目录

[一. jQuery 3](#_Toc439028106)

[1. 什么是jquery 3](#_Toc439028107)

[2. 宗旨和特点 3](#_Toc439028108)

[3. 出现的年代 3](#_Toc439028109)

[4. 其他相关的javascript框架包 3](#_Toc439028110)

[5. 简单使用 5](#_Toc439028111)

[二. 选择器 6](#_Toc439028112)

[1. 基本选择器 6](#_Toc439028113)

[2. 层次选择器 7](#_Toc439028114)

[2.1 $(s1 s2) [父子] 7](#_Toc439028115)

[2.2 $(s1 > s2) [父子] 7](#_Toc439028116)

[2.3 $(s1 + s2) [兄弟] 8](#_Toc439028117)

[2.4 $(s1 ~ s2) [兄弟] 8](#_Toc439028118)

[3. 并且(过滤)选择器 10](#_Toc439028119)

[4. 内容过滤选择器 11](#_Toc439028120)

[4.1 :contains(内容) 11](#_Toc439028121)

[4.2 :empty 12](#_Toc439028122)

[4.3 :has(选择器) 12](#_Toc439028123)

[4.4 :parent 12](#_Toc439028124)

[5. 表单域选中选择器 13](#_Toc439028125)

[三. 属性操作 14](#_Toc439028126)

[四. 快捷操作 16](#_Toc439028127)

[1. class属性值操作 16](#_Toc439028128)

[2. 标签包含内容操作 18](#_Toc439028129)

[2.1 html() 和 text()方法的区别： 18](#_Toc439028130)

[3. css样式操作 19](#_Toc439028131)

[3.1 css()样式操作特点： 19](#_Toc439028132)

[4. value属性快捷操作 20](#_Toc439028133)

[4.1 复选框操作 21](#_Toc439028134)

[4.2 下拉列表操作 22](#_Toc439028135)

[4.3 单选按钮操作 23](#_Toc439028136)

[5. 复选框操作 23](#_Toc439028137)

[五．$符号的由来 25](#_Toc439028138)

[六. jquery对象 与 dom对象关系 26](#_Toc439028139)

[1. jquery对象 和 dom对象 互相调用对方的成员 26](#_Toc439028140)

[2. jquery对象 如何封装的dom对象 26](#_Toc439028141)

[3. jquery对象 和 dom对象 的转化 27](#_Toc439028142)

[3.1 jquery对象-----》dom对象： $()[下标] 27](#_Toc439028143)

[3.2 dom对象-----》jquery对象：$(dom对象) 28](#_Toc439028144)

[七. jquery框架对象分析 29](#_Toc439028145)

[1. jquery对象 29](#_Toc439028146)

[2. $函数对象 30](#_Toc439028147)

[八. 遍历方法 30](#_Toc439028148)

[九. 加载事件 32](#_Toc439028149)

[1. jquery加载事件实现 32](#_Toc439028150)

[2. jquery加载事件与传统加载事件的区别 32](#_Toc439028151)

[2.1 设置个数 32](#_Toc439028152)

[2.2 执行时机不一样 33](#_Toc439028153)

[3. jquery加载事件原理 34](#_Toc439028154)

[十. 普通事件操作 34](#_Toc439028155)

[十一. jquery对文档的操作 35](#_Toc439028156)

[1. 节点追加 36](#_Toc439028157)

[1.1 父子关系追加 36](#_Toc439028158)

[1.2 兄弟关系追加 38](#_Toc439028159)

[2. 节点替换 38](#_Toc439028160)

[3. 节点删除 39](#_Toc439028161)

[4. 复制节点 40](#_Toc439028162)

[十二.属性选择器使用 42](#_Toc439028163)

[十三.事件绑定 44](#_Toc439028164)

[1. 事件绑定 44](#_Toc439028165)

[1.1 jquery事件绑定 44](#_Toc439028166)

[1.2 取消事件绑定 46](#_Toc439028167)

[2. 事件委派 47](#_Toc439028168)

[3. 事件对象、阻止浏览器默认动作、阻止事件冒泡 48](#_Toc439028169)

[十四. 动画效果 49](#_Toc439028170)

[1. 基本动画 49](#_Toc439028171)

[2. 垂直动画 50](#_Toc439028172)

[3. 颜色渐变动画 51](#_Toc439028173)

[十五. jquery封装的ajax 52](#_Toc439028174)

[十六. 地区三级联动 54](#_Toc439028175)

[1. 省份的获取和显示 54](#_Toc439028176)

[2. 显示对应城市信息 55](#_Toc439028177)

[十七. 迷你版jquery开发 56](#_Toc439028178)

[1. 制作选择器 57](#_Toc439028179)

[2. 各种方法的封装 57](#_Toc439028180)

[十八. jquery插件开发 58](#_Toc439028181)

[1. 什么是jquery插件 58](#_Toc439028182)

[2. 两种形式丰富方法 59](#_Toc439028183)

[3. 开发图片自动切换插件 59](#_Toc439028184)

[4. 使用”成品jquery插件”的步骤 60](#_Toc439028185)

# 一. jQuery

## 1. 什么是jquery

其是对javascript封装的一个框架包

简化对javascript的操作

javascript代码：获得页面节点对象、ajax元素节点对象实现、事件操作、事件对象

jquery代码：无需考虑浏览器兼容问题、代码足够少

## 2. 宗旨和特点



特点：

1. 语法简练、语义易懂、学习快速、丰富文档。
2. jQuery 是一个轻量级的脚本，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 IE 6.0+、FF 1.5+、Safari 2.0+和 Opera 9.0+。
5. 能将 JavaScript 脚本与 HTML 源代码完全分离，便于后期编辑和维护。
6. 插件丰富，除了 jQuery 自身带有的一些特效外，可以通过插件实现更多功能

## 3. 出现的年代

jQuery 是继 Prototype 之后又一个优秀的 JavaScript 框架，由 John Resig 于 2006 年初创建， 目前最新版本为 1.11.2，官方地址为：http://jquery.com/，中文地址为 http://jquery.org.cn/。

## 4. 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一致，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。

例如效果之一：标签切换对应内容



Extjs: 是目前js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。

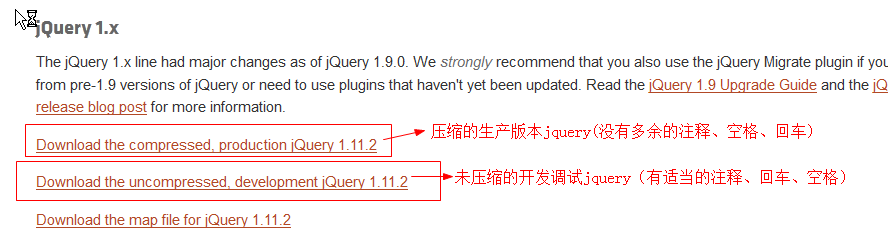
jQuery：javascript+query

使用前期，jquery侧重快速找到页面上各种节点。

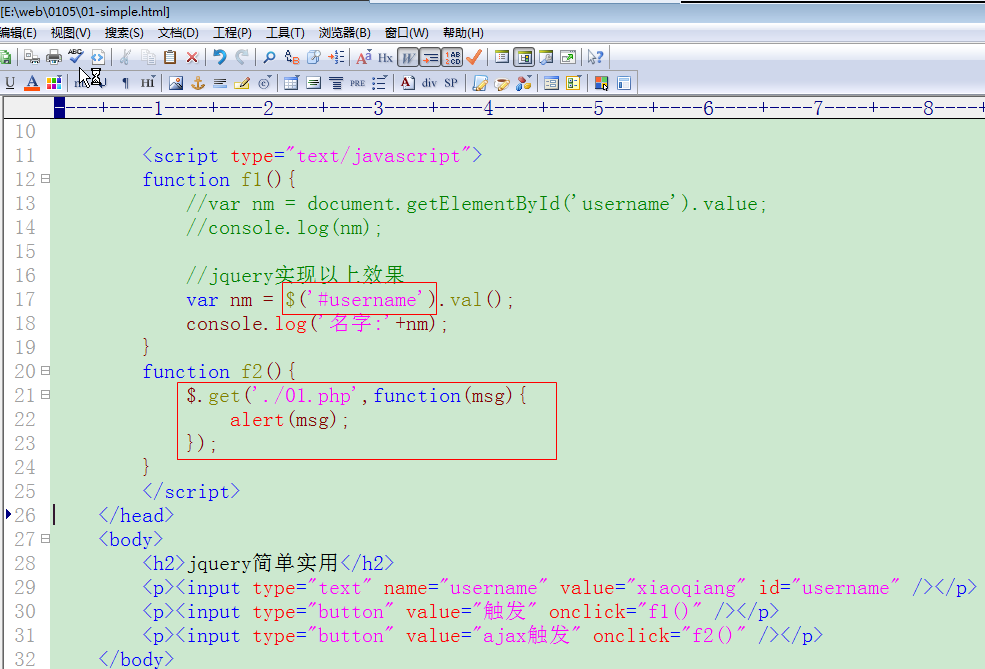
后期jquery丰富了事件操作、ajax操作、动画效果、DOM操作等等。

jquery的版本：

1.0.0 1.0.1 1.0.2 1.1.x 1.2.1 1.2.x 1.3.x 1.4.x 1.5.x.........1.11.2



## 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.getElementById()

document.getElementsByTagName();

document.getElementsByName();

## 基本选择器

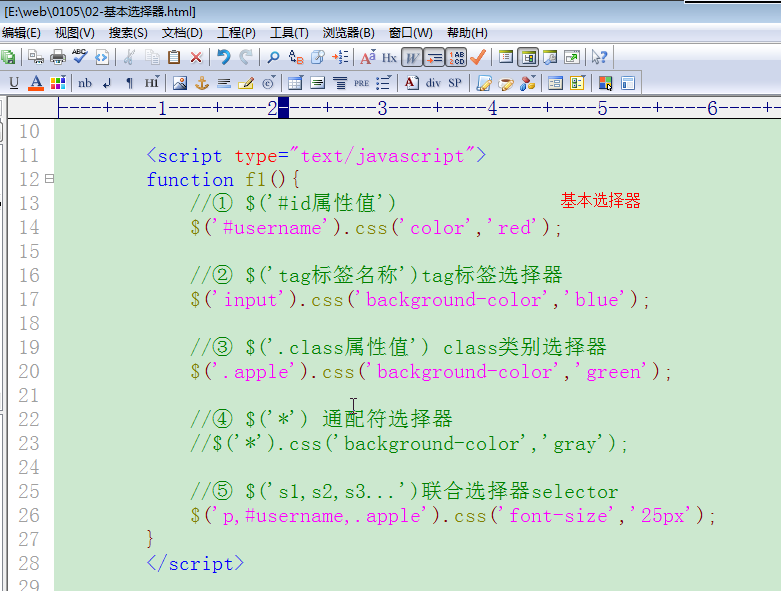
$(‘#id属性值’) ----------->document.getElementById()

$(‘tag标签名称’)----------->document.getElementsByTagName();

$(‘.class属性值’) class属性值选择器

$(‘\*’) 通配符选择器

$(‘s1,s2,s3’)联合选择器





## 2. 层次选择器

### 2.1 $(s1 s2) [父子]

派生选择器：在s1内部获得全部的s2节点(不考虑层次)

$(“div span”)

<div>

<span></span>

<p>

<span></span>

</p>

</div>

<span></span>

### 2.2 $(s1 > s2) [父子]

直接子元素选择器:在s1内部获得s2的子元素节点

$(“div > span”)

<div>

<span></span>

<p>

<span></span>

</p>

<span></span>

</div>

<span></span>

### 2.3 $(s1 + s2) [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

$(“div + span”)

<div>

<span></span>

<p>

<span></span>

</p>

<span></span>

</div>

<span></span>

<span></span>

<div></div>

<span></span>

### 2.4 $(s1 ~ s2) [兄弟]

后续全部兄弟关系节点选择器：在s1后边获得全部兄弟关系的s2节点

$(“div ~ span”)

