**javascript中匿名函数的多种调用方法**

Javascript中定义函数的方式有多种，函数直接量就是其中一种。如var fun = function(){},这里function如果不赋值给fun那么它就是一个匿名函数。好，看看匿名函数的如何被调用。

**方式1，调用函数，得到返回值。强制运算符使函数调用执行**

|  |  |
| --- | --- |
| 1  2  3  4 | (function(x,y){      alert(x+y);      return x+y;  }(3,4)); |

**方式2，调用函数，得到返回值。强制函数直接量执行再返回一个引用，引用再去调用执行**

|  |  |
| --- | --- |
| 1  2  3  4 | (function(x,y){      alert(x+y);      return x+y;  })(3,4); |

这种方式也是很多库爱用的调用方式，如[jQuery](http://jquery.com/" \t "_blank)，[Mootools](http://mootools.net/" \t "_blank)。

**方式3，使用void**

|  |  |
| --- | --- |
| 1  2  3  4 | void function(x) {        x = x-1;        alert(x);  }(9); |

**方式4，使用-/+运算符**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19 | -function(x,y){      alert(x+y);      return x+y;  }(3,4);    +function(x,y){      alert(x+y);      return x+y;  }(3,4);    --function(x,y){      alert(x+y);      return x+y;  }(3,4);    ++function(x,y){      alert(x+y);      return x+y;  }(3,4); |

**方式5，使用波浪符(~)**

|  |  |
| --- | --- |
| 1  2  3  4 | ~function(x, y) {      alert(x+y);     return x+y;  }(3, 4); |

**方式6，匿名函数执行放在中括号内**

|  |  |
| --- | --- |
| 1  2  3 | [function(){     console.log(this) // 浏览器得控制台输出window  }(this)] |

**方式7，匿名函数前加typeof**

|  |  |
| --- | --- |
| 1  2  3 | typeof function(){     console.log(this) // 浏览器得控制台输出window  }(this) |

**方式8，匿名函数前加delete**

|  |  |
| --- | --- |
| 1  2  3 | delete function(){     console.log(this) // 浏览器得控制台输出window  }(this) |

**方式9，匿名函数前加void**

|  |  |
| --- | --- |
| 1  2  3 | void function(){     console.log(this) // 浏览器得控制台输出window  }(this) |

**方式10，使用new方式，传参**

|  |  |
| --- | --- |
| 1  2  3 | new function(win){     console.log(win) // window  }(this) |

**方式11，使用new，不传参**

|  |  |
| --- | --- |
| 1  2  3 | new function(){      console.log(this) // 这里的this就不是window了  } |

**方式12，逗号运算符**

|  |  |
| --- | --- |
| 1  2  3 | 1, function(){      console.log(this) // window  }(); |

**方式13，按位异或运算符**

|  |  |
| --- | --- |
| 1  2  3 | 1^function(){      console.log(this) // window  }(); |

**方式14，比较运算符**

|  |  |
| --- | --- |
| 1  2  3 | 1>function(){      console.log(this) // window  }(); |

最后看看错误的调用方式

|  |  |
| --- | --- |
| 1  2  3  4 | function(x,y){      alert(x+y);      return x+y;  }(3,4); |