# jQuery简介

jQuery是一个轻量级的"写的少，做的多"的JavaScript库

核心理念：write less，do more

jQuery 库包含以下特性：

HTML 元素选取

HTML 元素操作

CSS 操作

HTML 事件函数

JavaScript 特效和动画

HTML DOM 遍历和修改

AJAX

Utilities

## jQuery下载

有两个版本的 jQuery 可供下载：

①Production version - 用于实际的网站中，已被精简和压缩。

②Development version - 用于测试和开发（未压缩，是可读的代码）

这两个版本都可以从 [jQuery.com](http://jquery.com/download/) 下载

## jQuery语法

jQuery 语法是为 HTML 元素的选取编制的，可以对元素执行某些操作

基础语法是：**$(selector).action()**

"$"定义 jQuery

"selector"查询和查找 HTML 元素

jQuery 的 action() 执行对元素的操作

# jQuery的选择器

jQuery的选择器是jQuery操作的基础

选择器允许对 DOM 元素组或单个 DOM 节点进行操作



## 基础选择器

①**全局选择器**，All Selector("\*")

描述：选择所有元素

语法：$("\*")

注：由于使用\*号选择器获取的是全部元素，浏览器会执行缓慢，需要谨慎使用

②**标签选择器**，Element Selector("element")

描述：根据给定(html)标记名称选择所有元素

语法：$("element")

③**类选择器**，Class Selector(".class")

描述：选择给定样式类名的所有元素

语法：$(".class")

④**ID选择器**，ID Selector("#id")

描述：选择一个具有给定id属性的单个元素

语法：$("#id")

⑤**复合选择器**，Multiple Selector("selector1,selector2...")

描述：将每一个选择器匹配到的元素合并后一起返回

语法：$("selector1,selector2 ... selectorN")

## 属性选择器

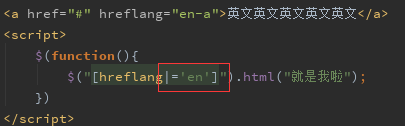
属性选择器,都是以"[..]"方括号来选择的

**①Attribute Contains Prefix Selector**

描述：选择属性值等于给定字符串或以该字符串为前缀+"-"的元素

语法：$("[attribute|='value']")

例如：

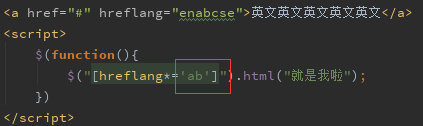


**②Attribute Contains Selector**

描述：选择属性值具有包含一个给定字符串的元素

语法：$("[attribute\*='value']")

例如：

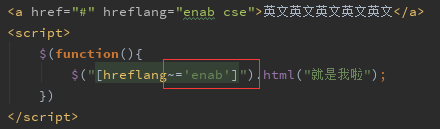


**③Attribute Contains Word Selector**

描述：选择属性值用空格分隔的值中包含一个给定值的元素

语法：$("[attribute~='value']")

例如：



**④Attribute Ends With Selector**

描述：选择属性值是给定值的元素

语法：$("[attribute='value']")

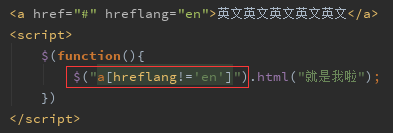
**⑤Attribute Not Equal Selector**

描述：选择不存在指定属性，或属性值不等于给定值元素

语法：$("attribute!='value'")

例如：注意需要加标签名,a[hreflang!='en']是a标签的hreflang属性值不等

于en的元素，而[hreflang!='en']是不存在en属性的元素

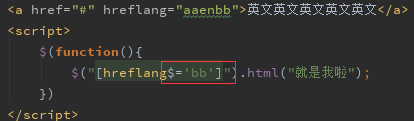


**⑥Attribute Ends With Selector**

描述：选择属性值是以给定值结尾的元素，比较是区分大小写的

语法：$("[attribute$='value']")

例如：

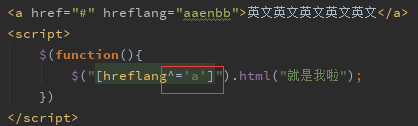


**⑦Attribute Starts With Selector**

描述：选择指定属性是以给定字符串开始的元素，跟上面的相反

语法：$("[attribute^='value']")

例如：

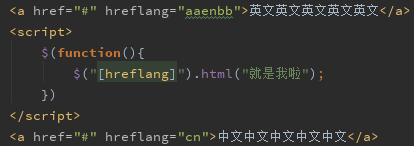


**⑧Has Attribute Selector**

描述：选择所有具有特定属性的元素，该属性可以是任何值

语法：$("[attribute]")

例如：只要具有hreflang属性的元素都会被选择



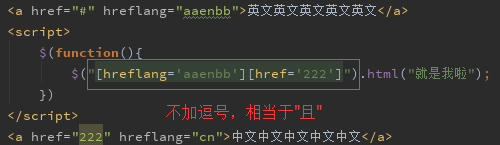
**⑨Multiple Attribute Selector**

描述：选择匹配所有指定的属性筛选器的元素

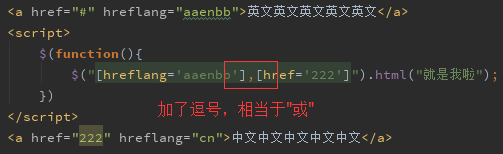
语法：$("[attribute1][attribute2] ... [attributeN]")

例如：注意区分逗号

不加逗号，相当于逻辑且，hreflang值为aaenbb且herf=222的元素



加了逗号，相当于逻辑或，hreflang=aaenbb或href=222的元素



## 基础过滤选择器

基础过滤选择器，都是以":"冒号来选择的

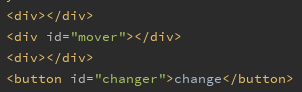
### :adnimated Selector

描述：选择正在执行动画效果的元素

语法：$(":animated")

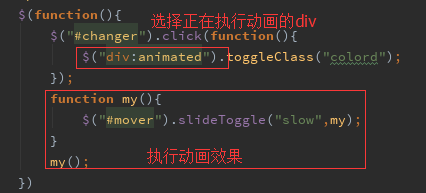
例如：

1、首先定义3个div，设置好样式。再定义一个按钮，用于设置点击事件



2、选择正在执行动画效果的div，将它的class设置为colord，

即将它的背景颜色改为浅蓝色(样式定义好了)



### :eq() Selector

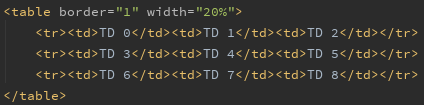
描述：在匹配的集合中选择索引值为指定值的元素

语法：①$("xx:eq(index)"),从0开始计数,选择匹配索引值为index的元素

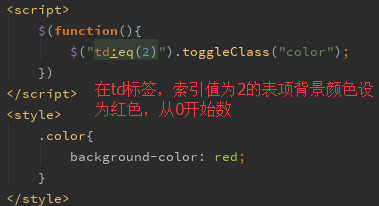
②$("xx:eq(-index)")选择从末尾开始计数第index个的元素

例如：

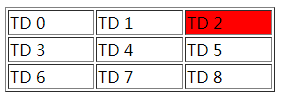
1、先准备一个3x3的表格，表项内容为td标签的索引



2、表示index=2的td标签的背景颜色设为红色



3、效果图：



### :even Selector

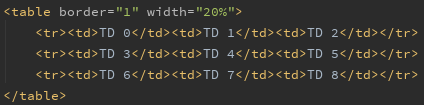
描述：选择索引值为偶数的元素，从0开始计数；改为odd，选择奇数的元素

语法：$(":even") $(":odd")

注意：因为从0开始计数,所以选择的是第1个元素、第3个元素...

