JQuery入门到精通

目录

[JQuery入门到精通 1](#_Toc472151434)

[一、jQuery概述 2](#_Toc472151435)

[1、jQuery历史 2](#_Toc472151436)

[2、Javascript代码库 2](#_Toc472151437)

[3、常用的Javascript代码库 2](#_Toc472151438)

[4、使用jQuery代码库 2](#_Toc472151439)

[5、下载jQuery代码库 2](#_Toc472151440)

[6、快速入门 3](#_Toc472151441)

[7、jQuery中的$内核分析 3](#_Toc472151442)

# 一、jQuery概述

## 1、jQuery历史

Javascript由网景公司（Netscape）开发的一款基于客户端浏览器、面向对象、事件驱动式的脚本语言。

Javascript有三个缺点：

① 复杂的DOM操作

② 不一致的浏览器体现

③ 调试工具相当匮乏

Ajax的出现，当时Google开发很多基于Ajax的Web应用（Gmail邮箱、Google搜索建议、Google Map…）

## 2、Javascript代码库

为了解决以上问题，很多JS开发者研发Javascript代码库（对Javascript原生代码进行封装）。

## 3、常用的Javascript代码库

Prototype代码库

Dojo代码库

YUI代码库（几个核心的CSS与Javascript）

Mootools代码库

jQuery代码库

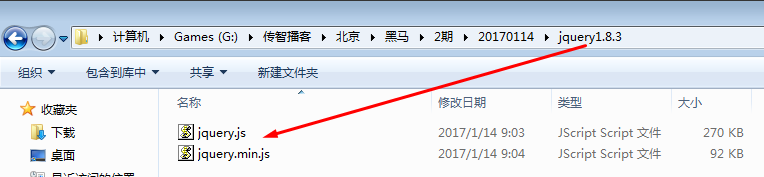
## 4、使用jQuery代码库

一款免费且开放源代码的JavaScript代码库

## 5、下载jQuery代码库

<http://www.jquery.com/> jQuery官网

下载后如下图所示：



jquery.js ：完整版的jquery源代码，具有可读性

jquery.min.js ：压缩版的jquery源代码，体积小，但是不具有可读性，一般用于生产环境

推荐书：《jQuery技术内幕》

## 6、快速入门

demo01\_rumen.html 示例代码：



## 7、jQuery中的$内核分析

问题：在原生代码中，我们都是通过document.getElementById获取元素的，为什么通过一个$就可以实现同样的效果呢？

① 为什么需要$获取元素（原生代码案例）



由于每次获取dom元素都需要通过document.getElementById过于麻烦，而且容易出错。

② 封装一个函数，用于获取指定id的dom元素



以上代码中我们封装的函数getDom的确可以用于获取指定id的dom对象，但是有一个缺点，因为其是有名函数，所以在实际项目开发中，可能会产生函数命名冲突。

③ 封装$函数，用于获取指定id的dom元素

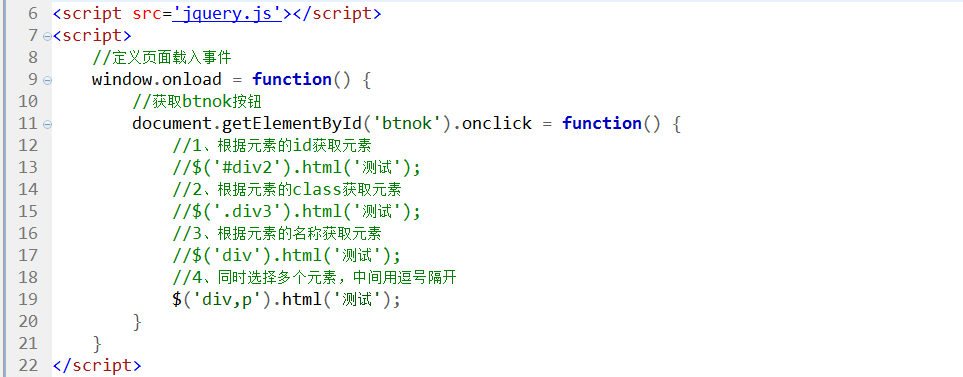


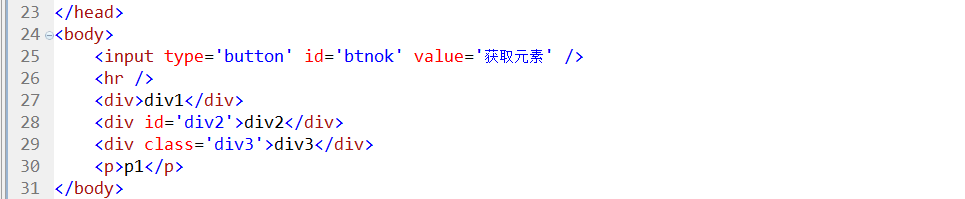
# 二、强大的选择器

## 1、基本选择器

* #id ：根据元素的id属性获取元素
* element ：根据元素的标签名称获取元素
* selector1,selector2 ：同时选择多个元素，中间用逗号隔开
* .class ：根据元素的class属性获取元素

