jQuery学习笔记



jQuery是一个快速、简洁的JavaScript框架

2018年3月2日

目录

[一、JQuery 4](#_Toc508362049)

[1. 什么是jquery 4](#_Toc508362050)

[2. Jquery宗旨和特点 4](#_Toc508362051)

[3. 出现年代 5](#_Toc508362052)

[4. 其他相关的JavaScript框架包 5](#_Toc508362053)

[5. 简单使用 6](#_Toc508362054)

[二、选择器 6](#_Toc508362055)

[1. 基本选择器 6](#_Toc508362056)

[2. 层次选择器 6](#_Toc508362057)

[2.1. $( s1 s1 ) [父子] 6](#_Toc508362058)

[2.2. $( s1 > s1 ) [父子] 7](#_Toc508362059)

[2.3. $( s1 + s1 ) [兄弟] 7](#_Toc508362060)

[2.4. $( s1 ~ s1 ) [兄弟] 8](#_Toc508362061)

[3. 并且（过滤）选择器 8](#_Toc508362062)

[3.1. 基本使用 8](#_Toc508362063)

[3.2. 复杂使用 9](#_Toc508362064)

[4. 内容过滤选择器 10](#_Toc508362065)

[4.1. :contains（内容） 10](#_Toc508362066)

[4.2. :empty（空选择器） 10](#_Toc508362067)

[4.3. :has（存在选择器） 11](#_Toc508362068)

[4.4. :parent（选择器） 11](#_Toc508362069)

[5. 表单域选中选择器 11](#_Toc508362070)

[四、属性操作 11](#_Toc508362071)

[五、快捷操作 12](#_Toc508362072)

[1. class属性值操作 12](#_Toc508362073)

[2. 标签包含内容操作 12](#_Toc508362074)

[2.1. html()和text()方法的区别 13](#_Toc508362075)

[3. CSS样式操作 13](#_Toc508362076)

[3.1. CSS()样式操作的特点 13](#_Toc508362077)

[4. value属性快捷操作 14](#_Toc508362078)

[4.1. 复选框value操作 14](#_Toc508362079)

[4.2. 下拉列表value操作 15](#_Toc508362080)

[4.3. 单选value操作 15](#_Toc508362081)

[5. 复选框快捷操作 15](#_Toc508362082)

[六、$符号的由来 17](#_Toc508362083)

[七、jQuery对象与DOM对象的关系 17](#_Toc508362084)

[1. 互调用对方成员 18](#_Toc508362085)

[2. jQuery对象封装DOM对象 18](#_Toc508362086)

[3. jQuery对象和DOM对象的相互转化 19](#_Toc508362087)

[3.1. jquery对象 —》DOM对象 19](#_Toc508362088)

[3.2. DOM对象 —》jquery对象 19](#_Toc508362089)

[八、jQuery框架对象分析 19](#_Toc508362090)

[1. jQuery对象 19](#_Toc508362091)

[2. $ 对象 20](#_Toc508362092)

[九、遍历方法 21](#_Toc508362093)

[十、加载事件 21](#_Toc508362094)

[1. Jquery加载事件实现 21](#_Toc508362095)

[2. Jquery加载事件与传统加载事件区别 22](#_Toc508362096)

[2.1. 设置个数 22](#_Toc508362097)

[2.2. 执行时机不一样 22](#_Toc508362098)

[2.3. jQuery加载事件的原理 23](#_Toc508362099)

[十一、普通（简单）事件操作 23](#_Toc508362100)

[十二、jQuery对网页文档的操作 24](#_Toc508362101)

[1. 节点追加 24](#_Toc508362102)

[1.1. 父子关系追加 24](#_Toc508362103)

[1.2. 兄弟关系追加 25](#_Toc508362104)

[2. 节点替换 26](#_Toc508362105)

[3. 节点删除 26](#_Toc508362106)

[4. 节点复制 26](#_Toc508362107)

[十三、属性选择器 26](#_Toc508362108)

[十四、事件绑定 27](#_Toc508362109)

[1. 事件绑定 27](#_Toc508362110)

[1.1. Jquery事件绑定 28](#_Toc508362111)

[1.2. 取消事件绑定 28](#_Toc508362112)

[2. 事件对象、阻止浏览器默认动作、阻止事件冒泡 28](#_Toc508362113)

[十五、动画效果 29](#_Toc508362114)

[1. 基本动画 30](#_Toc508362115)

[2. 垂直动画 31](#_Toc508362116)

[3. 颜色渐变动画 31](#_Toc508362117)

[十六、封装ajax使用 32](#_Toc508362118)

[十七、案例-地区三级联动 33](#_Toc508362119)

[1. 省份的获取和显示 33](#_Toc508362120)

[2. 显示对应城市信息 33](#_Toc508362121)

[十八、迷你版jQuery框架开发 34](#_Toc508362122)

[十九、jQuery插件 35](#_Toc508362123)

[1. 什么是jQuery插件 35](#_Toc508362124)

[2. 两种形式丰富方法 35](#_Toc508362125)

[3. 制作一个表格隔行换色的插件 36](#_Toc508362126)

[4. 使用“成品jQuery插件”的步骤 36](#_Toc508362127)

# 一、JQuery

## 什么是jquery

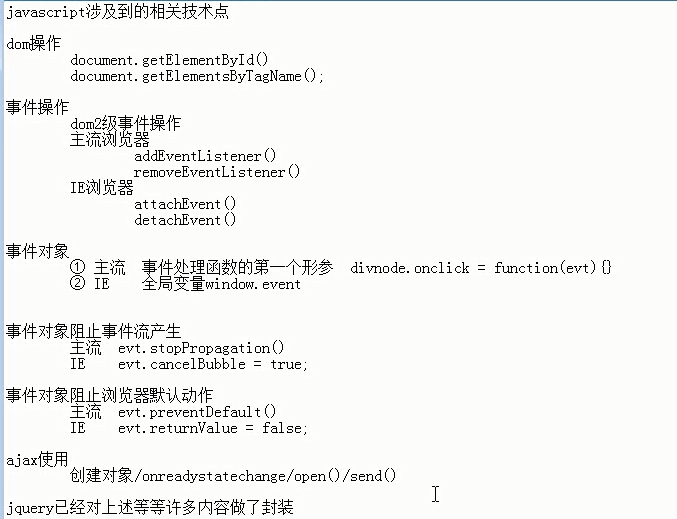
jQuery是一个快速、简洁的JavaScript框架，是继Prototype之后又一个优秀的JavaScript代码库（*或JavaScript框架*）。jQuery设计的宗旨是“write ，Do More”，即倡导写更少的代码，做更多的事情。它封装JavaScript常用的功能代码，提供一种简便的JavaScript设计模式，优化HTML文档操作、事件处理、动画设计和Ajax交互。

