

## 对网页添加视觉效果

当今，几乎所有的 Web 开发人员都希望对网站添加视觉效果。然而，对网页添加复杂的视觉效果需要编写多行 JavaScript 代码，这是耗时的活动。为了克服这一问题，引入了 jQuery。它是 HTML 支持的 JavaScript 库，将多行代码打包成可通过单个语句调用的方法。

本章介绍 jQuery。然后讨论 jQuery 选择器、事件和效果。此外还讨论如何向图像添加图像翻转效果以及如何创建图像库，因此使网站更具吸引力，提升用户的浏览体验。

### 目标

在本章中，您将学习：

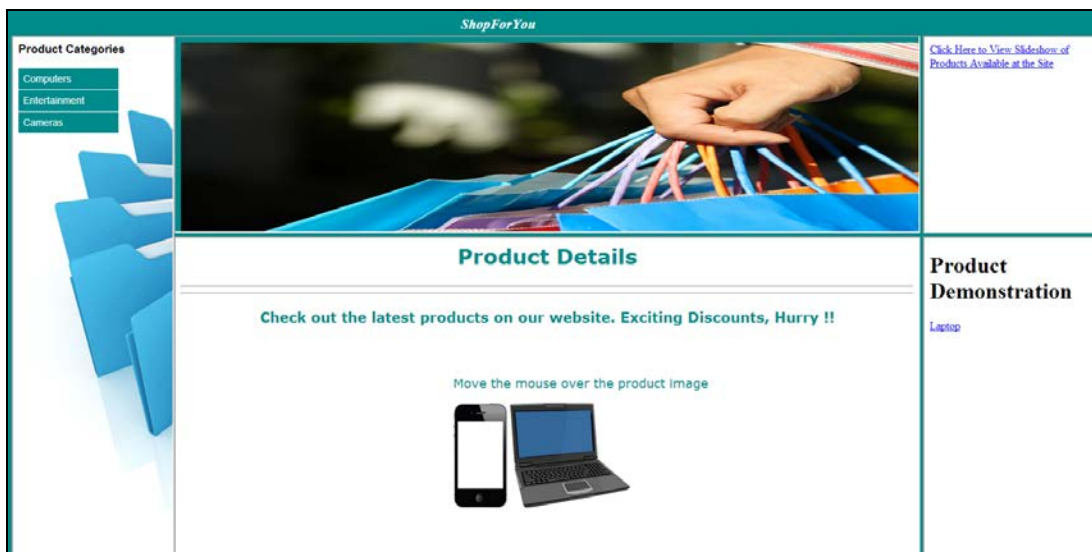
- 研究 jQuery
- 使用 jQuery 添加视觉效果
- 实现图像翻转
- 创建图像库



**NITT** | **training**  **-china**  
**.com**

# 研究 jQuery

思考在线购物网站 ShopForYou.com 的场景。网站的设计方式应该是以可滑动和可折叠菜单和子菜单的格式显示产品类别和产品，如下图所示。



示例网页

为此，Web 设计员被要求快速完成任务并使代码简单。对于此情况，设计员可使用预先编写的 JavaScript 库 jQuery。

jQuery 是开源和跨浏览器的 JavaScript 代码库，创建目的是使 JavaScript 编程更简单。jQuery 可供所有级别的程序员使用以改进网页的外观。jQuery 支持事件并且可绑定到网页上的 HTML 元素。jQuery 弥补了广泛编码和 Web 设计之间的差距，因为它便于理解并且提供许多预定义的特殊效果。通过 jQuery，Web 设计员可添加一系列使人印象深刻的视觉效果，并且还可扩展网页上的交互。

## 使用 jQuery 操作 HTML 元素

可在网页中使用 jQuery 以操作 HTML 元素，方法是下载轻量级 jQuery JavaScript 库并将它保存到您的系统。一旦下载了 jQuery JavaScript 库，可在网页中引用它，方法是在 Web 文档的头部分使用 `<SCRIPT>` 标记。

以下语法用于指定 jQuery 库：

```
<SCRIPT type="text/javascript" src="<path>/jquery-1.8.3.js"> </SCRIPT>
```

在上述语法中：

- `SCRIPT` 标记指示浏览器 HTML 文档使用脚本。
- `type` 属性指定所使用脚本的类型。

- `src` 属性指定所使用 jQuery 库的名称。如果 jQuery 库存储在与 HTML 文档相同的位置，则只能提供库的名称。然而，如果 jQuery 库和 HTML 文档存储在不同位置，必须指定 jQuery 库的完整路径。

jQuery 代码的执行只能发生在完全加载了网页以后。为了确保 jQuery 代码在网页上执行时网页已完全加载，jQuery 提供了 `document.ready()` 函数。此函数包含在浏览器中完全加载了网页以后要在 HTML 元素上执行的 jQuery 代码。

以下语法用于指定 `document.ready()` 函数：

```
<SCRIPT type="text/javascript">
$(document).ready(function(){

// jquery code...

});
</SCRIPT>
```

在以上语法中，美元符号 (\$) 表示 jQuery 代码块的开始，jQuery code 是为了在浏览器窗口中显示 HTML 元素要实现的代码。`document.ready()` 函数对于防止应该在 HTML 元素上执行的操作发生失败非常有用。例如，如果用于隐藏图像的 jQuery 代码在文档就绪前执行，将导致失败。因此，为了使 jQuery 代码只在要显示在浏览器窗口中的文档就绪时执行，应该在 `document.ready()` 函数内声明和定义所有 jQuery 方法和代码。

#### 注释

编写 jQuery 代码时，关键字 `jQuery` 可用于代替 `$` 符号。例如，以上语法也可写为：

```
jQuery(document).ready(function(){
//jquery code...
});
```

由于 jQuery 用于操作 HTML 元素，因此首先应该选择元素以在其上实现 jQuery 代码。

以下 jQuery 语法用于选择 HTML 元素并对它们执行所需操作：

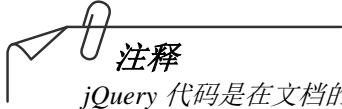
```
$(selector).action();
```

在以上语法中，`selector` 是要操作的元素，`action` 表示要在所选元素上执行的操作。jQuery 提供可应用到 HTML 元素的许多内置操作，例如隐藏、淡出、显示和上滑收起。思考在 HTML 文档中隐藏所有 `<H1>` 元素的以下代码：

```
<!DOCTYPE HTML>
<HTML>
<HEAD>
<SCRIPT type="text/javascript" src="jquery-1.8.3.js">
</SCRIPT>
<SCRIPT>
$(document).ready(function(){
$("h1").hide();
```

```
});
</SCRIPT>
</HEAD>
<BODY>
...
</BODY>
</HTML>
```

在上述代码段中，将在网页上隐藏所有 <H1> 元素。



*jQuery 代码是在文档的 <HEAD> 部分编写的。*

## 使用 jQuery 选择器

jQuery 使用选择器选择和操作一个或一组 HTML 元素。可通过以下方式选择 HTML 元素：

- 使用元素名
- 使用属性名
- 使用 CSS 选择器

### 使用元素名

jQuery 可用于通过元素名选择一个 HTML 元素或一组 HTML 元素。使用 jQuery，可通过引用 HTML 元素的名称、ID 或所属的类来选择它们。

思考选择文档中所有 <H1> 元素的以下代码段：

```
$( "h1" )
```

在上述代码段中，<H1> 元素是通过名称 h1 引用的。然而，也可使用类或 ID 选择 HTML 元素。要使用 jQuery 选择带有特定 ID 的元素，可使用以下语法：

```
$( "#Element_ID" )
```

在以上语法中，Element\_ID 表示要选择的元素的 ID。例如，要选择 ID 为 header 的元素，可使用以下代码段：

```
$( "#header" )
```

要选择具有特定类的元素，可使用以下语法：

```
$( ".Element_class" )
```

在以上语法中，Element\_class 表示要选择的元素的类。例如，要选择类属性值为 header 的元素，可使用以下代码段：

```
$( ".header" )
```

例如，您在 <H1> 标记上创建了类选择器 exClass 和 ID 选择器 exID，如以下代码段所示：

```
<H1 class="exClass">  
<H1 ID="exID">
```