demo03\_jiben.html 示例代码：

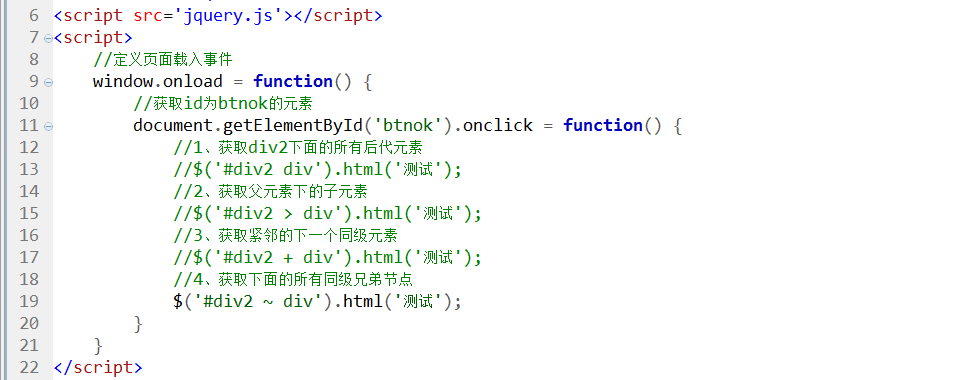


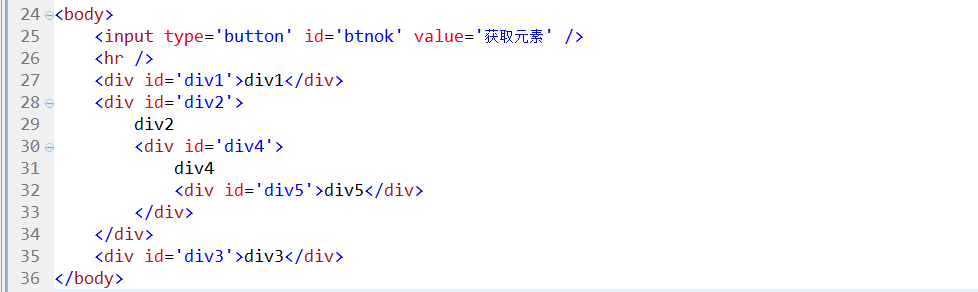


## 2、层级选择器

* ancetor descendant ：选择祖先元素下的所有后代元素
* parent > child ：获取父元素下的所有子元素
* prev + next ：获取当前元素紧邻的下一个同级元素
* prev ~ siblings ：获取当前元素下的所有同级元素

demo04\_cengji.html 示例代码：



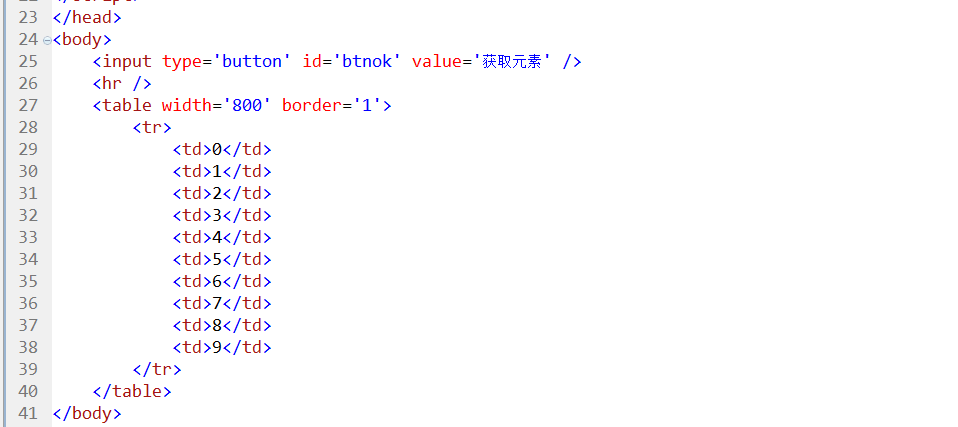


## 3、简单选择器

* :first ：获取第一个元素
* :last ：获取最后一个元素
* :eq(index) ：根据元素的index索引来获取元素，默认索引从0开始
* not(selector) ：获取除指定选择器以外的其他元素

