

15天学会jQuery (0-5)

■ 15 Days of jQuery(Day 0)---jQuery - What, Why, When, Where, Who

what

jQuery是一个了不起的javascript库，它可以是我们用很少的几句代码就可以创建出漂亮的页面效果。从网站的方面说，这使得javascript更加有趣。

如果你这样想：“孩子，我需要另外一个javascript库，就好比 I need another hole in my head” 那么加入这个俱乐部吧。这正是我第一次遇到的时候所想的。

我已经用过了Moo.fx, Scriptaculous, TW-SACK, 和 Prototype. 我曾参与了RICO, Yahoo YUI和其他一些库的开发。

没有了PHPjavascript和我一点也不亲近了。但是我还是尽全力保持头脑清醒，并尽量保持用AJAX去思考。

所以当我遇到jQuery的时候我想：“还需要另外一个javascript库吗？不了，谢谢...”

why

为什么我改变我对jQuery的看法，以及为什么你要考虑去使用它？很简单，只要你看一眼过使用jQuery的页面你就会发现它是如此的简单易用。只用很少的几行，就能表现出很优雅的效果。有一天当我突然看到一些用jQuery写的代码时我一下子豁然开朗了。早茶的过程中，我例行公务的去翻阅我的订阅，去看每日必看的设计博客的时候我看到了一个用jQuery写的javascript的例子。事实证明，这些代码还是有些和浏览器关联的bug，不过这些概念还是我以前从来没有见过的。

还有那些代码...

代码看起来很简单看起来不像我以前见过的。但也不无道理。

我开始通读文档，并且惊奇的发现用一点点代码竟然能做这么多事情。

when

你应当在你需要的时候使用jQuery.

给你一个小型的库文件 DOM强大的控制能力 不费吹灰之力的工作，和少许的努力。

或者

快速的通过AJAX 没有大量无用的代码 和一些基本的动画效果

但是

如果你需要超级花式效果，动画，拖放，和超级平稳动画，那么你可能想使用Prototype.他是一个有大量动画效果的类库。

where

你可以jQuery的官方网站下载到他的源代码(10K).

who

jQuery was created by John Resig.

■ 15 Days of jQuery(Day 1)---比window.onload更快一些的载入

window.onload()是传统javascript里一个能吃苦耐劳的家伙。它长久以来一直被程序员们作为尽快解决客户端页面载入问题的捷径。

但有时候等待页面载入还是不够快。

只有少数大型的图片文件会被快速的载入，而大部分大型的图片文件会使window.onload()载入的很慢。所以当我为最近的网络营销创建一个web应用程序的时候我不得希望更快一点。有一些围绕window.onload()的新研究(比如brother cake)的代码是一种快速的方式。如果你需要，可以试试。

但是如果你要做一些DOM(文档对象模型)javascript的编程，那么你为什么试试jQuery，它就像你自己亲自制作一个蛋糕，并品尝它。（双关Brother Cake，俏皮话）。

jQuery有一个用来作为DOM快速载入javascript的得心应手的小函数，那就是ready... 他在页面加载完成之后执行。

```
$(document).ready(function(){
// Your code here...
});
```

你可以用他来载入任何你想要载入的javascript，并不一定要保留jQuery的编码风格。让jQuery同时去执行多个函数也是可以的。

你以前可能见过类似于init()之类的函数，你会发现jQuery会快很多。

在以后的教程里我们会一遍又一遍的用到这个函数。

OK你现在可以尝试一下代码：(记得把jQuery引用进你的页面哦，不记得的话看看上篇。)

```
$(document).ready(function(){
alert("Congratluations!");
});
```

很Easy，不是吗？ 用几分钟时间做的双色表格。

■ 15 Days of jQuery(Day 2)---很容易的制作双色表格

这节本身没有太多的价值，重点在它提供的这个例子上。我将代码帖出来然后对重点部分注释一下：我们先来看看Thewatchmakerproject传统的做法：预览地址(你可以查看一下源代码)。再来看看jQuery是如何用5行代码来搞定的：