现在您希望操作带有指定类和 ID 的 <H1> 标记。需要使用以下代码：

```
$( "h1.exClass" )  
$( "h1#exID" )
```

此处，通过使用 \$( "h1.exClass" ) 语句，只有通过 class 属性具有 exClass 值指定的 <H1> 头元素被选中，通过使用 \$( "h1#exID" ) 语句，只有通过 ID 属性具有 exID 值指定的 <H1> 头元素被选中。

jQuery 还支持元素嵌套。这允许在嵌套于其他元素中的元素上应用操作。例如，列表项嵌套在列表内。思考 ShopForYou 在线购物网站的示例，Web 设计员希望在该网站实现菜单和子菜单。子菜单项必须是链接的，以在新文档中显示其详细信息。

菜单包含列表和列表项，它们分别由 <UL> 和 <LI> 标记表示。思考构建按钮、菜单和子菜单并向子菜单项附加链接的以下代码段：

```
<BODY>  
<P> Product Menu</P>  
<H4>Product Categories</H4>  
<UL ID="menu">  
<LI><A>Laptops</A>  
<UL>  
<LI><A href="">L-150</A></LI>  
<LI><A href="">L-160</A></LI>  
<LI><A href="">L-180</A></LI>  
<LI><A href="">L-260</A></LI>  
<LI><A href="">L-678</A></LI>  
</UL>  
</LI>  
<LI><A>Phones</A>  
<UL>  
<LI><A href="">P-1150</A></LI>  
<LI><A href="">P-1260</A></LI>  
<LI><A href="">P-1180</A></LI>  
<LI><A href="">P-2260</A></LI>  
<LI><A href="">P-6708</A></LI>  
</UL>  
</LI>  
<LI><A>Cameras</A>  
<UL>  
<LI><A href="">C-50</A></LI>  
<LI><A href="">C-60</A></LI>  
<LI><A href="">C-80</A></LI>  
<LI><A href="">C-60</A></LI>  
<LI><A href="">C-08</A></LI>  
</UL>  
</LI>  
<LI><A>Music Players</A>  
<UL>
```

```

<LI><A href="">MP-1150</A></LI>
<LI><A href="">MP-1260</A></LI>
<LI><A href="">MP-1180</A></LI>
<LI><A href="">MP-2260</A></LI>
<LI><A href="">MP-6708</A></LI>
</UL>
</LI>
</UL>
<BUTTON>Hide List</BUTTON>
</BODY>

```

思考以下代码段，它将在用户单击按钮时隐藏其 menu 指定为 ID 属性值的所有嵌套的 <UL> 项：

```

<HEAD>
<SCRIPT src="jquery-1.8.3.js">
</SCRIPT>
<SCRIPT>
$(document).ready(function(){
$("button").click(function(){
$('#menu ul').hide();
});
});
</SCRIPT>
</HEAD>

```

在上述代码段中，#menu 用于选择 ID 属性值为 menu 的 HTML 元素。#menu ul 选择嵌套在 ID 指定为 menu 的 HTML 元素中的所有 <UL> 项。当用户单击按钮时，将隐藏所有嵌套的列表项，包含项的菜单仅显示第一层：笔记本电脑、电话、照相机和音乐播放器将可见。

## 使用属性名

也可使用属性名选择 HTML 元素。以下语法用于通过属性名选择元素：

```

$("[attribute-name]")

```

在以上语法中，attribute-name 指定属性的名称。

考虑以下代码段：

```

$("[src]")

```

在上述代码段中，选择了带 src 属性的所有元素。

思考以下代码段：

```

$("[src='movie.jpg']")

```

在上述代码段中，选择了 src 属性为 movie.jpg 的所有元素。

## 使用 CSS 选择器

jQuery 提供 CSS 选择器以修改 HTML 元素或文档的 CSS 属性。

以下语法用于应用 CSS 选择器以修改文档或元素属性：

```
selector.css(PropertyName, PropertyValue );
```

思考将 <DIV> 元素文本颜色更改为绿色的以下代码段：

```
$("div").css("color","green");
```

在上述代码段中，文档中所有 div 元素都被选中，并且 css color 属性值被修改为 green。

通过 CSS 选择器，可对一个 HTML 元素设置多个属性。以下语法用于设置多个 CSS 属性：

```
css({ "propertyname": "value", "propertyname": "value", ... });
```

思考更改 <DIV> 元素文本颜色和字体大小的以下代码段：

```
$("div").css("color": "green", "font-size": "200%");
```

在上述代码段中，文档中所有 div 元素都被选中，并且 css color 属性值被修改为 green，font-size 设置为 200%。

## jQuery HTML

jQuery 库提供预定义函数以对 HTML 元素的内容执行修改。

下表列出用于修改 HTML 元素内容的最常用函数。

函数	描述	语法
html( )	更改 HTML 元素的内容，例如 <P> 和 <DIV> 标记中的文本。	<code>\$(selector).html(content)</code>
append( )	允许用户在选定元素内部内容后面追加内容。	<code>\$(selector).append(content)</code>
prepend( )	允许用户在选定元素的内部内容前面追加内容。	<code>\$(selector).prepend(content)</code>
remove( )	除去选定元素，包括其内部的所有文本和嵌套元素。	<code>\$(selector).remove( )</code>
empty( )	除去选定元素的所有文本和嵌套元素。然而，此方法不除去该元素。	<code>\$(selector).empty( )</code>
after( )	在选定元素的后面插入内容。	<code>\$(selector).after(content)</code>
before( )	在选定元素的前面插入内容。	<code>\$(selector).before(content)</code>
text( )	设置或返回选定元素的内容。	<code>\$(selector).text(content)</code>



函数	描述	语法
val()	设置或返回选定元素的属性值。	\$(selector).val(value)
attr()	设置或返回选定元素的值和属性。	\$(selector).attr(attribute,value)

### jQuery HTML 函数

思考用于修改 HTML 元素内部内容的以下代码：

```

<!DOCTYPE HTML><HTML>
<HEAD>
<SCRIPT type="text/javascript" src="jquery-1.8.3.js"></SCRIPT>
<SCRIPT type="text/javascript">
$(document).ready(function(){
$("button").click(function(){
$("#para").prepend(" <B>jQuery </B>");
$("#para").append(" <B>append() function</B>.");
$("#para").after('<H3> This is the newly inserted heading.</H3>');
$("#newhead").remove();
});
});
</SCRIPT>
</HEAD>
<BODY>
<DIV align="center">
</DIV>
<H3>Complete the sentence:</H3>
<P ID="para">is a javascript library that enables the
content to be appended after the inner content in the
selected element using the</P><BR>
<H2 ID="newhead">This heading would be removed.</H2>
<BUTTON ID="b">Click here</BUTTON>
</BODY>
</HTML>

```

在上述代码中，当用户单击 **Click here** 按钮时，将通过 `prepend()` 函数在 ID 为 `para` 的 `<P>` 标记前面追加内容 **jQuery**。类似地，通过 `append()` 函数，将内容 `append()` 函数追加到同一 `<P>` 标记后。

此外，通过 `after()` 函数将新的 `<H3>` 标记插入到 ID 为 `para` 的 `<P>` 标记后面。最后，使用 `remove()` 函数除去了 ID 为 `newhead` 的头。

单击 **Click here** 按钮前，将出现如下图所示的网页。

### Complete the sentence:

is a javascript library that enables the content to be appended after the inner content in the selected element using the

**This heading would be removed.**

Click here

单击按钮前的网页

单击 **Click here** 按钮后，将出现如下图所示的网页。

### Complete the sentence:

jQuery is a javascript library that enables the content to be appended after the inner content in the selected element using the **append() function**.