其是对JavaScript封装的一个框架包

简化对JavaScript的操作

JavaScript代码：dom操作，ajax操作、事件操作、事件对象

Jquery代码：无需烤炉浏览器兼容问题，代码非常少



## Jquery宗旨和特点

http://jquery.com



宗旨：写的代码很少，实现很多功能（write less，do more）

特点：

1. 语法简练、语义易懂、学习快速、丰富文档。
2. jQuery是一个轻量级脚本，其代码非常小巧
3. jQuery支持css1-css3定义的属性和选择器
4. jQuery是跨浏览器的，它支持的浏览器包括IE6.0+，主流浏览器
5. 能将JavaScript（行为）脚本与HTML（结构）源代码完全分离，便后期编辑和维护。
6. 插件丰富，除了jQuery自身一些特效外，可以通过插件实现更多功能

## 出现年代

jQuery是继prototype之后的一个优秀的JavaScript框架，有John Resig于2006年初创建，目前最新版本为1.11.3，其中文地址：<http://jquery.org.cn>

jquery版本:

1.x,x 兼容到IE6.0

2.x.x 对IE6/7支持不好

## 其他相关的JavaScript框架包

Prototype、YUI、Extjs、bindows、JSVM(国内)、mootools、jQuery

**Prototype**:与面向对象的原型继承关键字prototype—致，该框架的特点是功能扩展比较容易。

**YUI**: yahoo user interface雅虎用户接口，该框架可以实现各种页面布局效果。

**Extjs**:是目前is框架包里边最为时尚、前沿的。通过该框架包可以实现许多非常绚丽的效 果。

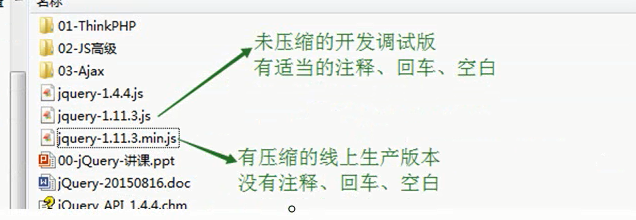
该框架可以实现效果之一：页面不同区域进行拖拽效果。

该框架由于实现的效果非常“绚丽”、导致其“实用”价值相对略低。

jQuery：JavaScript+jquery

使用前期，jquery侧重快速找到页面上各种节点。

后期jquery丰富了事件操作、ajax操作、动画操作、DOM操作



## 简单使用

# 二、选择器

在页面上获得各种元素节点对象而使用的条件就是选择器

Document.getElementBy··········——

Document.getElementBy··········——

Document.getElementBy··········——

## 基本选择器

$(‘ #id 属性值’) ------------> document.getELementById()

$(‘ tagName标签名’) ------------>document.getELementsByTagName()

$(‘ .class 属性值’) ------------>document.getELementByclass()

$(‘ \* ’) 通配符选择器

$(‘ s1,s2,s3 ’)联合选择器



其灵感来自于css选择器。

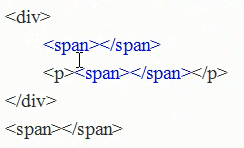
## 层次选择器

### $( s1 s1 ) [父子]

派生选择器：在s1内部获得全部的s2节点（不考虑层次）

Eg:

$(“div span”); //在div内部获得span节点



### $( s1 > s1 ) [父子]

直接子元素选择器：在s1内部获得子元素节点s2

Eg:

$(“div > span”); //在div内部获得子元素span节点

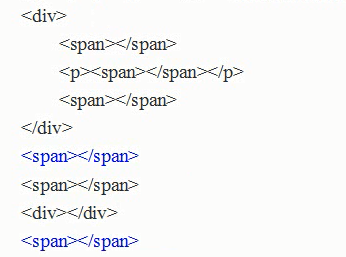


### $( s1 + s1 ) [兄弟]

直接兄弟选择器：在s1后边获得紧紧挨着的第一个兄弟关系的s2节点

Eg:

$(“div + span”); //获得div紧挨获得第一个子元素span节点，同一层级

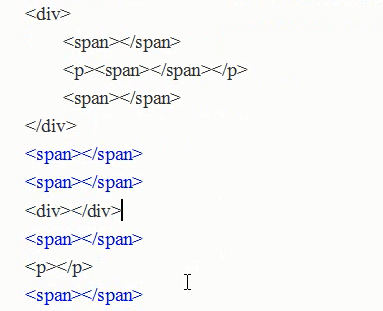


### $( s1 ~ s1 ) [兄弟]

后续全部兄弟关系节点选择器：在s1后面获得全部兄弟关系的s2节点

Eg:

$(“div + span”); //获得div获得所有子元素span节点，同一层级



## 并且（过滤）选择器

### 基本使用

#### : not(selector选择器)

:not(selector选择器) 去除某个节点



#### :first和 :last

:first 第一个 :last最后一个  


#### :eq(下标索引号码)

:eq(下标索引号码) equal等于意思，获得索引值下标引值的节点



#### :gt大于 和: lt小于

:gt(下标索引号码) great than节点索引值 大于 某个范围

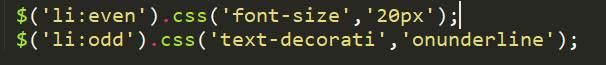
:lt(下标索引号码) less than节点索引值 小于 某个范围



#### :even偶数和 :odd奇数

:even 匹配到下标索引值为偶数的节点

:odd 匹配到下标索引值为奇数的节点

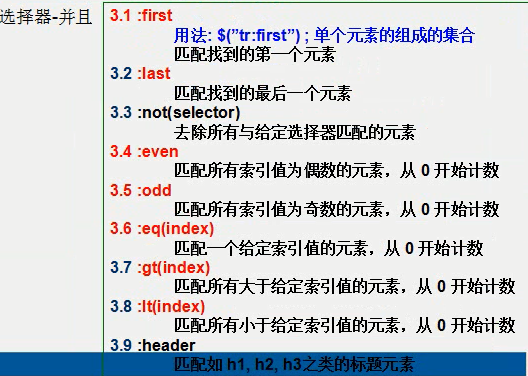


### 复杂使用

注意：

1. 并且选择器可以单独使用
2. 各种选择器都可以构成并且关系
3. 并且关系的选择可以使用多个，每个选择器使用前，已经获得节点的下标要“归位”（归零）
4. 并且关系的多个选择器，没有前后顺序关系，但是要避免"歧义",最好使用顺关系



