

1. 添加对jQuery 库的引用

1.1. NuGet

- 开源 jQuery 包管理工具
- 添加方式
 - 在当前项目中创建一个保存 jQuery 文件的文件夹，复制到该文件中
 - 在当前项目中添加一个引用，引用到其所在的绝对路径下
 - 修改 Web.config 文件进行配置
- 包管理
 - 解决方案资源管理器
 - 管理 NuGet 程序包

1.2. 设置引用位置

- 在需要的网页或者用户控件中添加对其的引用
- 在母版页添加引用

1.3. 包含方式

- 在某一网页或控件中添加

```
<script src="FileName.ext" type="text/javascript" > </script>
```

- 通过在当前项目根目录下创建的程序文件添加
 - 引用 Scripts 文件夹下的 jquery-2.1.3.min.js 文件

```
<script src="/Scripts/jquery-2.1.3.min.js" type="text/javascript"></script>
```

- 嵌入到母版页的 ScriptManager 控件中
 - 该控件中有一个 <Scripts> 子元素，用来注册 JavaScript 文件，该文件将添加到浏览器的最后一个页面上

```
<asp:ScriptManager ID="ScriptManager1" runat="server">
<Scripts>
<asp:ScriptReference Path="~/Scripts/jquery-2.1.3.min.js" />
</Scripts>
```

- `~/`: 表示当前目录下

- 测试例

```
<asp:Content ID="Content2" ContentPlaceHolderID="ContentPlaceHolder1" runat="server">
<input id="Button1" type="button" value="button" />

<script type="text/javascript">
    $(document).ready(function () {
        $('#Content2').css('background-color', 'green')

        $('#Button1').click(function () {
            $('#Content2').css('background-color', 'red')
            .animate({ width: '100px', height: '800px' })
        });
    });
</script>
</asp:Content>
```

2. jQuery 语法

2.1. 文档就绪函数

- 当整个页面加载完成后才能通过编程操作元素

```
<script type="text/javascript">
$(document).ready(function(){
    // do operation after doc is load and ready
});
</script>
```

- 当页面就绪后即执行的快捷方式

```
$(function(){
    // do after web loaded
});
```

2.2. jQuery 选择器

2.2.1. 选择器

- 通用选择器 - 匹配页面中的全部元素

```
$('.*').css('font-family' , 'Arial')
```

- 元素选择器 - 字符串形式传入元素名参数
- 所有 `h2` 元素应用 `css` 样式

```
$('#h2').css('padding-bottom' , '10px')
```

- ID 选择器 - `# ID` 形式传入字符串的 ID 参数

```
$('#Button1').addClass('typeclass')
```

- 类选择器
 - 获得与给定类名匹配的元素的引用

```
<h1 class = "Highlight">Heading 1</h1>
<h2>Heading 2</h2>
<p class = "Highlight">First paragraph</p>
<p>Second paragraph</p>
```

```
$('.Highlight').css('background-color','red')
```

- 分组和合并选择器
- 修改 `h1` 和 `h2` 的 `css` 样式

```
$('#h1,h2').css('color' , 'orange')
```

- 修改 `Content2` 区域中包含 `p` 元素的样式

```
$('#Content2 p').css('border' , '1px solid red')
```

- 应用

```
$(function(){
    //通用
    $('*').css('color','green');
    //ID
    $('#Sidebar').css('border-bottom','2px solid red');
    //元素
    $('h1').bind('click' , function(){alert('Hello')}} );
    //类
    $('.SampleClass').addClass('PlesWait').hide(5000);
    //合并
    $('footer,header').slideUp('slow').slideDown('slow');
    //分组
    $('#Sidebar img').fadeTo(5000,0.1);
});
```

2.2.2. 基本筛选器

- `:first / :last` : 选择匹配集的首项和末项

```
$('#DemoTable tr:first').css() // 设置表格第一行
$('#DemoTable tr:last').css() // 设置表格最后一行
```

- `:odd / :even` : 选择匹配集中的奇数偶数行

```
$('#DemoTable tr:odd').css()
```

- `:eq(index)` : 找到索引为 `index` 的集合
- `:lt(index)` : 返回 less than 参数 `index` 的所有集合
- `:gt(index)` : 返回 greater than 参数 `index` 的所有集合

```
$('#DemoTable tr:eq(0)').css()
$('#DemoTable tr:lt(2)').css()
$('#DemoTable tr:gt(2)').css()
```