demo05\_jiandan.html 示例代码：





扩展：Emmet插件，教程：<http://www.iteye.com/news/27580>

# 三、window.onload与ready方法

## 1、window.onload

window.onload代表页面载入事件，在原生代码中，我们可以通过window.onload实现页面载入，但是在jQuery中不建议使用此方法。

## 2、ready方法（jQuery代码库）

① ready完整写法

$(document).ready(function(){

//相关jQuery代码

});

demo06\_ready.html 示例代码：



② ready方法简写形式

$(function(){

//相关jQuery代码

});

demo06\_ready.html 示例代码：



## 3、window.onload与ready方法区别

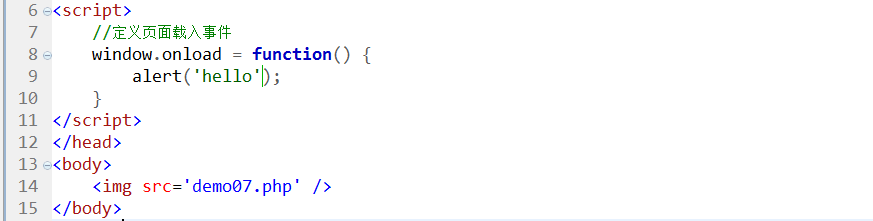
有区别的：

① 在html页面中，只能有一个window.onload，但是可以拥有多个ready方法。

② window.onload必须等待页面中的所有元素（包括外部元素）全部加载完毕后才开始执行。

ready方法不需要等待页面中的所有元素加载完毕，只要DOM结构渲染成功后，就会立即执行，可能外部元素还没有加载完毕。所以ready方法的执行效率要高于window.onload。

demo07\_qubie.html 示例代码：



demo07.php 生成一张背景颜色为绿色的画布 ：

讲个故事（爱情、悲伤、悬疑故事）：小明、小强

故事背景：小明（男）暗恋小强（男）

小明准备写一封情书给小强，写情书步骤：

第一步：准备一张白纸

第二步：润色

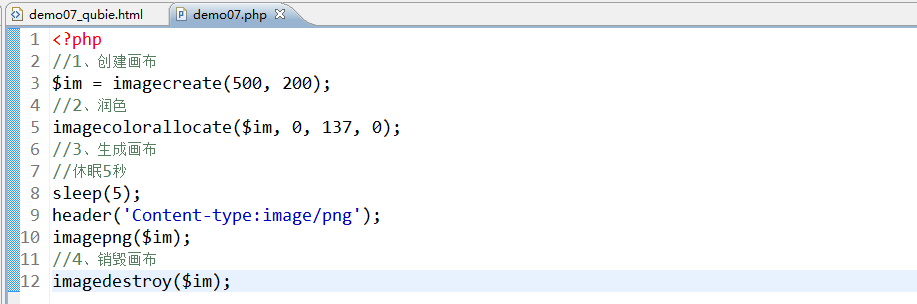
第三步：写字

第四步：写好了（生成画布）

传递给小强

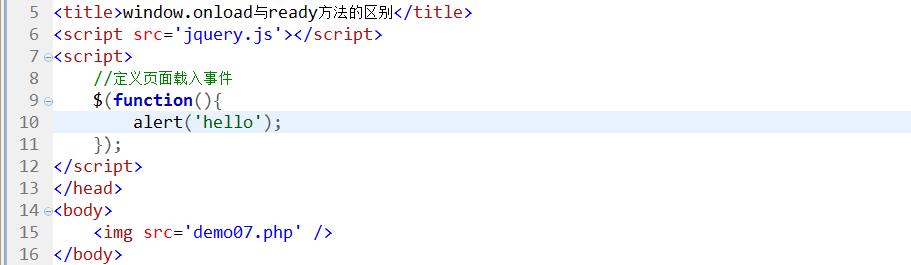
第五步：小强觉得恶心，销毁了

故事结束了…



window.onload运行结果：先等待图像加载完毕，然后弹出“hello”字符串。

demo07\_qubie.html 示例代码：



ready方法运行结果：先弹出“hello”字符簇，然后在继续加载外部图像资源。

# 四、DOM对象与jQuery对象

## 1、DOM对象

在Javascript中，通过document.getElement方式获取的对象都是dom对象。

var dom = document.getElementById(‘id’);