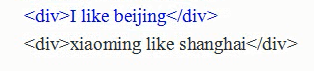


## 内容过滤选择器

### :contains（内容）

包含内容选择器，获得的节点内必须包含指定的内容,内部标签不构成影响

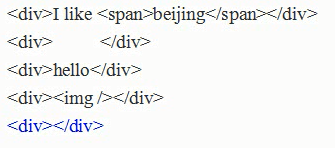
$(‘ div : contains(beijing) ’)



### :empty（空选择器）

获得空元素（内部没有任何元素/文本（空））节点对象

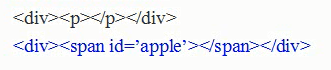
$(“ div:empty”)



### :has（存在选择器）

节点内部必须包含指定选择器对应的元素

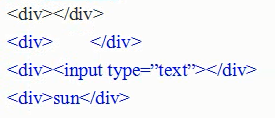
$(‘ div:has(#apple) ’)



### :parent（选择器）

寻找的节点必须作为父元素节点存在

$(“ div:parent ”)



## 表单域选中选择器

复选框、单选按钮、下拉列表

$(‘ :checked ’)复选框、单选按钮、选中选择器

$(“ :selected ”) 下拉列表 选中选择器

# 四、属性操作

<input type="checkbox" class="hby" name="hobby[]" value="混球">

**传统：**

Itnode.属性名

Itnode.属性名 = 值，

Itnode.getAttrubute(属性名称)

Itnode.setAttrubute(属性名称，值)

**Jquery方式：**

$().attr(属性名) //获得属性信息值

$().attr(属性名，值) //设置属性信息

$().removeattr(属性名) //删除属性

$().attr(json对象) //同时为多个属性设置信息值，json对象的键值对就是名称和值（批量操作）

$().attr(属性名，fn) //通过fn函数执行的return返回值对属性进行赋值

# 五、快捷操作

## class属性值操作

注意: class 属性值可以多个。

$().attr(‘class’,值)；

$().attr(‘class’)；

$().removeAttr(‘class属性’)； //删除class属性

Class具体快捷操作方法：

$().addClass(‘属性值’)； //给class属性追加信息值

$().removeClass(‘属性值’)； //删除class属性中某个信息值

$().toggleClass(‘属性值’)； //开关效果，有的删除，没有就添加

## 标签包含内容操作

<div><span>我们</span>正在学习jquery</div>

JavaScript传统操作：

dvnode.innerHTML; 获得div包含的信息

dvnode.innerHTML =XXX; 设置div包含的内容信息

（innerHTML不是w3c标准计算，许多浏览器对其支持而已）

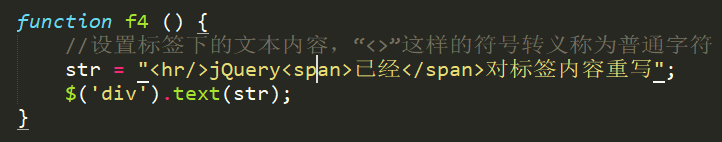
jQuery操作：

$().html( ); //获得节点包含的信息

$().html(信息); //设置节点包含的信息

$().text( ); //获得节点包含的“文本字符串信息”内容

$().textl(信息 ); //设置节点包含的内容（有html标签就把“><”符号转义符号实体）





### html()和text()方法的区别

1. 获取内容  
   前者可以获取html标签和普通字符串内容  
   后者只获取普通字符串内容（会自动过滤html标签）
2. 设置内容  
   前者可以设置html标签和普通字符串内容  
   后者指设置为普通字符串内容，如果内容有tag标签内容，就把其中的”<”, ”>”符号转变为符号实体（也就是转义的意思）：< ---- &lt; ，> ----- &gt; ，空格 -----&nbsp;

以上两种操作（获取/设置）如果针对的操作内容是纯字符串内容，则使用效果一致

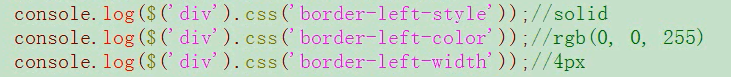
## CSS样式操作

$().css(name , value); //设置css样式

$().css(name ); //获取css样式

$().css(json对象); //批量设置css样式

注意：

1. jQuery不能操作复合样式，因此需要拆分进行操作  
   Eg: border : 4px solid red；  
    

### CSS()样式操作的特点

**获取内容：**

Jquery可以获取 行内、内部、外部样式

DOM只能获取行内样式。

**获取复合属性样式**

需要拆分为“具体样式”才能操作（有的浏览器是可以获得复合样式信息的，例如谷歌）



**样式设置：**

样式的设置，会被设置为“行内样式”

