



附录 A

关于\$(document).ready()函数

A.1 \$(document).ready()函数介绍

学习 jQuery 的第 1 个步骤就应该学习\$(document).ready()函数。例如要在页面上运行一个 JavaScript 函数，那么就应该将它写在\$(document).ready()函数内。在该函数内的所有代码都将在 DOM 加载完毕后，页面全部内容（包括图片等）完全加载完毕前被执行。

jQuery 代码如下：

```
$(document).ready(function() {  
    //将 jQuery 代码放在这里  
});
```

\$(document).ready()函数相比其他获得 JavaScript 事件并执行相应事件的函数有很多优势。

该函数不需要在 HTML 代码中进行任何“行为”标记，就可以分离 JavaScript 代码到每一个独立的文件中。因此很容易对与内容无关的代码进行维护。当光标在一个链接上悬停时，在浏览器的状态栏中会出现“javascript:void()”的信息，该信息就是由于在一个<a>标签中直接写入一个事件而产生的。

在一些传统的页面中，可能会在<body>标签里有 onload 属性，该属性限定只能执行一个函数，并且同时也将“行为”标记引入到了页面内容中。Jeremy Keith 所著的《DOM Scripting》展示了如何创建一个 addLoadEvent 函数以便将 JavaScript 分离成单独的文件，并允许加载多个函数。但是它需要一定数量且复杂的代码去实现这个功能。同时，用它加载的函数，也是必须要等页面所有内容被加载完毕后才执行。由于以上种种原因本书选择了更为先进的\$(document).ready()函数。

在\$(document).ready()函数括号中的所有代码都会提前（只要 DOM 在浏览器中被注册完毕）被执行，而不是在页面所有内容（例如图片等占用带宽的内容）加载完毕之后才执行事件。它允许用户在第一眼看到元素的时候，就能立即看到元素产生的一些隐藏效果、显示效

果和其他效果。

A.2 多个\$(document).ready()函数

本节介绍\$(document).ready()函数的另一个功能，即它可以被多次使用。如果不太在意代码的大小，那么可以将\$(document).ready()函数放到所有的 JavaScript 文件中。无论函数是在一个文件中还是在多个文件中，它都可以将这些函数归组。

例如在项目中，每个页面中都引用了一个公共.js 文件，并且每一个.js 文件仅仅在首页中被引用，而且它们都需要调用\$(document).ready()函数。那么可以在首页的<head>标签中，同时加载 3 个 JavaScript 文件，jQuery 代码如下：

```
<script type="text/javascript" src="/scripts/jquery.js"></script>
<script type="text/javascript" src="/scripts/common.js"></script>
<script type="text/javascript" src="/scripts/homepage.js"></script>
```

然后可以在每一个独立的.js 文件中反复调用该函数，jQuery 代码如下：

```
$(document).ready(function() {
    //将 jQuery 代码放在这里
});
$(document).ready(function() {
    //将 jQuery 代码放在这里
});
```

最后需要注意的是，可以使用\$(function){...}来代替\$(document).ready()函数，作为它的缩写形式，缩写方式如下：

```
$(function() {
    //将 jQuery 代码放在这里
});
```