**This is the newly inserted heading.**

Click here

单击按钮后的网页



注释

将在接下来的主题中涵盖 jQuery 事件。

## 处理 jQuery 事件

在 jQuery 中，可使用函数或预定义的事件方法来处理事件。事件方法用于检测事件并在发生该事件时触发函数。可通过函数或者事件方法操作元素。

下表列出 jQuery 提供一些事件方法。

事件方法	描述
<code>\$(document).load(function)</code>	用于在文档完成加载后执行函数。
<code>\$(selector).click(function)</code>	用于在所选元素上发生 Click 事件时执行函数。
<code>\$(selector).dblclick(function)</code>	用于在所选元素上发生 double-click 事件时执行函数。

<code>\$(selector).mouseenter(function)</code>	用于在鼠标指针移动到所选元素上面时执行函数。
<code>\$(selector).mouseleave(function)</code>	用于在鼠标指针从所选元素移出时执行函数。
<code>\$(selector).keydown(function)</code>	用于在所选元素上按键时执行函数。
<code>\$(selector).submit(function)</code>	用于在提交表单时执行函数。
<code>\$(selector).focus(function)</code>	用于在所选元素获得焦点时执行函数。
<code>\$(selector).blur(function)</code>	用于在所选元素失去焦点时执行函数。
<code>\$(selector).resize(function)</code>	用于在浏览器窗口改变大小时执行函数。
<code>\$(selector).unload(function)</code>	用于在用户导航离开页面时执行函数。
<code>\$(selector).mousedown(function)</code>	用于在所选元素上按下鼠标左键时执行函数。
<code>\$(selector).mouseup(function)</code>	用于在所选元素上释放鼠标左键时执行函数。

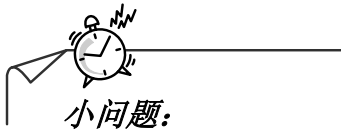
### jQuery 中的事件方法

思考处理 jQuery 事件的以下代码：

```
<!DOCTYPE HTML><HTML>
<HEAD>
<SCRIPT type="text/javascript" src="jquery-1.8.3.js"></SCRIPT>
<SCRIPT type="text/javascript">
$(document).ready(function(){
$("#hide_header").click(function(){ /* selects the hide_header button and
binds it to the click function */
$("h1").hide(); //hides the <h1> header
});
});
</SCRIPT>
</HEAD>
<BODY>
<H1>I am the header</H1>
<HR>
<INPUT type="button" ID="hide_header" value='Hide the header' />
</BODY>
</HTML>
```

在上述代码中，jQuery 在用户单击 **Hide the header** 按钮时识别 Click 事件。jQuery 识别事件后，将执行与该按钮关联的函数并隐藏 <H1> 元素。

有一些预定义的函数可附加到 jQuery 中的事件方法以使 Web 文档的视觉外观更丰富。这些函数称为 jQuery 效果。



### 小问题:

以下哪个函数除去所选 *HTML* 元素的所有文本和嵌套元素?

1. `remove()`
2. `empty()`
3. `prepend()`
4. `append()`

### 答案:

2.`empty()`

NIIT | training-china.com

## 使用 jQuery 添加视觉效果

借助 jQuery，可向 Web 文档的元素应用神奇的视觉效果。这些视觉效果有助于丰富用户的浏览体验。可用于增加网页的视觉吸引力的一些预定义 jQuery 效果是：

- 隐藏
- 显示
- 切换
- 滑动
- 淡化
- 动画

### 实现隐藏效果

hide() 函数用于使元素在发生事件（例如 click 或 double-click）时消失。

以下语法用于隐藏所选元素：

```
$(selector).hide(speed)
```

在以上语法中，speed 是可选参数，用于通过值 slow、normal 和 fast 表示速度，或者表示元素消失的持续时间（毫秒）。

思考用于实现隐藏效果的以下代码：

```
<!DOCTYPE HTML><HTML>
<HEAD>
<SCRIPT type="text/javascript" src="jquery-1.8.3.js"></SCRIPT>
<SCRIPT type="text/javascript">
$(document).ready(function(){
$("button").click(function(){
$("h1").hide("slow");
});
});
</SCRIPT>
</HEAD>
<BODY>
<H1>I am the header</H1>
<BUTTON>Hide the header</BUTTON>
</BODY>
</HTML>
```

在以上代码中，**I am the header** 将显示在 **Hide the header** 按钮上方。单击 **Hide the header** 按钮时，文本 **I am the header** 将从浏览器窗口缓慢消失。

## 实现显示效果

`show()` 函数用于在发生事件时使隐藏元素可见。以下语法用于显示已隐藏的所选元素：

```
$(selector).show(speed)
```

在以上语法中，`speed` 是可选参数。此参数用于通过值 `slow`、`normal` 和 `fast` 指定速度，或者指定元素重新出现的持续时间（毫秒）。

思考用于实现显示效果的以下代码段：

```
<!DOCTYPE HTML><HTML><HEAD>
<SCRIPT type="text/javascript" src="jquery-1.8.3.js"></SCRIPT>
<SCRIPT type="text/javascript">
$(document).ready(function(){
$(".view").click(function(){
$("h3").show(2000);
});
$(".conceal").click(function(){
$("h3").hide();
});
});
</SCRIPT>
</HEAD>
<BODY>
<H3> This text can be hidden and shown.</H3> <BR>
<BUTTON class="view">Click to view</BUTTON>
<BUTTON class="conceal">Click to hide</BUTTON>
</BODY>
</HTML>
```

在上述代码段中，单击 **Click to hide** 按钮时，`<H3>` 标记将在浏览器窗口中隐藏。单击 **Click to view** 按钮时，`<H3>` 标记将在 2000 毫秒（2 秒）内再次显示。

## 实现切换效果

`toggle()` 函数可用于在元素的显示和隐藏效果之间进行切换。此事件可用于在事件发生时交替地隐藏或显示元素。

以下语法用于显示或隐藏所选元素：

```
$(selector).toggle(speed)
```

在以上语法中，`speed` 是可选参数。它用于通过值 `slow`、`normal` 和 `fast` 指定速度，或者指定元素消失并重新出现的持续时间（毫秒）。

思考用于实现切换效果的以下代码段：

```
<!DOCTYPE HTML><HTML>
<HEAD>
<SCRIPT type="text/javascript" src="jquery-1.8.3.js"></SCRIPT>
<SCRIPT type="text/javascript">
```

```

$(document).ready(function(){
$(".view").click(function(){
$("h3").toggle('fast');
});
});
</SCRIPT>
</HEAD>
<BODY>
<H3> Show and Hide me Quickly</H3>
<HR>
<BUTTON class="view">Click here</BUTTON>
</BODY>
</HTML>

```

在上述代码段中，toggle() 函数交替地隐藏和显示 <H3> 标记。

### 实现滑动效果

滑动效果可用于在所选元素上产生滑动效果。可对所选元素应用三个滑动函数：slideDown()、slideUp() 和 slideToggle()。这些函数对所选元素的高度应用渐进的修改，以在网页上创造滑动动画效果。

下表列出滑动动画函数。

函数	描述	语法
slideDown()	产生向下滑动展开所选元素的效果。	\$(selector).slideDown(speed)
slideUp()	产生向上滑动收起所选元素的效果。	\$(selector).slideUp(speed)
slideToggle()	在交替单击时分别执行 slideup 和 slidedown 函数。	\$(selector).slideToggle(speed)

滑动效果函数

思考用于实现滑动效果的以下代码：

```

<!DOCTYPE HTML><HTML>
<HEAD>
<SCRIPT type="text/javascript" src="jquery-1.8.3.js"></SCRIPT>
<SCRIPT type="text/javascript">
$(document).ready(function(){
$(".flipbut").click(function(){
$(".thought").slideToggle();

```

```

});
});
</SCRIPT>
<STYLE type="text/css">
div.thought,.flipbut{ /*stylizing the div element with the class
attribute specified as "thought",
                        and the button with the class attribute
specified as "flipbut" */}
margin:0px;
padding:5px;
text-align:center;
background:beige;
border:solid 1px;
}
div.thought
{
height:120px;
display:none;
}
</STYLE>
</HEAD>
<BODY>
<BR><BR>
<BUTTON style="background-color:beige" class="flipbut" text-
align="center">Show/Hide the thought of the day</BUTTON>

<DIV class="thought">
<P>The thought of the day is:</P>
<P> Where there is a will, there is a way!</P>
</DIV>
</BODY>
</HTML>

```

在上述代码中，单击 **Show/Hide the thought of the day** 按钮时，将在 <DIV> 元素和 thought 类上实现在上滑收起和下滑展开之间切换的功能。