有则修改，无则添加（复合样式可以直接进行设置操作）

## value属性快捷操作

$( ).attr(‘ value’); //获取value属性的值

$( ).attr(‘ value’，信息值); //设置value属性的值

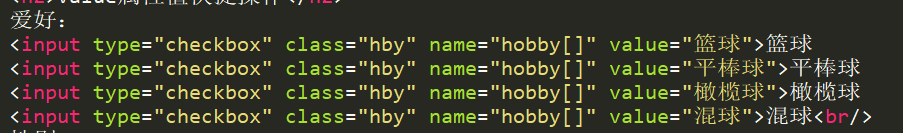
jquery快捷操作：

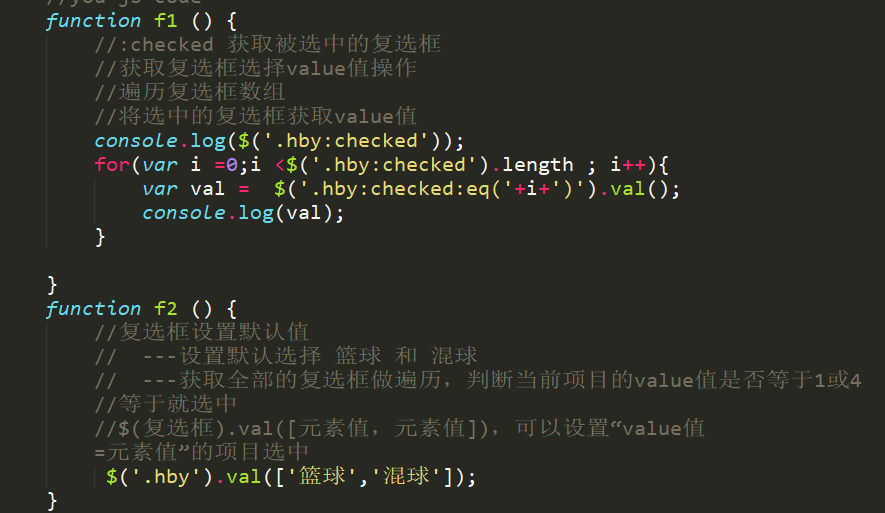
$( ).val( )； //获取value属性的值

$( ).val(信息值)； //设置value属性的值  
该val（）方法在复选框、单选框、下拉列表的使用有突出表现

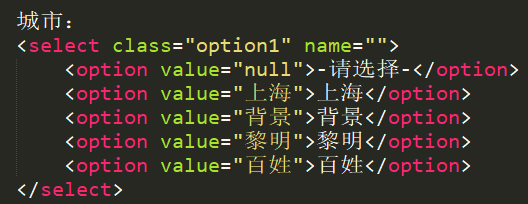
### 复选框value操作

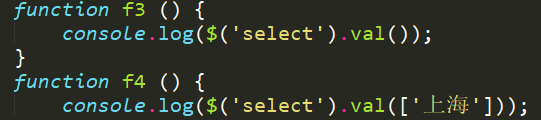
1. 获得被选中复选框的value属性值
2. 设置默认情况下哪个复选框被选中  
   $(复选框).val([元素值，元素值])，可以设置“value值



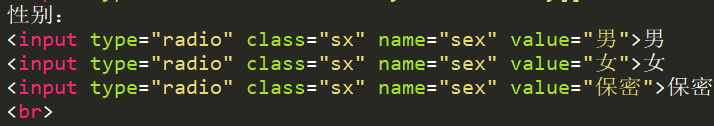


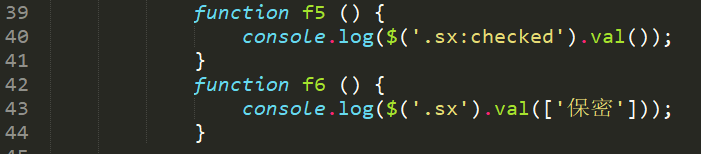
### 下拉列表value操作





### 单选value操作





## 复选框快捷操作

全选、反选、全不选

$().attr( ‘checked’,true ); //设置复选框选中

$().attr( ‘checked’,false ); //取消复选框选中

$().attr( ‘checked’ ); //获取复选框选中情况，返回boolean值

注意：Jquery1.6版本后，$().attr( ‘checked’ );不再返回boolean类型！而是“undefined”和“checked”

所以将一下方法改为如下：

$().prop( ‘checked’,true ); //设置复选框选中

$().prop( ‘checked’,false ); //取消复选框选中

$().prop( ‘checked’ ); //获取复选框选中情况，返回boolean值

网上查询资料如下：

什么时候使用attr()，什么时候使用prop()？

* 添加属性名称该属性就会生效应该使用prop();
* 是有true,false两个属性使用prop();
* 其他则使用attr();

官方建议：



# 六、$符号的由来

$( ‘ div ’ )、$( ‘ .apple ’ )、$( ‘\*’ )等等。

选择器使用的过程就是函数调用过程。

$符号就是一个函数，函数名为“ $ ”符号而已。也可以使用“jQuery” 符号。

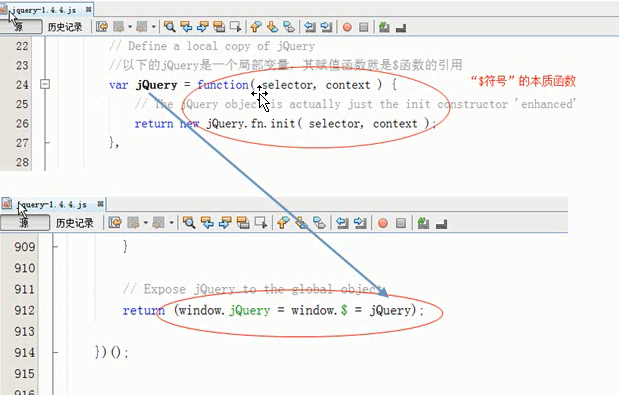


# 七、jQuery对象与DOM对象的关系

Jquery对象：$( ‘ li ’ ) $( ‘ .apple ’ )等选择器使用返回的对象信息就是jquery对象

dom对象： document.getElementById( )

document.getElementByTagName( )



$ 符号就是一个函数，同时其也是一个全局变量除此之外还可以使用jQuery符号，它们互为别名。

## 互调用对方成员

Jquery对象 和 dom对象不能相互调用对方成员。

以下方式都是会报错的：

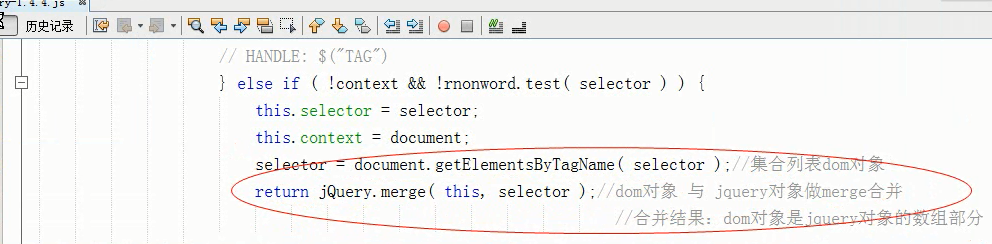


## jQuery对象封装DOM对象

DOM对象就是jQuery对象的数组组成部分：

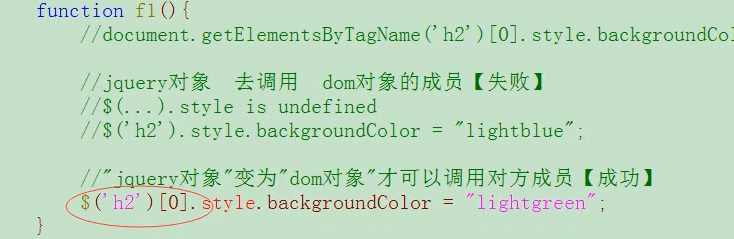


jQuery对象对document.getElementsById()等方法进行封装：



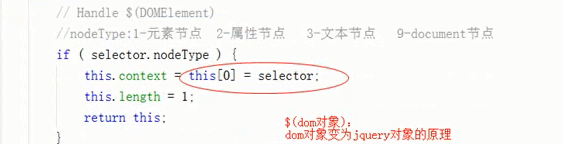
## jQuery对象和DOM对象的相互转化

### jquery对象 —》DOM对象

**$( )[下标]**

### DOM对象 —》jquery对象





dom对象变为jquery对象才能使用jquery方法

# 八、jQuery框架对象分析

Jquery框架对象类型：jquery对象 和 $对象。

1. Jquery对象（普通对象）：就是各种选择器创建出的来的对象。 $(‘ div ’)
2. $对象就是“函数对象” $.get( )

## jQuery对象



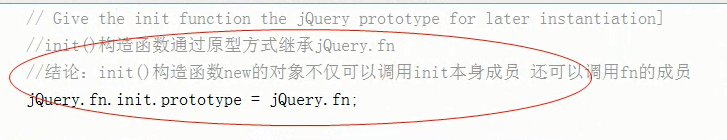
以上jquery对象，可以调用许多成员 css()、attr()、html()等等

