



链滴

JavaScript `apply` 与 `call` 的异同点

作者: [zhuzhuyule](#)

原文链接: <https://ld246.com/article/1520347285754>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

1.apply()方法 接收两个参数

- 一个是函数运行的作用域 (this)
- 另一个是参数数组。

语法: `apply([thisObj [,argArray]])`;

****说明:** **如果`argArray`不是一个有效 **数组** 或不是 **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**对象将用作 `thisObj`。

2.call()方法 接收多个参数

- 第一个参数和`apply()`方法的一样
- 后继传递给函数的多个参数必须列举出来。

语法: `call([thisObject[,arg1 [,arg2 [,...,argn]]]])`;

说明: `call` 方法可以用来代替另一个对象调用一个方法, `call` 方法可以将一个函数的对象上下文从初始的上下文改变为 `thisObj` 指定的新对象, 如果没有提供 `thisObj` 参数, 那么**Global**对象被用于 `thisObj`。

eg.1

```
window.color = 'red';
document.color = 'yellow';
var s1 = {
  color: 'blue'
};
function changeColor() {
  console.log(this.color);
}
//call()
changeColor.call(); //red (默认传递参数)
changeColor.call(window); //red
changeColor.call(document); //yellow
changeColor.call(this); //red
changeColor.call(s1); //blue
//apply()
changeColor.apply(); //red (默认传递参数)
changeColor.apply(window); //red
changeColor.apply(document); //yellow
changeColor.apply(this); //red
changeColor.apply(s1); //blue
```

eg.2

```
function add(c, d) {
  return this.a + this.b + c + d;
}
var s = {
```

```
a: 1,  
b: 2  
};  
console.log(add.call(s, 3, 4)); // 1+2+3+4 = 10  
console.log(add.apply(s, [5, 6])); // 1+2+5+6 = 14
```

eg.3

```
window.firstName = "Cynthia";  
window.lastName = "_xie";  
var myObject = {  
  firstName: 'my',  
  lastName: 'Object'  
};  
  
function getName() {  
  console.log(this.firstName + this.lastName);  
}  
  
function getMessage(sex, age) {  
  console.log(this.firstName + this.lastName + " 性别: " + sex + " age: " + age);  
}  
getName.call(window); //Cynthia_xie  
getName.call(myObject); //myObject  
getName.apply(window); //Cynthia_xie  
getName.apply(myObject); //myObject  
getMessage.call(window, "女", 21); //Cynthia_xie 性别: 女 age: 21  
getMessage.apply(window, ["女", 21]); //Cynthia_xie 性别: 女 age: 21  
getMessage.call(myObject, "未知", 22); //myObject 性别: 未知 age: 22  
getMessage.apply(myObject, ["未知", 22]); //myObject 性别: 未知 age: 22
```

原文: <http://blog.csdn.net/ganyingxie123456/article/details/70855586>