对于在线购物网站 ShopForYou.com，菜单和子菜单必须具有滑动效果。当用户单击产品类别时，应该使用滑动切换效果显示该类别的详细产品。

要在 ShopForYou.com 网站中实现此功能时，需要在文件中编写以下代码并将文件另存为 **menu.html**：

```

<!DOCTYPE HTML><HTML>
<HEAD>
<SCRIPT type="text/javascript" src="jquery-1.8.3.js"></SCRIPT>
<SCRIPT type="text/javascript">
$(document).ready(function(){
$('#menu ul').hide();
$('#menu li a').mouseenter(function() {
$(this).next().slideToggle('slow');
}
);
});
</SCRIPT>
<STYLE type="text/css">

```



```

body{
background-image:url('background.jpg');
}
li a
{
display:inline;
}
body {
font-family:Arial, sans-serif;
font-size:0.9em;
background-color:silver;
}
p
{
line-height:1.5em;
}
#menu, ul#menu ul { /* stylizing the menu and the list nested in the menu */
list-style-type:none;
margin:0;
padding:0;
width:10em;
}
ul#menu a { /* stylizing the <a> elements nested in the menu */
display: inline;
text-decoration: none;
}
ul#menu li { /* stylizing the list items nested in the menu */
margin-top:1px;
}
ul#menu li a { /* stylizing the <a> elements nested in list items of the menu
*/

background: darkcyan;
color:#fff;
padding:0.5em;
}
ul#menu li a:hover { /* stylizing the <a> elements nested in list items of
the menu when mouse is placed or moved over them */

background: black;
}
ul#menu li ul li a {
background: grey;
color: white;
padding-left:20px;
}

ul#menu li ul li a:hover {
background:#aaa;
color: white;
border-left:5px grey solid;
padding-left:15px;
}
.st
{

```

```

display:inline;
}
</STYLE>
</HEAD>
<BODY>
<H3>Product Categories</H3>
<UL ID="menu">
<LI class=".st"><A>Computers</A>
<UL>
<LI><A href="">e-Book Readers</A></LI>
<LI><A href="">Tablets and i-Pads</A></LI>
<LI><A href="">Laptops</A></LI>

</UL>
</LI>
<LI class=".st"><A>Entertainment</A>
<UL>
<LI><A href="">LED-TVs</A></LI>
<LI><A href="">LCD-TVs</A></LI>

</UL>
</LI>
<LI class=".st"><A>Cameras</A>
<UL>
<LI><A href="">Digital Cameras</A></LI>
<LI><A href="">Digital SLR Cameras</A></LI>
</UL>
</LI>
</UL>
</BODY>
</HTML>

```

在上述代码中，使用 CSS 创建了显示 **Computers**、**Entertainment** 和 **Cameras** 产品类别的菜单并使其带有样式。产品类别下包含不同的产品项。

所有的产品项都是隐藏的，只有 **Computers**、**Entertainment** 和 **Cameras** 产品类别可见，如下图所示。



产品菜单

单击产品类别时，将使用 jQuery 提供的滑动切换效果使属于该特定产品类别的产品项呈现下滑展开的效果。

单击 **Entertainment** 产品类别后，将显示子菜单，如下图所示。



Computers 产品类别的子菜单

再次单击该产品类别时，产品项将呈现上滑收起的效果。

## 实现淡化效果

jQuery 淡化效果用于逐渐降低所选元素的不透明度。可使 HTML 元素逐渐淡化为可见或者不可见。下表列出产生淡化效果的四个函数。

函数	描述	语法
<code>fadeOut()</code>	用于使所选元素逐渐淡化消失。	<code>\$(selector).fadeOut(speed)</code>
<code>fadeIn()</code>	用于使已淡化消失的所选元素恢复原样。	<code>\$(selector).fadeIn(speed)</code>

函数	描述	语法
<code>fadeTo()</code>	用于指定所选元素淡化的不透明度。	<code>\$(selector).fadeTo(speed,opacity)</code>
<code>fadeToggle()</code>	用于在淡出时使所选元素逐渐淡化消失，并且在淡入时使所选元素恢复原样。	<code>\$(selector).fadeToggle(speed)</code>

### 淡化效果函数

思考用于实现淡化效果的以下代码：

```

<!DOCTYPE HTML><HTML>
<HEAD>
<SCRIPT type="text/javascript" src="jquery-1.8.3.js"></SCRIPT>
<SCRIPT type="text/javascript">
$(document).ready(function(){
$("div").click(function(){
$(this).fadeOut(1200);
});
$("button").click(function(){
$("div").fadeIn(1200);
});
});
</SCRIPT>
</HEAD>
<BODY>
<DIV style="background:yellow;width:200px">FADE ME AWAY!</DIV><BR>
<BUTTON>Restore</BUTTON>
</BODY>
</HTML>

```

在上述代码中，<DIV> 标记是通过名称引用的。在 <DIV> 标记的 Click 事件处理程序中，this 关键字引用当前 HTML 元素，即 <DIV> 标记。当用户单击 <DIV> 标记时，文本 **FADE ME AWAY!** 将淡化消失。当单击 **Restore** 按钮时，文本 **FADE ME AWAY!** 重新出现。

## 实现动画效果

jQuery 动画效果用于创建自定义动画。可使用以下语法创建动画函数：

```
animate({params},speed,callback);
```

其中，

params 指定要带动画效果的 CSS 属性。

speed 指定动画效果的持续时间。

callback 指定完成动画以后要执行的函数。这是可选参数。

思考在 <DIV> 元素上应用动画效果的以下代码段：

```
<!DOCTYPE HTML>
<HTML>
<HEAD>
<SCRIPT src="jquery-1.8.3.js">
</SCRIPT>
<SCRIPT>
$(document).ready(function(){
$("div").mouseenter(function(){
$("div").animate({
center:'250px',
opacity:'0.5',
height:'250px',
width:'250px'
});
});
$("div").mouseleave(function(){
$("div").animate({
opacity:'1',
height:'100px',
width:'100px'
});
});
});
</SCRIPT>
</HEAD>
<BODY>

<DIV style="background:#98bf21;height:100px;width:100px;position:absolute;">
</DIV>
</BODY>
</HTML>
```

上述代码将在鼠标移动到 <DIV> 元素上时根据指定的宽度和高度增加其大小，并且将在鼠标离开它时根据指定的宽度和高度减小 <DIV> 元素的大小。



### 小问题:

以下哪个函数用于在发生事件时使隐藏元素可见?

1. `hide()`
2. `show()`
3. `toggle()`
4. `slide()`

### 答案:

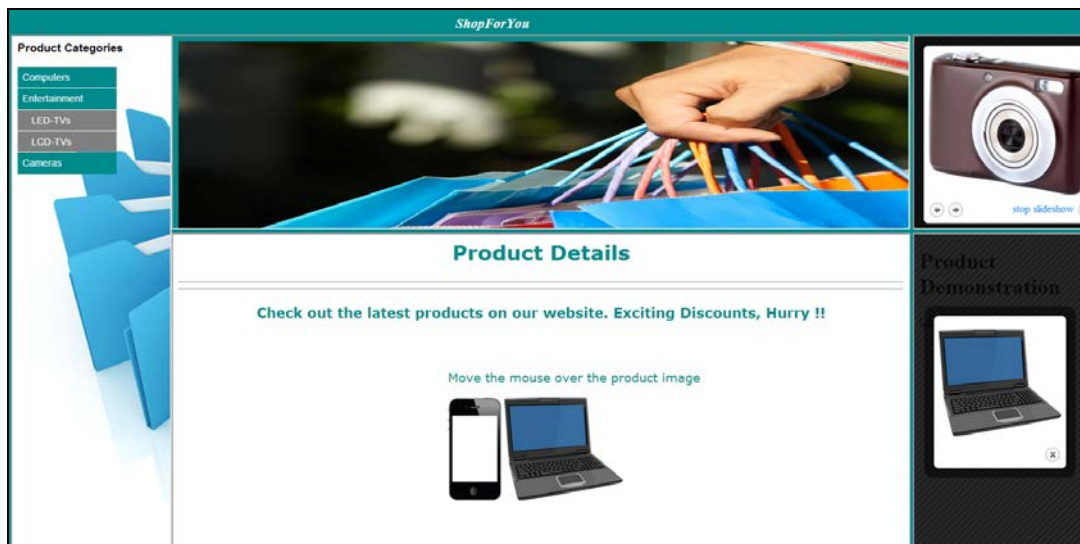
2.`show()`



## 活动 7.1：研究 jQuery

## 实现图像翻转

思考 ShopForYou.com 网站的场景，它销售各种类别的产品，例如电话和笔记本电脑。为了使客户能够预览这些产品，将在产品页面上显示其图像。为了使这些产品的预览吸引客户，您可能希望对显示的图像添加效果，例如当客户将鼠标指针放到或移到图像上时，它应该放大显示或者显示时带有该产品的信息。这可通过将图像替换为同一产品的更大图像或者该产品的纯文本信息来实现，如下图所示。