<div>

<span></span>

<p>

<span></span>

</p>

<span></span>

</div>

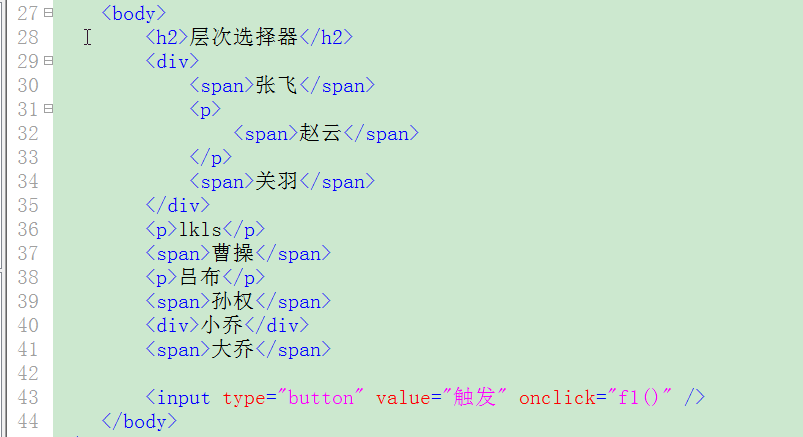
<span></span>

<span></span>

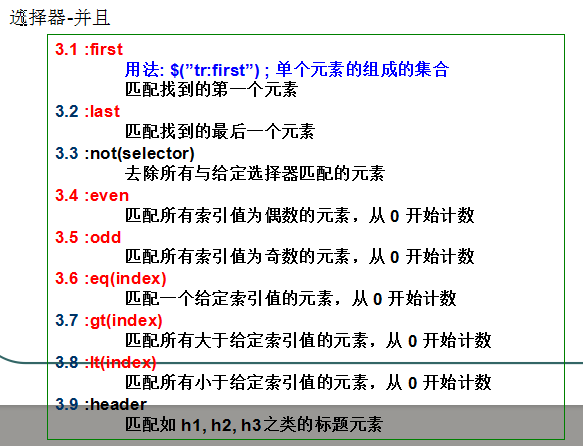
<p></p>

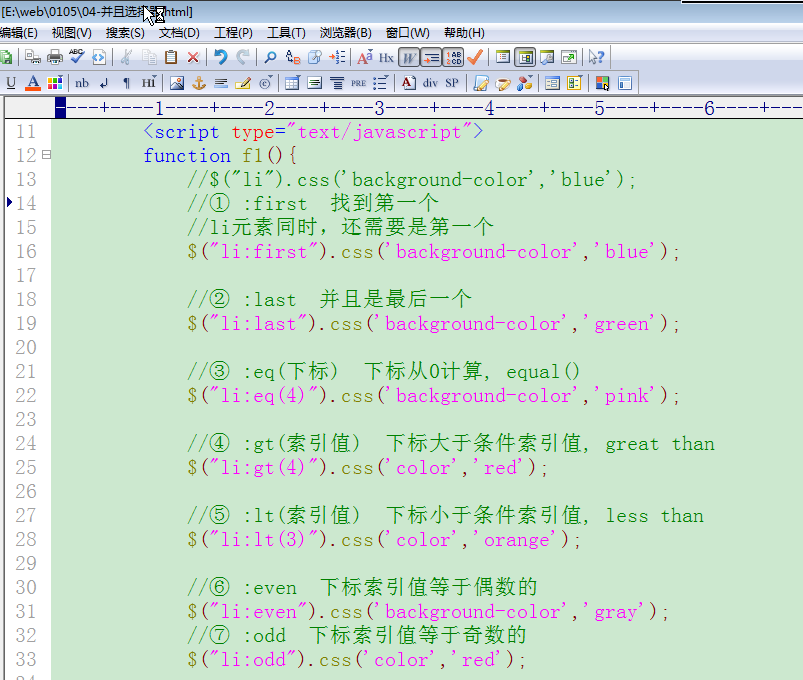
<span></span>

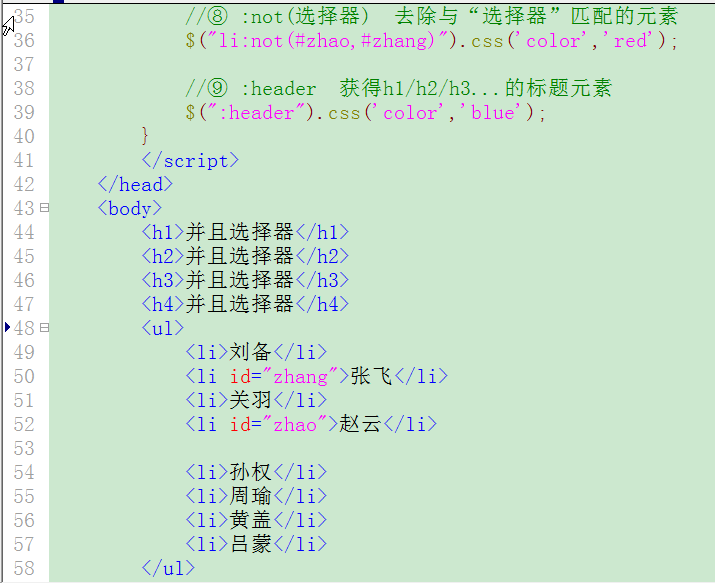


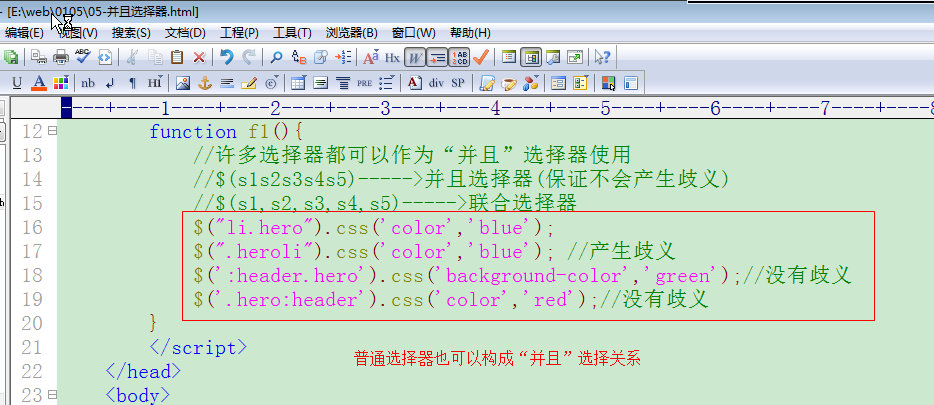


## 并且(过滤)选择器









## 4. 内容过滤选择器

### 4.1 :contains(内容)

包含内容选择器，获得节点内部必须通过标签包含指定的内容

$(“div:contains(beijing)”)

<div>linken love beijing</div>

<div>jack love shanghai</div>

### 4.2 :empty

获得空元素(内部没有任何元素/文本(空) )节点对象

$(“div:empty”)

<div>linken love beijing</div>

<div>jack love shanghai</div>

<div></div>

<div><img /></div>

<div> </div>

### 4.3 :has(选择器)

内部包含指定元素的选择器  
 $(“div:has(#apple)”)

<div>hello</div>

<div><p></p></div>

<div><span id=”apple”></span></div>

<div><span class=”apple”></span></div>

### 4.4 :parent

寻找的节点必须作为父元素节点存在

$(“div:parent”)

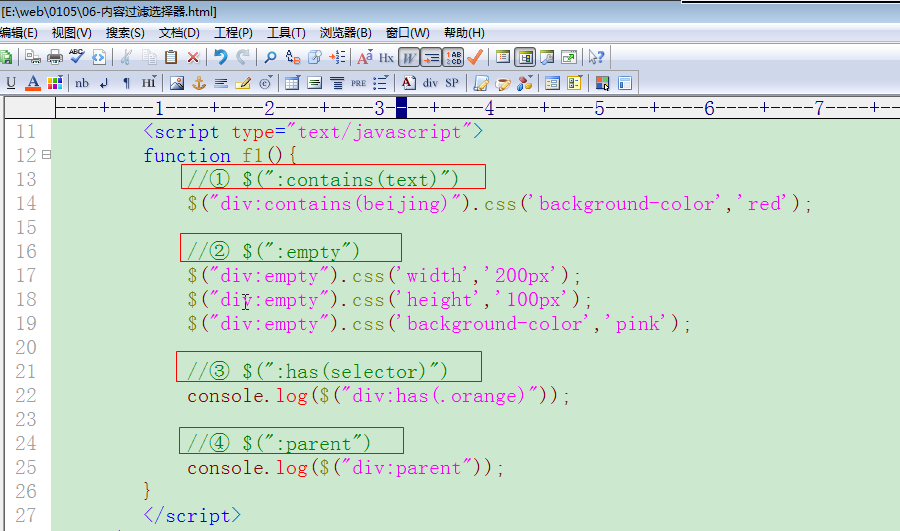
<div>linken love beijing</div>

<div>jack love shanghai</div>

<div></div>

<div><img /></div>

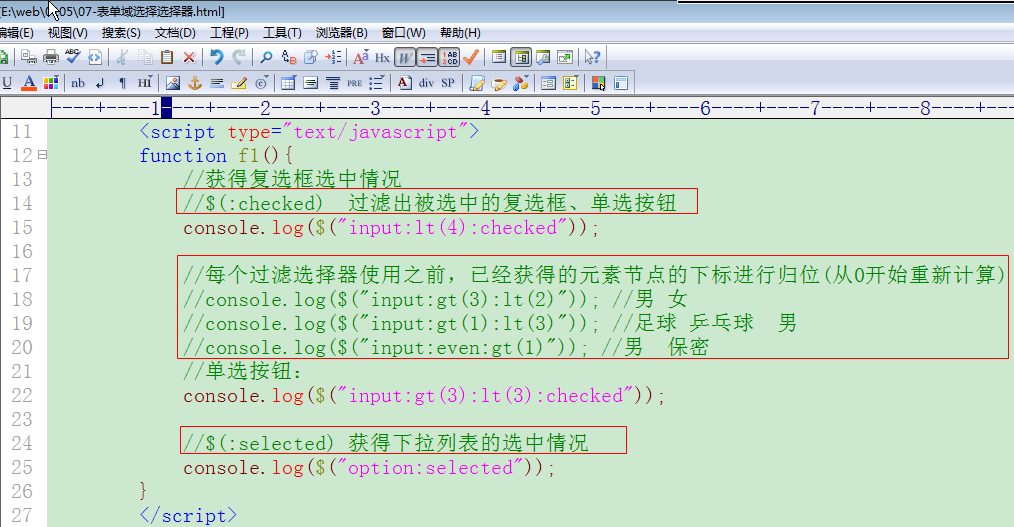
<div> </div>

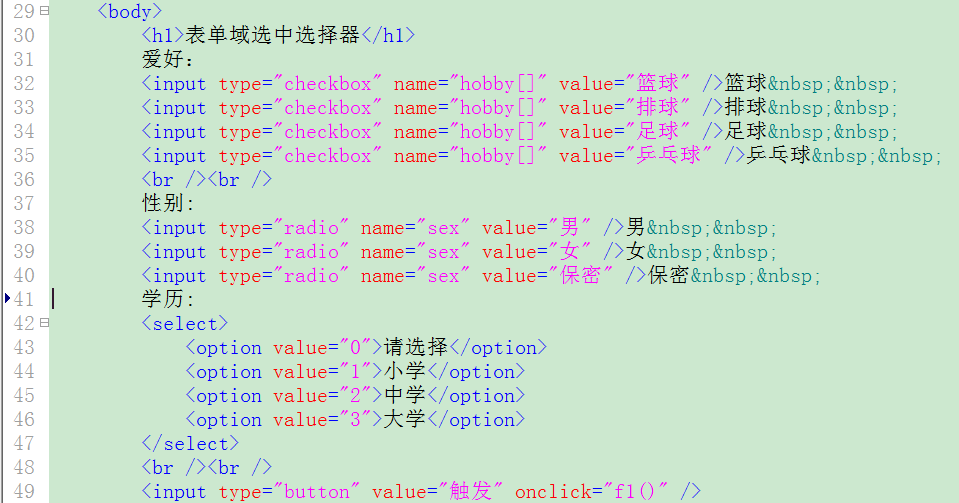




## 表单域选中选择器

复选框、单选按钮、下拉列表





# 三. 属性操作

<input type=”text” class=”apple” id=”username” name=”username” value=”tom”

address=”beijing”

/>

itnode.属性名称

itnode.属性名称= 值;

itnode.getAttribute(属性名称);

itnode.setAttribute(属性名称，值);

jquery方式操作属性(attribute)：

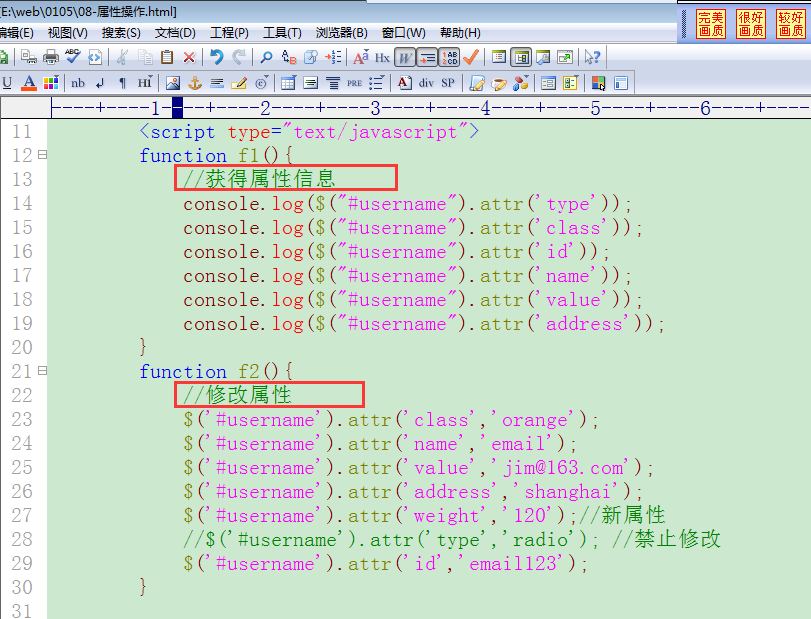
$().attr(属性名称); //获得属性信息值

$().attr(属性名称，值); //设置属性的信息

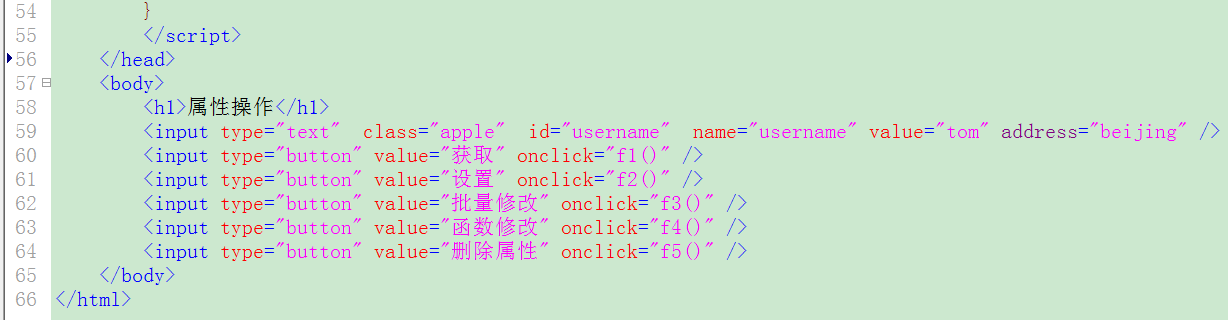
$().removeAttr(属性名称); //删除属性

$().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

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







# 四. 快捷操作

## 1. class属性值操作

$().attr(‘class’,值);

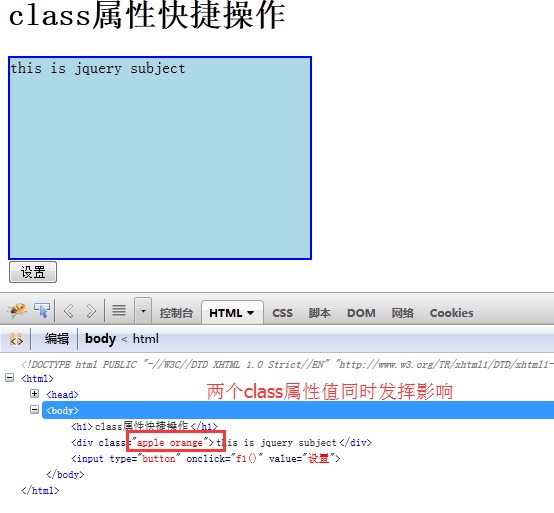
$().attr(‘class’);

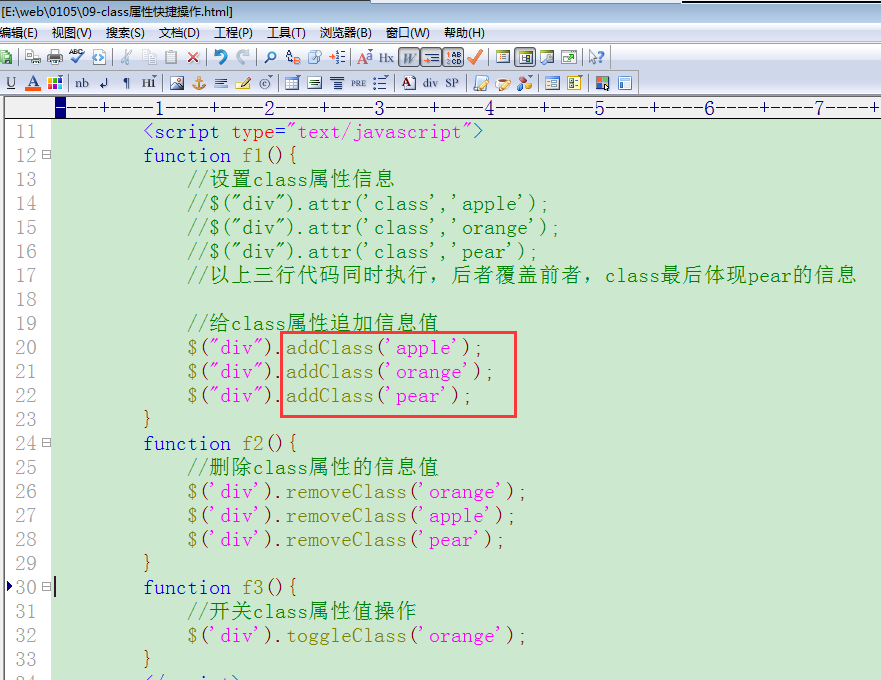
$().removeAttr(‘class’); //删除class的属性

$().addClass(值); //给class属性追加信息值

$().removeClass(值); //删除class属性中的某个信息值

$().toggleClass(值); //开关效果，有就删除，没有就添加







## 2. 标签包含内容操作

<div>hello<span>world</span></div>

javascript操作：

dvnode.innerHTML 获得div包含的信息

dvnode.innerHTML = XXX； 设置div包含的内容

(innerHTML不是w3c标准技术，许多浏览器对其有支持而已)