例如：

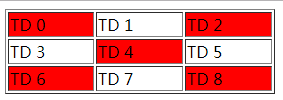
1、先准备一个3x3的表格，表项内容为td标签的索引



2、选择索引值为偶数的td标签



3、效果图



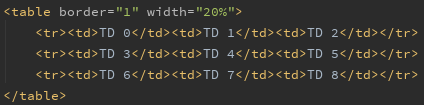
### :first Selector

描述：选择第一个匹配的元素

语法：$(":first")

例如：

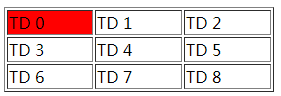
1、先准备一个3x3的表格，表项内容为td标签的索引



2、选择第一个的td标签



3、效果图



### :focus Select

描述：选择当前获取焦点的元素

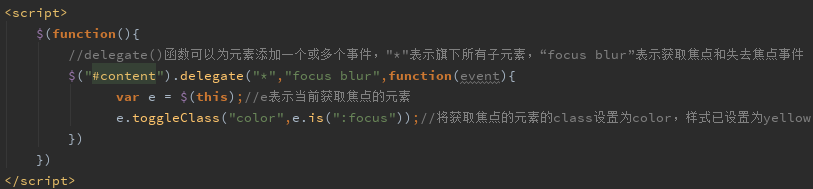
语法：$(":focus")

例如：

1、定义div标签和输入框...



2、设置获取焦点的元素，背景颜色设置为黄色



3、效果图







### :header Selector

描述：选择所有标题元素，想h1,h2,h3..等

语法：$(":header")

### :last Selector

描述：选择最后一个匹配的元素

语法：$(":last")

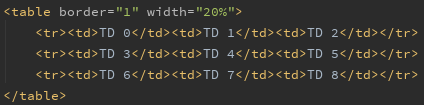
### :gt() Selector

描述:选择匹配集合中所有大于给定Index(索引值)的元素，注：不包括index

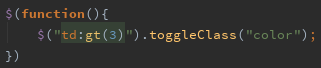
语法：$(":gt(index)")或$(":gt(-index)")

例如：

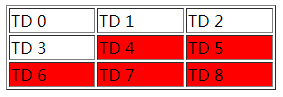
1、先准备一个3x3的表格，表项内容为td标签的索引



2、选择索引值大于3的td标签，注意不包含等于3的情况



3、效果图



### :lt() Selector

描述：选择匹配集合中所有索引值小于给定Index的元素，同样不包括index

语法：$(":lt( index )")或$(":lt( index )")



## 子元素过滤选择器

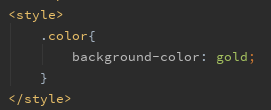
### :first-child Selector

描述：选择所有父级元素下的第一个子元素

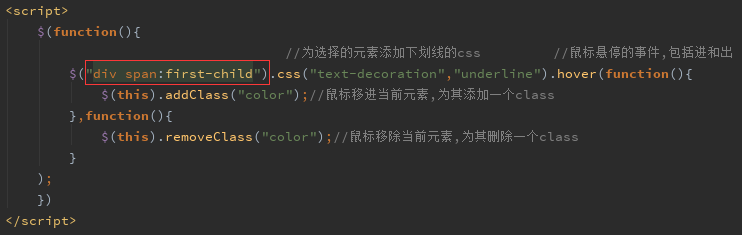
语法：$(":first-child")

例子：

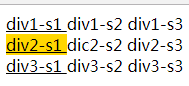
1、定义3个div，每个div里面包含3个span标签,并定义一个类的css样式

2、表示所有div下的第一个span标签(注意span标签要在div的包含内)



3、效果图



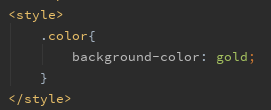
### :last-child Selector

描述：选择所有父级元素下的最后一个子元素

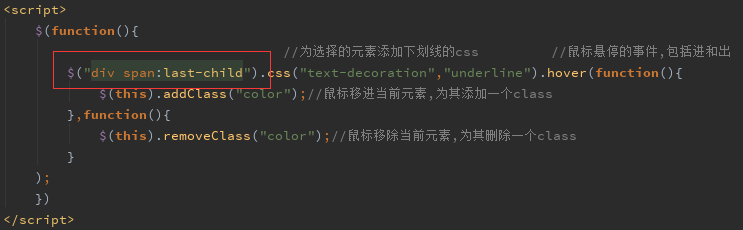
语法：jQuery(":last-child")

例子：

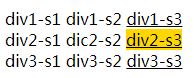
1、定义3个div，每个div里面包含3个span标签,并定义一个类的css样式

2、选择所有div标签下的最后一个span标签



3、效果图：



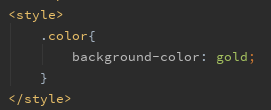
### :first-of-type Selector

描述：选择所有相同的元素名称的第一个兄弟元素

语法：$(":first-of-type")

例子：

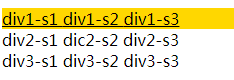
1、定义3个div，每个div里面包含3个span标签,并定义一个类的css样式

2、在所有div标签组成的集合中,选择第一个div



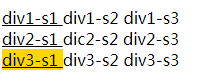
效果为：



3、在所有div中,旗下的子标签span组成的集合中,选择第一个span标签



效果图：



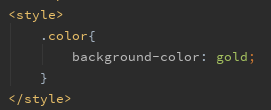
### :last-of-type Selector

描述：选择所有元素之间具有相同元素名称的最后一个兄弟元素

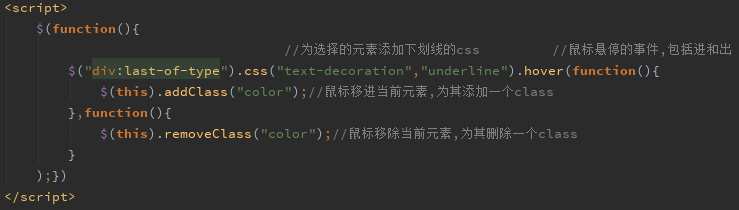
语法：$(":last-of-type")

例子：

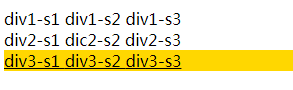
1、定义3个div，每个div里面包含3个span标签,并定义一个类的css样式