- `:header` : 匹配出页面中的所有标题

2.2.3. 高级筛选器

- `:contains(text)` : 匹配所有文本中包含 `text` 值的元素

```
$('td:contains("Row 3")').css()
```

- 如果没有 `td` 则会匹配整张表，因为当前整个 `table` 中包含了 `Row 3` 这一元素

- `:has(element)`：匹配出包含 `element` 元素的集合

```
$('.header:has("span")').css()
```

- 只会匹配出包含 `span` 的标题元素

- `[attribute]`：匹配所有包含给定的元素的集合

```
$('#[id]').css()  
$('input[type]').css()
```

- `[attribute=value]`：匹配包含给定的元素且值为 `value` 的集合

```
$('#id=fuck').css()
```

- 表单元素筛选器
 - `:input`、`:text`、`:password`、`:radio` 等等
 - 通过 `.attr` 设置表单元素的属性

3. 通过jQuery修改DOM

3.1. CSS方法

- `css(name,value)`

```
$('#h3').css('background-color','green');
```

- `css(name)`

```
alert($('#h1 span').css('font-style'));
```

- `css(properties)`：同时设置多个元素

```
$('#tbl td').css({'color':'blue' , 'font-family':'times new roman' , 'padding':'5px'  
});
```

- `addClass`、`removeClass`、`toggleClass`
- 都允许传递多个类，类间用 `,` 分隔

```
$('#h1').addClass('PleaseWait');
$('#h1').removeClass('PleaseWait');
```

- `attr(name)`：获取 HTML 元素的属性

3.2. 事件处理

- 在标记中直接定义

```
<input type="button" onclick="alert('你好');" value="确定"/>

<!-- 设定一个函数 -->

<input type="button" onclick="writeName();" value="确定"/>
```

- 将事件挂到整个匹配集
 - 即匹配集的每一个标记或元素都挂上当前的事件
- 实现对表格的每一行绑定 `mouseover` 事件，鼠标放置时候颜色改变
 - 但当鼠标移过的时候颜色会保留

```
$(function{
    $('#DemoTable tr')
    .bind('mouseover',function(){$(this).css({}); // this 设置当前的css
});
```

- 为了解决颜色滞留这类问题，有了链接
 - 即只需要对当前应用了 `jQuery` 的数据集在应用一下即可

```
$('#DemoTable tr')
    .bind('mouseover', function() {$(this).css('background-color', 'yellow')})
    .bind('mouseout', function() {$(this).css('background-color', '')});

// 即如下结构
$('#DemoTable tr').bind('mouseover', ...).bind('mouseout', ...);
```

4. jQuery 的其它功能

4.1. each 方法

- 用于遍历一个匹配集合
- 将匹配集的每一项作为参数传递给 `each`

```
$('#DemoTable td').each(function(){  
    alert(this.innerHTML); // 显示当前表格的内容  
});
```

4.2. parent 和 prev 方法

- 用于在文档树中上下移动找出上一或下一层次的元素
- `prev` 方法选择匹配元素的直系兄弟

```
alert($('#id td:contains("Row 1 Cell 2")').prev()[0].innerHTML)
```

- 发返回的结果就是 **Row 1 Cell 1**
- 返回的结果只有一项，返回的是一个集合，需要使用下标索引获得元素

- `parent` 返回的是与当前同一层次的元素

```
alert($('#id td:contains("Row 1 Cell 2")').parent()[0].innerHTML)
```