示例网页

可通过创建图像翻转效果在运行时将图像替换为其他图像。JavaScript 帮助您在网页上添加图像翻转效果。

### 创建图像翻转

图像翻转是当鼠标指针放在它上面或移动到它上面时图像外观更改的效果。外观的更改可通过使用 JavaScript 将原始图像替换为其他图像来实现。

在 ShopForYou.com 网站中，**Product1.png** 和 **Product2.png** 图像分别在产品页面上显示笔记本电脑和移动电话的图像。需要在网站上添加功能以便在客户将鼠标指针移动到或者放到笔记本电脑或移动电话的图像上时，显示其详细信息。要完成此任务，需要向 **Product1.png** 和 **Product2.png** 图像添加图像翻转效果。

要实现翻转效果，需要在文件中编写以下代码并将它另存为 **imagerollover.html**：

```
<!DOCTYPE HTML><HTML>
<HEAD>
<STYLE>
body{
background-color:white;
```



```

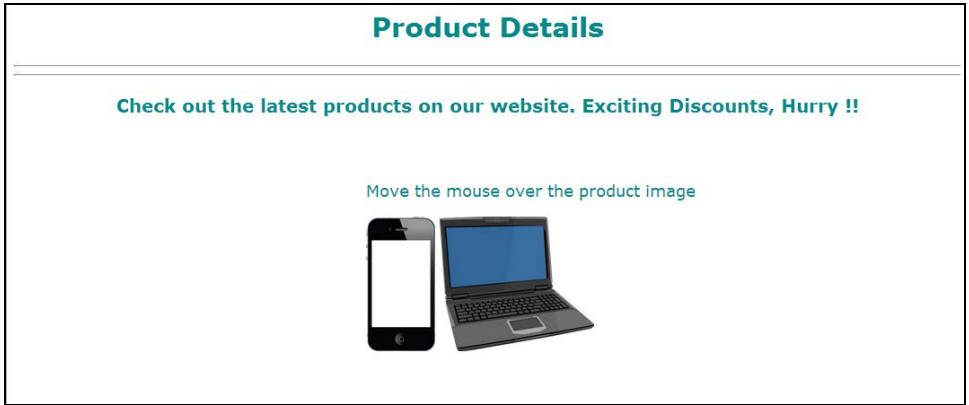
}
h2
{
font-family:verdana;
font-size:30px;
color:darkcyan;
text-align:center;
}
h3
{
text-align:center;
font-family:verdana;
color:darkcyan;
font-size:20px;
}
#para1
{
font-family:verdana;
font-size:18px;
text-align:center;
color:darkcyan;
}
</STYLE>
<SCRIPT>
function over(img,imgsrc)
{
img.src=imgsrc;
}
function out(img,imgsrc)
{
img.src=imgsrc;
}
</SCRIPT>
</HEAD>
<BODY>
<H2> Product Details </H2>
<HR><HR>
<H3> Check out the latest products on our website.Exciting Discounts,
Hurry !!</H3>
<DIV style="position:absolute; top:180px; left:400px;">
<P ID="para1"> Move the mouse over the product image </P>
<IMG ID="productImage1" src="Product1.png" height="150"
onmouseover="over(this,'Product1Details.jpg')" onmouseout="out
(this,'Product1.png')"/>
<IMG ID="productImage2" src="Product2.png" height="150"
onmouseover="over(this,'Product2Details.jpg')" onmouseout="out
(this,'Product2.png')"/>
</DIV>
</BODY>
</HTML>

```

在上述代码中，对 <IMG> 标记的 onmouseover 和 onmouseout 事件分别调用了两个函数 over() 和 out()。**Product1.png** 和 **Product2.png** 图像是通过 <IMG> 标记添加到网页的。当客户将鼠标指针移动到 **Product1.png** 图像上时，将调用 over() 函数，它将网页上图像的 src 属性值更改为

**Product1Details.jpg**。因此，**Product1.png** 图像将替换为 **Product1Details.jpg** 图像。然而，当客户将鼠标指针从 **Product1Details.jpg** 图像移开时，将调用 `out()` 函数，它将文档中图像的 `src` 属性值更改为 **Product1.png** 图像。因此，**Product1.png** 图像将重新显示在页面上。类似地，**Product2.png** 图像将替换为 **Product2Details.jpg** 图像。

下图显示了上述代码的输出。



显示产品图像的网页

在以上 **Product Details** 网页上，如果鼠标指针移动到笔记本图像上，它将替换为包含产品信息的另一图像，如下图所示。



显示图像翻转效果的网页

## 创建后向兼容翻转

在大多数浏览器（例如 Firefox 和 Internet Explorer）中，可使用鼠标事件在图像上创建翻转效果。然而，在早期版本的浏览器中，不能捕获图像上的鼠标事件。因此，不能通过这些浏览器图像上的鼠标事件添加翻转效果。为了克服这一限制，需要创建后向兼容的翻转。

例如，在 ShopForYou.com 网站中，需要添加对于所有浏览器都兼容的翻转效果，这样就可对所有客户生成所需的输出，无论他们使用什么浏览器。

后向兼容的翻转效果有助于捕获所有浏览器图像上的鼠标事件。为了实现此效果，需要使用 <A> 标记并在其中使用 <IMG> 标记来放置所有图像。<A> 标记可捕获图像上的鼠标事件，例如 onmouseover 和 onmouseout。

思考创建后向兼容翻转效果的以下代码段：

```
<A href="#" onmouseover="over()" onmouseout="out()" >
<IMG ID="image1" src="SampleImage1.gif">
</A>
```

在上述代码段中，<IMG> 标记放在 <A> 标记内。<A> 标记捕获 **SampleImage1.gif** 图像上的鼠标事件。捕获的事件可用于创建翻转效果。

在以上示例中，您只使用图像 ID 和 src 属性来引用单个图像并将它替换为其他图像。然而，要在多个图像上实现翻转效果，可使用 document.images 数组来轻松实现。document.images 数组将所有图像的引用存储在当前文档中。如果任何图像的引用存在于 document.images 数组中，document.images 数组返回 true。因此，document.images 数组可用于检查图像是否存在。

document.images 数组中的各个图像可通过将图像 ID 或索引用作数组的下标来进行引用。例如，文档中的第一个图像可通过图像索引进行引用，如以下代码段所示：

```
document.images[0]
```

类似地，第二个图像引用为 document.images[1]，以此类推。

数组中的图像也可通过将图像 ID 属性用作数组下标来进行引用，如以下语法所示：

```
document.images [ID];
```

例如，在 ShopForYou.com 网站中，为了在多个图像上实现翻转效果并确保与所有浏览器的兼容性，可在 ShopForYou.com 网站的产品页面上使用以下代码：

```
<!DOCTYPE HTML><HTML>
<HEAD>
<STYLE>
h2
{
font-family:verdana;
font-size:30px;
text-align:center;
}
h3
{
font-family:verdana;
color:orange;
font-size:20px;
text-align:center;
}
#para1
{
font-family:verdana;
```

```

font-size:14px;
color:blue;
}
</STYLE>
<SCRIPT>
function over(img,imgsrc)
{
if(document.images)
{
document.images[img].src = imgsrc;
}
}
function out(img,imgsrc)
{
if (document.images)
{
document.images[img].src = imgsrc;
}
}
</SCRIPT>
</HEAD>
<BODY>
<H2> Product Categories </H2>
<HR><HR>
<H3> Check out the latest products on our website.Exciting Discounts, Hurry !!
</H3>

<DIV style="position:absolute; top:180px; left:450px;">
<P ID="para1"> Choose the desired product image to view the product details
</P>
<A href="#" onmouseover="over('productImage1','Product1Details.jpg'); "
onmouseout="out('productImage1','Product1.png');">
<IMG ID="productImage1" src="Product1.png" border="0"/>
</A>
<A href="#" onmouseover="over('productImage2','Product2Details.jpg'); "
onmouseout="out('productImage2','Product2.png');">
<IMG ID="productImage2" src="Product2.png" border="0"/>
</A>
</DIV>
</BODY>
</HTML>

