

Javascript 的 this 用法

作者：阮一峰

日期：2010年4月30日

感谢 [腾讯课堂NEXT学院](#) 赞助本站，[腾讯官方的前端培训](#) 正在招生中。



this 是 JavaScript 语言的一个关键字。

它是函数运行时，在函数体内部自动生成的一个对象，只能在函数体内部使用。

```
function test() {  
  this.x = 1;  
}
```

上面代码中，函数 test 运行时，内部会自动有一个 this 对象可以使用。

那么，this 的值是什么呢？

函数的不同使用场合，this 有不同的值。总的来说，this 就是函数运行时所在的环境对象。下面分四种情况，详细讨论 this 的用法。

情况一：纯粹的函数调用

这是函数的最通常用法，属于全局性调用，因此 this 就代表全局对象。请看下面这段代码，它的运行结果是1。

```
var x = 1;  
function test() {  
  console.log(this.x);  
}  
test(); // 1
```

情况二：作为对象方法的调用

函数还可以作为某个对象的方法调用，这时 `this` 就指这个上级对象。

```
function test() {  
  console.log(this.x);  
}  
  
var obj = {};  
obj.x = 1;  
obj.m = test;  
  
obj.m(); // 1
```

情况三 作为构造函数调用

所谓构造函数，就是通过这个函数，可以生成一个新对象。这时， `this` 就指这个新对象。

```
function test() {  
  this.x = 1;  
}  
  
var obj = new test();  
obj.x // 1
```

运行结果为1。为了表明这时`this`不是全局对象，我们对代码做一些改变：

```
var x = 2;  
function test() {  
  this.x = 1;  
}  
  
var obj = new test();  
x // 2
```

运行结果为2，表明全局变量 `x` 的值根本没变。

情况四 **apply** 调用

`apply()` 是函数的一个方法，作用是改变函数的调用对象。它的第一个参数就表示改变后的调用这个函数的对象。因此，这时 `this` 指的就是这第一个参数。

```
var x = 0;
function test() {
  console.log(this.x);
}

var obj = {};
obj.x = 1;
obj.m = test;
obj.m.apply() // 0
```

`apply()` 的参数为空时，默认调用全局对象。因此，这时的运行结果为 `0`，证明 `this` 指的是全局对象。

如果把最后一行代码修改为

```
obj.m.apply(obj); //1
```

运行结果就变成了 `1`，证明了这时 `this` 代表的是对象 `obj`。

(完)

文档信息

- 版权声明： 自由转载-非商用-非衍生-保持署名（[创意共享3.0许可证](#)）
- 发表日期： 2010年4月30日