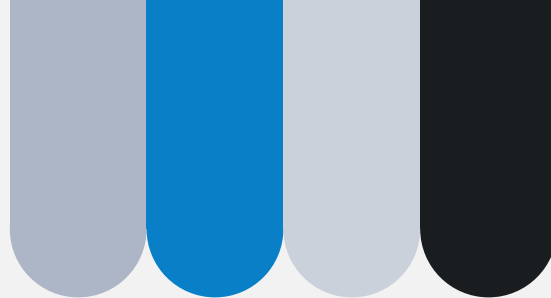


# 为什么要JavaScript库？

- JavaScript 高级程序设计（特别是对浏览器差异的复杂处理），通常很困难也很耗时。
- 为了应对这些调整，许多的JavaScript (helper) 库应运而生。
- 这些 JavaScript 库常被称为 JavaScript 框架。
- 目前，比较流行的JavaScript库有：
- jQuery、Prototype、MooTools
- 所有这些框架都提供针对常见JavaScript 任务的函数，包括动画、DOM 操作以及 Ajax 处理。



# jQuery概述

管理科学与工程学科  
耿方方



# 主要内容

- jQuery简介
- 搭建jQuery环境
- jQuery的简单应用

# jQuery简介

- jQuery是一个集JavaScript、CSS、DOM、Ajax于一体的强大框架体系。它是众多JavaScript类库中的一种。
- 脚本库能够帮助用户完成编码逻辑,实现业务功能。使用jQuery将极大地提高编写JavaScript代码的效率,让写出来的代码更加简介,更加健壮。同时,网络上丰富的jQuery插件也让开发者的工作变得更为轻松,让项目的开发效率有了质的提升。

- jQuery是继Prototype之后又一个优秀的JavaScript类库，是由美国人John Resig于2006年1月创建的一个开源项目，它的主旨是：用更少的代码，实现更多的功能（Write less, do more）。
- 2006年8月jQuery的第一个稳定版本，已经可以支持CSS选择符，时间交互以及Ajax交互。
- 到了2007年7月，jQuery1.1.3版本发布，这次小版本的变化包含了对jQuery选择符引擎执行速度的显著提升，也是从这个版本开始，jQuery的性能达到了Prototype、Mootools以及Dojo等同类JavaScript类库的水平。
- jQuery随着被人们熟知，越来越多的程序高手加入其中，完善并壮大其项目内容，现在已成为一个深受Web前端开发者喜爱的JavaScript类库。

- 便捷操作DOM元素

jQuery可以很方便的获取和修改页面中的某些元素，无论是删除、移动还是复制，jQuery都提供了一整套方便、便捷的方法，既减少了代码的编写，又大大提高了页面的体验度。

- 控制页面样式

jQuery可以十分方便的控制页面中的CSS样式，浏览器对页面文件的兼容性一直都是Web前端开发者最为头疼的事情，而使用jQuery操作页面的样式却可以很好的兼容各种浏览器。

- 对页面事件的处理

引用jQuery之后，可以使页面的表现层与功能开发分离，开发者更多的专注于程序的逻辑与功效；页面设计人员侧重于页面的优化与用户体验。然后通过事件绑定机制，轻松的将二者结合起来。

- 大量的插件在页面中的运用

在引用jQuery库之后，还可以使用大量的插件来完善页面的功能和效果，如表单插件、UI插件等，这些插件的使用不会与jQuery造成冲突，还会极大的丰富页面的展示效果，简化代码的开发成本、降低开发难度。

- 与Ajax技术完美结合

Ajax的异步读取服务器数据的方法，极大地方便了程序的开发，提升了用户的页面体验度；而引用jQuery库后，不仅完善了原有的功能，而且减少了代码的书写量，通过其内部对象或函数，就可以实现复杂的Ajax功能。