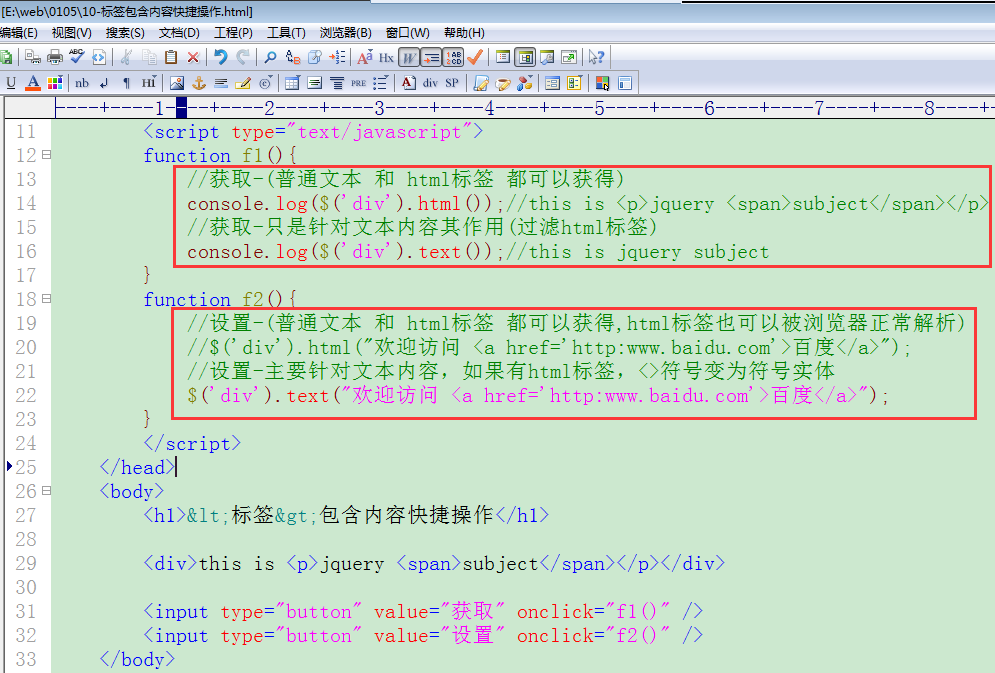
jquery操作：

$().html(); //获得节点包含的信息

$().html(信息); //设置节点包含的内容

$().text(); //获得节点包含的“文本字符串信息”内容

$().text(信息); //设置节点包含的内容(有html标签就把“><”符号变为符号实体)





### 2.1 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

1. 设置内容

前者可以设置html标签 和 普通字符串内容

后者只设置普通字符串内容，如果内容里边有tag标签内容，就把其中的”<”“>”符号转变为符号实体 <-----&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。

## 3. css样式操作

$().css(name，value); //设置

$().css(name); //获取

$().css(json对象); //同时修改多个css样式

### 3.1 css()样式操作特点：

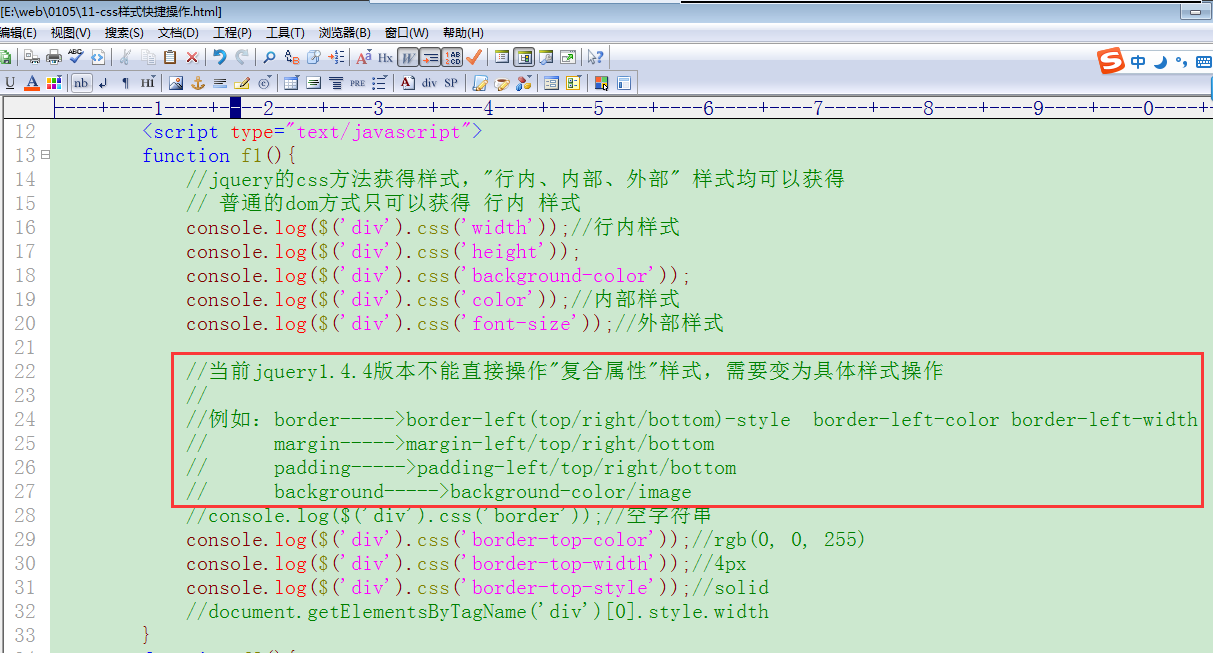
① 样式获取，jquery可以获取 行内、内部、外部的样式。

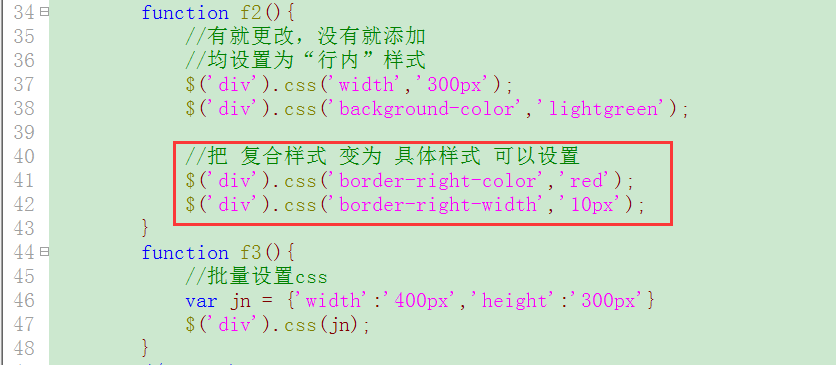
dom方式只能获得行内样式

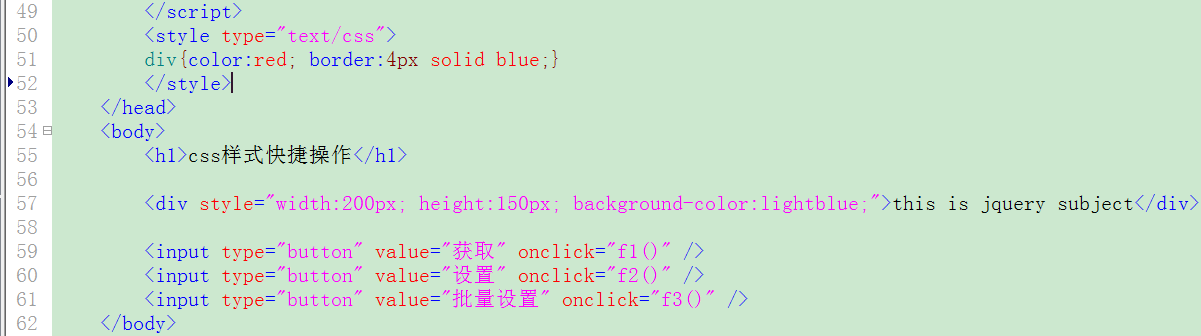
1. 一些jquery版本不支持复合属性操作，所以需要把复合属性样式需要拆分为"具体样式"才可以操作。  
   例如： background 需要拆分为 background-color background-image 等进行操作

border： border-left-style border-left-width border-left-color 等

margin： margin-left margin-top 等







## 4. value属性快捷操作

$().attr(‘value’);

$().attr(‘value’,信息值);

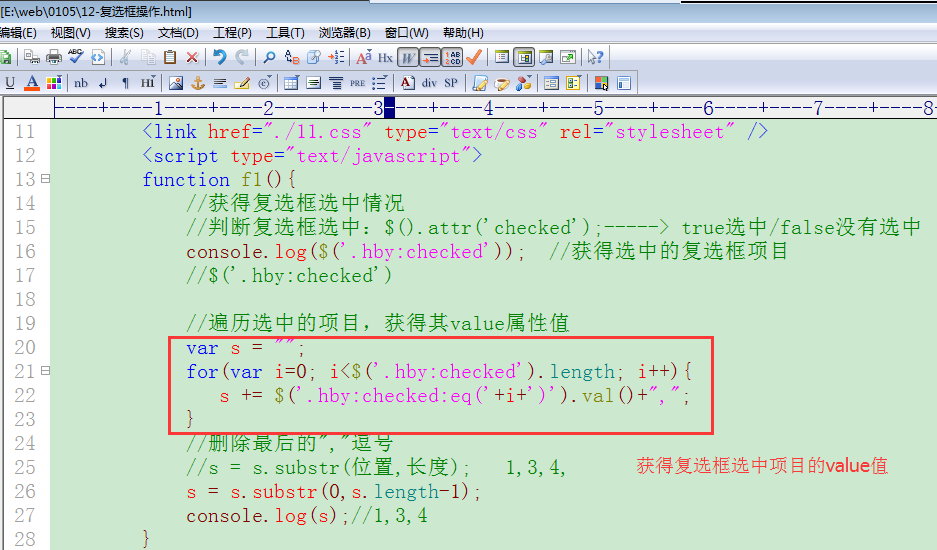
快捷操作：

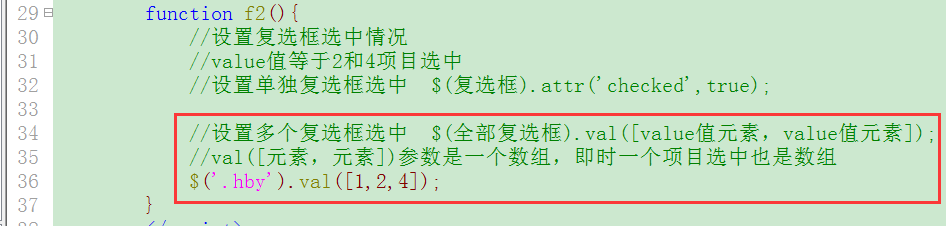
$().val(); //获得value属性值

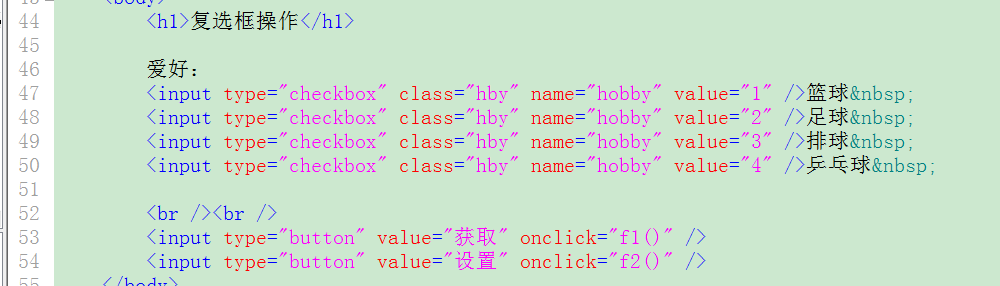
$().val(信息值); //设置value属性的值

该val方法在 复选框、单选按钮、下拉列表 的使用有凸出表现。

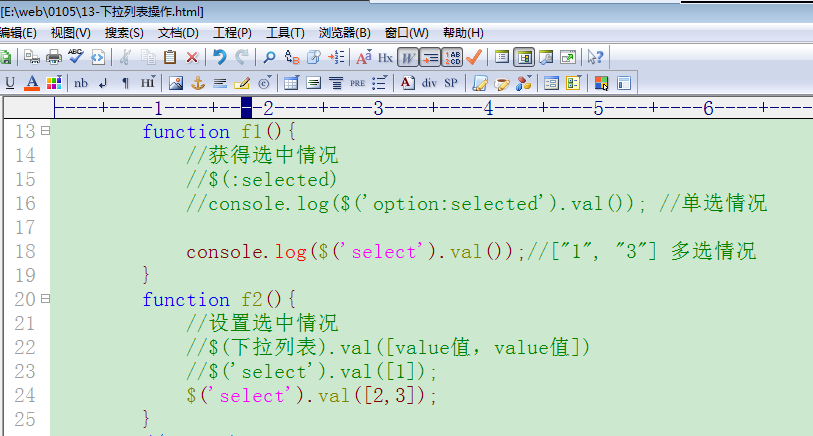
### 4.1 复选框操作

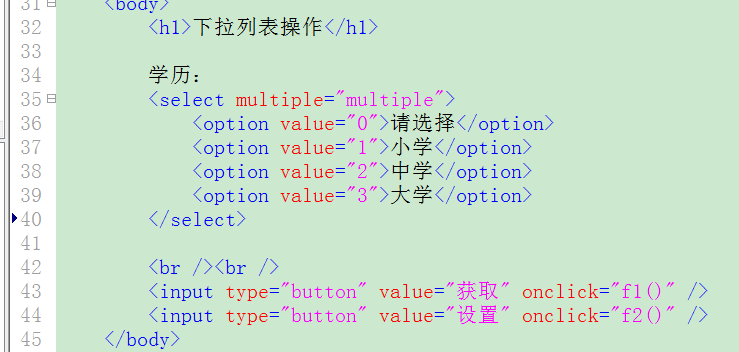






### 4.2 下拉列表操作





### 4.3 单选按钮操作



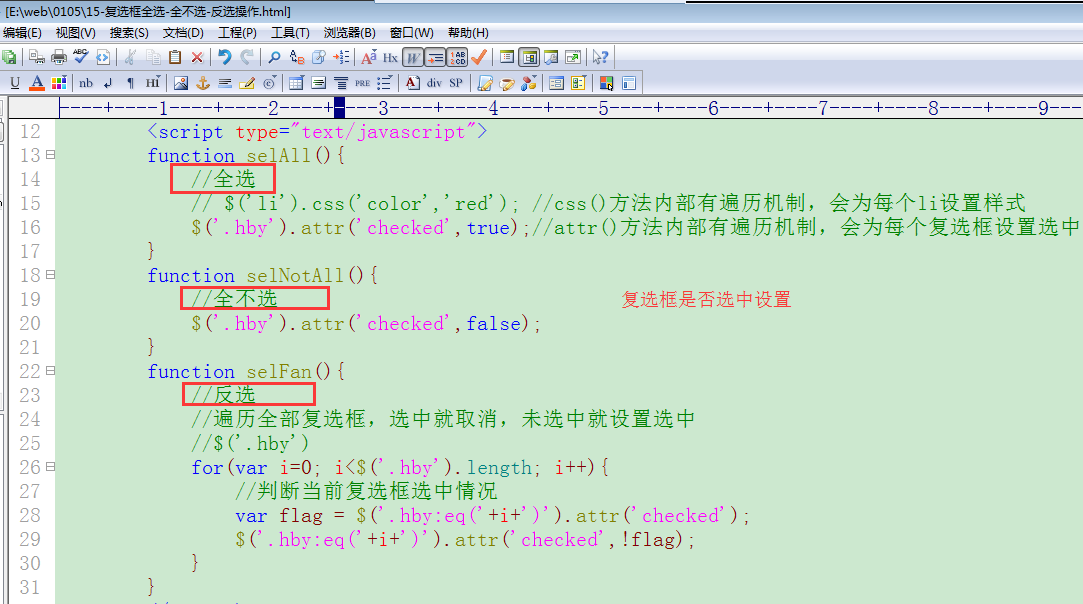
## 5. 复选框操作

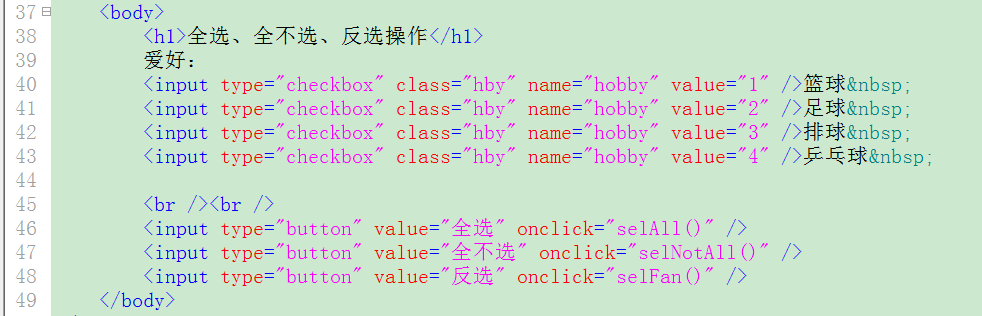
全选、反选、全不选

$().attr(‘checked’,true); //设置复选框选中

$().attr(‘checked’,false); //取消复选框选中

$().attr(‘checked’); //判断复选框选中情况，返回布尔值





总结：

1. 属性操作

$().attr(name)