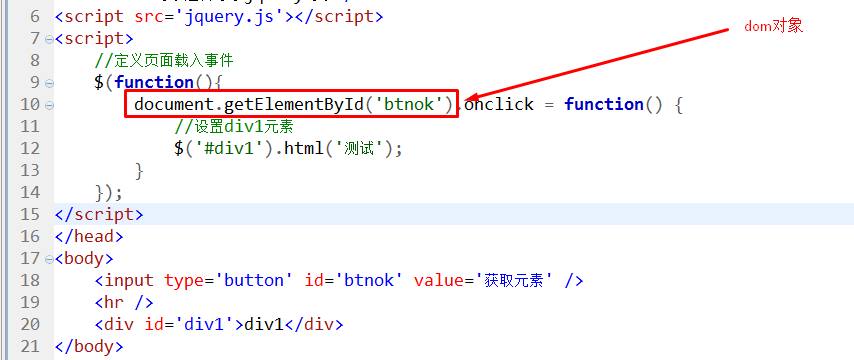
## 2、jQuery对象

在jQuery中，通过$符号获取的对象都是jquery对象。

var jq = $(‘#id’);

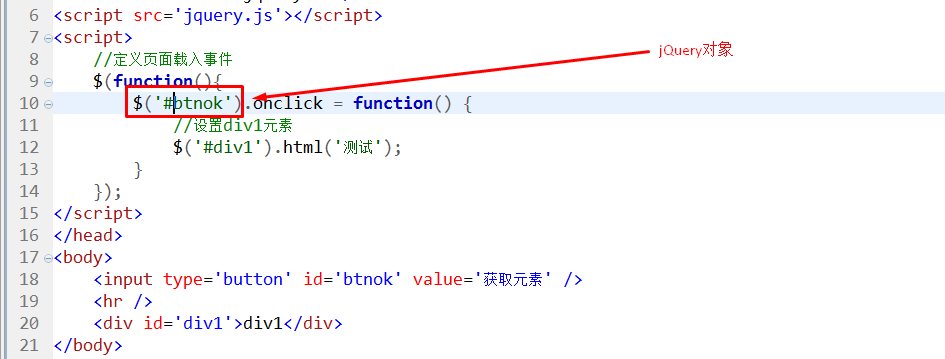
## 3、DOM对象是否等于jQuery对象呢？

DOM对象：



运行结果：正常

jQuery对象：



运行结果：异常，无法执行。

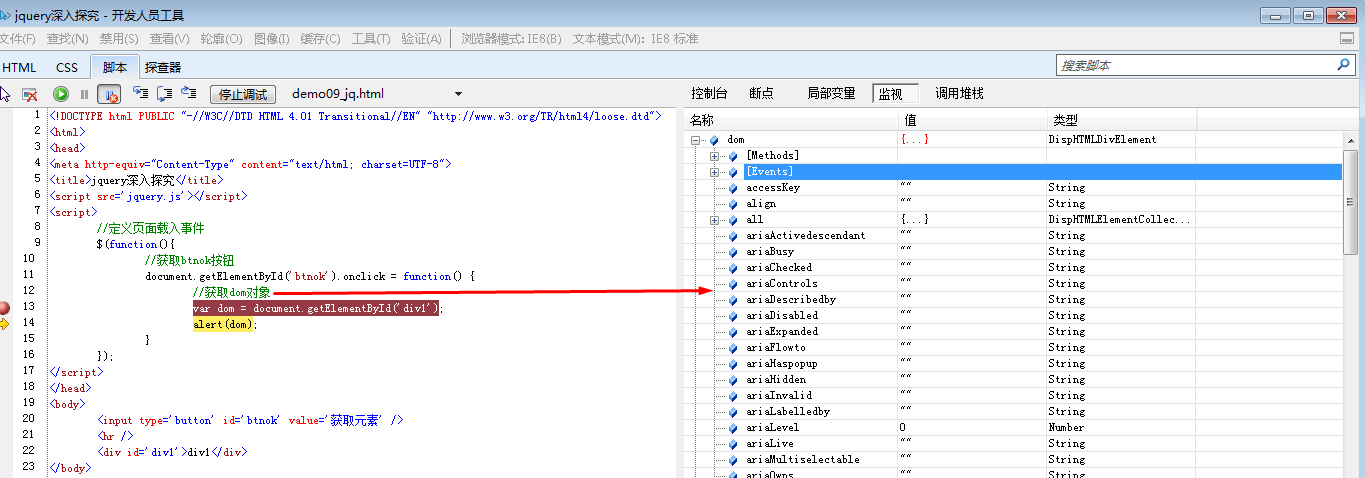
所以可以得出结论“DOM对象与jQuery对象是两个完全不同的对象，不能交换使用”。