- JavaScript是一种为网站添加互动以及自定义行为的客户端脚本语言，有关JavaScript的信息我们在前章中已经详细的介绍过了。JavaScript的出现使得网页与用户之间实现了一种实时的、动态的和交互的关系。使得网页包含更多活跃的元素和更加精彩的内容。JavaScript本身存在三个弊端，即复杂的文档对象模型（DOM）、不一致的浏览器实现和缺乏便捷的开发、调试工具。
- 而jQuery则是JavaScript的一个类库，jQuery封装了很多预定义的对象和使用函数，能帮助使用者很轻松的建立有高难度交互的页面，并兼容各大浏览器，以方便Web前端开发者直接使用，而不需要再使用JavaScript语句书写大量的代码，是当前最流行的JavaScript类库之一。

- jQuery强调的理念是用更少的代码，实现更多的功能。jQuery独特的选择器、链式操作、事件处理机制和封装完整的Ajax都是其他JavaScript类库难以企及的，总结起来jQuery共有几种优势：轻量级、强大的选择器、出色的DOM操作封装、可靠的事件处理机制、完善的Ajax、不污染顶级变量、出色的浏览器兼容性、链式操作方式、隐式迭代、行为层与结构层的分离、丰富的插件支持、完善的文档、开源。

- ✓ 轻量级

jQuery非常轻巧，总大小只有30KB左右。

- ✓ 强大的选择器

jQuery允许开发者使用从CSS1到CSS3几乎所有的选择器，以及jQuery独创的高级而又复杂的选择器。另外还可以加入插件使其支持更多的选择器，甚至网页Web前端开发者可以编写属于自己的选择器。由于jQuery的这一特性，因此有一定CSS经验的Web前端开发者可以很容易的切入到jQuery的学习中来。

- ✓ 出色的DOM操作封装

jQuery封装了大量常用的DOM操作，使Web前端开发者在编写DOM操作相关程序的时候能够更加得心应手。jQuery可以轻松地完成各种原本十分复杂的操作，让JavaScript新手也能写出优秀的程序。

- ✓ 可靠的事件处理机制

jQuery在预留退路、循序渐进以及非入侵式编程思想方面，做的十分出色。

- ✓ 完善的Ajax

jQuery将所有的Ajax操作封装在一个函数里面，使得Web前端开发者处理Ajax的时候能够更加专心的处理业务逻辑，而无需考虑关系复杂的浏览器兼容性与Ajax使用问题。

### ✓ 不污染顶级变量

jQuery只创建一个名为jQuery的对象，其所有的函数方法都在该对象下，其别名\$也可以随时交出控制权，绝不会污染其它的对象。该特性使jQuery可以与其它JavaScript类库共存，而不用考虑后期可能冲突的问题。

### ✓ 出色的浏览器兼容性

作为一个流行的JavaScript类库，浏览器的兼容性是必须具备的条件之一。jQuery能够在IE 6.0+、FF 2+、Safari 2.0+和Opera 9.0+下正常运行，同时修复了一些浏览器之间的表现差异性问题，使Web前端开发者不必在为浏览器兼容性问题而苦恼。

### ✓ 链式操作方式

jQuery中最有特色的莫过于它的链式操作，即对发生在同一个jQuery对象上的一组动作，可以直接连写而无需重复获取对象。

### ✓ 隐式迭代

jQuery里的方法都被设计成自动操作对象集合，而不是单独的对象，这使得大量的循环结构变得不再必要，从而大幅的减少了代码量。

### ✓ 行为层与结构层的分离

Web前端开发者可以使用jQuery选择器选中元素，然后直接给元素添加事件。这种将行为层与结构完全分离的思想，可以使jQuery开发人员和HTML或其它页面开发人员的工作职能相分离，摆脱过去开发冲突或个人单干的开发模式。

- ✓ 丰富的插件支持

jQuery的易扩展性，吸引了来自全球的开发者来编写jQuery的扩展插件。目前已经有超过几百种的官方插件支持。而且还不断有新插件面世。

- ✓ 完善的文档

jQuery的文档非常丰富，减少了学习成本。

- ✓ 开源

jQuery是一个开源的产品，允许所有Web前端开发者自由的使用与修改。

- Query不需要安装，要使用它只需要该文件的一个副本，该副本可以放在外部站点上，也可以放在自己的服务器上。

以下代码为引用jQuery库的方法：

✓ `<script src="jquery/jquery.js"></script>`。

- 在引用样式表文件的代码之后，是包含JavaScript文件的代码，这里需要注意的是，引用jQuery库文件的`<script>`标签，必须放在引用自定义脚本文件的`<script>`标签之前。否则，在编写的代码中将引用不到jQuery框架。