2、选择所有div标签组成的集合中的最后一个div



3、效果图：



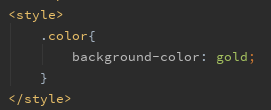
### :nth-child() Selector

描述：选择所有父元素的第n个子元素

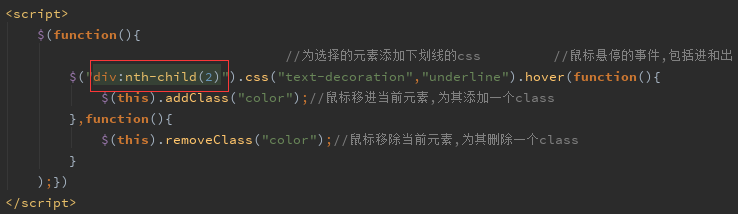
语法：$(":nth-child(index)")

例子：

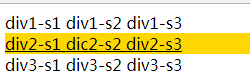
1、定义3个div，每个div里面包含3个span标签,并定义一个类的css样式

2、在所有div标签组成的集合中，选择第2个div



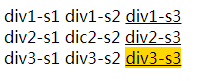
效果图：



3、在所有div标签下的span标签组成的集合中，选择第3个span标签



效果图：



## 内容过滤选择器

### :contains() Selector

描述：选择所有包含指定文本text的元素

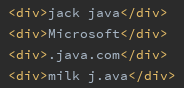
语法：$(":contains( text )")

注意：text是用来匹配的一个文本字符串，区分大小写，匹配的文本可以直接

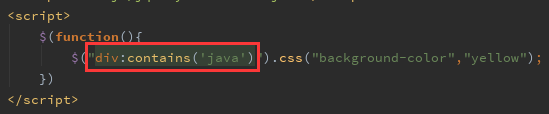
出现在该元素中，或在该元素的任何后代中，或两者兼有

例子：

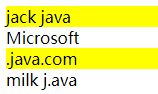
1、定义4个div标签，分别为这4个div添加文本



2、引入jQuery后，选择包含"java"文本的div标签，将其渲染成黄色



3、效果图



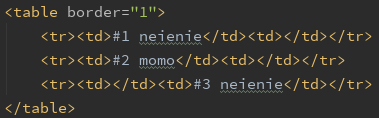
### :empty Selector

描述：选择所有没有子元素的元素(包括文本节点)

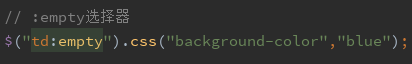
语法：$(":empty")

例子：

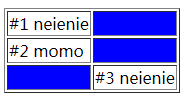
1、定义一个2x3表格，其中3个有文本，3个为空白



2、选择这些空白文本的td标签，将其渲染成蓝色



3、效果图



### :has() Selector

描述：选择元素，它至少包含指定选择器匹配的一种元素

语法：$(":has( selector )")

例子：

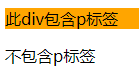
1、定义2个div标签，一个包含牌p标签，一个不包含



2、将包含p标签的div渲染为橘色



3、效果图



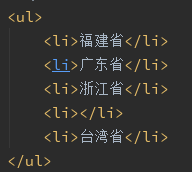
### :parent Selector

描述：选择所有含有子元素或者文本的父级元素

语法：$(":parent")

例子：

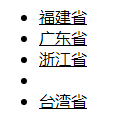
1、创建无序列表，其中4个有写文本内容，1个没写文本内容



2、选择有文本的li标签，为他们加上下划线



3、效果图



## 表单选择器

表单选择器用来选取表单里面的组件，格式为：“:”

①:button Selector

描述：选择所有按钮元素和类型为按钮的元素

语法：$(":button")

②:checkbox Selector

描述：选择所有类型为复选框的元素

语法：$(":checkbox")

③:checked Selector

描述：匹配所有勾选的元素

语法：$(":checked")

④:disabled Selector

描述：选择所有被禁用的元素

语法：$(":disabled")

⑤:enabled Selector

描述：选择所有可用的(未被禁用的)元素

语法：$("enabled")

⑥:file Selector

描述：选择所有类型为文件(file)的元素

语法：$(":file")

⑦:focus Selector

描述：选择当前获取焦点的元素

语法：$(":focus")

⑧:image Selector

描述：选择所有图像类型的元素

语法：$(":image")

⑨:input Selector

描述：选择所有input、textarea、select、button元素

语法：$(":input")

⑩:password Selector

描述：选择所有类型为密码的元素

语法：$(":password")

⑾:radio Selector

描述：选择所有类型为单选框的元素

语法：$(":radio")

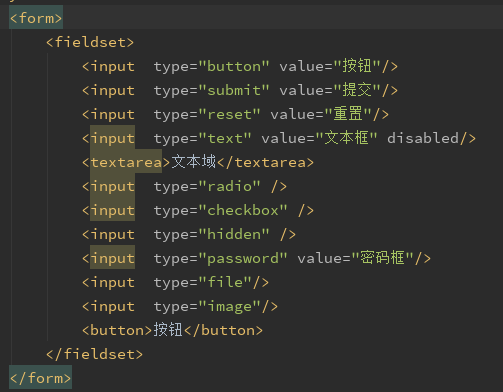
⑿:submit Selector

描述：选择所有类型为提交的元素

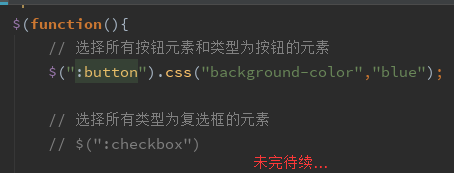
语法：$(":submit")

例子：

1、定义一个表单，并组装好各组件



2、定义脚本文件，使用表单选择器



3、效果图



## 层级选择器

### 子元素选择器

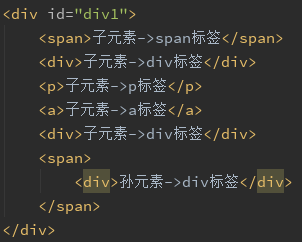
描述：选择所有指定父元素中指定的直接子元素

语法：$(" parent > child")

注：parent是任何有效的选择器，child是用来筛选子元素的选择器

例子：

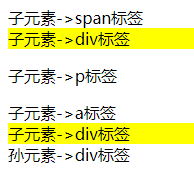
1、在一个div中定义多个标签，包含span、div、a、p



2、选择外层div下的div子标签，添加点击事件和css样式



3、效果图



### 后代选择器

描述：选择给定祖先元素的所有后代元素

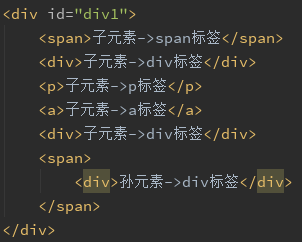
语法：$(" ancestor descendant")

注：ancestor指任何有效的选择器

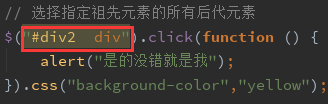
descendant指一个用来筛选后代元素的选择器

例子：

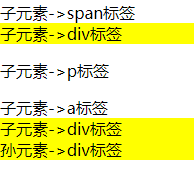
1、在一个div中定义多个标签，包含span、div、a、p



2、选择外层div元素下的所有后代div元素



3、效果图



### 相邻选择器

描述：选择所有紧邻在" prev "元素后的" next "元素

语法：$(" prev + next")