$().attr(name,value)

$().attr(json对象);

$().removeAttr(name)

$().attr(name,function)

1. 快捷操作
   1. class属性值的操作

$().addClass(value)

$().removeClass(value)

$().toggleClass(value)

b) 包含内容操作

$().html() $().html(value)

$().text() $().text(value)

c) css样式操作

$().css(name,value)

$().css(name)

$().css(json对象)

d) val()

$().val()

$().val(value)

e) 复选框选中操作

$().attr(‘checked’,true);

$().attr(‘checked’,false);

作业：

1. 利用过滤选择器实现如下效果



1. 利用样式操作(height()、width())设置动画效果

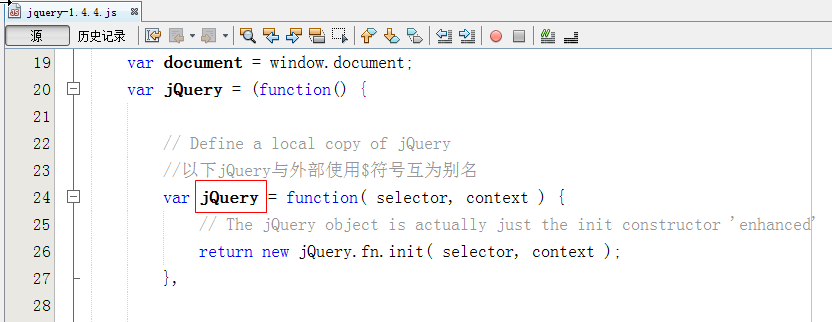


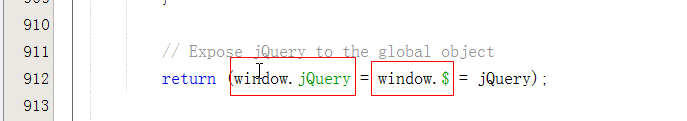
1. 制作一个“复选框”控制其他复选框的选中、不选中情况

# 五．$符号的由来

$(‘div’) $(‘.apple’) $(‘\*’)等等。

$符号就是一个函数，函数名称为”$”符号而已。也可以使用jQuery符号。





使用jquery过程中用到的$符号，其在jquery框架内部体现为是函数，其和jQuery关键字互为别名。

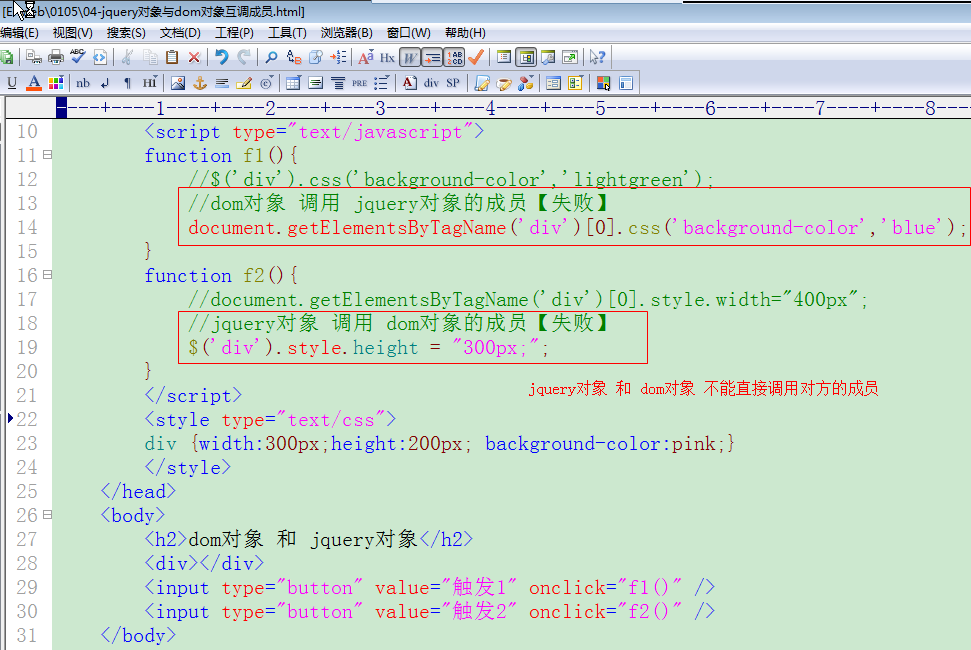
# 六. jquery对象 与 dom对象关系

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

dom对象： document.getElementById()

document.getElementsByTagName();

## jquery对象 和 dom对象 互相调用对方的成员



jquery对象 与 dom对象 互相调用对方成员失败了

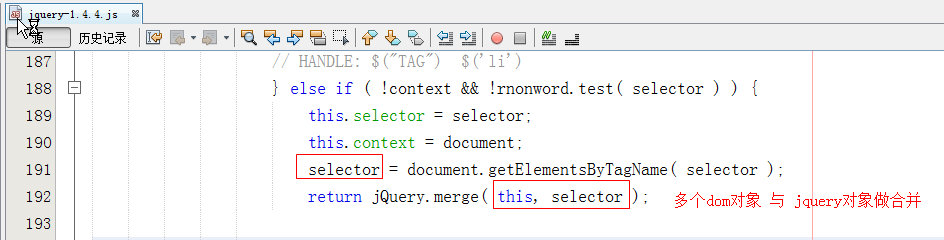
原因：它们不是平等关系，导致其成员也有层次划分。

jquery对象 是 包含 dom对象的

## 2. jquery对象 如何封装的dom对象

dom对象就是jquery对象的数组组成部分

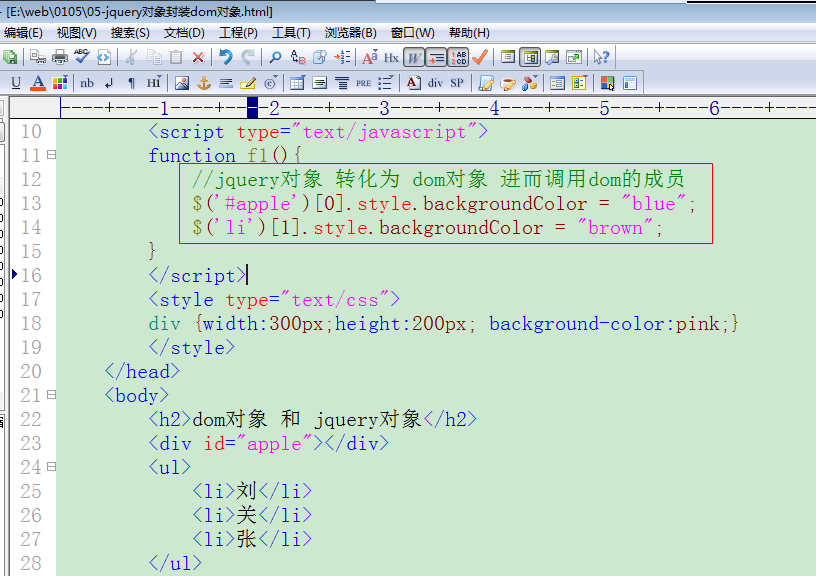




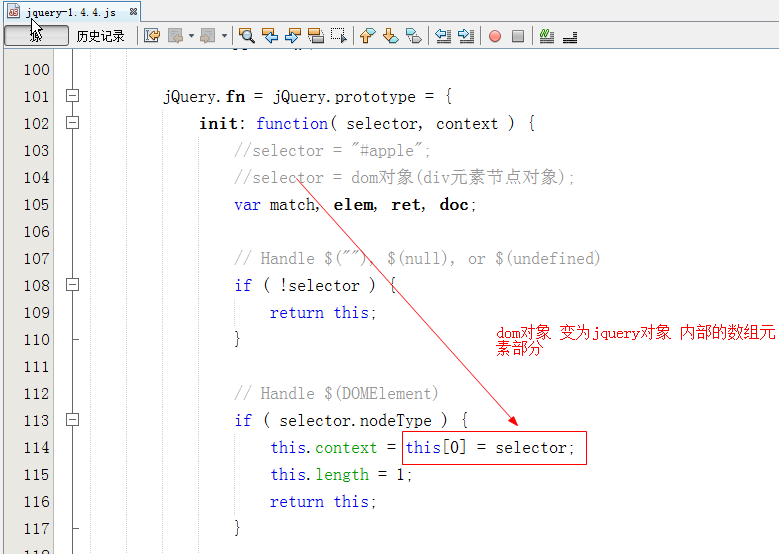
通过上图，我们看到dom对象 和jquery对象做合并，dom对象 是jquery对象 的数组组成部分。

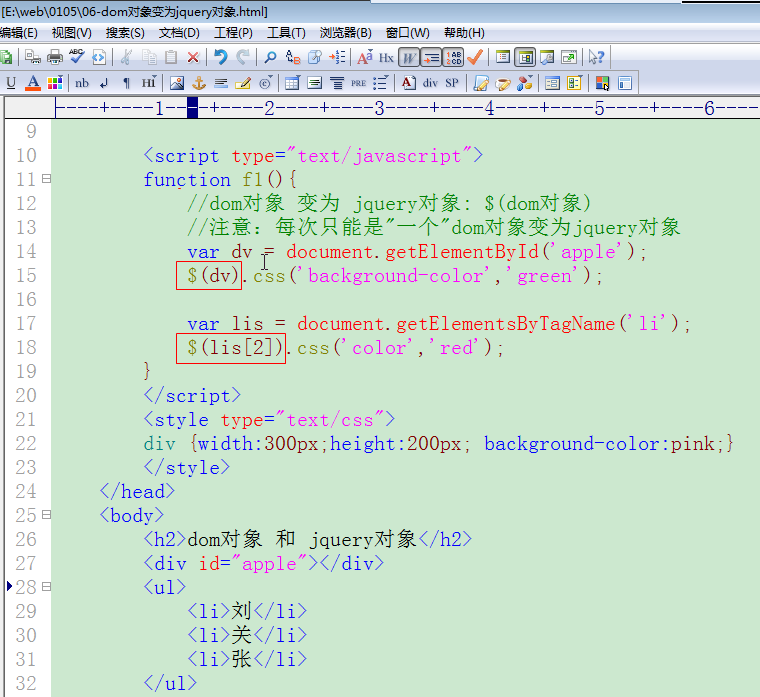
## 3. jquery对象 和 dom对象 的转化

### 3.1 jquery对象-----》dom对象： $()[下标]



### 3.2 dom对象-----》jquery对象：$(dom对象)





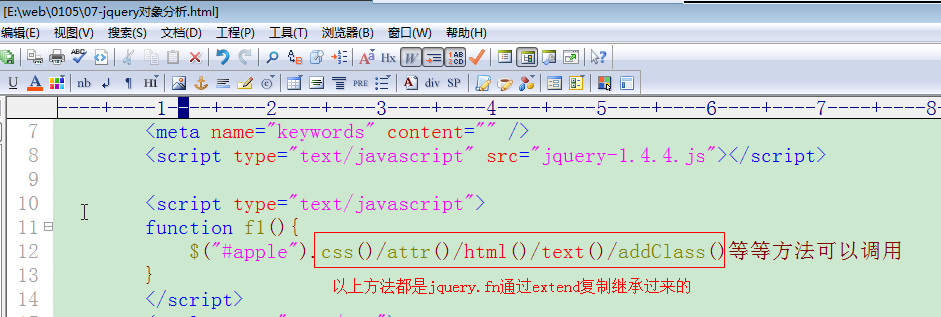
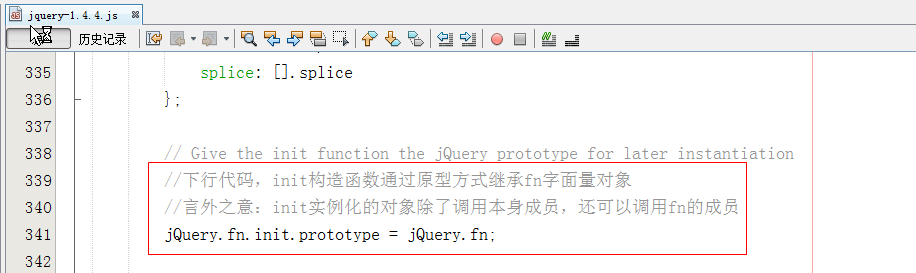
# 七. jquery框架对象分析