问题：

Jquery对象，可以调用的各种成员是从哪来的！



jQuery.fn本身有extend复制继承，可以自己丰富很多成员



普通的jquery对象调用成员的90%以上都是jquery.fn.extend复制继承过来的。

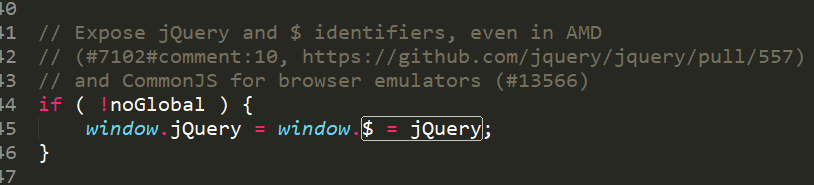
## $ 对象

$对象---->函数对象

该$对象可以调用的成员都是extend复制继承过来的。

$.get() $.post() $.ajax()

$对象也是通过extend复制继承，来丰富成员的。



# 九、遍历方法

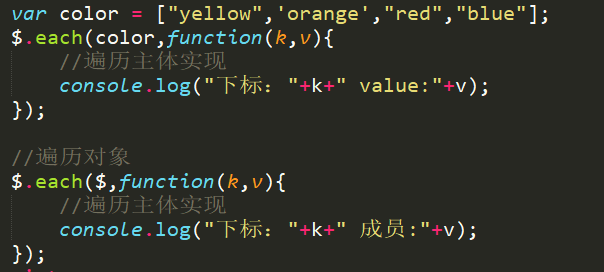
**each( )遍历方法：**

$.each(数组/对象，funcation处理)； //$对象 调用的

$(选择器).each(funcation处理)； //jquery对象 调用的

**$().each(funcation ( k , v ){ })；**  k是dom对象下标，v 是dom对象下标对应的值

each()方法遍历如下：



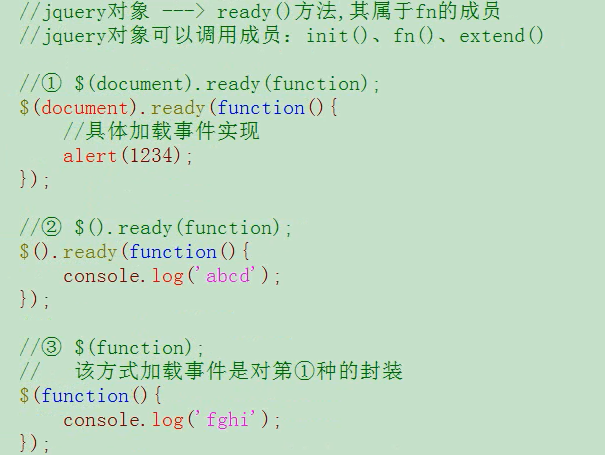
# 十、加载事件

<body onload="函数()">

window.onload = function () {}

## Jquery加载事件实现

1. $(document).ready( funcation处理 )；  
   $(document)是把document对象变为jquery对象
2. $().ready( funcation处理 )；  
   $()也是创建jquery对象，不过内部没有dom对象的组成部分
3. $( funcation处理 )；  
   对**第一种加载**的封装而已



## Jquery加载事件与传统加载事件区别

### 设置个数

在同一个请求里面，jquery加载事件的课一设置多个，而传统方式只能加载一个，

传统方式加载事件是给onlaod事件属性复制，多次赋值后，后者会覆盖前者。

Jquery方式加载事件是把每一个加载事件都存入一个数组里面成为数组的元素，执行的时候遍历该数组执行每个元素即可，因此其可以设置多个加载事件。

### 执行时机不一样

传统方式加载事件，是全部内容（文字、图片、样式）在浏览器显示完毕再给执行加载事件。

广告图片小叉隐藏图片实现（在加载事件里边给图片的小叉设置oneIick事件）

用户名输入框有点击隐藏灰色的文字(在加载事件里边给输入框设置oneIick事件，隐藏灰色文字）

Jquery方式加载事件，只要全部内容（文字、图片、样式）在内存里面对应的DOM树结构绘制完毕就给执行，有可能对应的内容在浏览器里面还没有显示。

Eg：广告图片、用户名输入框（提示）

得知：jquery加载事件时机 快于 传统方式加载事件。

### jQuery加载事件的原理

jQuery并不是对onload事件进行封装，而是对DOMContentLoaded的封装。



# 十一、普通（简单）事件操作

① dom1级事件设置

<input type=” text” oneIick=” 过程性代码” value=’ tom’/>

<input type=” text” oneIick=” 函数 0 ” />

itnode. onclick = function ( ) {}

itnode. onclickf 函数；

1. dom2级事件设置1

itnode. addEventListener (类型，处理，事件流）；

itnode. removeEventListener (类型，处理，事件流）；

node. attachEvent( ) ;

node. detachEvent( ) ;

1. jquery事件设置

$( ).事件类型（funcation处理函数）； //设置事件

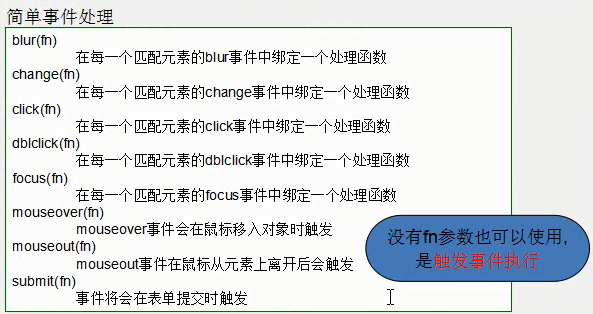
$( ).事件类型（）； //触发事件执行

注意：事件类型：click、keyup、keydown、mouseover、mouseout、blur、focus 等等

例如：

$ (form). |sUbmit 0可以使得表单进行提交。

$( ‘div’）. click (function 0{事件触发过程 this});