注：prev指任何有效的选择器，next指紧跟在"prev"后的选择器

例子：

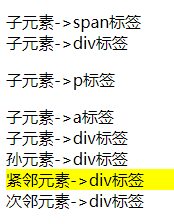
1、定义外层div标签，并定义其子元素和相邻元素



2、选择"#div3"的紧邻div元素



3、效果图



### 兄弟选择器

描述：匹配" prev "元素之后的所有兄弟元素，它们具有相同的父元素

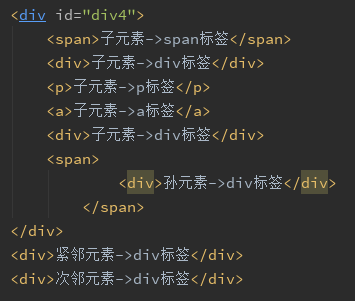
语法：$(" prev ~ siblings")

注： prev：任何有效的选择器

siblings：一个选择器来过滤第一选择器以后的兄弟元素

例子：

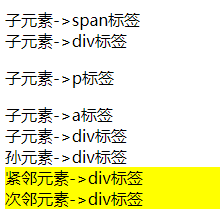
1、定义外层div标签，并定义其子元素和相邻元素



2、选择" #div4 "的同级兄弟元素



3、效果图



# jQuery的事件

## 鼠标事件

①.click()：点击事件（鼠标点击一次触发事件）

②.dblclick()：双击事件(鼠标快速点击2次触发)

③.mousedown()：鼠标按下（鼠标一按下就触发）

④.mouseup()：鼠标按键被释放

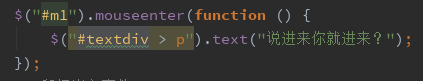


⑤.mouseenter()：鼠标进入元素

⑥.mouseleave()：鼠标离开元素

⑦.mouseover()：鼠标进入元素（支持事件冒泡）

⑧.mouseout()：鼠标离开元素（支持事件冒泡）



⑨.hover：鼠标悬停和离开(鼠标进入和离开分别触发)

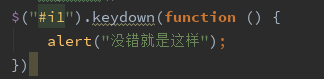
⑩.mousemove()：鼠标在元素内移动

## 键盘事件

①.keydown()：键盘按下触发

②.keyup()：键盘弹起触发

③.keypress()：键盘按下触发,着重按下键盘输入了哪个字符



如果3个键盘事件一起存在于同一个元素，keyup()事件不会触发

先触发keydown()事件再触发keypress()事件

keypress事件会在按下键盘按键时触发。它与keydown事件类似，但keypress着重的是按下该键输入了哪个字符(只有能够打印字符的键才会触发keypress)，keydown着重的是按下了哪个键(按下任何键都可触发keydown)。对于诸如Ctrl、Alt、Shift、Delete、Esc等修饰性和非打印键，请监听keydown事件

## 浏览器事件&文档加载事件

1、浏览器事件

①.scroll()，元素发生滚动时触发



②.resize()



2、文档加载事件

js中ready和onload事件：

\*ready事件，在DOM结构绘制完成之后就会执行，保证了即使有大量媒体

文件未加载，也可以执行js

\*load事件，必须等到网页中所有内容全部加载完毕之后才被执行

文档加载过程：

①解析HTML结构

②加载外部脚本和样式表文件

③解析并执行脚本代码

④构造HTML DOM模型 // ready()触发

⑤加载图片等外部文件

⑥页面加载完毕 // load()触发

jQuery文档加载的3种写法：

①$().ready(function(){..})

②$(document).read(function(){..})

③$(function(){..})

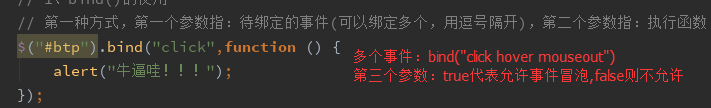


## 绑定事件处理器

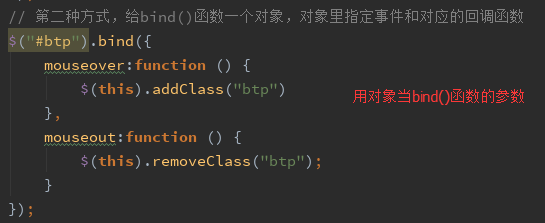
### .bind()与.unbind()

① .bind()函数可以为指定的元素绑定一个或多个事件，它有2种写法：

1、语法：bind( eventType ，[eventData ], handler )



2、语法：bind({event:fun(),event:fun(),...})



②.unbind()，解除bind()函数绑定的事件



### .delegate()与.undelegate()

①.delegate()，语法：delegate( selector, eventType, handler )

delegate() 为指定的元素（属于被选元素的子元素）添加一个或多个事件处

理程序，并规定当这些事件发生时运行的函数。使用 delegate() 的事件处

理程序适用于当前或未来的元素（比如由脚本创建的新元素）

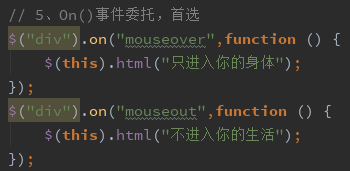


②.undelegate(),解除delegate()绑定的1个事件或多个事件

### .on()与.off

①.on()，语法：$(selector).on(event,childSelector,data,function)

on()的效果和bind()、delegate()差不多，但是首推使用on()



②.off()就是取消On()委托的事件，可以取消一个，也可以取消多个

### .one()

.one()方法为被选元素附加一个或多个事件处理程序，并规定当事件发生时运行的函数。当使用 one() 方法时，每个元素只运行一次事件处理器函数

语法：one( events [, data ], handler )



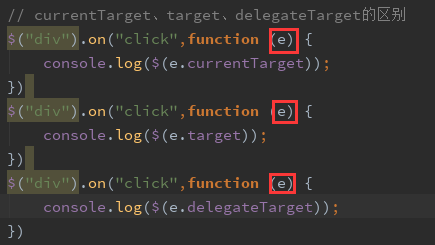
## 事件对象

### currentTarget、target、delegateTarget

event.currentTarget 返回其监听器触发事件的节点

event.target 指明哪个 DOM 元素触发了该事件

event.delegate 返回负责绑定当前被调用的事件处理函数的DOM元素



### pageX、pageY

event.pageX： 鼠标指针的位置，相对于文档的左边缘

event.pageY： 鼠标指针的位置，相对于文档的上边缘



### type

event.type ： 事件的类型



### preventDafault()、stopPropagation

event.preventDafault() :阻止事件的默认行动

比如，a标签会跳转到指定Url页面，使用这个函数就可以阻止

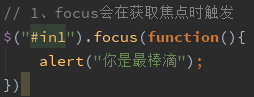


stopPropagation是阻止事件冒泡。

## 表单事件

### focus()

当元素获得到焦点的时候，就会触发



### blur()



### change()



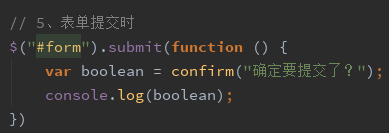
### select()