## 4、jQuery对象到底是什么东东?

demo09\_jq.html 示例代码：



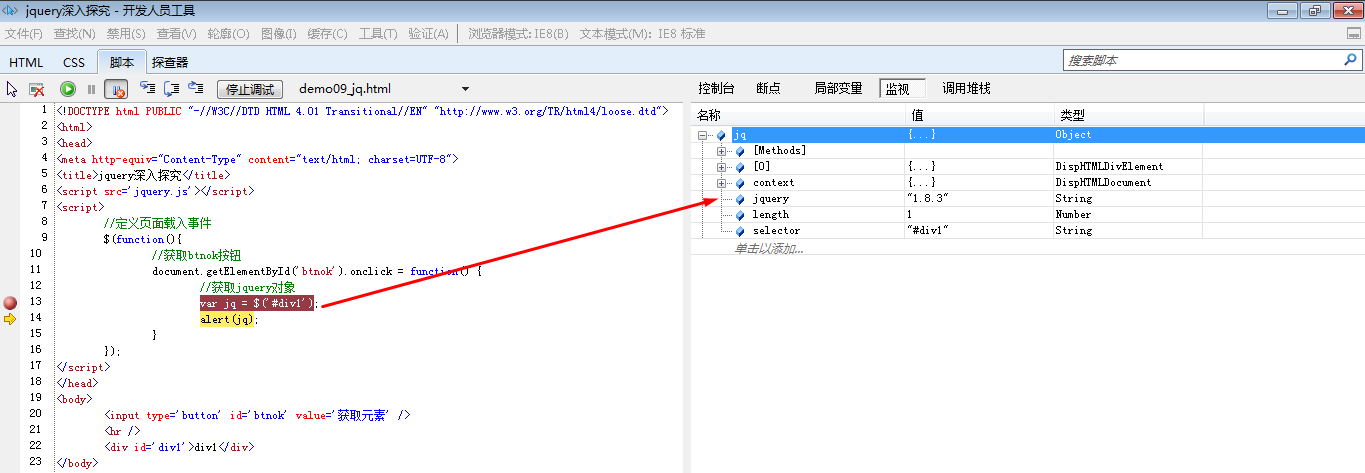
调试结果：



demo09\_jq.html 示例代码：



调试结果：

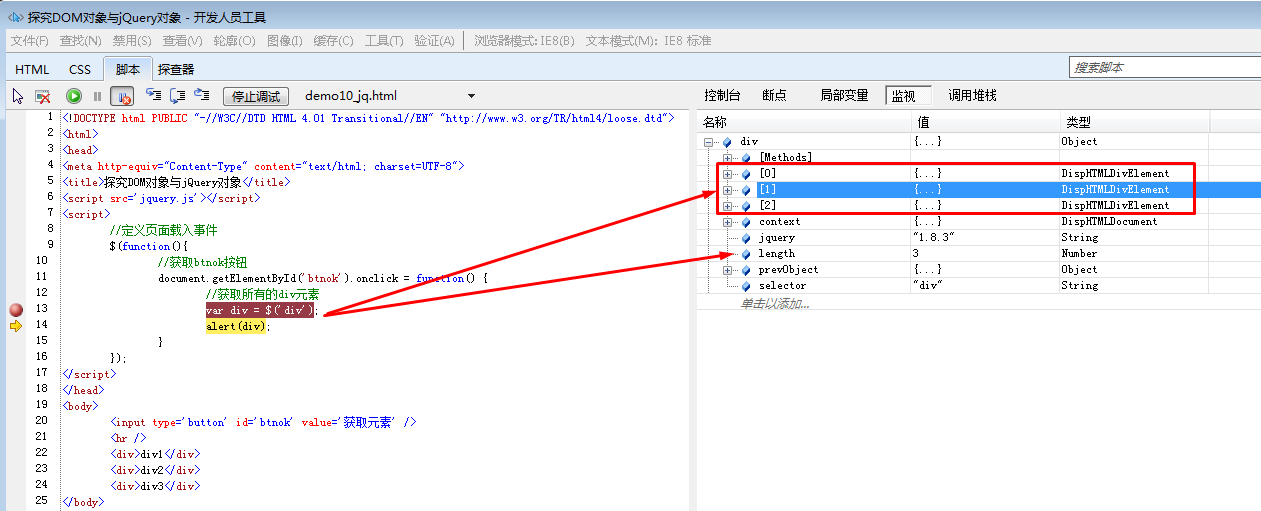


## 5、探究DOM对象与jQuery对象之间的关系

demo10\_jq.html 示例代码：



调试结果：



由上图可知：jQuery对象的本质是一个数组，数组中的每个元素都是一个DOM对象。

## 6、DOM对象与jQuery对象的相互转化

① jQuery对象转化为DOM对象：

获取DOM对象 = jQuery对象[索引下标];

