

## 漫谈apply和call

成为一个优秀的javascript编程者，灵活的运用call和apply是必须掌握的技能。

谈到apply和call，那就必须结合this对象。

什么是this对象呢？this指的是当前函数的作用域，一个函数作为一个对象的方法时，那么this就指向这个对象，当一个函数作为一个普通函数进行调用时，那么this指向window（或global）对象。

来看几个例子：

```
global.id = "global";
var obj = {
  id: "obj",
  sayId: function () {
    console.log(this.id);
  }
};
var sayId = obj.sayId;
obj.sayId();
sayId();
```

运行结果为：

```
obj
global
```

上面这个例子我们可以看到，我们把obj对象中的sayId方法给了变量sayId，所以当执行这个sayId方法时，其this指向了global，从运行结果可以很清晰的看到不同。

而apply和call就是为了扩大函数的作用域而生，来看看：

```
var sayId = obj.sayId;
obj.sayId();
sayId();
sayId.apply(obj);
```

其结果为：

```
obj
global
obj
```

这个时候通过apply将原本指向global的this变成了指向obj，成功的改变了函数依赖的作用域。

但咱们的apply和call可不仅仅来干这个，咱们还有第二个参数呢，那是干啥的呢？

第二个参数就是该方法的实参集合，来个例子把：

我们给一个方法函数加一个参数试试：

```
global.name = "lcm";
var obj = {
  name: "yh",
  greet: function (name) {
    console.log("Hello, " + name);
  }
};
var greet = obj.greet;
obj.greet(obj.name);
greet(global.name);
greet.apply(obj, ["qyc"]);
```

结果为：

```
Hello, yh
Hello, lcm
Hello, qyc
```