### submit()

submit()只能用在form表单标签，当发起表单提交请求时触发



# jQuery的DOM属性

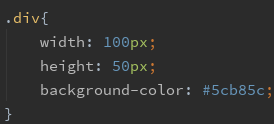
## .addClass()和.removeClass()

①addClass为匹配元素添加指定的样式类名

语法：addClass( className )



需要事先在<style>标签中定义 class="div"的样式



②removeClass移除匹配元素一个、多个或全部的样式

语法：remove( [className] )

移除一个样式，参数就写一个；移除多个，参数就写多个，空格隔开

不带参数就是移除所有的样式



## .attr()和.removeAttr()

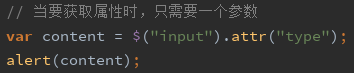
①.attr有2种使用方式：一是添加属性，二是获取属性值

当用来为匹配元素添加属性时，如果是一个属性，直接写：参数名+参数值

如果是多个属性，需要将 参数+参数值 封装成一个对象，当做参数



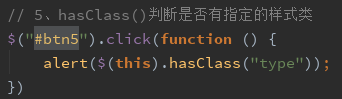
如果用于获取属性值，只需要一个参数，就是匹配元素的参数名



②removeAttr()移除其属性

## .hasClass()

确实匹配元素是否有指定的样式类，有即返回true，否则返回false





因为此时的button并没有一个类名为type的样式类，所以返回false

## .html()

html()也有2种用法：一是获取匹配元素的Html内容，二是设置其html内容

<p>111</p>，像“111”这种的就是html()可以读取到的内容





## .prop()和.removeProp()

prop()有2种使用方式：一是获取属性的值，二是添加属性

是不是发现，它和attr()很类似，确实2个的作用是一样的，但是：

处理HTML元素本身就带有的固有属性，使用prop方法

处理HTML元素自自定义的DOM属性，使用attr方法

剩下的语法和attr()一样，同理removeProp()也是一样的

## .toggleClass()

在匹配元素添加或删除一个或多个样式类，如果此元素存在该样式类则删除，如果此元素不存在该样式类则添加



## val()

val()获取匹配元素的当前值或者设置该元素的值

与html()不同的是:  
html()一般是获取的是，像<p>aaa</p>夹在2个标签的html内容

val()一般是获取，诸如<input value="bb">这样的值



# jQuery的DOM操作

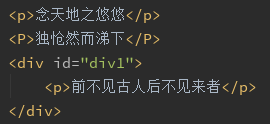
## DOM包裹

### .wrap()

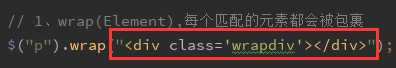
描述：在每个匹配的元素外面包裹一个html元素

语法：wrap(Element)

例如：1、定义p元素



2、在每个p元素外面包裹一层div



3、结果：



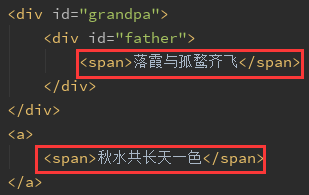
### .unwrap()

描述：将匹配元素的父级元素删掉，保留自己和兄弟元素的原本位置

语法：unwrap()

例子：

1、定义2个<span>并将其放在<div>和<a>之中

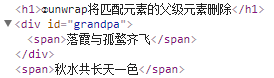


2、删除<span>的父级元素



3、结果：

父级元素都被去掉，父级以上元素就不会被删掉



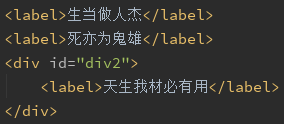
### .wrapAll()

描述：所有匹配元素集中起来，在它们外面包裹Html元素

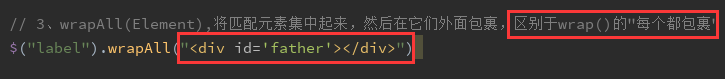
语法：wrapAll(Element)

例子：

1、定义label标签

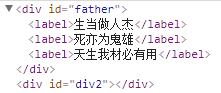


2、将所有label标签集中起来，包裹一层div



3、效果：

就连已经在<div>标签下的<label>天生我材必有用</label>也会被集中到一起



### .wrapInner()

描述：在匹配元素的内容外包裹一层html元素，注意是内容

语法：wrapInner(Element)

例子：

1、定义一个button标签



2、在button标签的内容外包裹一层div



3、效果图



## DOM插入现有元素内

### .append()和.appendTo()

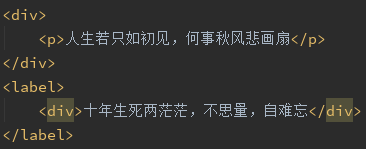
描述：都是将参数嵌入到匹配元素内部的末尾，写法相反

语法：$(目标).append(Element)

$(Element).appendTo($(目标))

例子：

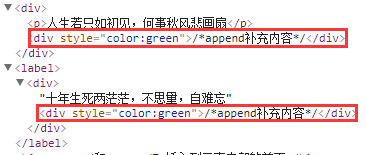
1、定义如下的div和label标签



2、将新参数嵌入到div标签内的末尾，注意写法不一样



3、结果



### .prepend()和.prependTo()

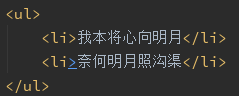
描述：都是将参数嵌入元素内部的最前面，写法相反

语法：$(目标).prepend(Element)

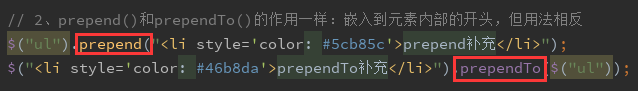
$(Element).prependTo($(目标))

例子：

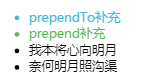
1、定义无序标签ul



2、将参数嵌入到ul的最前方



3、结果



### .html()和.text()

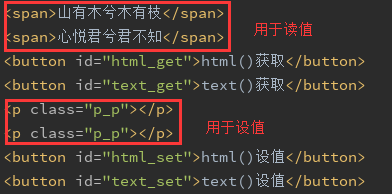
描述：都既可以获取匹配元素的内容，又可以设置匹配元素的内容

但html()只获取所有匹配元素的第一个匹配元素的内容,text()获取全部

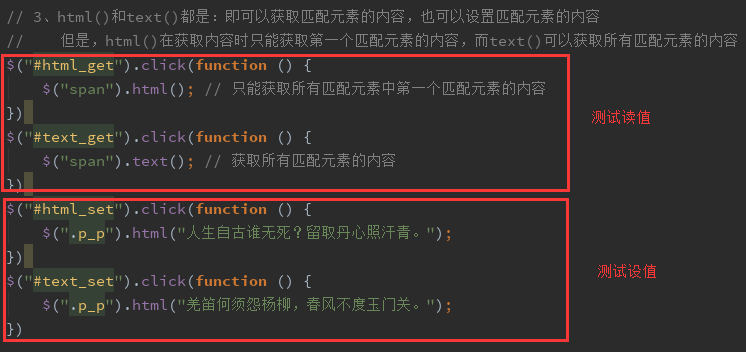
如果在设置匹配元素的内容，html()和text()没有区别，设置所有匹配元素

例子：

1、创建span标签和p标签，用于读值和设值，在创建4个按钮，作用如图



2、分别测试html()和text()的读取和设值



3、效果展示去集成开发环境看

## DOM插入现有元素外

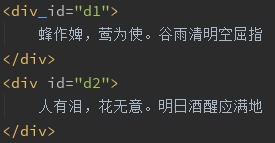
### .clone()

