

你可曾见过如此简单粗暴的JavaScript解说 -- if 判断的正确打开方式？

在JavaScript中，对于 **if else** 的逻辑判断你肯定非常熟悉，本文罗列了几种你不一定知道的简写方式，仅供参考。

例子：

已知小明考了68分，小于60分为不及格，大于60分为及格，问：小明是否通过了考试？

1.传统的写法

```
if(score < 60){
    alert('不及格');
}else{
    alert('及格');
}
```

这没什么好说的，最基本的**if else**语法。可读性应该是最好的。

2.利用三目运算符来进行判断

```
score < 60 ? alert('不及格') : alert('及格') ;
```

这种写法应该还算是比较多见的，而且，如果只有单纯的一个**if**判断的话，可读性也不是很差。

3.利用&& 来判断

```
score < 60 && alert('不及格') ;
score >= 60 && alert('及格') ;
```

&&的用法： 如果左边的结果为真，那么运行右边的表达式，如果右边的也为真，那么就返回真。其中，只要有一个为假，结果就是假。并且，一旦左边的结果为假，右边的就不执行了。

这里就是取了一个巧，因为我只是想要执行右边的表达式，无所谓返回值是什么。

那么，如果score<60 为真，我不管三七二十一，都去执行一下右边的代码。我们的目的已经达到了。

这不就相当于这样吗：

```
if(score < 60 == true){
    alert('不及格') ;
}

if(score >= 60 == true){
    alert('及格') ;
}
```

简写：

```
if(score < 60){
    alert('不及格') ;
}

if(score >= 60){
    alert('及格') ;
}
```

就是说，一旦 () 里面的表达式为真，就执行 {} 里面的内容。否则就不执行。

4.利用 || 来判断

```
score < 60 || alert('及格') ;
score >= 60 || alert('不及格') ;
```

|| 的用法： 如果左边的结果为真，那么直接返回true，右边的就忽略了！

如果左边为假，那么就运行右边的，看看右边的结果如何？如果右边为真，就返回真，否则返回假。

也就是说，不管右边的结果怎样，只要左边为假了，右边就无论如何会被执行一次的。这边也就是利用这个原理，相当于这样：

```
if(score < 60 == false){
    alert('及格') ;
}

if(score >= 60 == false){
```

```
    alert('不及格') ;  
}
```

如果分数 < 60 是假的，是骗你的，不就是说分数 > 60 才是真相吗？哦，超过60分就是及格了，所以alert出来的是及格。

如果分数 >= 60 是假的，是骗你的，不就是说分数 < 60 才是真相吗？哦，不满60分就是不及格了，所以alert出来的是不及格。

这样应该说的比较清楚了吧。

再说得白一点，就是：

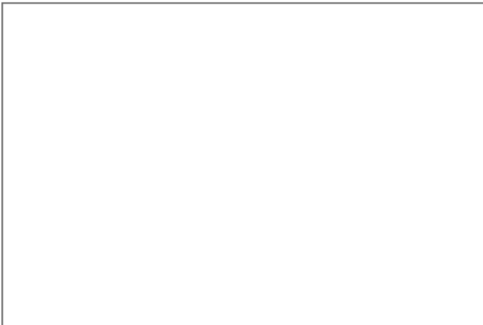
你要么给我 < 60 ,要么给我 alert('及格')；

你要么给我 >= 60 ,要么给我 alert('不及格')；

5.实战演练

我们来一个小案例，看看可以怎么运用这些知识点？

```
<!DOCTYPE html>  
<html>  
  <head>  
    <meta charset="UTF-8">  
    <script src="http://libs.baidu.com/jquery/2.0.0/jquery.min.js"></script>  
    <style type="text/css">  
      .box1 {  
        height:200px;  
        width:300px;  
        border: 1px solid #666;  
      }  
  
      .box2 {  
        background: green;  
      }  
    </style>  
    <script>  
  
    </script>  
  </head>  
  <body>  
    <div id='myBox' class='box1'></div>  
  </body>  
</html>
```



我们引入了jQuery，现在希望自己来编写一个toggleClass，如何实现呢？

当我们点击这个div的时候，就自动将class换成box2。再次点击的时候，就把box2的样式去掉，依此循环。

上代码：

```
$( 'document' ).ready(function() {  
  function toggleClass(dom , className) {  
    //判断该class是否已经存在  
    if (dom.hasClass(className)) {  
      //如果存在，就去掉这个class  
      dom.removeClass(className);  
    } else {  
      //如果不存在，就加上这个class  
      dom.addClass(className);  
    }  
  }  
  
  $( '#myBox' ).click(function() {  
    toggleClass( $(this), 'box2' );  
  });  
});
```

```
});  
});
```



我猜想，jQuery对toggleClass的实现应该也差不多是这个意思。现在，我们考虑做如下简化：

第一种方式，用三目运算符简化：

```
function toggleClass(dom , className){  
    dom.hasClass(className) ? dom.removeClass(className) : dom.addClass(className);  
}
```

第二种方式，用&& 和 ||：

```
!dom.hasClass(className) && dom.addClass(className) || dom.removeClass(className);
```

可能有点绕，而且我也不是很推荐这种方式，一般来说，这种方式用于一些给变量赋值的场合。

我大概解释一下：

```
!dom.hasClass(className) && dom.addClass(className)
```

这表示如果dom元素不包含你指定的className，那么 !dom.hasClass(className) 是真，于是无论如何我都要执行右边的代码：dom.addClass(className)，也就是说，给dom元素添加这个class。

这样一来，

```
!dom.hasClass(className) && dom.addClass(className)
```

这句话是不是已经成立了，OK，那么因为右边是 ||， 左边都已经成立了，那么我右边的代码是不是可以不执行了呀？

好的，第二次点击：

!dom.hasClass(className)的意思是说，你这个dom元素是不是没有指定的class啊，不，我们刚才已经添加了，所以是有的。

那么，这句话就不成立了，&& 左边的是假，那么右边的就不执行了、

就是说：

```
dom.addClass(className)
```

这句话不执行了。

综合来看：

```
!dom.hasClass(className) && dom.addClass(className)
```

这个条件就是不成立的。

好的，|| 左边的不成立，是不是说，我们无论如何都要执行一下||右边的代码呢？

恩， dom.removeClass(className) 就被执行了。

本章结束 ...

剽悍一小兔，电气自动化毕业。

参加工作后对计算机感兴趣，深知初学编程之艰辛。
希望将自己所学记录下来，给初学者一点帮助。

我的个人博客：<http://s-335245.gotocdn.com:8080/index.html>

简书地址：http://www.jianshu.com/users/8875b36caaec/latest_articles

免责声明： 博客中所有的图片素材均来自百度搜索，仅供学习交流，如有问题请联系我，侵立删，谢谢。