# 十二、jQuery对网页文档的操作

通过jQuery方式实现网页各种节点的追加、删除、复制、替换等操作

## 节点追加

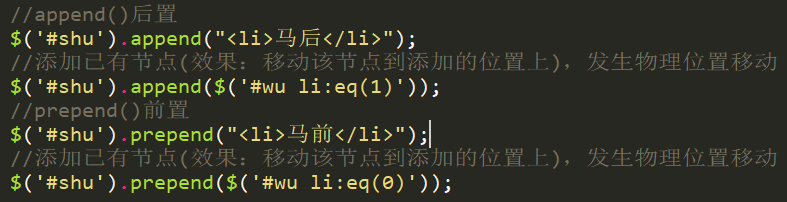
### 父子关系追加



**$('itnode').append( 节点html信息 )；** //后置

**$('itnode').prepend( 节点html信息 )；** //前置

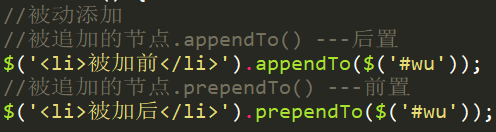
Eg：



**$( 节点html信息 ). appendTo($( ‘ itnode ’ ))；** //后置

**$( 节点html信息 ). prependTo($( ‘ itnode ’ ))；** //前置

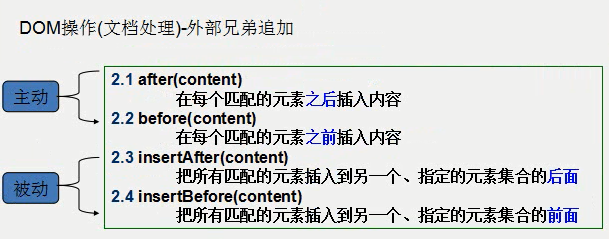
Eg：



、注意：

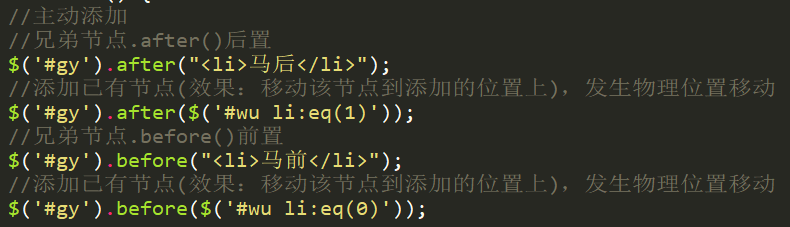
追加已有的节点，就是将该节点移动到追加的位置上。

### 兄弟关系追加



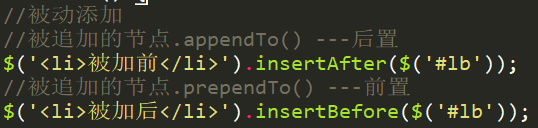
**$('itnode').after( 节点html信息 )；** //后置

**$('itnode').before( 节点html信息 )；** //前置



**$( 节点html信息 ). insertAfter ($( ‘ itnode ’ ))；** //后置

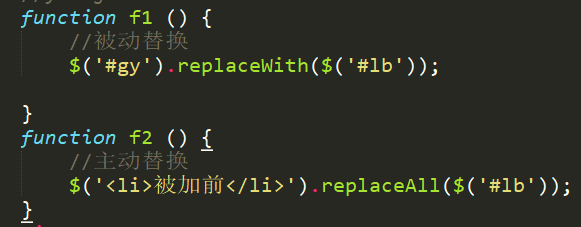
**$( 节点html信息 ). insertBefore ($( ‘ itnode ’ ))；** //前置



## 节点替换

**$( ) . replaceWith( )；** //被动替换

**$( ) . replaceAll( )；** //主动替换



## 节点删除

**$( 父节点 ) . empty( )；** //父节点清空所有子节点，

**$( 匹配节点 ).remove( )；** //删除指定的节点

## 节点复制

**$( ) . clone( true )；** //节点和其身上的事件都给复制

**$( ) . clone( false )；** //只复制节点本身（包括节点内部信息）

# 十三、属性选择器



* [attribute]用法:**$(”div[id]“); （常用）**

匹配包含给定属性的元素

* [attribute=value]用法：**$(‘ input[name = newsletter] ‘); （常用）**

匹的属性是某个特定值的^

* [attribute!=value]用法：**$(”input[name != ’newsletter’ ] ”);**

匹配給定的属性是不包含某个特定值的元素

* [att0buteA=value]用法：**$(”input[name ^= ‘news’ ]“);**

匹配給定的属性是些值开始的元素

* [attribute$=value]用法：**$(”input[name $= ‘letter’ ]“);**

匹配給定的属性是以某些值结尾的元素

* [attribute\*=value]用法：**$(”input[name \*= ‘man’ ]“);**

匹配給定的属性是以包含某些值的元素

[selector1][selector2][selectorN]

* **$(“ input[ id ] [ name$=‘man’ ]“);**

并且选择器，需要同时满足多个条件时使用



# 十四、事件绑定

## 事件绑定

$( ).事件类型（funcation处理函数）； //设置事件

$( ).事件类型（）； //触发事件执行

### Jquery事件绑定

$( ).bind(事件类型，funcation事件处理 )； //绑定一个事件

$( ).bind(类型1 类型2 类型3 ，事件处理 )； //给许多不同类型的事件绑定同一个处理

$() .bind( json对象 ); //同时绑定多个不同类型的事件



（事件处理：click、mouseover、mouseout、focus、blur等等）

事件处理：有名函数、匿名函数

### 取消事件绑定

传统事件事件取消：

node.onclick = null ;

node.removeEventListener( 类型，（有名函数）事件处理 ，事件流)；

node.detachEvent( 类型， ( 有名 )处理);

jquery方式取消事件绑定：

1. $( ).unbind(); //取消所有事件绑定
2. $( ).unbind( 事件类型 ); //取消指定的事件类型的全部绑定
3. $( ).unbind( 事件类型 ，(有名)处理函数); 取消指定类型的指定处理事件绑定

注意：第③中取消事件绑定，处理事件必须是有名函数

## 事件对象、阻止浏览器默认动作、阻止事件冒泡

$( ).bind( ‘ click ’ , function( event ){ })