描述：创建一个匹配元素的深度拷贝副本

语法：$(selector).clone()

例子：

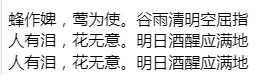
1、创建2个div



2、拷贝下面的div内嵌到上面的div中



3、效果



### .after()和.insertAfter()

描述：将参数外嵌到匹配元素的末尾，注意是外嵌

语法：$(Selector).after(Element)

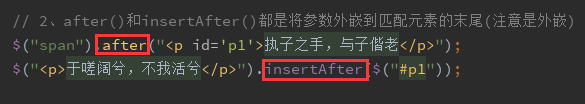
$(Element).insertAfter($(Selector))

例子：

1、创建一个span标签



2、在span标签后面添加2个p标签



3、结果



### .before()和.insertBefore()

描述：将参数外嵌到匹配元素的前面，注意是外嵌

语法：$(Selector).before(Element)

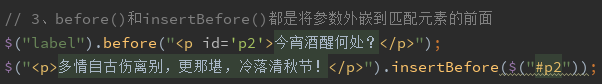
$(Element).insertBefore($(Selector))

例子：

1、创建一个label标签



2、外嵌2个p标签到label标签前面



3、结果



## DOM移除和替换

### .detach()和.remove()

描述： detach()：从DOM中去掉所有匹配的元素

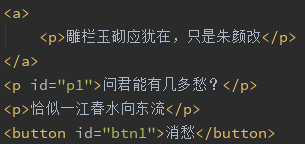
remove()：将匹配元素集合从DOM中删除并移除其事件和数据

语法： $(Selector).detach( [detail] )

$(Selector).remove()

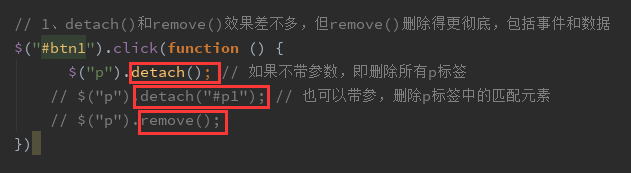
例子：

1、定义几个P标签，并定义一个按钮



2、删除dom中的所有P标签，不带参数默认删除所有，如果带参数，则

只删除匹配参数的P标签



### .empty()和.unwrap()

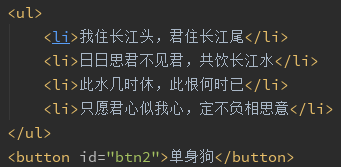
描述： .empty()：从DOM中移除集合中匹配元素的所有子节点

.unwrap()：将匹配元素结合的父级元素删除，保留自身和兄弟结点

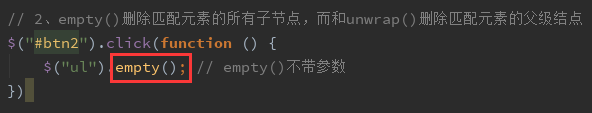
语法： $(Selector).empty() 不带参数，unwrap()前面已经提到过

例子：

1、定义ul标签和一个按钮



2、删除ul标签下的所有子节点



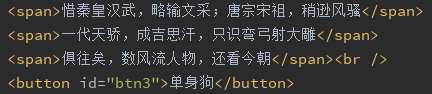
### .replaceAll()

描述：用参数去替换掉匹配的所有元素

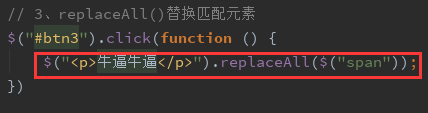
语法：$("参数").replaceAll($(Selector))

例子：

1、定义3个span标签



2、用p标签去替换span标签



# jQuery的CSS操作

## .css()

描述: 获取匹配元素集合中第一个元素的样式属性的值、设置每个匹配元素的

一个或多个css属性

语法: $( Selector ).css("属性")-----获取css属性值

$( Selector ).css("属性","属性值")---设置一个css样式

$( Selector ).css({对象})----设置多个css样式

## height()、innerHeight()、outerHeight()

.height(): 获取或设置匹配元素的height属性

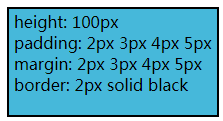
.innerHeight() :获取匹配元素的height+padding(包括上下内边距)

.outerHeight() :获取匹配元素的height+padding+border(上下边距)

.outerHeight(true):获取的height+padding+border+margin(上下边距)

例如：

1、一个div的各项属性如下：



2、height()返回100--------height的值

innerHeight()返回106---height的值+padding的上边距(2)和下边距(4)

outerHeight()返回110---height的值+padding+border的上下距离(2)

outerHeight(true)返回116--height + padding + border + margin

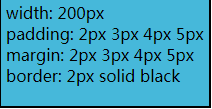
## width()、innerWidth()、outerWidth()

width() : 获取匹配元素的width属性值

innerWidth() : 获取匹配元素的width+padding的左右边距

outerWidth() : 获取匹配元素的width+padding+border的左右边距

outerWidth(true) : 获取width+padding+border+margin的左右边距

跟上面算高度的思想是一样的

## offset()、position()

offset() :获取或设值匹配元素的坐标，此坐标相对于document对象

position() :获取或设值匹配元素的坐标，此坐标相对于最近的祖先元素

(祖先元素首先需要被定位过)

## scrollLeft()、scrollTop()

scrollLeft() : 获取或设置匹配元素的水平滚动条的位置

scrollTop() : 获取或设置匹配元素的垂直滚动条的位置

# jQuery的遍历和过滤

## 过滤

### eq()、filter()

.eq() : 在匹配元素集合中选择符合索引值的元素(仅能选取一个)

从头算起，索引值从0开始算；从尾算起，索引值从-1开始算



filter() :在匹配元素集合中选择符合要求的若干个元素，此方法接收

函数、正则表达式、选择器 作参数



### first()、last()

first() : 匹配元素集合中的第一个元素

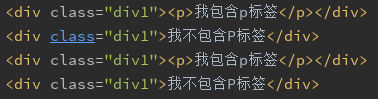
last() : 匹配元素集合中的最后一个元素

### has()

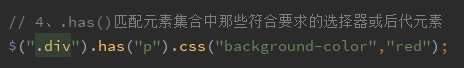
描述：筛选匹配元素集合中符合参数要求的选择器或后代元素

例子：

1、div元素集中，若干个包含p标签



2、选择包含P标签的div

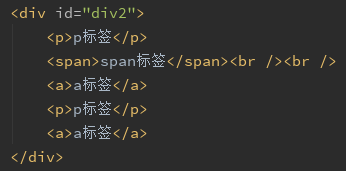


### is()

描述：判断匹配元素是否符合参数过滤要求，符合则返回true，反之返回false

例子：

1、创建一个div标签，div中包含其他标签



2、给div一个点击事件，如果点击对象是P标签，则弹框



### not()

描述：在匹配元素集合中选择不符合参数要求的若干元素

例子：

在一个由类为"div3"的元素集合中，选择索引不是偶数的其他元素

even代表偶数。odd代表奇数



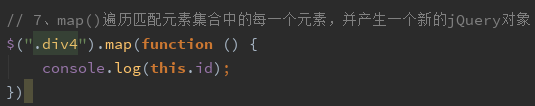
### map()