```

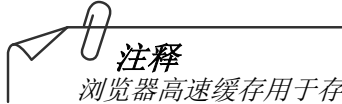
在上述代码中，<IMG> 标记定义笔记本电脑和移动电话的图像源，它们分别是 **Product1.png** 和 **Product2.png**。此标记放在 <A> 标记内。因此，<A> 标记使您能够捕获 **Product1.png** 和 **Product2.png** 图像上的鼠标事件。over() 和 out() 函数是通过 <A> 标记调用的。over() 函数是在用户将鼠标移动到图像上时调用的。out() 函数是在客户将鼠标从图像上移开时调用的。这两个函数都取两个参数，即要被替换的图像的 ID 和名称。over() 和 out() 函数通过使用数组 document.images 检查文档中的图像。

当客户将鼠标指针移动到 **Product1.png** 图像上时，将调用 over() 函数。此函数将 **Product1.png** 图像替换为 **Product1Details.jpg** 图像，方法是将 src 属性值更改为 **Product1Details.jpg** 图像。

当客户将鼠标从图像上移开时，将调用 out() 函数，它将 src 属性的值复位为 **Product1.png** 图像。类似地，**Product2.png** 图像可替换为 **Product2Details.jpg** 图像。

## 预加载图像

预加载图像是用于在执行网页脚本并且在浏览器窗口中显示网页前，将图像加载到浏览器高速缓存的技术。这将有助于避免从各自位置加载图像然后在网页上显示这些图像时发生任何延迟。Image 对象可用于在应用程序中预加载图像。要使用 Image 对象，需要创建 Image 对象的一个实例，实际图像需要附加到头部分的 Image 对象。这确保图像先加载到内存，然后再加载页面主体。



### 注释

浏览器高速缓存用于存储下载的文件，这样当用户重新访问页面时就不需要再次下载这些文件，而是可以通过浏览器高速缓存使用它们。

以下语法用于创建 Image 对象的实例：

```
var Imagename= new Image([Width], [Height]);
```

在上述语法中：

- Imagename：是 Image 对象的新实例的名称。
- [Width]：是以像素为单位的图像宽度。
- [Height]：是以像素为单位的图像高度。

实例化 Image 对象时，[Width] 和 [Height] 参数都是可选的。

思考创建 Image 对象实例的以下代码段：

```
var img = new Image(40,30);
```

在上述代码段中，img 是图像对象的实例名称。图像的宽度和高度被分别赋值 40 和 30。

实例化图像对象后，需要将图像与 Image 对象关联。为此，使用 Image 对象的 src 属性。将图像附加到 Image 对象的实例的语法如下：

```
Imagename.src="ImageUrl"
```

在上述语法中：

- Imagename：是 Image 对象的实例。
- ImageURL：是图像的 URL。

思考将图像附加到 Image 对象的以下代码段：

```
img.src= SampleImage1.gif;
```

在上述代码段中，使用 Image 对象的 src 属性将 SampleImage1.gif 图像附加到 img 对象。

如果 HTML 文档包含多个图像，需要在预加载图像时引用文档中的所有图像。

例如，在 ShopForYou.com 网站中，可使用 Image 对象预加载图像并实现翻转效果。为此，您可以使用以下代码：

```

<!DOCTYPE HTML><HTML>
<HEAD>
<STYLE>
table{
text-align:center;
}
h2
{
font-family:verdana;
font-size:30px;
text-align:center;
}
h3
{
font-family:verdana;
color:orange;
font-size:20px;
text-align:center;
}

#para1
{
font-family:verdana;
font-size:14px;
color:blue;
text-align:center;
}
</STYLE>
<SCRIPT>
var images;
function preloadimages()
{
images = new
Array('product1.png','product2.png','product1details.jpg','product2details.jp
g');
preload(images);
}

function preload(img)
{
for(var i=0;i<img.length;i++)
{
var im=new Image();
im.src=img[i];
}
}

function over(a)
{
document.images[a].src=images[a+2];
}

function out(a)
{

```

```

        document.images[a].src=images[a];
    }
</SCRIPT>
</HEAD>
<BODY onLoad=preloadimages()>
<H2> Product Categories </H2>
<HR><HR>

<H3> Check out the latest products on our website.Exciting Discounts,
Hurry !!</H3>

<DIV style="position:absolute; top:180px; left:400px;">
<P ID="para1"> Move the mouse cursor over the product image to view the
product details
</P>
<TABLE border="0">
<TR>
<TD>
<IMG src="product1.png" onmouseover="over(0)" onmouseout="out(0)" />
</TD>
<TD>
<IMG src="product2.png" onmouseover="over(1)" onmouseout="out(1)" />
</TD>
</TR>
</BODY>
</HTML>

```

在上述代码中，在 `preloadimages()` 函数中创建了图像数组。此数组作为参数传递到 `preload()` 函数。在 `preload()` 函数中，通过 `Image` 对象预加载图像。文档的图像是通过 `over()` 和 `out()` 函数中的 `document.images` 数组引用的。

## 创建图像库

思考 ShopForYou.com 的场景：它使用图像库来显示网站可以提供的产品的图像。图像库是一组缩略图或图像链接。所有这些图像都可分别单击以在另一网页上提供相应产品的大视图。此图像库是使用 JavaScript 实现的。此功能的缺点是预览图像时，客户需要反复导航到上一页面。

为了避免反复从一个页面导航到上一页面，可在同一页面上显示全尺寸图像。为此，需要使用 jQuery 插件来创建图像库。jQuery 插件是用于增强 jQuery 代码的预先编写且易于使用的程序。通过使用 jQuery 插件，可通过在运行时在同一网页上动态显示图像来创建图像库。jQuery 插件可用于通过以下方式创建图像库：

- 使用 colorbox 插件
- 使用 gallery 插件

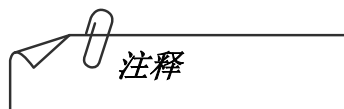
### 使用 Colorbox 插件

轻量级 colorbox 插件支持通过“上一个”和“下一个”按钮逐个显示一组图像。colorbox 插件可用于在优美框架的中间显示具有一系列转换效果（例如淡化或弹性）的图像。此框架称为 **lightbox**，是在网页中心位置创建的，其周围覆有一层半透明的膜。可有效使用 colorbox 插件以在网页上创建和显示图像库或图像幻灯片，其中图像顺序可以通过“上一个”和“下一个”按钮控制。要在网页上创建优美图像库或幻灯片，可下载 **colorbox.zip** 文件，该文件包含实现 colorbox 插件功能所需的文件夹和文件。然后可通过 **colorbox.zip** 文件抽取 colorbox 文件夹。colorbox 文件夹包含以下文件夹：

- **colorbox**：此文件夹包含 **jquery.colorbox.js** 文件，该文件是 colorbox 插件文件。此 colorbox 插件文件用于实现 **lightbox**、幻灯片和转换效果来显示图像库中的图像。
- **example1**、**example2**、**example3**、**example4** 和 **example5**：这些文件夹包含 colorbox 插件的示例实现。每个文件夹演示 colorbox 插件的唯一自定义实现以创建图像库或幻灯片。其中每个文件夹都包含以下子文件夹和文件：
  - **colorbox.css**：此样式表包含在使用 colorbox 插件创建图像库时用于样式化、定位和修饰图像库的格式化指令。例如，网页背景、图像边框以及“上一个”、“下一个”和“关闭”按钮。
  - **images 文件夹**：此文件夹包含可用于创建和控制图像库或幻灯片的图像。可根据 **colorbox.css** 样式表中指定的指令，在网页上对这些图像进行样式化和定位。
  - **index.html**：此网页用于演示如何使用弹性、淡化或无转换效果来显示图像库。此外，此网页用于显示图像幻灯片。

Colorbox 是 jQuery 插件。因此，要实现此插件提供的功能，需要 **jquery-1.8.3.js** 文件。此外，需要 jQuery 事件方法 **document.ready()**，它是调用 colorbox 插件以实现幻灯片或图像库功能所必需的。



