[1js闭包 3](#_Toc476905972)

# js匿名函数与闭包

## 1.1解决大量全局变量的问题

<http://www.jb51.net/article/79238.htm>

**注意**

1 闭包允许内层函数引用父函数中的变量，但是该变量是最终值

[?](http://www.jb51.net/article/79238.htm)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15 | /\*\*  \* <body>  \* <ul>  \* <li>one</li>  \* <li>two</li>  \* <li>three</li>  \* <li>one</li>  \* </ul>  \*/  var lists = document.getElementsByTagName('li');  for(var i = 0 , len = lists.length ; i < len ; i++){  lists[ i ].onmouseover = function(){  alert(i);  };  } |

你会发现当鼠标移过每一个<li>元素时，总是弹出4，而不是我们期待的元素下标。这是为什么呢？注意事项里已经讲了（最终值）。显然这种解释过于简单，当mouseover事件调用监听函数时，首先在匿名函数（ function(){ alert(i); }）内部查找是否定义了 i，结果是没有定义；因此它会向上查找，查找结果是已经定义了，并且i的值是4（循环后的i值）；所以，最终每次弹出的都是4。

解决方法一：

[?](http://www.jb51.net/article/79238.htm)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8 | var lists = document.getElementsByTagName('li');  for(var i = 0 , len = lists.length ; i < len ; i++){  (function(index){  lists[ index ].onmouseover = function(){  alert(index);  };  })(i);  } |

解决方法二：

[?](http://www.jb51.net/article/79238.htm)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7 | var lists = document.getElementsByTagName('li');  for(var i = 0, len = lists.length; i < len; i++){  lists[ i ].$$index = i; //通过在Dom元素上绑定$$index属性记录下标  lists[ i ].onmouseover = function(){  alert(this.$$index);  };  } |

解决方法三：

[?](http://www.jb51.net/article/79238.htm)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9 | function eventListener(list, index){  list.onmouseover = function(){  alert(index);  };  }  var lists = document.getElementsByTagName('li');  for(var i = 0 , len = lists.length ; i < len ; i++){  eventListener(lists[ i ] , i);  } |

2 内存泄露

使用闭包十分容易造成浏览器的内存泄露，严重情况下会是浏览器挂死