或

获取DOM对象 = jQuery对象.get(索引下标);

demo10\_jq.html示例代码：



② 把DOM对象转化为jQuery对象

获取jQuery对象 = $(DOM对象);

demo11\_dom.html 示例代码：



# 五、jQuery中的相关属性与方法

## 1、attr属性

* attr(name) ：获取name属性的值
* attr(key,value) ：设置元素的key属性值为value
* removeAttr(name) ：移除元素的name属性

demo12\_attr.html 示例代码：



## 2、class属性

* addClass(class) ：向元素中添加class属性
* removeClass(class) ：移除元素的class属性

demo13\_class.html 示例代码：



## 3、css属性（style属性）

css(name) ：获取name属性的值

css(key,value) ：设置元素的key属性的value值

demo14\_css.html 示例代码：



## 4、文本与值

html() 方法：相当于innerHTML，用于设置元素中间的内容

val() 方法：相当于value属性，专门用于设置form表单元素的value属性

text() 方法：相当于innerText，用于设置元素中间的内容，只设置文本内容

demo15\_text.html 示例代码：



# 六、jQuery中的事件编程

## 1、使用bind方法实现事件绑定（重点）

* bind(type,[data],fn) ：为元素的某个事件绑定相应的事件处理程序

参数说明：

type ：触发的事件类型，不带’on’前缀，如click，mouseover，mouseout

[data] ：可省略，代表事件发生时所传递的参数

fn ：通常是一个匿名函数，代表事件发生时所执行的处理程序

* bind({type:fn,type:fn}) ：同时为元素的某些事件绑定相应的事件处理程序

参数说明：

bind方法中的参数要求是一个JSON格式的数据，里面可以拥有多个属性。

type：事件的类型，不带’on’前缀

fn：通常是一个匿名函数

* one(type,[data],fn) ：为元素绑定某个事件，但是只触发一次

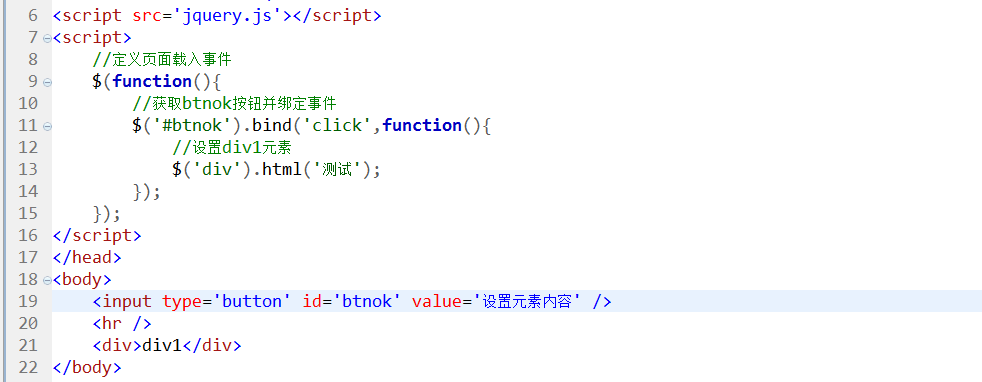
参数说明：

type ：触发的事件类型，不带’on’前缀，如click，mouseover，mouseout

[data] ：可省略，代表事件发生时所传递的参数

fn ：通常是一个匿名函数，代表事件发生时所执行的处理程序

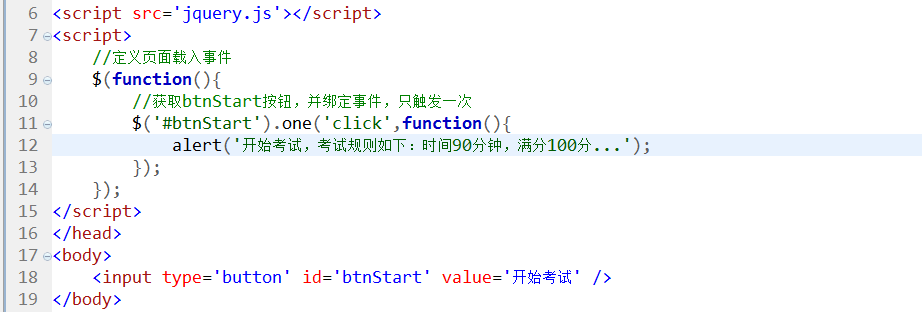
demo16\_bind.html 示例代码：



demo17\_bind.html 示例代码：



demo18\_one.html 示例代码：



## 2、回顾Javascript中常用的事件

* blur(fn) ：当失去焦点时触发
* change(fn) ：当下拉菜单改变时触发
* click(fn) ：当单击时触发
* dblclick(fn) ：当双击时触发
* focus(fn) ：当获得焦点时触发
* keydown(fn) ：当键盘按下时触发（捕获功能键）
* keyup(fn) ：当键盘弹起时触发
* keypress(fn) ：当键盘按下时触发（不能捕获功能键）
* load(fn) ：当页面载入完毕后触发，等价于ready方法
* unload(fn) ：当页面关闭时触发
* mousedown(fn) ：当鼠标按下时触发
* mouseup(fn) ：当鼠标弹起时触发
* mousemove(fn) ：当鼠标移动时触发
* mouseover(fn) ：当鼠标悬浮时触发
* mouseout(fn) ：当鼠标离开时触发
* resize(fn) ：当窗口大小改变时触发
* scroll(fn) ：当滚动条滚动式触发
* select(fn) ：当文本框中的文本被选中时触发
* submit(fn) ：当表单提交时触发

