区别只是参数不同。call其实是apply的语法糖。

题目描述

合并数组 arr1 和数组 arr2。不要直接修改数组 arr，结果返回新的数组

示例1

输入

```
[1, 2, 3, 4], ['a', 'b', 'c', 1]
```

输出

```
[1, 2, 3, 4, 'a', 'b', 'c', 1]
```

方法一：

```
function concat(arr1, arr2) {
    return arr1.concat(arr2);
}
```

方法二：用apply

```
function concat(arr1,arr2){
    var newArr = arr1.slice(0);
    [].push.apply(newArr,arr2);
    return newArr;
}
```