描述：遍历匹配元素集合中的每一个元素，并产生一个新的jQuery对象

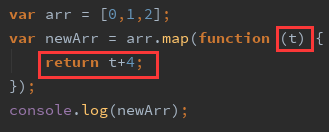
参数为一个函数

例如：

遍历一个类名为"div4"的元素集合，分别读出该集合中每一个元素的id属性



也可以遍历一个数组，将数组的每个元素值+4，它会返回一个新的数组



### slice()

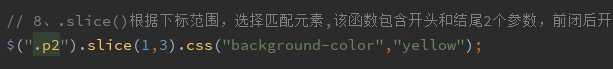
描述：根据下标范围，选择匹配元素

语法：slice( a,[b])如果只有一个参数，表示选择从索引a开始到结束的元素

如果是2个参数，表示选择区间[a,b)的元素

例子：

选择类名为"p2"的元素集合中索引为1~2的元素



## 普通遍历

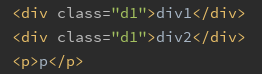
### .add()

描述： 将参数指定的元素添加到已选择的元素集合中

语法： $( Selector ).add( param )，支持大部分参数如：selector、html

例子：

1、现在有2个div和1个p



2、要把div和p一起做相同的样式处理，就可以使用add()方法

表示选择类名为d1的div元素和p元素，一起做css处理



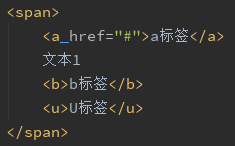
### .contents()

描述： 获取匹配元素集合中每个元素的子元素，包括其文本和注释节点

和child()方法相似，都获取后代元素，但contents()还能获取文本

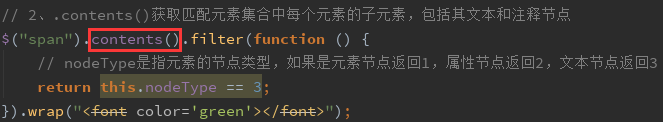
例子：

1、span标签里面包含a标签、b标签、u标签和文本



2、content()方法可以遍历span标签里面所有元素，判断如果是文本元素，

为其包裹上一层绿色的font标签



### .end()

描述： 结束当前链条中的最近的筛选操作，并将匹配元素集还原为之前的状态

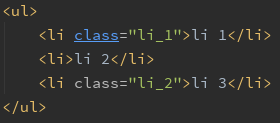
意思说：比如正在对一个元素集合里面的元素做过滤操作，当完成其中

一个元素的操作，可以使用end()方法让元素集返回到最初状

态，而不用重新匹配元素集再过滤

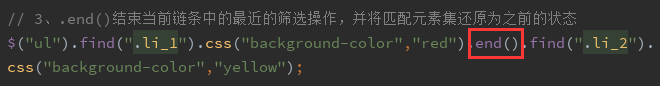
例子：

1、如下的ul标签，先对li 1操作，再对li 3操作



2、完成第一个li标签的样式后，不用重新去选择ul标签，直接使用

end()方法就可以回到选择ul标签的初始状态，继续操作li标签



## 树遍历

树遍历，不单可以遍历自己，也可以遍历子元素

### childrean()、find()

描述：children()遍历匹配元素寻找它的直接子元素(儿子关系)

语法： .children()寻找所有直接子元素

.childrean( Selector )寻找指定的直接子元素

例子：



描述： find()遍历匹配元素寻找它的所有子元素(儿子、孙子、祖孙...)

语法： .find( [ selector/html/object ] )

例子：



### next()、nextAll()、nextUntil()

①.next() : 获取匹配元素集合中每个元素紧邻的同辈元素，如果提供选择器

则紧邻的同辈元素符合该选择器时，才会被返回

语法：.next( [Selector] )



②.nextAll() : 获取匹配元素集合中每个元素的所有跟随的同辈元素，如果提

供选择器，筛选符合选择器的同辈元素集合

语法 : .nextAll( [Selector] )



③.nextUntil() : 获取匹配元素到指定参数之间的符合要求的兄弟元素集合

可以无参、有一个参、有两个参这3种情况

语法 : .nextUntil()

.nextUntil( Selector )

.nextUntil( Selector , fliter)



### prev()、prevAll()、prevUntil()

①.prev() : 获取匹配元素集合中每个元素紧邻的前一个同辈元素，可以用选择

器Selector筛选，与上面的next()相反(找后一个同辈元素)

语法: prev( [Selector] )

例子: 如果加了参数，紧邻的同辈元素符合参数要求才会被返回



②.prevAll() : 获取匹配元素集合中每个元素前面的所有同辈元素，可以用选

择器筛选，与上面的nextAll()相反(找后面所有同辈元素)

语法：prevAll( [Selector] )

例子：找到匹配元素前面的，类名为"dd1"的所有同辈元素



③.prevUntil() : 选择匹配元素集合中每一个元素到参数1指定的元素之间

(不包含头和尾)的符合参数2要求的所有同辈元素

与上面的nextUntil()相反(它是往后找)

语法 : prevUntil( [Selector] , [filter])

例子 : 从id="start"的元素开始找，到id="div5"之间符合id="div4"的

同辈元素



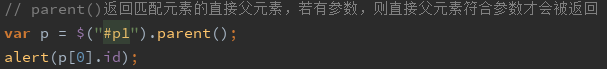
### parent()、parents()、offsetParent()、parentsUntil()

①.parent():获取匹配元素集合中每个元素的直接父元素，可以使用选择器进行

筛选，仅当符合选择器的直接父元素才会被返回

语法 : parent() 返回匹配元素的直接父元素

parent( Selector ) 返回匹配元素符合Selector的直接父元素

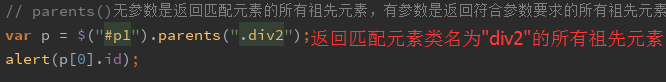


②.parents() : 获取匹配元素集合中每个元素的所有祖先元素，当使用选择器

进行筛选时，选择符合选择器要求的祖先元素

语法 : parents() : 选择匹配元素的所有祖先元素

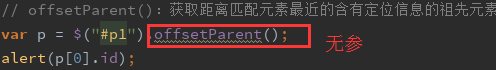
parents( Selector ):选择匹配元素符合Selector的所有祖先元素



③.offsetParent() : 获取距离匹配元素最近的含有定位信息的祖先元素

所谓定位是指css中的position属性被定义

语法： offserParent()



④.parentUntil() : 获取匹配元素直至参数指定的父级元素之间且满足参数要

求的祖先元素集合

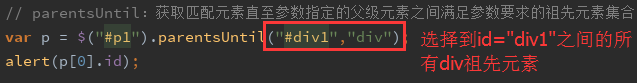
语法 : 类比上面的nextUntil()方法

.parentUntil():选择匹配元素所有的父元素

.parentUntil(Selector)选择自身到Selector之间的祖先元素（不包含首尾）

.parent(Selector,filter)选择自身到Selector之间的符合filter要求的

所有祖先元素(不包含本身和Selector)