## 3、两个特殊的事件方法

① hover(over,out) ：鼠标悬浮与鼠标离开切换事件

参数说明：

over ：匿名函数，鼠标悬浮时所触发的程序

out ：匿名函数，鼠标离开时所触发的程序

② toggle(fn,fn) ：单击切换事件，第一次单击触发第一个fn，第二次单击触发第二个fn，第三次触发第一个fn，fn可以拥有多个。

demo19\_hover.html 示例代码：



demo20\_toggle.html 示例代码：



# 七、jQuery中的动画效果

## 1、基本效果

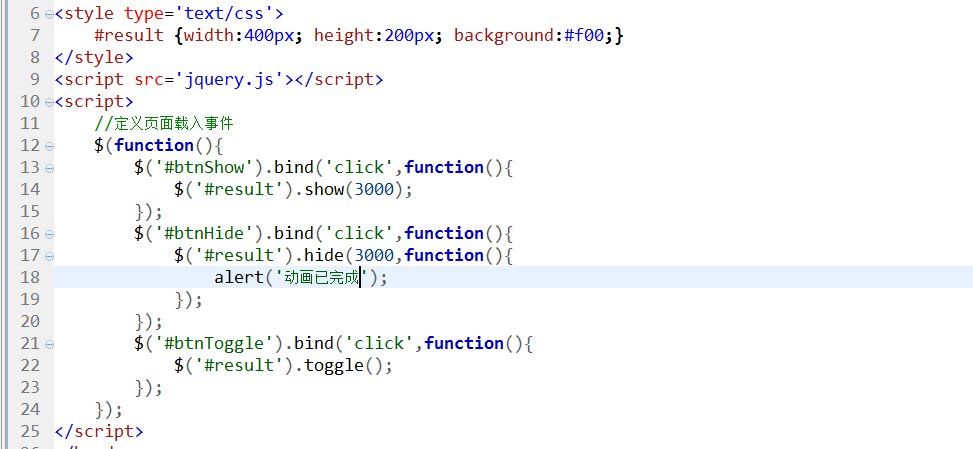
* show(speed,[callback]) ：以动画效果显示
* hide(speed,[callback]) ：以动画效果隐藏
* toggle(speed,[callback]) ：以动画效果显示或隐藏

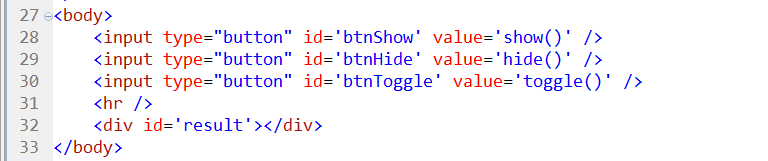
参数说明：

speed ：动画持续时间

[callback] ：可选参数，动画完成时所触发的程序

demo21\_jiben.html 示例代码：





## 2、滑动特效

* slideDown(speed,[callback]) ：以动画效果向下滑动
* slideUp(speed,[callback]) ：以动画效果向上滑动
* slideToggle(speed,[callback]) ：以动画效果向上或向下滑动

demo22\_slide.html 示例代码：



## 3、自定义动画

基本语法：

jQuery对象.animate(params,[speed]);