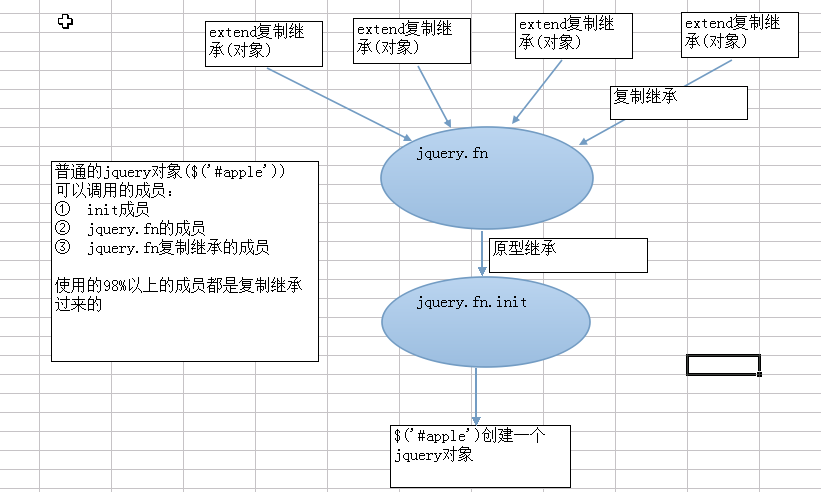
jquery框架对象类型：jquery对象 和 $对象

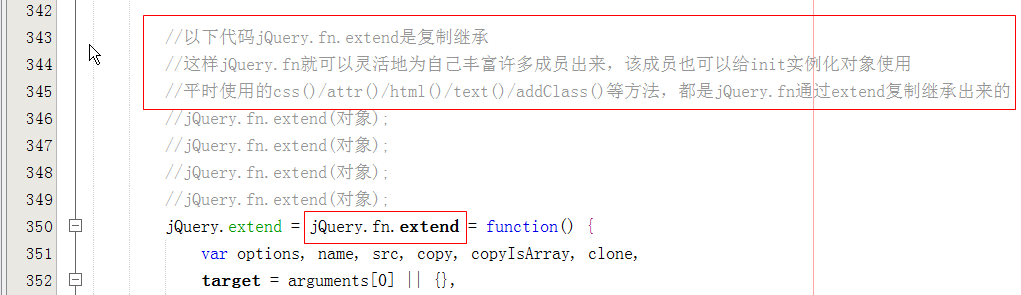
① jquery对象就是各种选择器创建出来的对象

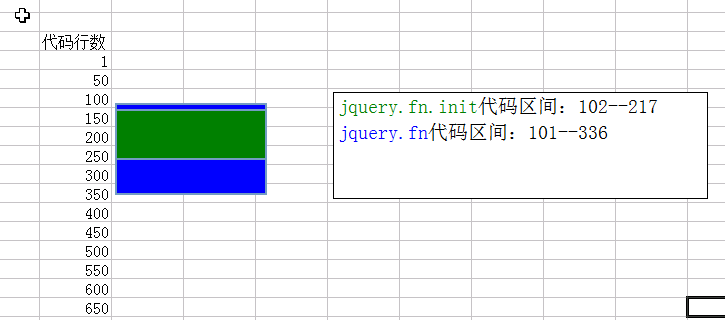
② $对象就是函数对象

## jquery对象



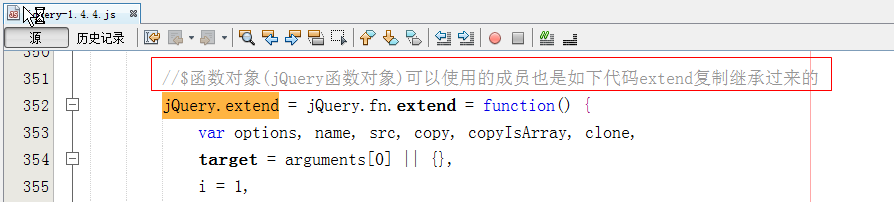




## 2. $函数对象

$函数对象可以调用的许多成员也是通过extend复制继承过来的

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



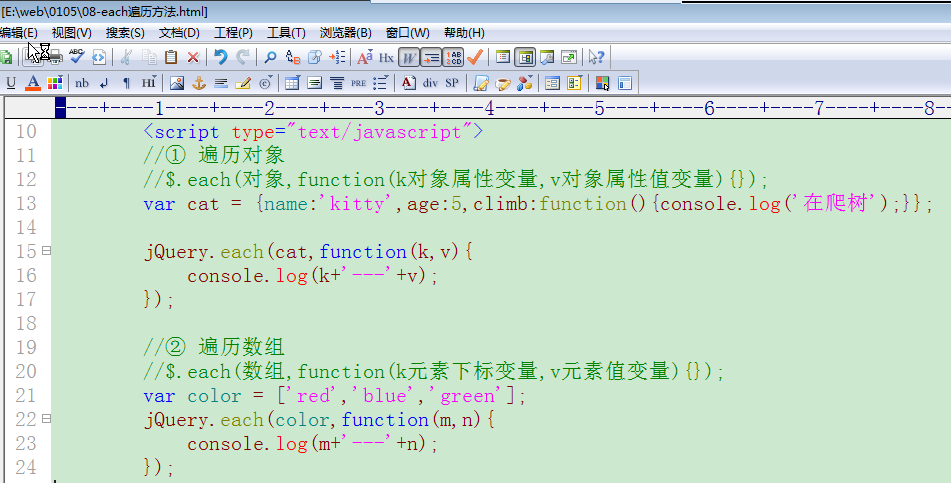
以上无论是jquery对象 还是 $对象，他们98%以上的成员都是通过各自的extend复制继承过来的。

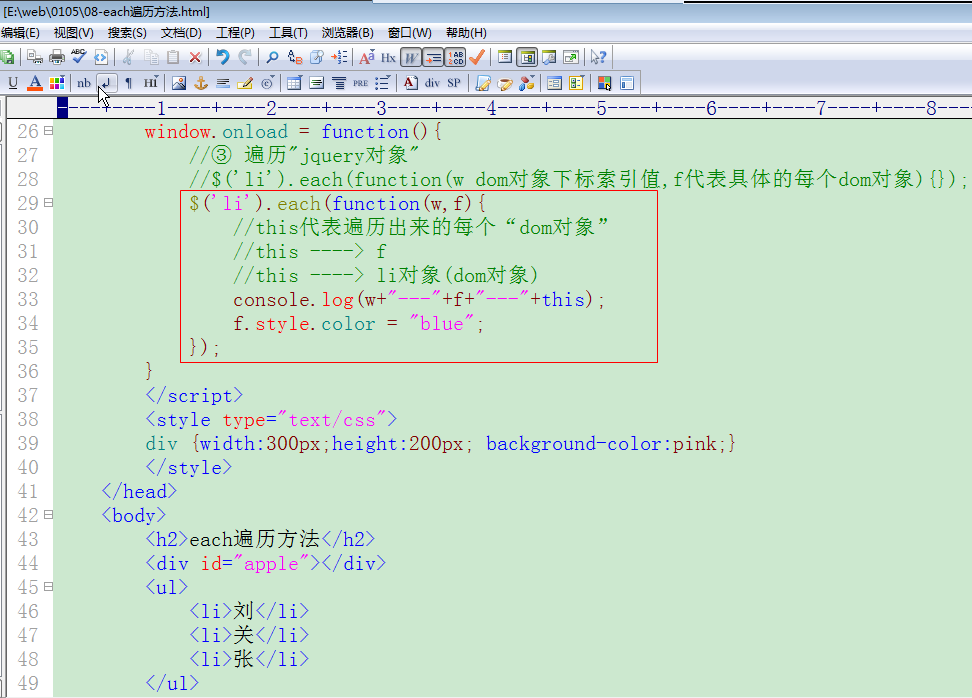
# 八. 遍历方法

each()方法：

$.each(数组/对象,function处理); //$对象 调用的

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





总结：

1. $符号的由来
2. jquery对象 和 dom对象的关系

dom对象 是jquery对象的数组组成部分

jquery对象---》dom对象：$()[下标]

dom对象----》jquery对象：$(dom对象)

1. jquery框架对象分析

类型：jquery对象 $(jQuery)对象

jquery对象(new jQuery.fn.init)----原型----->jQuery.fn-----复制继承--->extend(许多对象)

$对象 -----复制继承------->extend(许多对象)

jquery框架对象：

jquery对象:通过各种选择器获得的对象都是jquery对象，例如$(“div”) $(“.apple”)

$对象:其是函数对象，$.get() $.post() $.ajax() $.each

# 九. 加载事件

javascript的加载事件：

<body onload = “函数()”>

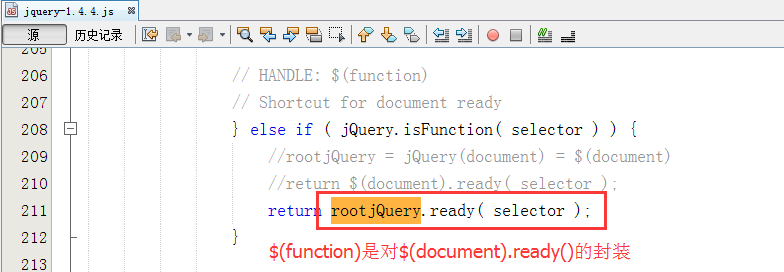
window.onload = function(){}

## 1. jquery加载事件实现

① $(document).ready(function处理);

② $().ready(function处理);

③ $(function处理); 对第一种加载的封装而已



## 2. jquery加载事件与传统加载事件的区别

### 2.1 设置个数

在同一个请求里边，jquery的可以设置多个，而传统方式只能设置一个

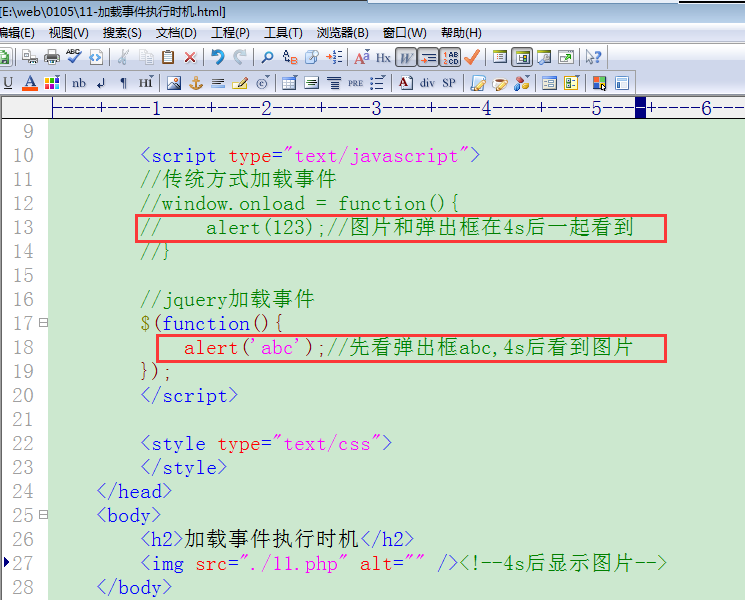
传统方式加载事件是给onload事件属性赋值，多次赋值，后者会覆盖前者。

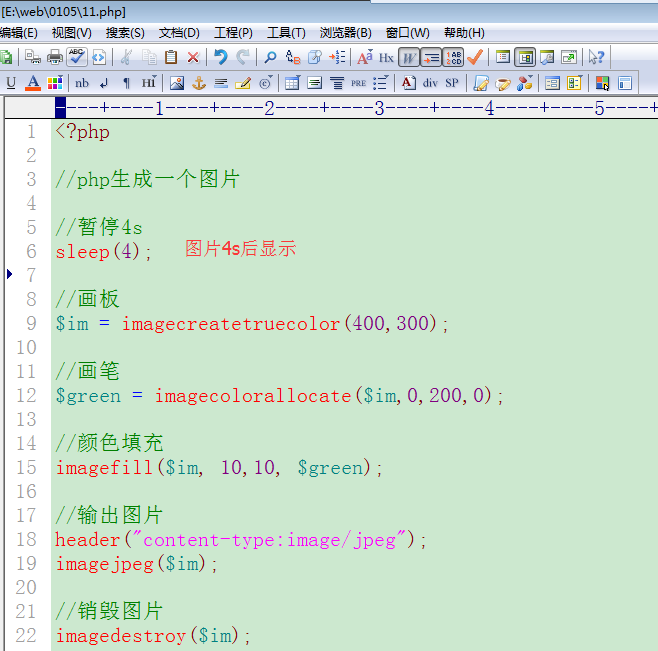
jquery方式加载事件是把每个加载事件都存入一个数组里边，成为数组的元素，执行的时候就遍历该数组执行每个元素即可，因此其可以设置多个加载事件。

### 2.2 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。

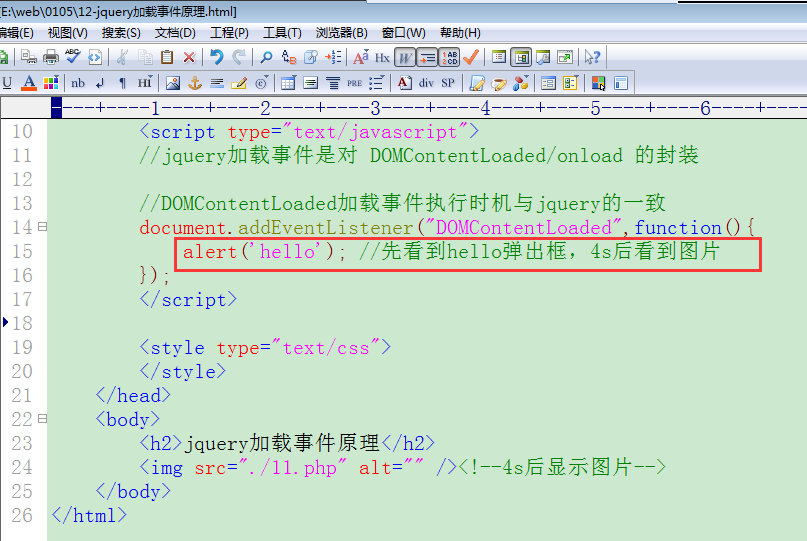
jquery方式加载事件，只要全部内容(文字、图片、样式)在内存里边对应的DOM树结构绘制完毕就给执行，有可能对应的内容在浏览器里边还没有显示。





## 3. jquery加载事件原理

jquery加载事件是对DOMContentLoaded的封装(而非onload)



# 十. 普通事件操作

① dom1级事件设置

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

<input type=”text” onclick=”函数()” />

itnode.onclick = function(){}

itnode.onclick = 函数;

② dom2级事件设置

itnode.addEventListener(类型，处理，事件流);

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

node.attachEvent();

node.detachEvent();

③ jquery事件设置

$().事件类型(事件处理函数fn); //设置事件

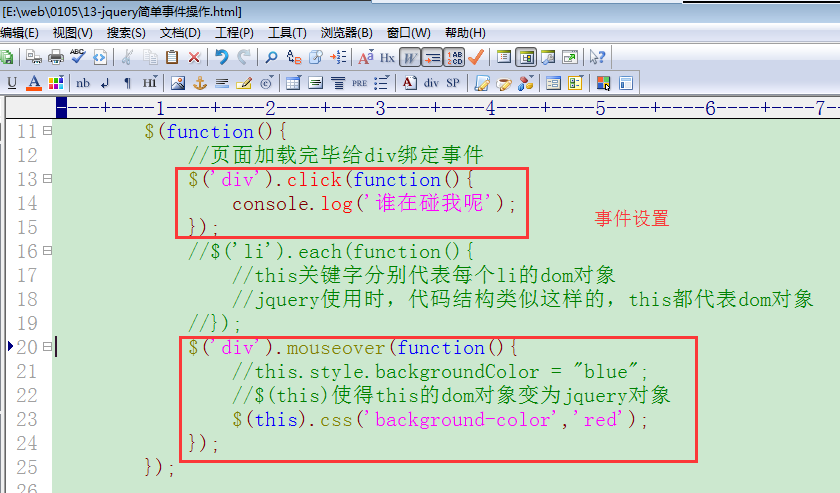
$().事件类型(); //触发事件执行

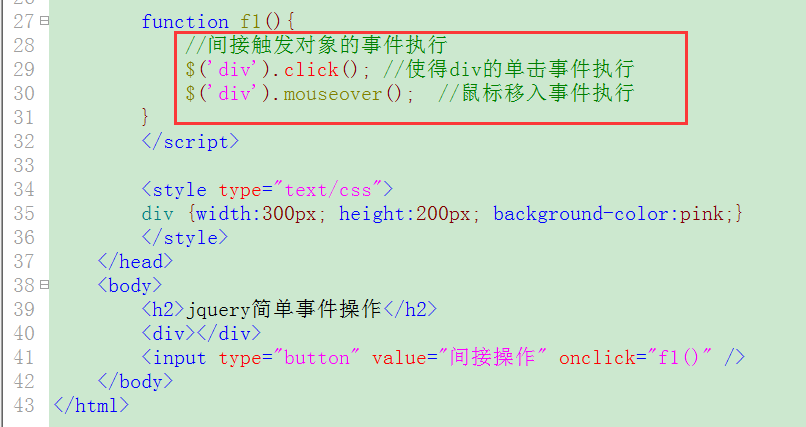
事件类型：click、keyup、keydown、mouseover、mouseout、blur、focus等等