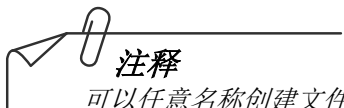


### 注释

幻灯片用于在不使用“上一个”和“下一个”按钮的情况下在定义的持续时间之后逐个显示图像。而在图像库中，用户必须使用“上一个”和“下一个”按钮在所有图像之间滚动。

思考以下示例：为图像 **Product1.png** 实现 colorbox 插件并创建 lightbox。为此，您需要执行以下步骤：

1. 在系统任何驱动器上创建文件夹，例如 **demo**。
2. 将下载的 **colorbox** 文件夹复制到此文件夹。
3. 在 **demo** 文件夹中创建 **jQueryS**、**CSS** 和 **contentFiles** 文件夹。这样可确保要使用的 jquery 文件 (**jquery-1.8.3.js**)、jquery 插件文件 (**jquery.colorbox.js**)、样式表 (**colorbox.css**) 和图像 (**Product1.png**) 都分开存储以供轻松实现。



### 注释

可以任意名称创建文件夹。事实上，不必要创建新的文件夹。这样可确保所有相关文件都存储在一个位置。根据文件的存储位置，需要在 `<SCRIPT>` 标记的 `src` 属性中指定对应文件的相对或绝对路径。

4. 将 **jquery.colorbox.js** 文件从下载的 **colorbox** 插件文件夹中提供的 **colorbox** 文件夹复制到 **jQueryS** 文件夹。
5. 将 **colorbox.css** 文件从下载的 **colorbox** 插件文件夹中提供的 **example1** 文件夹复制到 **CSS** 文件夹。
6. 将下载的 **jquery-1.8.3.js** jQuery 文件复制到 **jQueryS** 文件夹。
7. 将 **Product2.png** 文件复制到 **contentFiles** 文件夹。
8. 将 **images** 文件夹从下载的 **colorbox** 插件文件夹中提供的 **example1** 文件夹复制到 **CSS** 文件夹。
9. 在 **Notepad** 中输入以下代码，并将文件另存为 **LightboxDemo.html** 放置于计算机上创建了 **jQueryS**、**CSS** 和 **contentFiles** 文件夹的位置：

```
<!DOCTYPE HTML>
<HTML>
  <HEAD>
    <LINK media="screen" rel="stylesheet" href="CSS/colorbox.css" />
    <SCRIPT src="jQueryS/jquery-1.8.3.js"></SCRIPT>
    <SCRIPT src="jQueryS/jquery.colorbox.js"></SCRIPT>
    <SCRIPT type="text/javascript">
      $(document).ready(function(){
        $("a").colorbox();
      });
    </SCRIPT>
  </HEAD>
  <BODY>
```

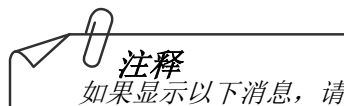
```

<H1>Product Demonstration</H1>
<A href="contentFiles/Product2.png">Laptop</A>
</BODY>
</HTML>

```

在上述代码中，a 表示 <BODY> 元素内定义的 <A> 元素。colorbox 插件的 colorbox() 方法用于在 lightbox 中显示 **Product1.png** 图像。

10. 在 Microsoft Internet Explorer 中打开 **LightboxDemo.html** 文件。

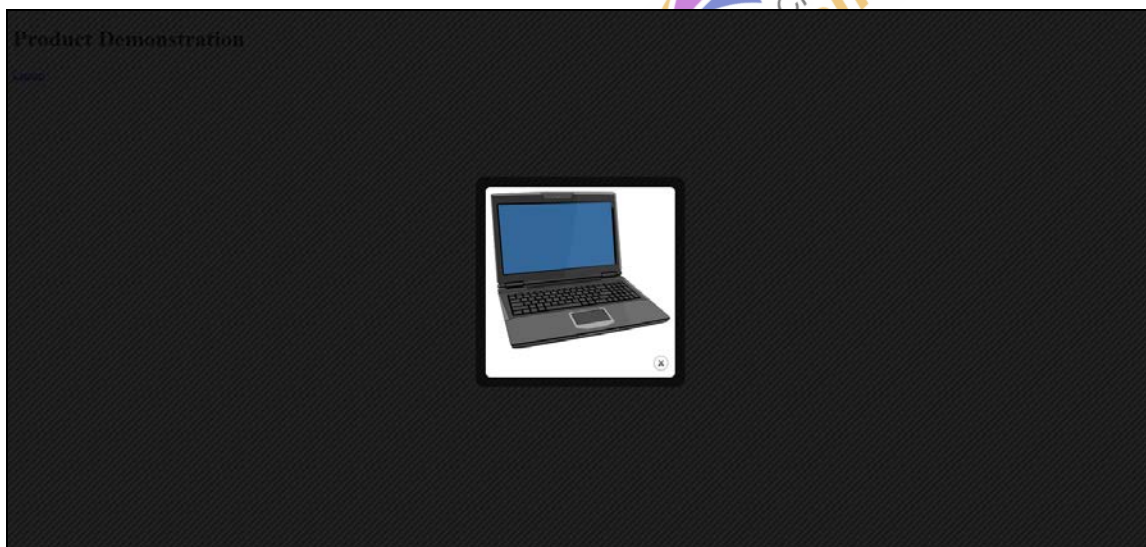


#### 注释

如果显示以下消息，请单击**允许阻止的内容**按钮。

**Internet Explorer 已限制此网页运行 ActiveX 控件的脚本。**

11. 单击 **Laptop** 超链接。将通过 lightbox 以默认转换效果显示产品图像的幻灯片视图，如下图所示。



显示产品图像的 Lightbox

在上图中，lightbox 显示带有默认转换效果的单个图像。然而，您可使用 colorbox 插件自定义图像在图像库或幻灯片中的显示。为此，colorbox 插件提供多个键，需要将它们作为参数传递到 colorbox() 方法。

下表列出可用于 colorbox() 方法的键。

键	描述
transition	它指定网页图像所需的转换类型，例如 fade、elastic 或 none。默认值是 elastic。
speed	它指定显示图像时的转换速度。默认值是 350 milliseconds。
width	它指定在其中显示图像的框架的总宽度。默认值是 false，表示框架的宽度不固定，将根据图像宽度而变化。
height	它指定在其中显示图像的框架的总高度。默认值是 false，表示框架的高度不固定，将根据图像高度而变化。
innerWidth	它指定在其中显示图像的框架的内宽度。这不包含边框和按钮。
innerHeight	它指定在其中显示图像的框架的内高度。这不包含边框和按钮。默认值是 false。
current	它指定使用 lightbox 显示多个图像时出现的文本（例如在显示 5 个图像中的第一个时显示文本 image 1 of 5）。默认值是 image{current} of {total}。{current} 将进行检测并替换为 ColorBox 运行时正在显示的图像的实际编号。{total} 将进行检测并替换为图像库中图像的实际数目。
slideshow	它指定将多个图像组合起来以供预览时是否需要创建图像幻灯片。默认值是 false。

### Colorbox 插件的键

思考以下代码，它通过指定在其中显示图像的框架的高度和宽度来创建 lightbox：

```

<!DOCTYPE HTML>
<HTML>
  <HEAD>
    <LINK media="screen" rel="stylesheet" href="CSS/colorbox.css" />
    <SCRIPT src="jQueryS/jquery-1.8.3.js"></SCRIPT>
    <SCRIPT src="jQueryS/jquery.colorbox.js"></SCRIPT>
    <SCRIPT type="text/javascript">
      $(document).ready(function(){
        $("a").colorbox({transition:"fade", height:"250", width:"220"});
      });
    </SCRIPT>
  </HEAD>

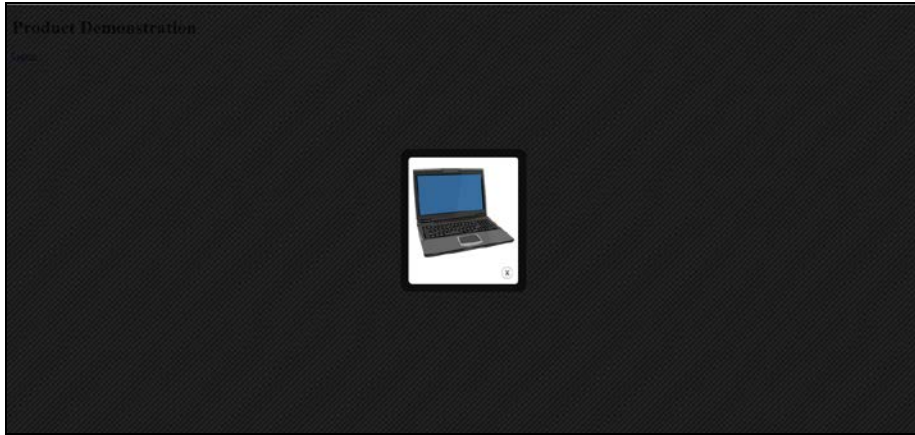
```

```

<BODY>
  <H1>Product Demonstration</H1>
  <A href="contentFiles/Product2.png">Laptop</A>
</BODY>
</HTML>