参数说明：

params ：要求是一个JSON格式的数据{}

speed ：动画的持续时间

demo23\_animate.html 示例代码：



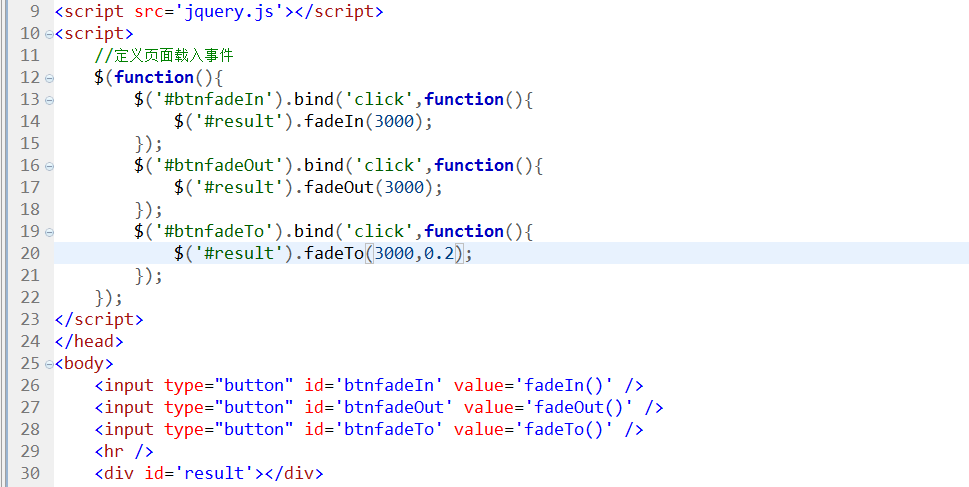
## 4、透明度动画

* fadeIn(speed,[callback]) ：以动画效果淡入
* fadeOut(speed,[callback]) ：以动画效果淡出
* fadeTo(speed,opacity,[callback]) ：设置元素的透明度

参数说明：

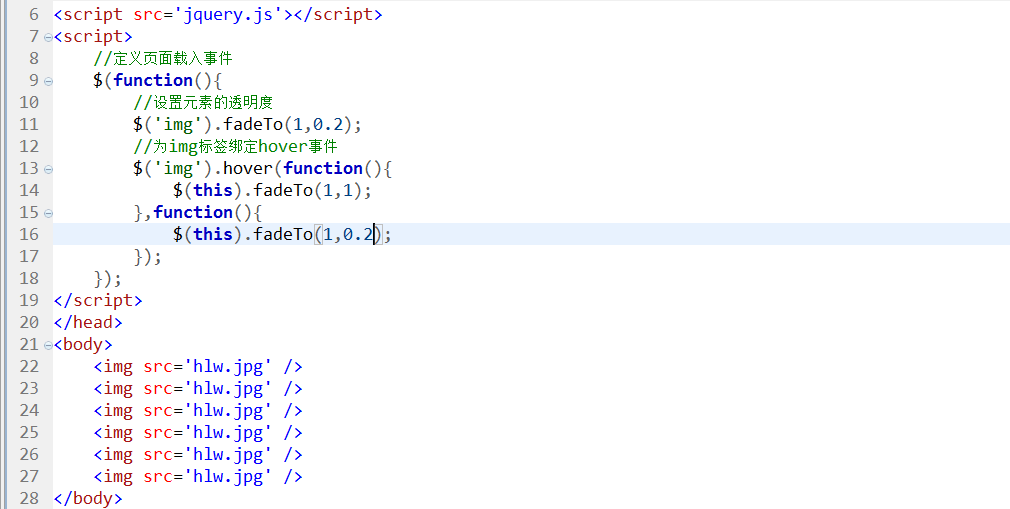
opacity ：设置元素的透明度，从0到1,0代表全透明，1代表不透明，0-1之间半透明

demo24\_fade.html 示例代码：



## 5、案例

一组图片的淡入淡出，demo25\_anli.html 示例代码：



特别注意：事件绑定中的this关键词，在jQuery事件绑定中，其事件处理程序的内容都有一个特殊的关键词叫做this，这个this代表当前正在操作的dom对象，一定要记住，this是dom对象，如果想使用jQuery中的属性或方法必须使用$(this)进行封装。

# 八、jQuery中的文档操作（节点操作）

## 1、内部插入

* append(content) ：在元素内容的尾部插入content
* appendTo(content) ：把匹配的内容插入到content元素内容的尾部
* prepend(content) ：在元素内容的头部插入content
* prependTo(content) ：把匹配的内容插入到content元素内容的头部

<div>content文本信息content</div>

demo26\_neicha.html 示例代码：



## 2、外部插入

* after(content) ：在元素尾部插入内容
* before(content) ：在元素头部插入内容
* insertAfter(content) ：把匹配到的内容插入到content元素的尾部
* insertBefore(content) ：把匹配到的内容插入到content元素的头部

content <div>hello</div>content

demo27\_waicha.html 示例代码：



## 3、删除操作

* empty() ：只清空元素的内容，但是元素本身不删除
* remove() ：删除元素内容同时删除元素

demo28\_del.html 示例代码：



## 4、查询操作（类似选择器）

* eq(index) ：根据元素的索引来获取元素
* filter(expr) ：筛选指定元素
* not(expr) ：匹配除指定选择器以外的其他元素
* children([expr]) ：获取当前元素的所有子元素
* find(expr) ：获取当前元素的所有后代元素
* next([expr]) ：获取当前元素紧邻的下一个同级元素
* prev([expr]) ：获取当前元素紧邻的上一个同级元素
* parent([expr]) ：获取当前元素的父元素
* siblings([expr]) ：获取当前元素的所有同级兄弟元素

demo29\_find.html 示例代码：