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=gb2312" />
<title>StripingTable</title>
<script src="jquery-latest.pack.js" type="text/javascript"></script>
<!--将jQuery引用进来-->
<script type="text/javascript">
$(document).ready(function(){ //这个就是传说的ready
    $(".stripe tr").mouseover(function(){
        //如果鼠标移到class为stripe的表格的tr上时，执行函数
        $(this).addClass("over");}).mouseout(function(){
        //给这行添加class值为over，并且当鼠标一出该行时执行函数
```

```
$(this).removeClass("over");}) //移除该行的class
$(".stripe tr:even").addClass("alt");
//给class为stripe的表格的偶数行添加class值为alt
});
</script>
<style type="text/css">
th {
    background:#0066FF;
    color:#FFFFFF;
    line-height:20px;
    height:30px;
}

td {
    padding:6px 11px;
    border-bottom:1px solid #95bce2;
    vertical-align:top;
    text-align:center;
}

td * {
    padding:6px 11px;
}

tr.alt td {
    background:#ecf6fc; /*这行将给所有的tr加上背景色*/
}

tr.over td {
    background:#bcd4ec; /*这个将是鼠标高亮行的背景色*/
}

</style>
</head>

<body>
<table class="stripe" width="50%" border="0" cellspacing="0" cellpadding="0">
<!--用class="stripe"来标识需要使用该效果的表格-->
<thead>
<tr>
<th>姓名</th>
<th>年龄</th>
<th>QQ</th>
<th>Email</th>
</tr>
</thead>
<tbody>
<tr>
<td>邓国梁</td>
<td>23</td>
<td>31540205</td>
<td>gl.deng@gmail.com</td>
</tr>
<tr>
<td>邓国梁</td>
<td>23</td>
<td>31540205</td>
<td>gl.deng@gmail.com</td>
</tr>
<tr>
<td>邓国梁</td>
<td>23</td>
<td>31540205</td>
<td>gl.deng@gmail.com</td>
</tr>
<tr>
<td>邓国梁</td>
<td>23</td>
<td>31540205</td>
<td>gl.deng@gmail.com</td>
</tr>
<tr>
<td>邓国梁</td>
<td>23</td>
```

```
<td>31540205</td>
<td>gl.deng@gmail.com</td>
</tr>
</tbody>
</table>
<p>PS: 飘飘说我的table没有<thead>,我知错了...</p>
</body>
</html>
```

预览地址

这里有一个jQuery的技巧不得不提一下: jQuery的链式操作, 什么是链式操作呢? 我们来看看, 本来应该写成这样的:

```
$(".stripe tr").mouseover(function(){
    $(this).addClass("over");})
$(".stripe tr").mouseout(function(){
    $(this).removeClass("over"); })
```

但是我们写成了:

```
$(".stripe tr").mouseover(function(){
    $(this).addClass("over");}).mouseout(function(){
    $(this).removeClass("over");})
```

在jQuery中, 执行完mouseover或者mouseout等方法之后, 都会返回当前的对象, 所以可以进行链式操作

*

15 Days of jQuery(Day 3)---巧妙的伪装链接

今天的教程是草草完成的.我想把一些东西放在这15天的前面简单的讲讲,这样以来就可以使一些js新手不至于被一堆代码搞的晕头转向.事实上我是在即将结尾的时候才做出的这个决定.

目标

我们要使用jQuery去创建一小段代码,这段代码会把一个页面所有的超链接转换并且伪装起来.

为什么?

一些下属经销商认为,一部分用户有足够的悟性发现会员链接,并能尽量避免通过点击URL链接直接进入浏览器,从而“欺骗”下属经销商脱离代理(假设购买行为已经发生)

“老”办法

有很多下属经销商千方百计的掩饰他们的链接,避免人们通过链接找到他们.这些伎俩涉及到.htaccess和服务器的代码.但对于本教程,我会将重点放到“老学校”的javascript上:

```
<a onMouseOver='window.status="http://www.merchant-url-here.com";
return true;' onMouseOut='window.status="Done"; return true;'
href="http://www.affiliate-url-here.com"
target="_blank">Link Text Here</a>
```

这段代码被用来在浏览器状态栏显示用户鼠标指向的链接地址.比如实际链接是www.website.com?aff=123,则可以在状态栏显示www.website.com.

问题

你是一个很活跃的下级经销商,你可能会以疯狂的速度给很多个页面添加链接,并且还要给每个链接添加这种效果那么你会肯定会的.加入有一天你要修改任务栏里显示的链接的时候估计你会疯掉的.

jQuery的解决办法

首先,我们想让javascript尽快的掩饰我们的链接所以我们应该先从这里开始:

```
<script src="jquery.js"></script>
<script type="text/javascript">
$(document).ready(function(){
//code goes here
});
</script>
```

当DOM准备好的时候我们放在ready里的代码就开始执行了.

接下来

要给所有我们想伪装的链接添加一个class,class有助于jQuery帮我们找到需要伪装的链接而撇开其它不需要伪装的链接.title有两个作用:当鼠标划过链接的时候会有一个小小的盒状提示显示URL:www.affsite.com并且同样的信息会显示在浏览器的状态栏(IE Only).

```
<p><a href="http://www.affsite.com?id=123" title="http://www.affsite.com"
class="affLink">Super Duper Product</a></p>
```

告诉jQuery找到有class= “affLink” 的链接

```
$('.a.affLink')
```

添加一个鼠标划过事件

```
$('.a.affLink').mouseover(function(){window.status=this.title;return true;});
```

简单的说:找到class= “affLink” 的所有链接,当鼠标划过它们的时候改变浏览器状态栏信息为该链接title的内容.这个在Firefox并不能正常的工作,只是在IE里起作用.在Firefox的状态栏只是显示一个” Done”,或者更准确的说,鼠标划过超链接对状态栏并没有任何影响.我没有更多的浏览器测试.

继续-mouseout jQuery可以让我们用”链”的方式,像这样:

```
$('.a.affLink').mouseover(function(){window.status=this.title;return true;})
.mouseout(function(){window.status='Done';return true;});
```

这点代码告诉jQuery改变浏览器状态栏信息,或者当鼠标不再停留在链接上时返回” Done”. 如果你不适应jQuery的这种链的工作方式,那么你可以这样写:

```
$('.a.affLink').mouseover(function(){window.status=this.title;return true;});
$('.a.affLink').mouseout(function(){window.status='Done';return true;});
```

这就看你了.

把这些代码放到一起:

```
<script src="jquery.js"></script>
<script type="text/javascript">
$(document).ready(function(){
```