$( ).click( function( event ){ } )

$( ).bind( ‘mouseover’, f1);

Function f1 ( event ){ }

事件对象：就使用红色的event即可，在jQuery框架内部有做浏览器兼容处理

以上红色的event对主流的事件对象 和 IE的事件对象有封装

阻止浏览器默认动作、阻止事件冒泡：

DOM2级浏览器默认动作阻止：

事件对象.preventDefault( ); //主流浏览器

事件对象.returnValue = false ; //ie浏览器

DOM2级事件冒泡阻止：

事件对象.stopPropagation( ); //主流浏览器

事件对象.cancelBubule = false ; //ie浏览器

在jQuery里：

$( ).bind( ‘ click’ , function( evt ){

evt.preventDefault(); //阻止事件默认动作

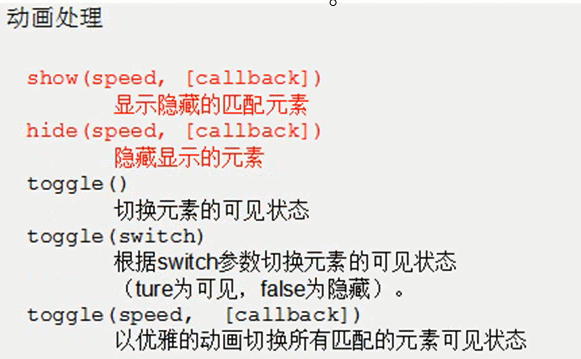
evt.stopPropagation(); //冒泡阻止

});

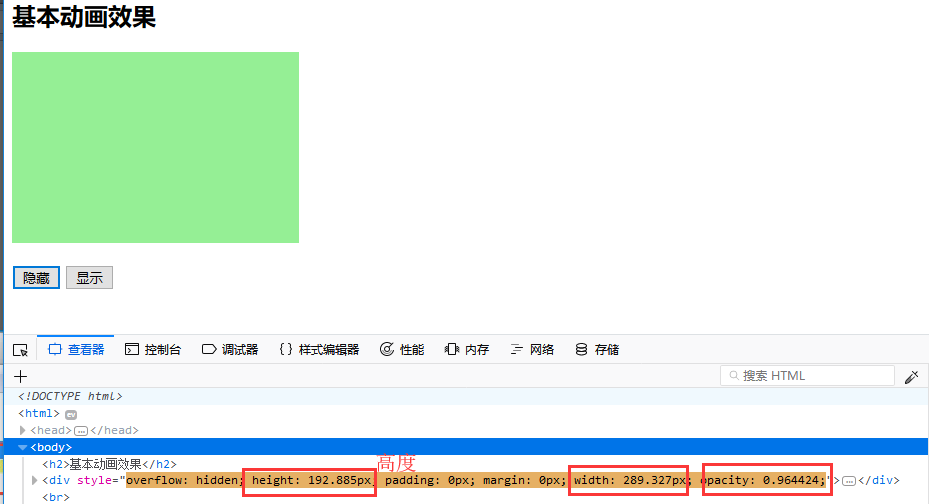
preventDefault( )方法是jQuery的方法，名字与js底层代码的名字一致而已。并具浏览器兼容。

stopPropagation( )方法是jQuery的方法，名字与js底层代码的名字一致而已。并具浏览器兼容。

# 十五、动画效果



## 基本动画



具体实现：通过修改style样式的高度、宽度、透明度。

动画处理：show ( speed , [callback] )

显示隐\_的匹配元素：hide ( speed , [callback] )

隐藏显示的元素·：toggle ( )

切换元素的可见状态：toggle ( switch )

根据switch#数切换元素的可见状态 (ture为可见，false为隐藏）。

toggle(speed, [callback])

以优雅的动画切换所有匹配的元素可见状态

## 垂直动画

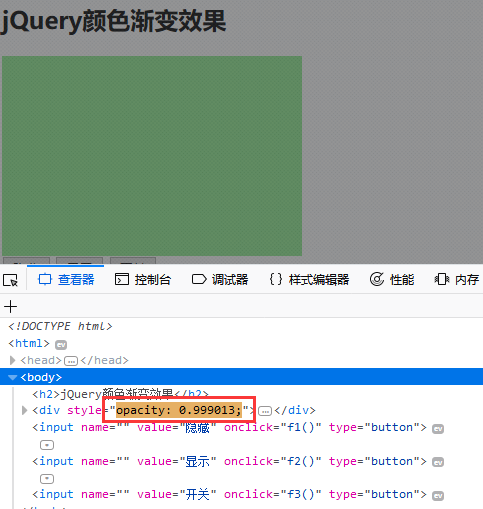


具体实现：通过修改style样式的高度。

$().slideUp([速度毫秒，回调函数])

$().slideDown([速度，回调函数])

## 颜色渐变动画

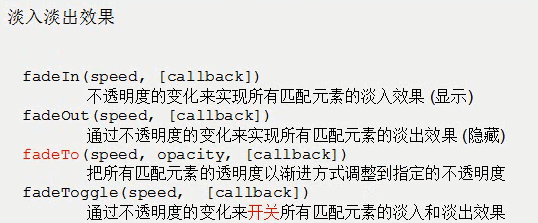


具体实现：通过修改opacity透明度。

隐藏：$().fadeOut([速度，回调函数])

显示：$().fadeIn([速度，回调函数])

开关效果：$().fadeToggle([速度，回调函数])



# 十六、封装ajax使用

具体操作：

1. $.get (url [,date] [,function(msg){}回调函数] [,dataType]);

data: 给服务器传递的数据，请求字符串、json对象 都可以设置

function(msg){}回调函数：ajax请求完成后调用该函数，可以在此函数完成ajax的后续处理，msg泛指从服务器传递回来的信息。

dataType：服务器返回数据类型，html、text、xml、json

1. $.post(url [,date] [,function(msg){}回调函数] [,dataType]);