- 结果是围绕当前匹配值的单元格

5. 有效性验证

- 将校验规则写到控件中
- 将校验规则写到代码中

6. 课后习题

利用Timer 控件和局部更新技术，实现广告显示一段时间后自动关闭

```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="TimerExample.aspx.cs"
Inherits="Exise.Ch08.TimerExample" %>
<!DOCTYPE html>
<head id="Head1" runat="server">
<title></title>
<link href="jQuery/ui-1.8.18/redmond/jquery-ui.css" rel="stylesheet" type="text/css"
/>
<style>
body { text-align: center; font-size: 14px; }
#UpdatePanell { width: 600px; margin: 0 auto; border: 1px solid; height:100px; }
</style>
</head>
<body>
<form id="form1" runat="server">
<h3>
习题:这是一个广告，将播放 10 秒，10 秒后自动停止
</h3>
<asp:ScriptManager ID="ScriptManager1" runat="server"></asp:ScriptManager>
<asp:Timer ID="Timer1" runat="server" ontick="Timer1_Tick" Interval="1000">
</asp:Timer> <asp:UpdatePanel ID="UpdatePanell" runat="server">
<ContentTemplate>
<hr />
<div id="div4" style="width: 200px; text-align: center; border: 1px solid;"
runat="server"> 这是一个
广告</div>
<div id="div1" runat="server" style="width: 100px;font-size:larger;color:Red;text-
align:center
font-weight:bolder;" >0</div>
</ContentTemplate>
<Triggers>
<asp:AsyncPostBackTrigger ControlID="Timer1" /> </Triggers>
53 </asp:UpdatePanel> </form>
</body>
</html>
```

```
namespace Exise.Ch08 {
    public partial class AdExample : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e){
            if (IsPostBack==false) {
                Timer1.Interval = 1000;
                Timer1.Enabled = true;
            }
        }
    }
    protected void Timer1_Tick(object sender, EventArgs e) {
        string str = div1.InnerText;
        int v1 = int.Parse(str) + 1;
        if (v1 <= 10){
```



```

        div1.InnerText=v1.ToString();
        div4.InnerText = "广告播放第" + v1.ToString() + "次";
    }
    else {
        Timer1.Enabled = false; div4.InnerText = "广告播放完毕";
    }
}
}
}

```

服务器端每隔1s处理一次业务，10s处理完毕。设计一个页面利用局部更新技术，用进度条自动显示每次处理的速度

```

<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="TimerExample.aspx.cs"
Inherits="Exise.Ch08.TimerExample" %>
附录 各章习题参考答案 • 19 •

<!DOCTYPE html>
<html>
<head id="Head1" runat="server">
<title></title>
<link href="../../../jQuery/ui-1.8.18/redmond/jquery-ui.css" rel="stylesheet"
type="text/css" /> <style>
body { text-align: center; font-size: 12px; }
#UpdatePanel1 { width: 600px; margin: 0 auto; border: 1px solid; }
#div4 { width: 200px; margin: 0 auto; border: 1px solid; }
.ui-progressbar { display: inline-block; width: 90%; }
.ui-progressbar-value { text-align: right; }
</style>
</head>
<body>
<form id="form1" runat="server">
<h3>
习题:每隔 1 秒处理一次业务, 10 秒处理完毕
</h3>
<asp:ScriptManager ID="ScriptManager1" runat="server"></asp:ScriptManager> <asp:Timer
ID="Timer1" runat="server" ontick="Timer1_Tick"></asp:Timer> <asp:UpdatePanel
ID="UpdatePanel1" runat="server">
<ContentTemplate>
<hr />
进度 1:
<div class="ui-progressbar ui-widget ui-widget-content ui-corner-all">
<div id="div1" runat="server" style="width: 0%;" class="ui-progressbar-value ui-
widget-header ui-corner-left">
</div>
</div>
<hr />
</ContentTemplate>
<Triggers>
<asp:AsyncPostBackTrigger ControlID="Timer1" />
</Triggers>

```

```
</asp:UpdatePanel>
</form>
</body>
</html>
```

```
using System;
using System.Collections.Generic; using System.Linq;
using System.Web;
using System.Web.UI;

using System.Web.UI.WebControls; using System.Web.UI.HtmlControls; namespace
Exise.Ch08{
    public partial class TimerExample : System.Web.UI.Page {
        protected void Page_Load(object sender, EventArgs e){
            if (IsPostBack == false) {
                div1.InnerText = "0"; Timer1.Interval = 1000; Timer1.Enabled = true; }
        }
        protected void Timer1_Tick(object sender, EventArgs e) {
            int v1 = int.Parse(div1.InnerText.TrimEnd('%')) + 10;
            if (v1 <= 100){
                div1.InnerText = div1.Style["width"] = v1 + "%";
            }
            else{
                div1.InnerText = div1.Style["width"] = "100%"; Timer1.Enabled = false;
            }
        }
    }
}
```