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

（该方式事件函数内部this都代表jquery对象内部的dom节点对象）。

jquery调用的大部分方法里边的this关键字都代表其对应的dom对象。



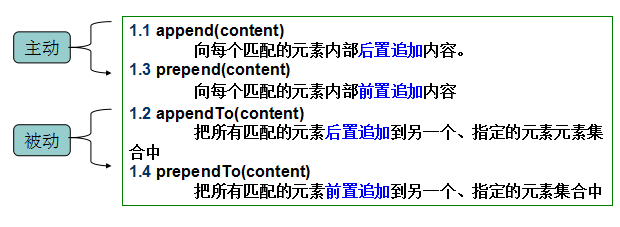


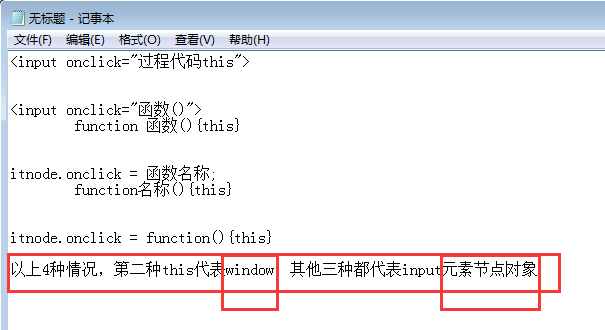
# 十一. jquery对文档的操作

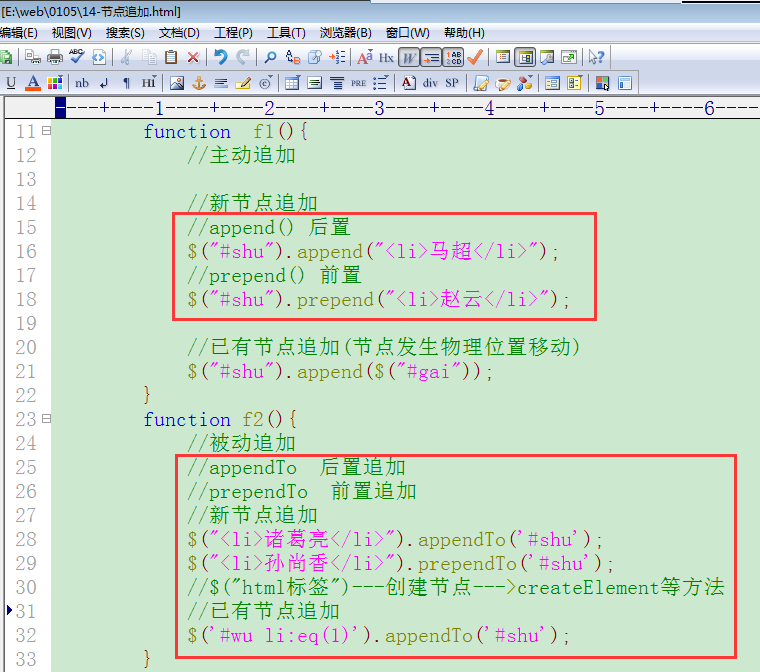
通过jquery方式实现页面各种节点的追加、修改、删除、复制等操作

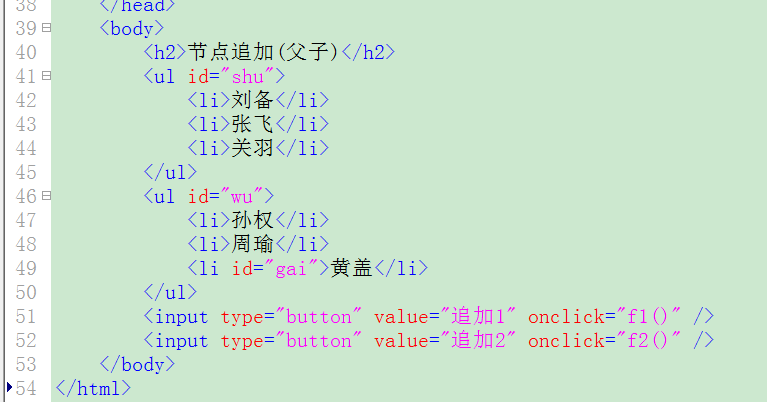
## 1. 节点追加

### 1.1 父子关系追加

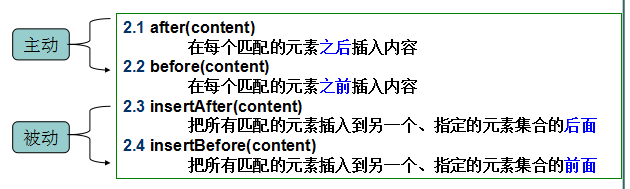


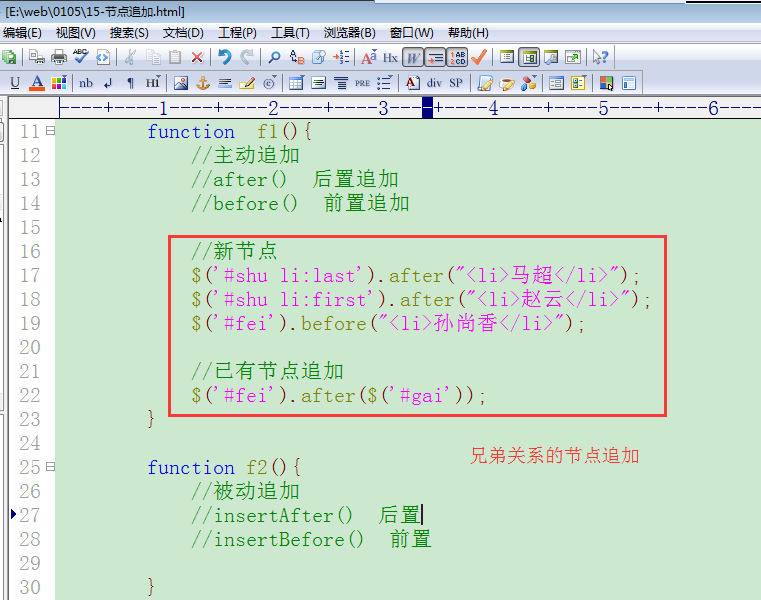






### 1.2 兄弟关系追加



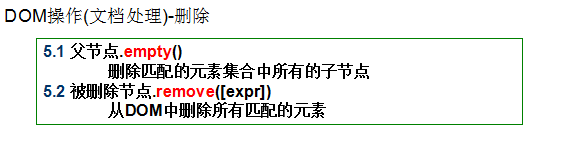


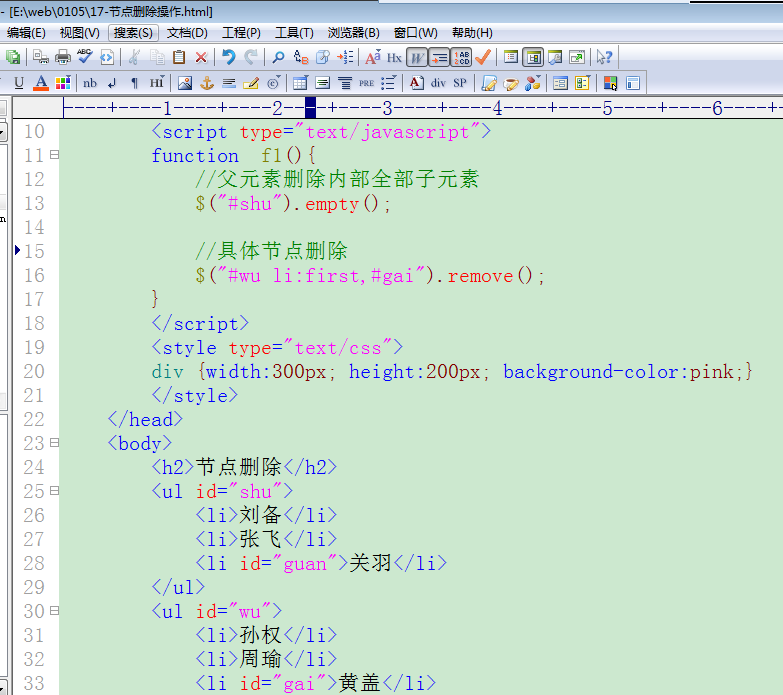
## 节点替换



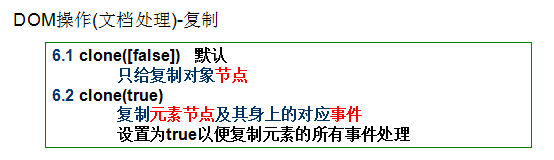


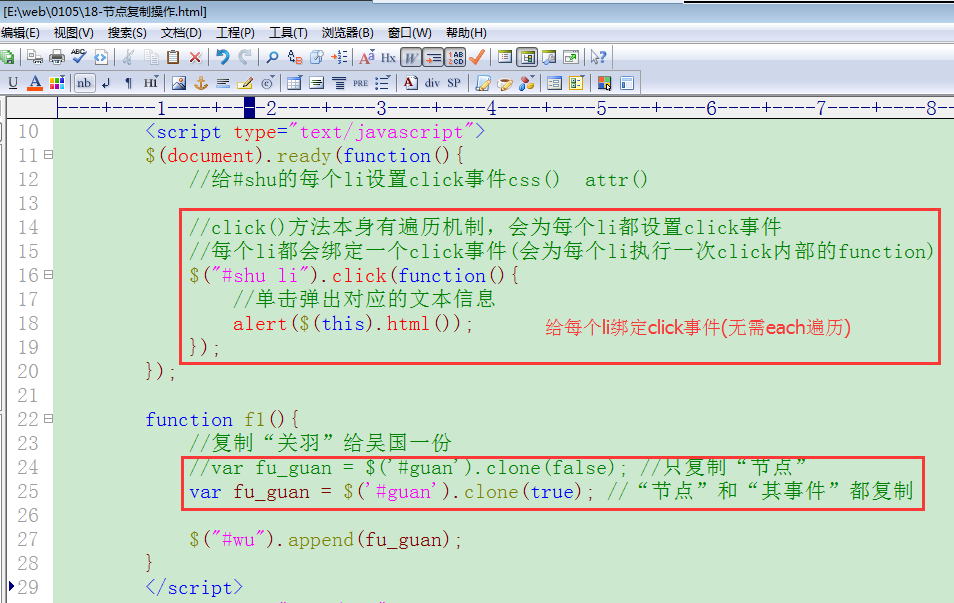
## 节点删除





## 复制节点







# 十二.属性选择器使用







总结：

1. 加载事件使用

$(document).ready(funciton)

$().ready(function)

$(function);

与传统加载事件的区别：

① 加载事件个数：jquery可以设置多个、传统的只有一个生效

② 执行时机

jquery加载事件是对DOMContentLoaded的封装。

1. 简单事件设置

$().事件类型([function])

1. 文档DOM的处理

节点追加：

append() prepend() appendTo() prependTo()

after() before() insertAfter() insertBefore()

替换：

replaceWith() replaceAll()

复制：

clone(true/false)

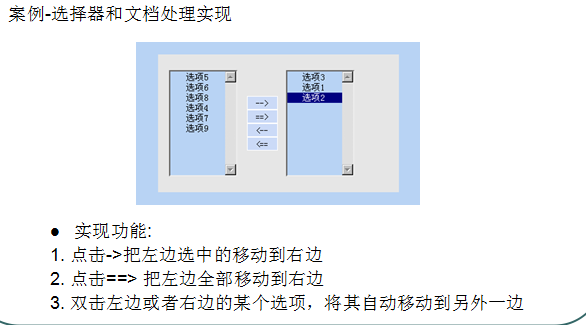
删除：

empty()

remove()

作业：

1. 两个下拉列表项目切换(dom节点操作)



1. 制作无序列表，利用mouseover和mouseout实现鼠标当前行高亮显示
2. 练习，图片4秒后显示，在加载事件里边给图片的属性(宽度、高度、图片)做更改

4. 练习

广告图片右上角有×图标，点击广告图片小时，实现效果，看到图片×但是不能关闭图片(display:none)，需要5s后才可以关闭。【传统加载事件】

×上边有onclick事件。

看到×就要单击关闭【jquery加载事件】

# 十三.事件绑定

## 1. 事件绑定

jquery事件的简单操作：

$().事件类型(function事件处理);

$().事件类型();

### 1.1 jquery事件绑定

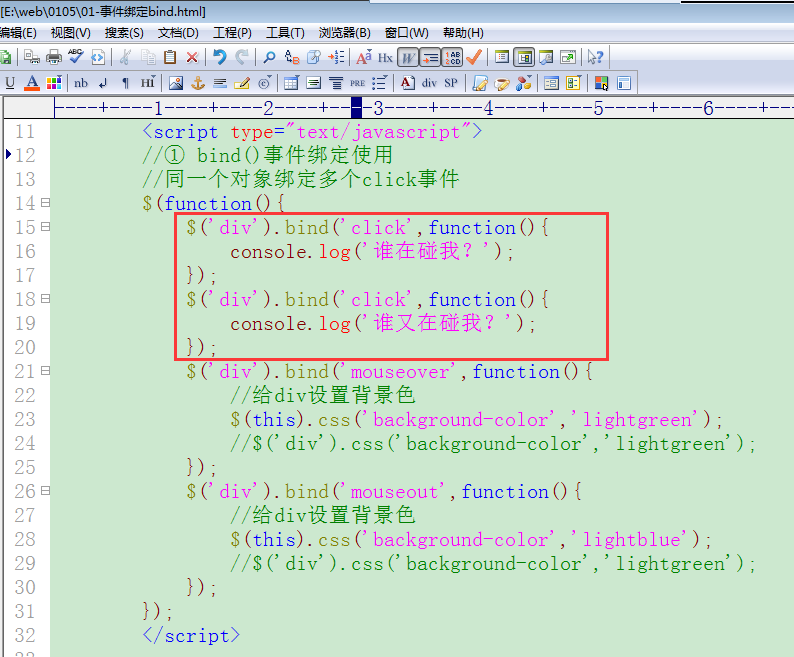
$().bind(事件类型，function事件处理);

$().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

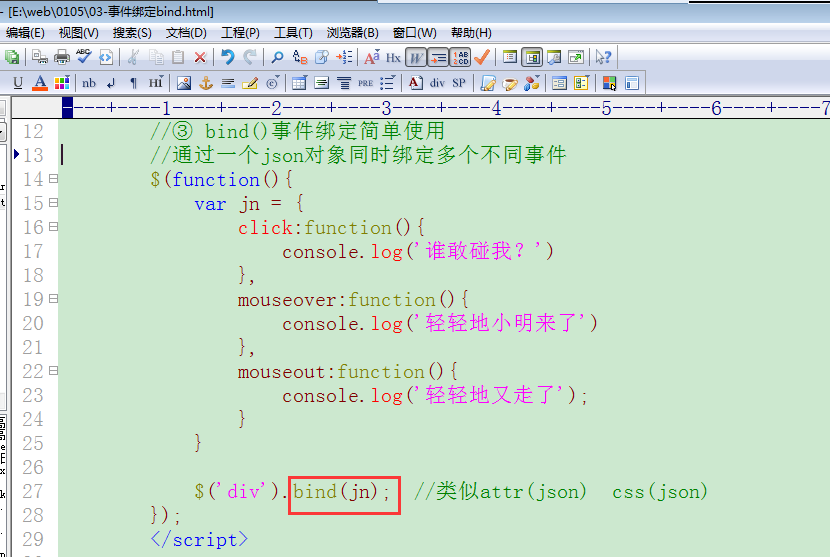
$().bind(json对象); //同时绑定多个不同类型的事件

(事件类型：click mouseover mouseout focus blur 等等)