该方法与$.get()方法使用完全一致，不同的是其为post请求方式

给服务器传递数据的时候，不需要设置header头

（以上两种ajax请求是异步的，如果需要设置同步请求，就换其他方法）

1. $.ajax（{ //json对象

url：请求地址，

[data]：给服务器传递的数据（请求字符串json对象）

[dataType]：默认字符串返回信息，数据从服务器返回格式html、text、xml、json

[type]：get/post请求方式

[success]：function（msg）{ } ajax请求成功后的回调函数，可以做后续处理使用

msg泛指服务器返回的信息

async：true异步/false同步

cache：true缓存/false不缓存

}）

# 十七、案例-地区三级联动

技术点：jquery+ajax+xml/json

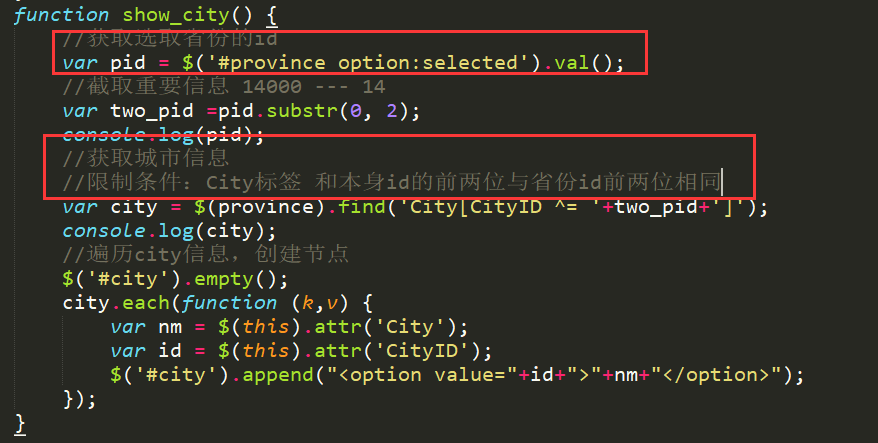
## 省份的获取和显示

通过jquery+ajax获取xml信息从中解析出数据



## 显示对应城市信息

根据获取的省份信息，获取城市信息

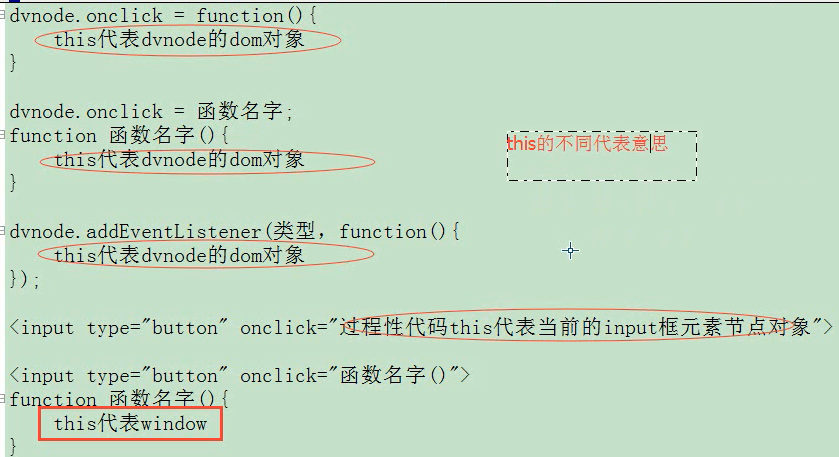
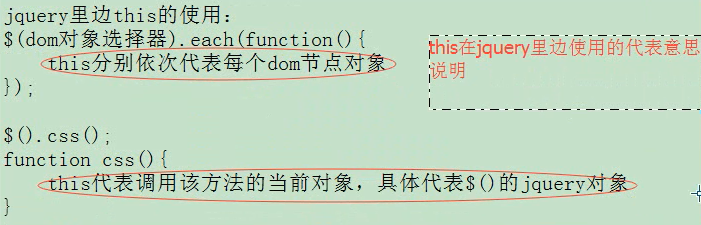


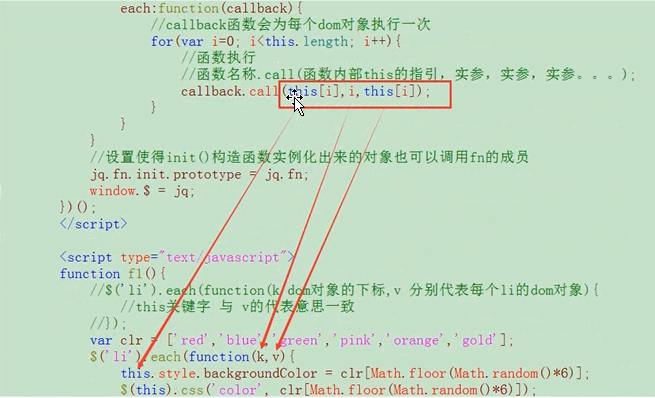
# 十八、迷你版jQuery框架开发

选择器：#id、tag标签 、dom节点选择器

方法：css()、attr()、each()方法

学习“迷你版jQuery”可以认识到的地方：

1. jQuery里边的大部分方法有“遍历机制”
2. jQuery方法里面的参数this代表dom对象  
     
   
3. 函数的封装



# 十九、jQuery插件

## 什么是jQuery插件

jQuery框架本身给我提供了一些方法提供使用，但是方法的数目是有限的，其不能任意满足我们队各种功能的需求。那么我们自己可以来给jQuery框架开发、扩展一些额外功能方法。

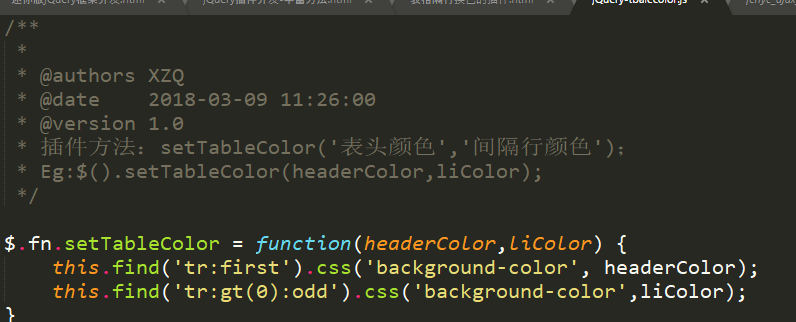
**给jQuery框架开发、扩展一些额外方法的过程“插件开发”**

## 两种形式丰富方法

1. 给jQuery.fn丰富成员（该成员可以给jquery对象使用）  
   $.fn.成员 = 值；  
   $.fn.extend（json对象）；
2. 给 $对象丰富成员（该成员可以给$对象使用）  
   $.成员 = 值；  
   $.extend(json对象)；

## 制作一个表格隔行换色的插件

在项目开发中，如果有一些功能大家都需要使用，就可以开发出一个插件，使得每个人只要调用一个方法就可以实现。



## 使用“成品jQuery插件”的步骤

1. 引入jquery的插件文件
2. 引入对应的css样式、img图片、辅助的相关js文件
3. 检查jquery插件对本身jquery是否有要求