```
$('#a.affLink').mouseover(function(){window.status=this.title;return true;})  
.mouseout(function(){window.status='Done';return true;});  
});  
</script>
```

最后的想法 你们当中可能觉得今天的课程太简单了,有些还可能还是有点不太明白,因为你们不是二级经销商。

In “Days” to come you’ ll see me tackle issues that get more involved and apply to almost anyone with a website - whether you monetize your traffic or not.

■ 15 Days of jQuery(Day 4)---安全邮件列表

规则提到如何防止垃圾邮件:不要把你的邮件地址放到任何一个mailto:链接中. 在与垃圾邮件恶魔做斗争的过程中我们的网页设计师和程序员总结出了一些有创意的解决办法,让我们快速的看一些这些常见方法的缺点(或多或少有一些).

name [at-no-spam] website.com

问题:链接式的更方便,而且把邮件地址敲入收件人栏还有可能会出错.

联系方式

问题:你冒着这么大的风险就是因为有一个垃圾邮件借用你的帐户发送大量的垃圾邮件(除非你使用真正的安全邮件脚本).而这样就扼杀了那些只想给你发个简单邮件的用户.

javascript加密

问题:你的邮件仍然暴露在光天化日之下,即使你使用了复杂的密码技术给它带上面具.还有谁希望为了发送一封邮件而启用第三方的解密网站,反正我是不会.

藏在一种简单的形式后面 (有一个例子,但是打不开了.)<http://simon.incutio.com/contact/>我能想到的没有人...但是让我们看看是否我们能改进观念。

可能的解决办法:AJAX

我提供的解决方案将比目前设计师们使用的方法有如下优势:

- 易于实施
- 易于修改
- 还有一些小小的花哨的效果
- 不用第三方工具来加密邮件地址
- 没有邮件地址暴露在光天化日之下

最后我想说明一点,我认为电子邮件靠这种闪烁其词的加密手段来躲避垃圾邮件还是非常不明智的.在实践中,我认为电子邮件加密是相对安全的,但是客观事实是,电子邮件还是在html源代码里.

概念

1. 1.用jQuery从服务器上把html文件内容抓下来.
2. 2.把包含邮件链接的html文件放到好的容器中是一种简单的保护机制.

示例

我要示范一些例子来显示邮件链接地址,当访客点击一个按钮或者一个链接的时候,页面就会跳转到对应的那个例

子里.

按钮点击--立即显示

链接点击--淡出

页面载入--淡出

页面载入--立即显示

(说明:所谓的立即显示,我的意思是说”没有花哨的效果而尽快的显示电子邮件地址”)

代码

这里发表非商业共创使用许可,如果你希望将代码使用在你的商业产品中时,请联系我.我正在一个新的CMS for web designers中使用它.

为什么这种方式比普通的mailto链接安全?

真正的问题是垃圾邮件制造者会使用自动化软件从html源文件中寻找电子邮件链接,这种做法和google一样:使用相关链接. 他么就和我们大部分人一样懒惰.所以很难所他们不会拿个本子放在键盘旁边记下你的电子邮件地址. 请查看我提供的示例的源代码,你将不会在html里找到任何的邮件地址. 这儿坚实的保证了你绝对不会收到垃圾邮件,只会从朋友或者浏览者那里收到邮件. 但是从页面中移除邮件地址,.....

最后一点说明

先仔细看看前面三个例子,你会看到我使用了AJAX回调函数来触发slideDown() 和 show() 效果. 换句话说,我希望AJAX调用收到信息(html)时jQuery才开始slideDown() 效果.把秘密粘贴到我们简单的服务段脚本并且等待服务器返回信息.

正确的方法:

```
$(document).ready(function(){
$.post('mailtoInfo.php',{
pass: "secret"
},function(txt){
$('div.email').html(txt);
$('div.email').slideDown("slow");
});
});
```

错误的方法:

```
$(document).ready(function(){
$.post('mailtoInfo.php',{
pass: "secret"
},function(txt){
$('div.email').html(txt);
});
$('div.email').slideDown("slow");
});
```

■ 15 Days of jQuery(Day 5)---包起来--懒人用Jquery生成的HTML

这个让我们轻松的纪念日已经到来 - 我恨我在计算机前已经花了48个小时,我希望能够有另外一个jQuery来结束我的噩梦,并且让我上网更快.

当我一边“在用Jquery方法编写”和一边“进行复杂的文件上传”，我已经筋疲力尽。然而这两种操作的代码是一种较浅的，它只不过是才刚刚开始解决的一些简单问题。

所以下来我开始介绍：

尽管我在我的网站用所有的CSS样式表去进行表格设计（也许这要花费两年半的时间或者更多），我已经用了许多我能找到的在这方面的信息。回到2004年5月(古代史)A list a part有一篇《关于创建阴影的伟大教程(洋葱皮投影)》可以应用到任何图片,无论尺寸多大。

那片文章的应经不能再评论了,但还是有些人希望能够再出篇教程。

问题

一些css工程师用一些”不相干”的标记,就是为了使背景图片能够应用到每一个元素上.例如：

这里是A list a part用到的代码：

```
<div class="wrap1">
<div class="wrap2">
<div class="wrap3">

</div>
</div>
</div>
```

所有这些divs充当一个给图片添加投影效果的”钩子”。这不见得好,我们可以把源代码精简成这样：

```

```

按着这个思路...

目标

在这里,我要想你展示如何用jQuery轻而易举的将多于的标记从你的源代码中剔除.运用这个方法,让你的代码更加干净(更重要的是)将使以后变换布局容易的多。

解

这里,看看jQuery是如何击退这个问题的。