事件处理：有名函数、匿名函数







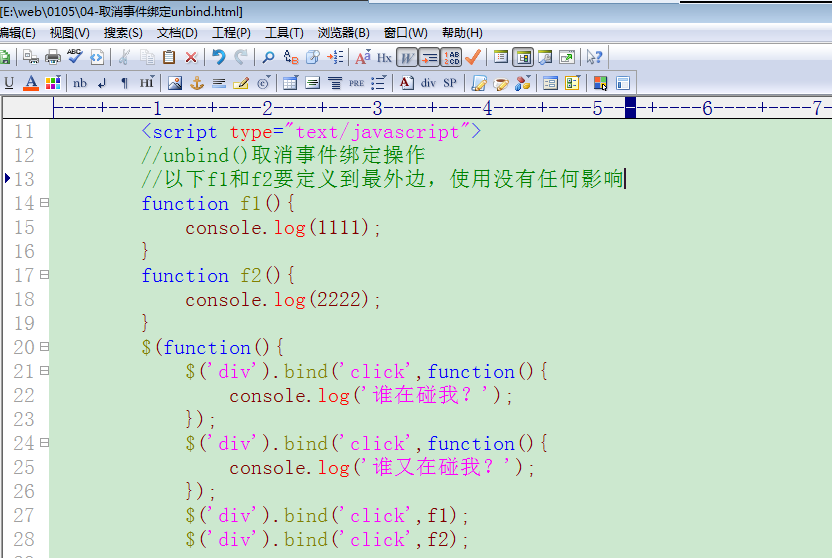
### 1.2 取消事件绑定

① $().unbind(); //取消全部事件

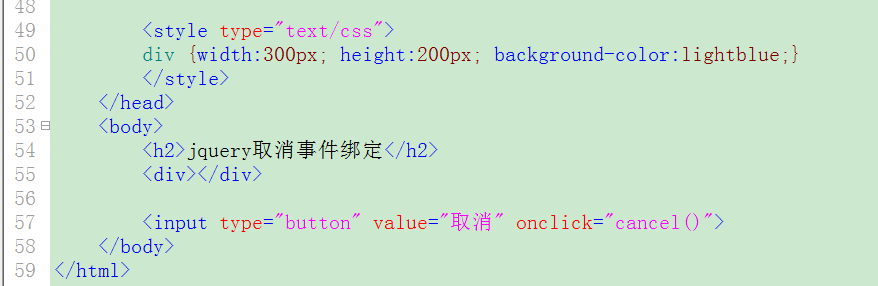
② $().unbind(事件类型); //取消指定类型的事件

③ $().unbind(事件类型，处理); //取消指定类型的指定处理事件

注意：第③种取消事件绑定，事件处理必须是有名函数。







事件绑定是丰富事件操作的形式而已。

从dom2级事件操作开始，同一个对象可以绑定多个同类型的事件，具体如下：

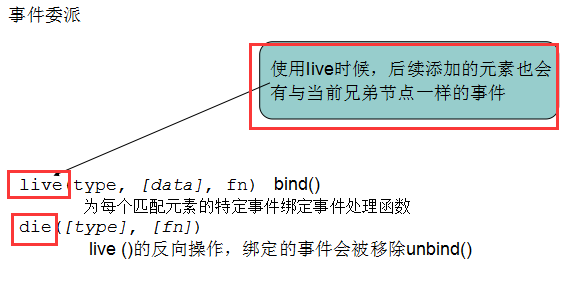
dvnode.addEventListener(‘click’,fn);

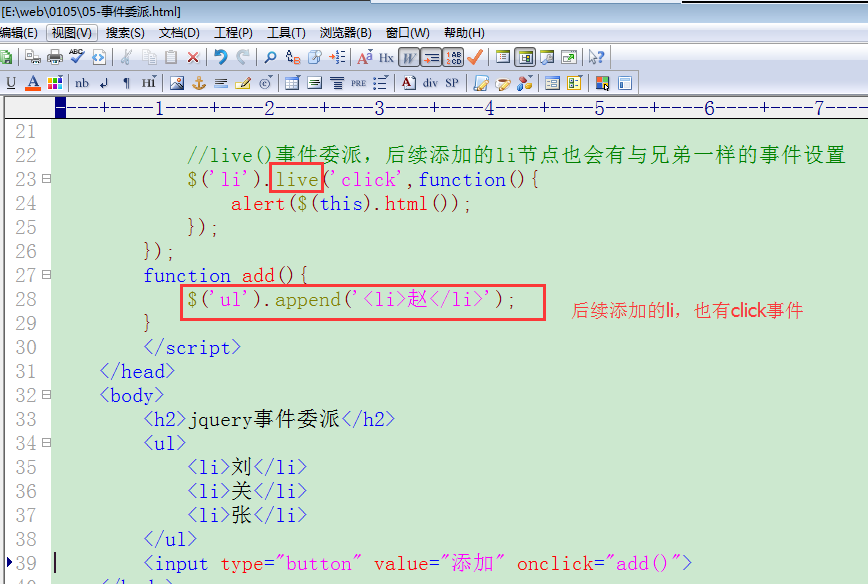
dvnode.addEventListener(‘click’,fn);

$(‘div’).click(function);

$(‘div’).click(function);

## 事件委派





## 3. 事件对象、阻止浏览器默认动作、阻止事件冒泡

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

$().click(function(evt){});

$().bind(‘mouseover’,f1);

function f1(evt){}

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

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 = true; IE浏览器

在jquery里边：

$().bind(‘click’,function(evt){

evt.preventDefault();

evt.stopPropagation();

});

preventDefault()方法是jquery的方法，名字与js底层代码的名字一致而已。

并且其有做浏览器兼容处理

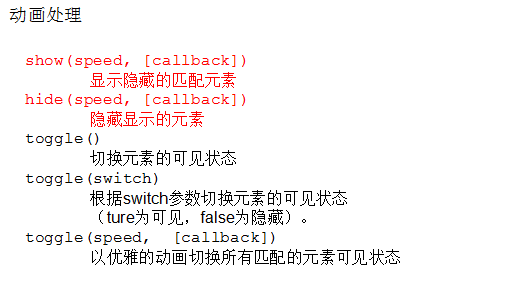
stopPropagation()方法是jquery的方法，名字与js底层代码的名字一致。

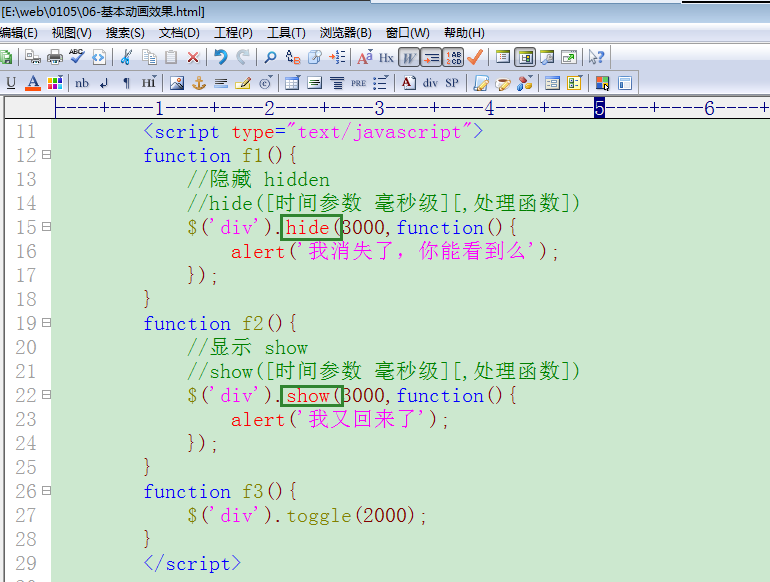
其有做浏览器兼容处理

# 十四. 动画效果

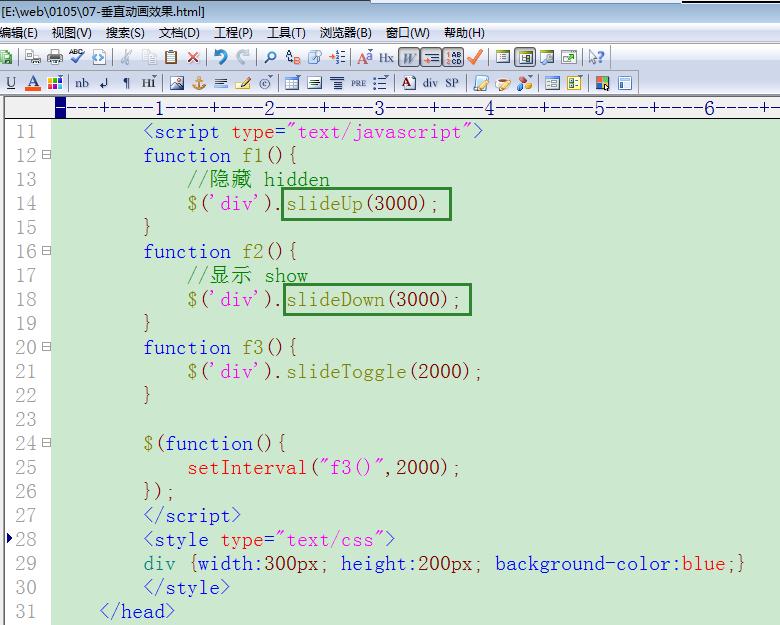
## 1. 基本动画



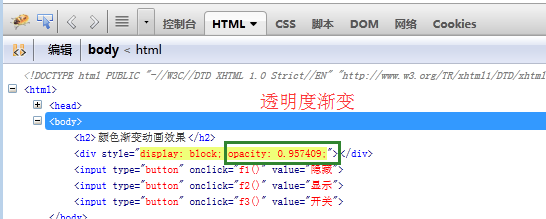


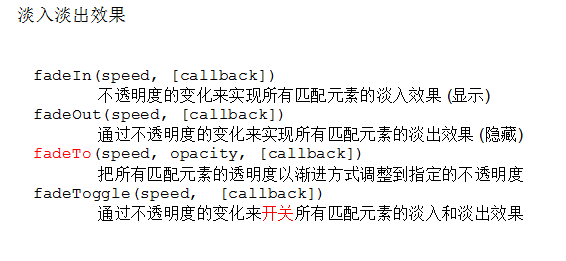


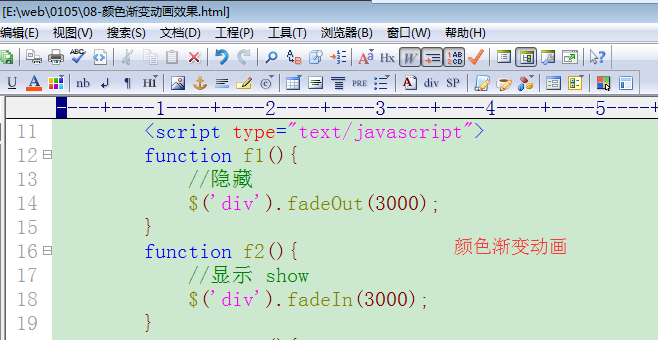
## 垂直动画

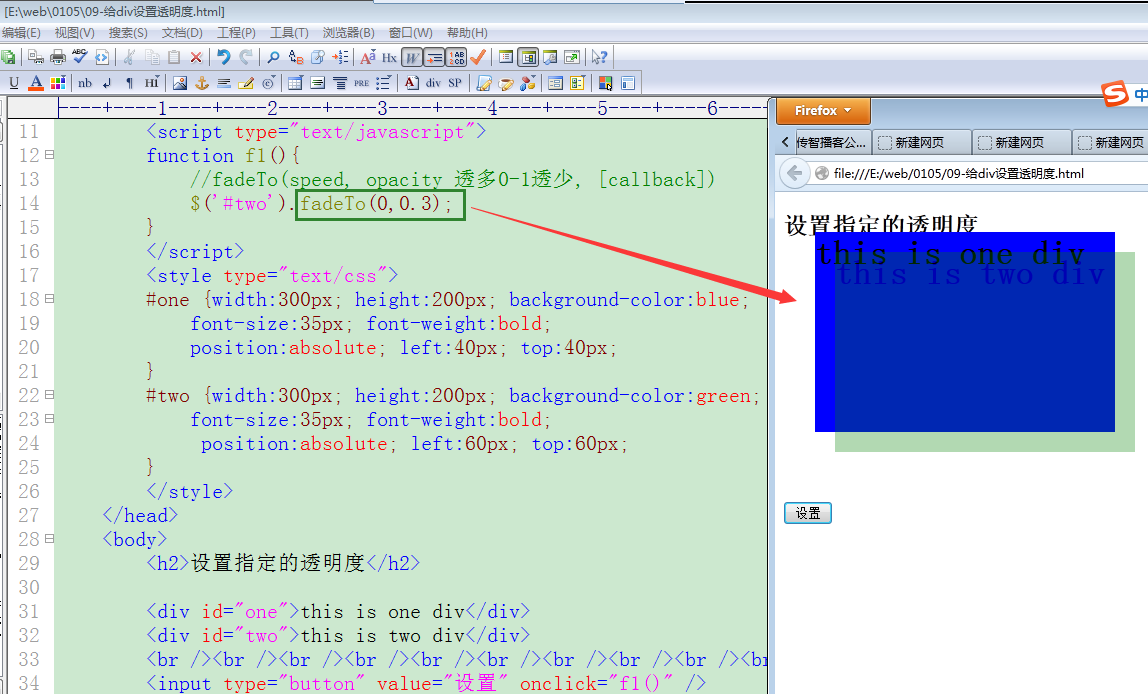


## 颜色渐变动画









# 十五. jquery封装的ajax

具体操作：

$.get(url [,data] [,fn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

fn：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

dataType：服务器返回数据类型，html、text、xml、json

(该ajax是异步的get方式请求)

$.post(url[,data][,fn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方式请求

$.ajax({ //json对象

url:请求地址,

data:给服务器传递的数据,

dataType:数据从服务器返回格式html、text、xml、json

type:get/post请求方式

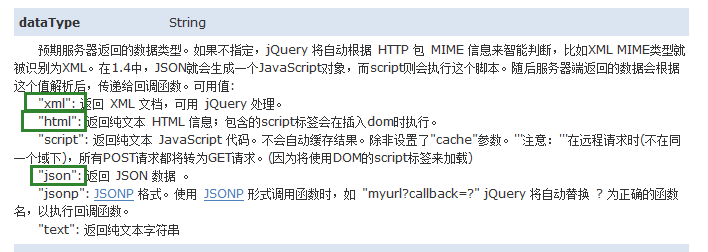
success:function(){} ajax成功请求后的回调函数，可以做后续处理使用

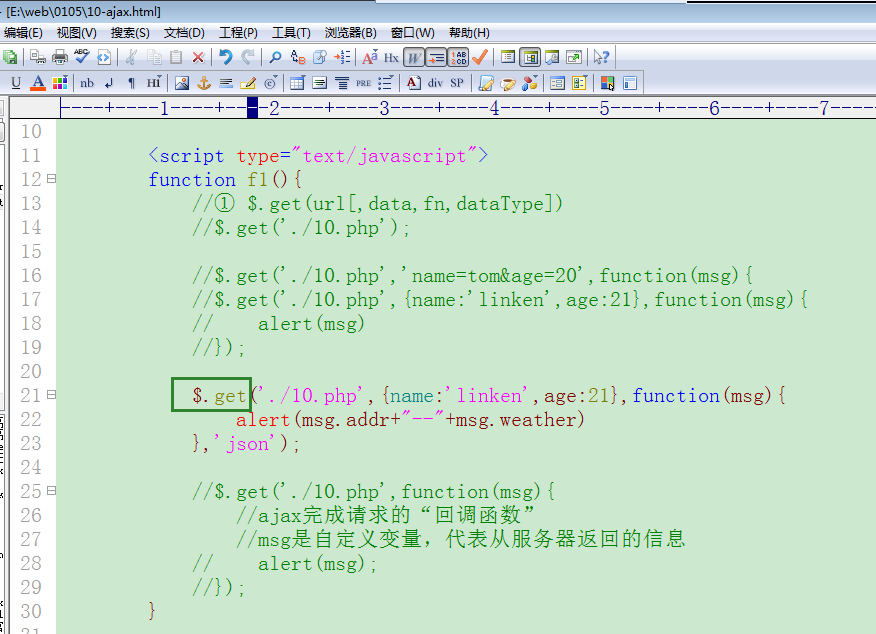
async:[true]异步/false同步，

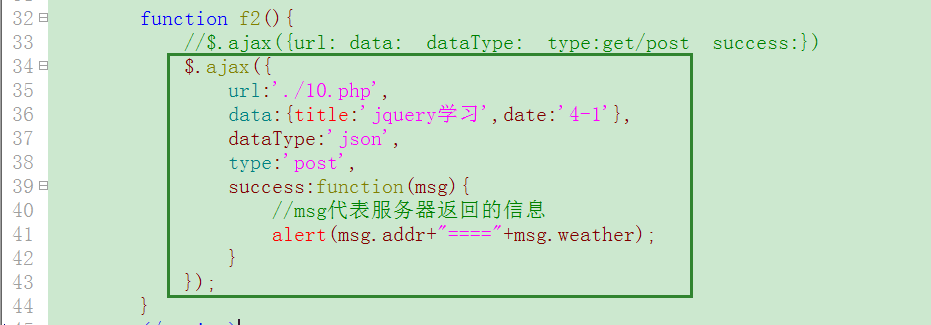
cache:[true]缓存/false不缓存,

}

)