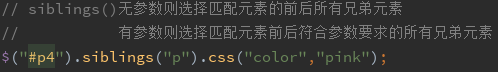


### siblings ()

描述：获取匹配集合中每个元素的兄弟元素(包括前和后)，提供可选的选择器

语法：siblings( [Selector] )

例子：siblings() = nextAll() + prevAll()



### closest()

描述：从匹配元素本身开始，逐级向上级元素匹配，返回最先匹配的祖先元素

语法： .closest( selector ),必写参数，表示哪一种类型的祖先元素

例子：

如果.closest()的参数是自身元素的类型，则匹配的就是自己，因为是从

自身开始匹配的

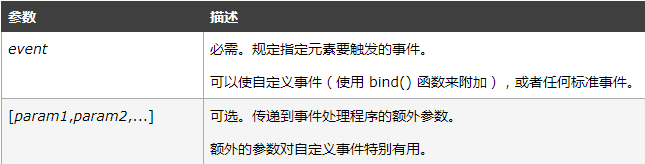


# jQuery的函数(补充)

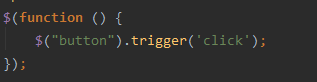
## trigger()

描述：trigger()触发被选元素的指定事件类型

语法：$(selector).trigger(event,[param1,param2...])



例子：触发<button>标签的点击事件：



## delegate()

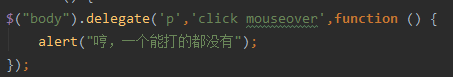
描述：delegate()为指定的元素添加一个或多个事件处理程序，并规定当这些事

件发生时运行的函数

语法：$(selector).delegate(childSelector,event,data,function)



例子：为body元素下的p元素绑定点击事件和鼠标移动事件



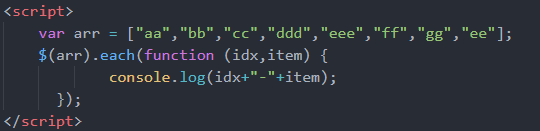
## each()

描述：each()为每个匹配元素规定运行的函数

语法：$(selector).each(function(index,element))



例子：

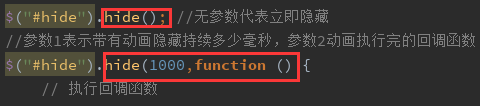


# jQuery特效

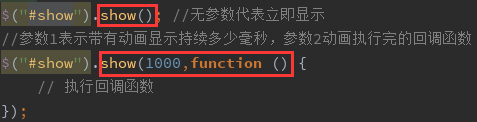
## 隐藏与显示

隐藏和显示特效一共有3种方法：

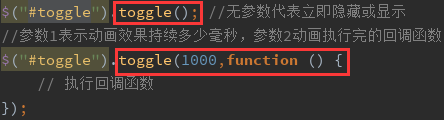
1、hide():隐藏元素



2、show()：显示元素



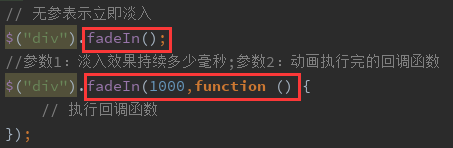
3、toggle()：隐藏或显示元素



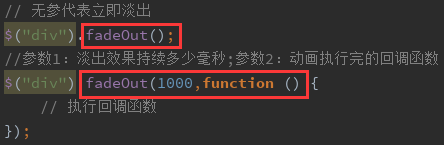
## 淡入淡出

淡入淡出一共有4种方法：

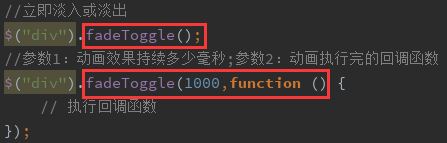
①fadeIn():将元素淡入



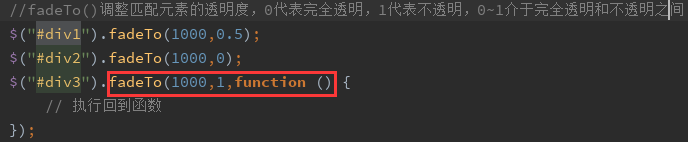
②fadeOut()：将元素淡出



③fadeToggle()：元素淡入或淡出



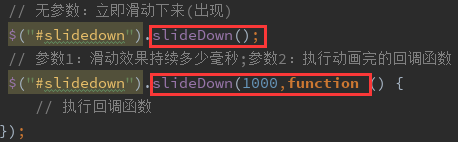
④fadeTo()：调整元素的透明度



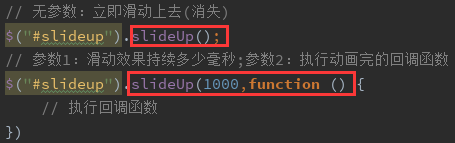
## 滑动

滑动特效有3个方法：

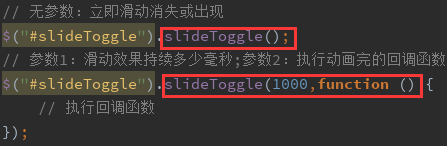
①slideDown()：让元素滑动出现



②slideUp()：让元素滑动消失



③slideToggle()：让元素滑动消失或出现



## 自定义特效

### .animate()

描述：根据一组css属性值，执行自定义的动画效果

animate()会将元素从一个状态改变为另一个状态,css属性值是逐渐改

变的，所以可以表现出动画效果

注：

只有数字值才可创建动画（比如 "margin:30px"），字符串值无法创建动画

（比如 "background-color:red"）

使用 "+=" 或 "-=" 来创建相对动画（relative animations）

语法：.animate([css],[time],[access],[function]),接收4个参数

css ：一个或一组css样式(多个css样式用对象表示)

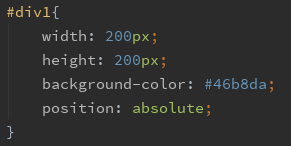
time : 动画执行的持续时间，单位是毫秒

accsess: 动画执行的过渡类型：线性"linear"或曲线性"swing"

function:回调函数

例子：

1、定义一个div，并配好样式

2、为该div创建2个动画效果，动画执行顺序按照代码声明顺序

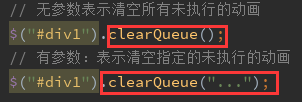
一个动画执行完成才去执行下一个动画，以此类推



### .clearQueue

描述 : 将当前动画执行完并清空后续正在等候执行的动画

例子：stop()只适用于动画，而clearQueue() 能够清除任何排队的函数



### .delay()

描述：在当前动画执行结束后，设置一个延迟，暂缓下一个动画的执行

例子：

本次动画结束后，5S后，才会去执行下一个动画



### .finish()

描述 : 停止所有动画执行(包括正在执行和等待执行)，将元素跳转到所有动画

执行后的效果



### .stop()

描述 : 立即停止当前正在执行的动画,执行下一个动画

0