```
$(document).ready(function(){
$("img.dropshadow")
.wrap("<div class='wrap1'><div class='wrap2'>" +
"<div class='wrap3'></div></div></div>");
});
```

图片就可以保持这样了：

```

```

仔细看看

\$(document).ready() 是jQuery版的window.onload()

\$(“img.dropshadow”) 告诉jQuery找到带有class= “dropshadow” 的图片,如果你想用一个id你可以这样: \$(“img#dropshadow”)wrap() (wrap() tells jQuery to use the DOM (Document Object

Method Model) to wrap the images with the class="dropshadow" in the html inside the parenthesis.)

最后的结果

傻乎乎的图片,但是和original Onion Skinned Drop Shadows用的是一样的.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title>Onion Skin DropShadwo with jQuery</title>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />

<style>
.wrap0, .wrap1, .wrap2, .wrap3 {
display:inline-table;
/* \*/display:block;/**/
}
.wrap0 {
float:left;
background:url(shadow.gif) right bottom no-repeat;
}
.wrap1 {
background:url(shadow180.gif) no-repeat;
}
.wrap2 {
background:url(corner_bl.gif) -18px 100% no-repeat;
}
.wrap3 {
padding:10px 14px 14px 10px;
background:url(corner_tr.gif) 100% -18px no-repeat;
}
body { background: #fff;}
</style>
<script src="jquery.js" type="text/javascript"></script>

<script>
$(document).ready(function(){
$("img.dropshadow")
.wrap("<div class='wrap0'><div class='wrap1'><div class='wrap2'>" +
"<div class='wrap3'></div></div></div></div>");
});
</script>
</head>

<body>
<h1>Onion Skinned - With jQuery</h1>
<p>First, the old-school, multiple divs hard coded into the html as seen on the <a href="http://www.f
<div class="wrap0">
<div class="wrap1">
<div class="wrap2">
<div class="wrap3">

</div>
</div>
</div>
</div>

<p style="clear:both;">And now, the jQuery method, which uses javascript to wrap the image at runtime

<p>View the source of this page and you'll see the huge difference in markup!</p>
</body>
</html>
```

(这里只是代码,没有图片.要看效果去这里)

jQuery和其它解决方法的比较

jQuery的网站上有一个到Ajaxian网站的链接,那里有用另外一个javascript库创建的Onion Skin Drop Shadow,我相信他的代码复杂程度和代码量现在来自不待言.我宁愿使用jQuery.(怎么?你猜到了..)

平心而论,没有一个库是对于每一个工作或每一段代码都是合适的.本教程不是为了证明jQuery是一切javascript类库中的老大.试试Prototype, Scriptaculous, YUI, Rico, Behaviour, Moo.fx 和 the dozens 或者其它的.如果你找到了一个你用起来比较顺手的,那就去用它吧.

jQuery对于我来说只是一个工具.我只是希望这个教程能够提供给你更多使用它的方法.

有关jQuery的工具

jQuery用难以置信的简单来操作DOM. 你应该花些时间看看jQuery能用来做什么,用下append(), prepend(), before(), after(), html(), and remove().

来自jQuery Docs

wrap(String html)

把所有匹配的元素用其他元素的结构化标记包装起来.这种包装对于在文档中插入额外的结构化标记最有用,而且它不会破坏原始文档的语义品质.

这个函数的原理是检查提供的第一个元素(它是由所提供的HTML标记代码动态生成的),并在它的代码结构中找到最上层的祖先元素——这个祖先元素就是包装元素.

当HTML标记代码中的元素包含文本时无法使用这个函数.因此,如果要添加文本应该在包装完成之后再行添加.

示例:

```
$( "p" ).wrap( "<div class='wrap'></div>" );
```

HTML

```
<p>Test Paragraph.</p>
```

结果

```
<div class='wrap'><p>Test Paragraph.</p></div>
```