- 在jQuery程序中，使用最多的莫过于“\$”美元符了，\$就是jQuery的一个简写格式，无论是页面元素的选择，还是功能函数的前缀都必须使用该符号，可以说它是jQuery程序的标志。
- 事件操作连接式书写。

```
<script>
  $(document).ready(function(){
    alert("欢迎使用 jQuery");
  })
</script>
```



- 在上面的例子中，用到了`$(document).ready`其类似于JavaScript中的`window.onload`，不同之处主要有两点：
- 一是执行时间不同，`$(document).ready`在页面框架加载完毕后就执行；而`window.onload`必须在页面全部加载完毕后（包含图片下载）后才能执行，明显是前者执行效率更快一点。
- 二是执行数量不同，`$(document).ready`可以重复写多个，并且每次执行结果不同；但是`window.onload`尽管可以执行多个，但是仅输出最后一个的执行结果，无法完成多个结果的输出。
- `$(document).ready(function() {})`可以简写为 `$(function() {})`

- 案例1: `<html>`
- `<head>`
- `<title>第一个简单的jQuery案例</title>`
- `<meta charset="UTF-8">`
- `<meta name="viewport" content="width=device-width, initial-scale=1.0">`
- `<script src="jQuery/jquery-3.2.1.js" type="text/javascript"></script>`
- `<script>`
- `$(function() {`
- `alert('您好, 我的第一个jQuery代码!');`
- `});`
- `</script>`
- `</head>`
- `<body>`
- `</body>`
- `</html>`

- 什么是DOM对象？
- DOM的每一个页面都是一个DOM对象，通过传统的JavaScript方法访问页面中的元素，就是访问DOM对象。
- 例如：<div id=" divTmp" >测试对象</div>
- <div id=" divOut" ></div>
- 通过JavaScript代码可以访问DOM对象和获取或设置其内容值。

```
var tDiv=document.getElementById( "divImp" );获取DOM对象
```

```
var oDiv=document.getElementById( "divOut" );获取DOM对象
```

```
var cDiv=tDiv.innerHTML; 获取DOM对象中的内容
```

```
oDiv.innerHTML=cDiv 设置DOM对象的内容
```

- 什么是jQuery对象？
- 在jQuery库中，通过本身自带的方法获取页面元素的对象，我们称之为jQuery对象；为了同样实现在ID为“divOut”的标记中显示ID为“divTmp”的标记内容，采用jQuery访问页面元素的方法。

```
var tDiv=$( "#divImp" );获取jQuery对象
```

```
var oDiv=$( "#divOut" );获取jQuery对象
```

```
var cDiv=tDiv.html(); 获取jQuery对象中的内容
```

```
oDiv.html(cDiv) 设置jQuery对象的内容
```

通过代码的对比，发现jQuery对象访问方法比DOM对象访问方法更简单、高效，它们实现同样的功能。

- 在jQuery框架中，通过自带的JavaScript编程，提供全部的CSS3下的选择器，开发者可以首先定义自己的样式文件，然后通过jQuery中的addClass（）方法将该样式轻松地添加到页面指定的某元素中，而不用考虑浏览器的兼容性。

- 案例2: `<html>`
- `<head>`
- `<title>jQuery控制CSS</title>`
- `<meta charset="UTF-8">`
- `<meta name="viewport" content="width=device-width, initial-scale=1.0">`
- `<script src="jQuery/jquery-3.2.1.js" type="text/javascript"></script>`
- `<style>`
- `.divDefault {width:260px;font-size:10px;padding:5px;}`
- `.divClick {width:260px;border:1px solid #666;font-size: 10px;`
- `background-color:#eee;padding:5px;}`
- `</style>`

```
<script>
    $(function(){
        $(".divDefault").click(function(){
            $(this).toggleClass("divClick")
            .html("点击后的样式");
        });
    });
</script>
</head>
<body>
    <div class="divDefault">点击前样式</div>
</body>
</html>
```