```

上述代码创建了一个高度为 250 px、宽度为 220 px 的 lightbox 并在其中通过 fade 转换显示图像。单击 **Laptop** 链接后，上述代码的输出如下图所示。



显示产品图像的 Lightbox

也可通过“上一个”和“下一个”按钮控制一组图像在图像库中的显示。要使用 lightbox 显示一组图像，需要使用 <A> 标记对每个图像添加链接，并为图像所定义的每个 <A> 标记的 rel 属性指定相同值。rel 属性指示当前文档与 href 属性指定的路径之间的关系。通过对所有图像的 rel 属性提供相同值，可将它们相关到一个组中。

假设您需要为一组图像 **Product1.png**、**Product2.png** 和 **Product3.png** 创建 lightbox。为此，您需要执行以下步骤：

1. 将 **Product2.png** 和 **Product3.png** 文件复制到 **contentFiles** 文件夹。
2. 在 **Notepad** 中输入以下代码，并将文件另存为 **LightboxGroupDemo.html** 放置于计算机上创建了 **jQueryS**、**CSS** 和 **contentFiles** 文件夹的位置：

```

<!DOCTYPE HTML>
<HTML>
  <HEAD>
    <LINK media="screen" rel="stylesheet" href="CSS/colorbox.css" />
    <SCRIPT src="jQueryS/jquery-1.8.3.js"></SCRIPT>
    <SCRIPT src="jQueryS/jquery.colorbox.js"></SCRIPT>
    <SCRIPT type="text/javascript">
      $(document).ready(function(){
        $("a").colorbox();
      });
    </SCRIPT>
  </HEAD>
  <BODY>
    <H1>Product Demonstration</H1>

```

```

<A rel="collection" href="contentFiles/Product1.png">Click here to view</A>
<A rel="collection" href="contentFiles/Product2.png"></A>
<A rel="collection" href="contentFiles/Product3.png"></A>
</BODY>
</HTML>

```

在上述代码中，在 `body` 标记内定义每个 `<A>` 标记的 `rel` 属性都指定了值 `collection`。这确保 `rel` 属性具有相同值的图像将分在一组，这样它们就可作为一个图像库的一部分而显示。

在 HTML 文件中保存上述代码然后在 Microsoft Internet Explorer 中打开时，将显示 **Click here to view** 超链接。单击超链接时，将显示带有“上一个”和“下一个”按钮的输出，如下图所示。



带有“上一个”和“下一个”按钮的产品图像



#### 注释

如果显示以下消息，请单击**允许阻止的内容**按钮。

**Internet Explorer 已限制此网页运行 ActiveX 控件的脚本。**

可使用 `colorbox` 插件通过将一些图像分在一组来创建这些图像的幻灯片。为此，需要在 `<A>` 标记中为要显示的所有图像的 `rel` 属性使用相同值，并将 `colorbox()` 方法的 `slideshow` 键设置为 `true`。

思考以下代码，它创建幻灯片并将它另存为 **SlideShow.html**：

```

<!DOCTYPE HTML>
<HTML>
  <HEAD>
  <STYLE>
  body{
  Background-color:"silver";}
</STYLE>

```

```

<LINK media="screen" rel="stylesheet" href="css/colorbox.css" />
<SCRIPT src=" jQueryS/jquery-1.8.3.js"></SCRIPT>
<SCRIPT src=" jQueryS/jquery.colorbox.js"></SCRIPT>
<SCRIPT>
    $(document).ready(function(){
        $("a").colorbox({slideshow:true,current:false});

    });
</SCRIPT>
</HEAD>
<BODY>

<A href=" contentFiles/Product1.png" rel="collection">Click Here to View
Slideshow of Products Available at the Site
</A>
<A href=" contentFiles/Product2.png" rel="collection"></A>
<A href=" contentFiles/Product3.png" rel="collection"></A>
</BODY>
</HTML>

```

在上述代码中，slideshow 键设置为 true 以创建图像幻灯片。current 键的值设置为 false。这防止在幻灯片显示期间显示描述当前图像编号的文本（例如 image 2 of 5）。

上述代码的输出如下图所示。



图像幻灯片

## 使用 Galleria 插件

Galleria 插件是用于创建外观优美专业的图像库的 jQuery 插件。此插件用于显示图像库并创建要在库中自动显示的图像缩略图。当鼠标指针移动到缩略图上时，将在同一网页上通过转换效果显示该图

像的更大版本。与 colorbox 插件不同，galleria 插件可用于创建图像幻灯片，但是它还能够创建要在图像库中显示的图像的缩略图。

要使用 galleria 插件创建图像库，需要从 Internet 下载 **galleria.zip** 文件，并将 **galleria.zip** 文件中的 galleria 文件夹抽取到系统上的所需位置。galleria 文件夹包含 **galleria-1.2.8.js** 文件，该文件是用于实现创建图像库功能的 galleria 插件。



## 活动 7.2：创建图像库

---

NIIT | training-china.com



## 练习问题

1. 以下哪个符号用于表示 jQuery?
  - a. #
  - b. \$
  - c. &
  - d. \*
2. 应该使用以下哪个语法选择和隐藏网页上的所有 <H1> 元素?
  - a. `$("h1").hide();`
  - b. `$(h1).hide;`
  - c. `("h1").hide;`
  - d. `$"h3".hide()`
3. Rick 希望在网页上 <P> 元素的内容结束处插入一个单词。他应该使用以下哪个函数执行此任务?
  - a. `append()`
  - b. `prepend()`
  - c. `insert()`
  - d. `after()`
4. John 是 Web 开发人员。他希望在网页上创建 lightbox。他可使用以下哪个方法创建 lightbox?
  - a. `colorbox();`
  - b. `lightbox();`
  - c. `color();`
  - d. `light();`
5. 以下哪个选项用于通过 `colorbox()` 方法创建图像幻灯片?
  - a. `slideshow`
  - b. `slideshowview`
  - c. `showslides`
  - d. `viewslideshow`



## 小结

在本章中，您学习了：

- jQuery 是开源且跨浏览器的 JavaScript 代码库，创建用途是使 JavaScript 编程更简单。
- 可在网页中使用 jQuery 以操作 HTML 元素，方法是下载轻量级 jQuery JavaScript 库并将它保存到您的系统。
- 为了确保 jQuery 代码在网页上执行时网页已完全加载，jQuery 提供了 `document.ready()` 函数。
- 使用 jQuery，可通过引用 HTML 元素的名称、ID 或所属的类来选择它们。
- jQuery 提供 CSS 选择器以修改 HTML 元素或文档的 CSS 属性。
- 在 jQuery 中，可使用函数或预定义的事件方法来处理事件。
- 可用于增加网页的视觉吸引力的一些预定义 jQuery 效果是：
  - 隐藏
  - 显示
  - 切换
  - 滑动
  - 淡化
  - 动画
- 图像翻转是当鼠标指针放在它上面或移动到它上面时图像外观更改的效果。
- 预加载图像是用于在执行网页脚本并且在浏览器窗口中显示网页前，将图像加载到浏览器高速缓存的技术。
- Image 对象可用于在应用程序中预加载图像。
- 图像库是一组缩略图或图像链接。
- 通过使用 jQuery 插件，可通过在运行时在同一网页上动态显示图像来创建图像库。
- 轻量级 colorbox 插件支持通过“上一个”和“下一个”按钮逐个显示一组图像。
- galleria 插件是用于创建外观优美专业的图像库的 jQuery 插件。