总结：

1. 事件绑定

$().bind(事件类型，function处理)

$().bind(类型1 类型2 类型3.., function处理);

$().bind(json对象);

取消绑定

$().unbind();

$().unbind(类型)

$().unbind(类型，有名函数处理)

1. 事件委派

$().live()

$().die()

与事件绑定不同：后续添加的节点也会有与兄弟一样的时间设置

1. 动画效果

基本：show(时间) hide(时间)

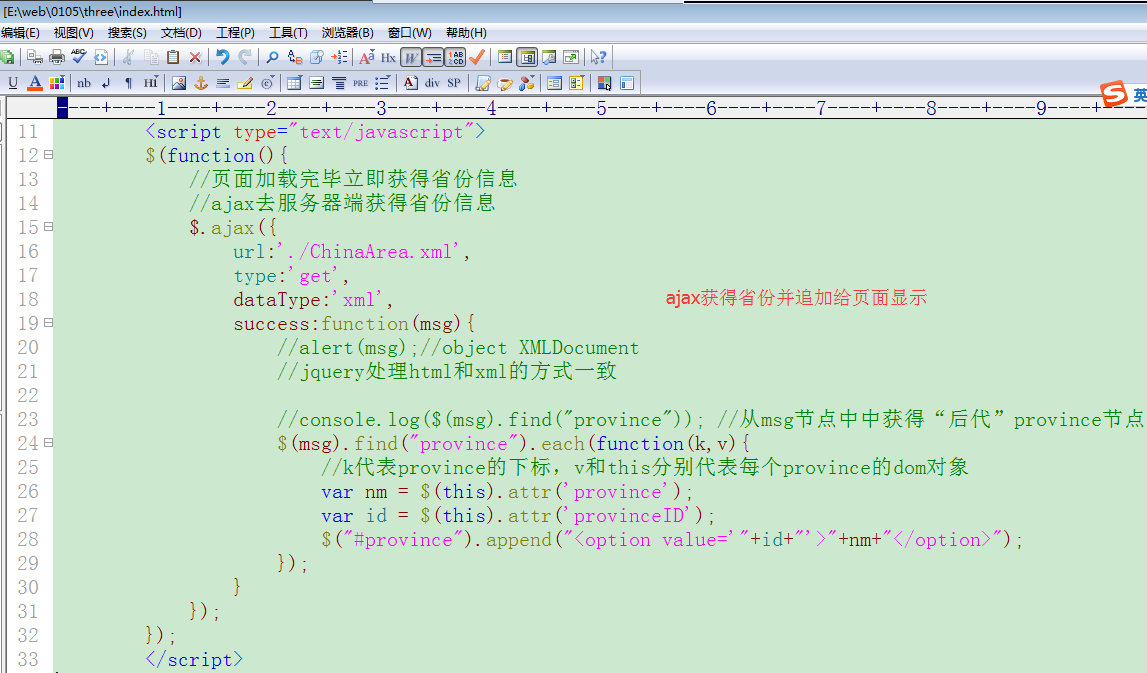
垂直: slideDown() slideUp()

颜色渐变：fadeIn() fadeOut() fadeTo()

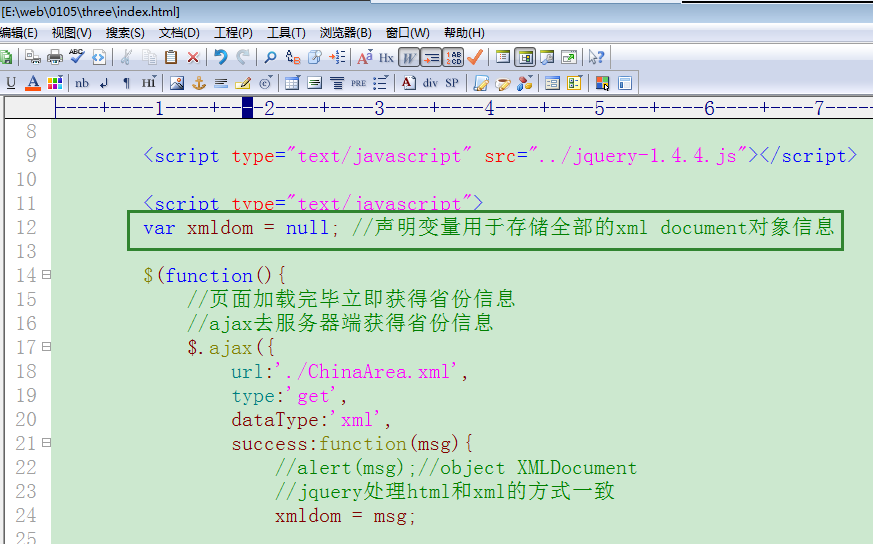
# 十六. 地区三级联动

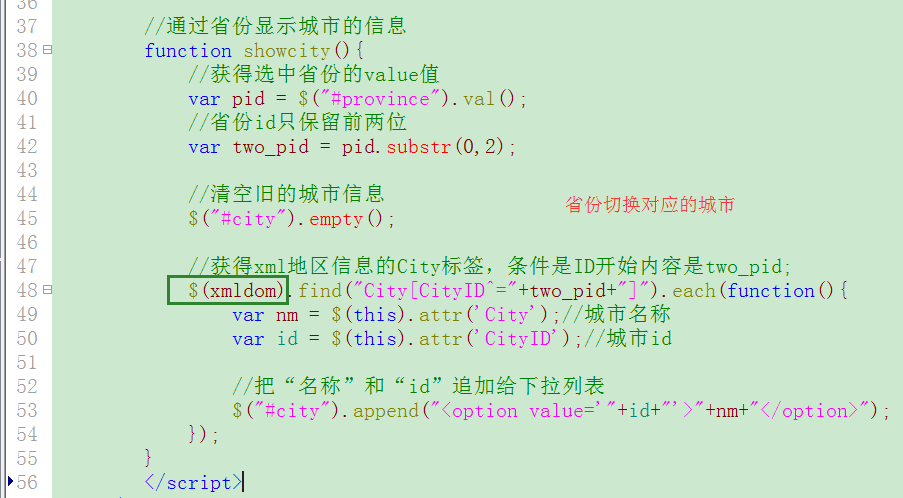
技术：jquery + ajax + xml/json

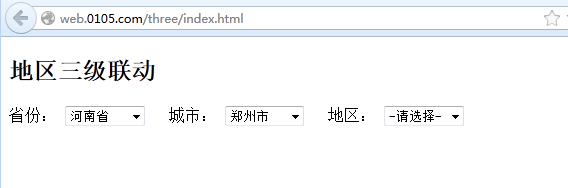
## 省份的获取和显示



## 显示对应城市信息





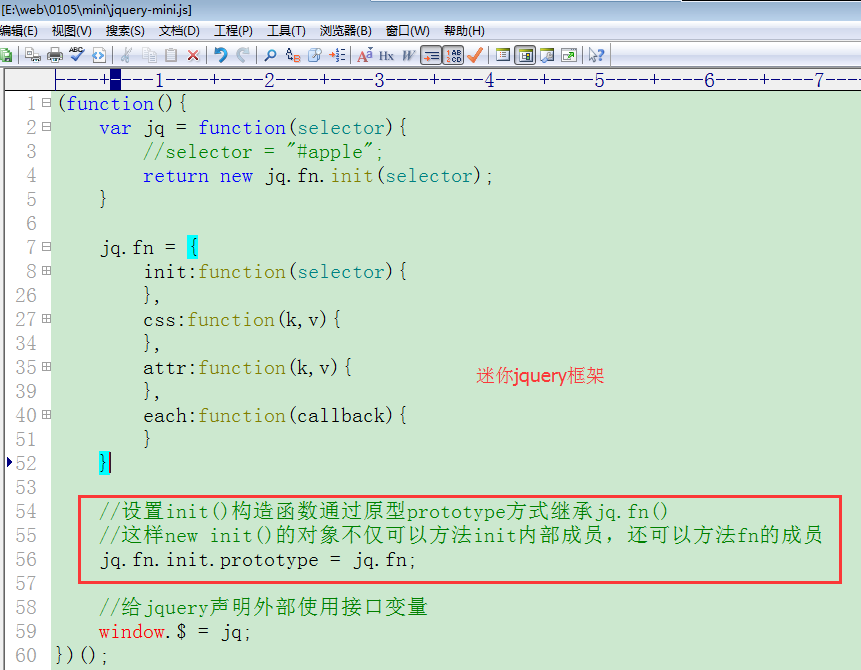


# 十七. 迷你版jquery开发

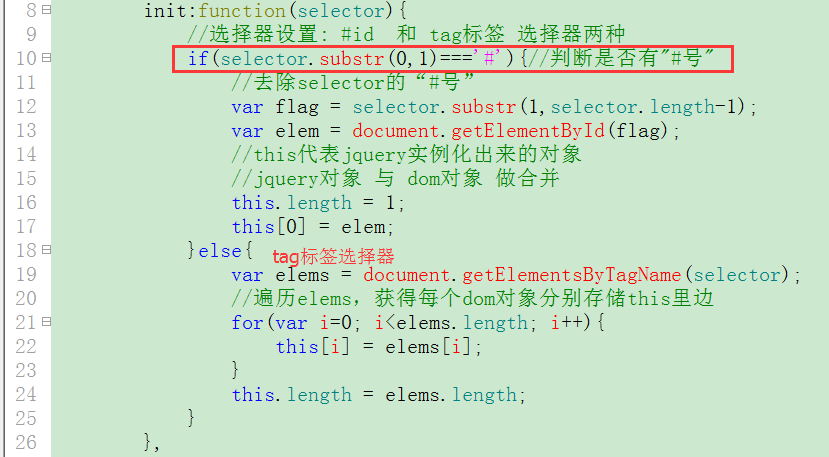
选择器：#id、tag标签

方法：css()、attr()、each()方法

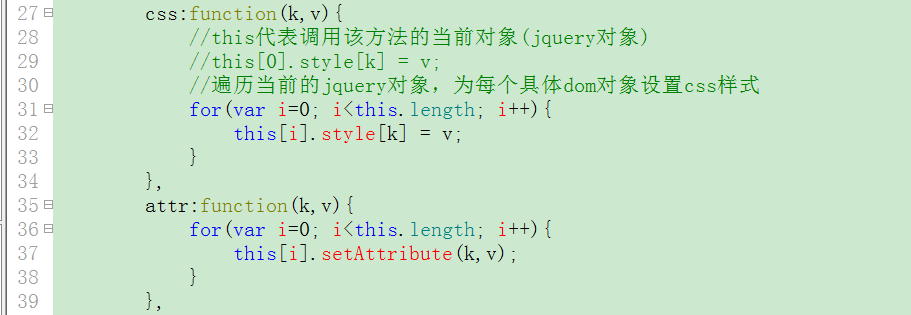
(在each()方法里边可以清楚知道jquery里边this如何代表dom对象)

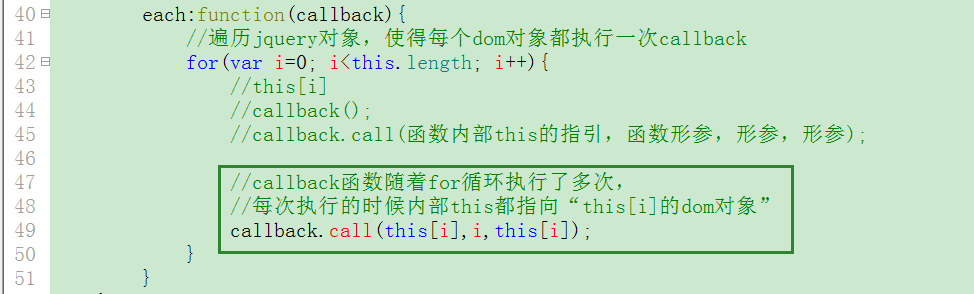


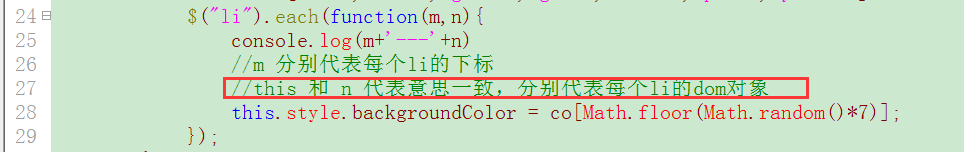
## 制作选择器



## 各种方法的封装







上图：this[i]-------指向------->this

i----------指向----------->m

this[i]-------指向------->n

学习”迷你版jquery”要认识到的地方：

1. jquery里边的大部分方法有“遍历机制”
2. jquery方法里边的参数this是代表dom对象

# 十八. jquery插件开发

## 1. 什么是jquery插件

jquery框架本身给我们提供了一些方法供使用。但是方法的数目是有限的，其不能任意满足我们对各种功能的需求。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。

开发、扩展jquery框架额外方法的过程就是“插件开发”

## 2. 两种形式丰富方法

① 给$.fn(给jquery对象)丰富

$.fn.成员 = 值;

$.fn.extend(json对象);

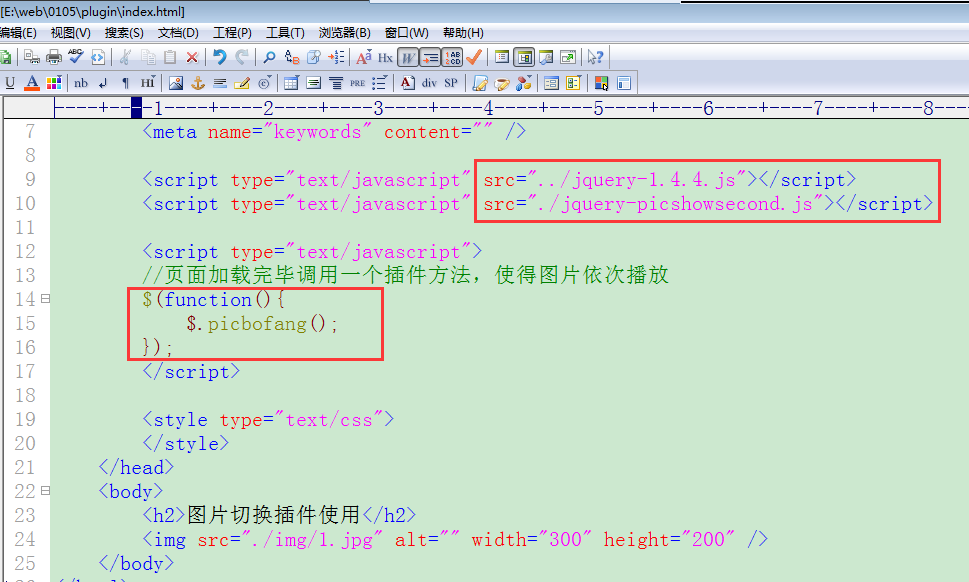
② 给$(给$对象)丰富

$.成员 = 值;

$.extend(json对象);

## 开发图片自动切换插件





## 4. 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

总结：

1. 地区三级联动

jquery+ajax+xml

$().find(选择器)；

1. 迷你版jquery框架开发

选择器：#id选择器、tag标签选择器

方法：css()/attr()/each()

目的：

① jquery内部方法的遍历实现机制

② jquery方法内部this关键字分别代表dom对象的理解

1. jquery插件开发

就是给 jquery对象 或 $对象 增加一些额外的方法。

① jquery对象增加方法：

$.fn.成员= 值;

$.fn.extend(json对象);

② $对象增加方法：

$.fn.成员= 值;

$.fn.extend(json对象);

作业：

1. 完成地区三级联动：城市-----切换-----》地区显示
2. 练习使用jquery成品插件(时间选取器、表单验证等)