**jQuery课程二**

**第一章**

1. 选择元素
2. $() = jQuery（）

这是选择元素的操作，二者是等效的，之所以使用jQuery是因为$在其他的语言中有可能被使用了造成冲突，使用jQuery来等效替换是解决冲突的其中一种方法，因为在jQuery中还提供了专门的解决冲突的方法。

1. 就绪函数

$（function（）{//开始写代码}）——就和onload函数是一样的效果，只是一个是原生js中的用法，一个是jQuery中的用法而已。

1. css（）——设置样式。

使用这个方法可以对元素设置样式，设置的样式直接当作参数传进去，如果不传参就是获取样式属性的值。（**五星级**）

1. jQuery可以省略原生js的循环操作，例如原生js中需要将选出来的一组元素遍历才能对其进行操作，但是jQuery默认直接对这一组符合条件都的元素直接操作了，省去了遍历的过程，如果需要遍历那就使用each方法来实现。
2. 使用jQuery的关键点就是理解“jQuery方法函数化”，在使用的时候，方法就是像使用某一种方法一样进行调用。
3. 理解jQuery和JavaScript的关系

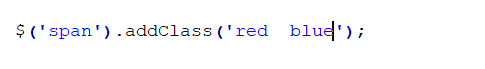
在使用jQuery的时候是可以使用JavaScript语句的，但是需要注意的是不能讲js语句和jQuery的语句进行混用。一个语句中，必须是完全是jQuery语句或者是JavaScript语句，而不能是将两种语句进行混合使用；如果是特护情况的话，那必须要使用get（）方法来进行转换。get（）前面的内容是jQuery的内容，后面就是原生js的内容。

1. 在jQuery中，this还是一个原生的JavaScript的内容，因此，如果是需要使用this的话，那么就需要使用 **$(this)** 来进行转换，否则就会报错。
2. jQuery中的值的设置或者获取是通过方法的参数个数决定的。如果在CSS（）方法中传入一个参数，那么就是获取；如果传入两个参数（一个是属性，一个是值）的话，那就是设置相应的值。

html（）方法中传入参数就是将传入的参数添加到选中元素中，如果为空就是获取当期元素的内容。很多的方法都有这样的特性。

1. 操作属性的方法attr（）：在使用这个方法的时候，如果传入一个参数那么就是获取属性，如果传入一个属性一个值就是进行属性的设置。
2. class的操作
3. 添加、删除class

添加：addClass（）方法，参数就是要添加的class，可以传入多个参数中间用空格隔开，有自动去重的功能。



删除：removeClass（）方法，参数就是需要删除的class，可以传入多个参数。如果传入的参数没有找到，那就不会产生其他效果。

toggleClass（）方法，自行调用添加或者删除方法，比较智能。如果在参数中添加的内容和已有的是相同那么就执行删除操作，否则就执行添加的操作。

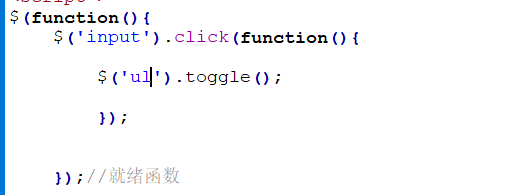
1. 显示隐藏show、hide（**重要**）

1、显示隐藏方法

显示：show（）

隐藏：hide（）

toggle（）——自动切换显示隐藏效果，注意和toggleClass写法进行区别



1. 与css（）方法的区别（**最好使用hide和show方法**）

相比之下，使用jQuery的情况下要实现显示隐藏的效果的话，最好是使用上面的显示隐藏的方法而不是css（），虽然在效果上面不会有明显的区别，但是还是在一些细节上面有不同。

在css来设置显示隐藏的时候，由于不同元素的可能是块级元素也可能是行内元素，那么在使用display属性的时候就有可能会在切换之后改变元素的属性，影响到布局。但是使用show和hide方法的话，他会自己进行判断元素的类型，然后自行进行调整，就不会说造成将行内元素变成块级元素的情况，这是比较好的选择。

总之就是CSS方法会改变元素的种类，而hide和show不会。

1. 节点的选择
2. 对兄弟节点的操作

Prev（）——选择当前元素的前一个兄弟节点；（**五星级**）

next（）——选择当前元素的后一个兄弟节点；（**五星级**）

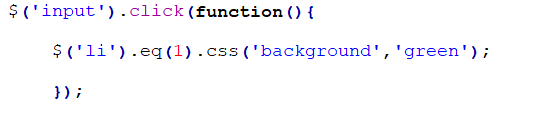
prevAll（）——选择当前元素的前面所有兄弟节点；（**三星级**）

nextAll（）——选择当前元素的后面所有的兄弟节点；（三**星级**）

siblings（）——查找当前元素所有的兄弟节点；（面试经常会问到）

1. 参数的筛选功能

上面的方法传入的参数可以为空，也可以是元素选择器，传入相关条件对要选择的节点进行筛选。

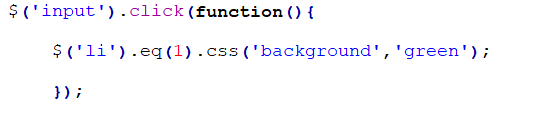


1. **实战技巧**

有时候可能在开发的时候会遇到某一个元素在结构比较简单的时候选择起来比较方便，但是如果中间添加了其他结构就会破坏这种选择，那么可以使用参数来筛选，并且将选择范围扩大（例如从next（）变成nextAll（）等）；

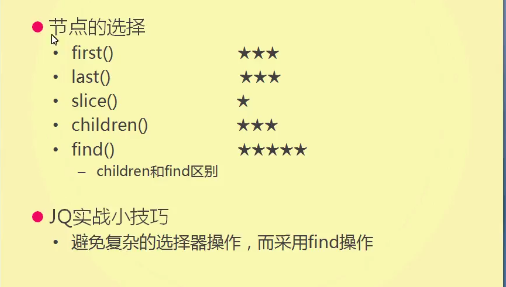
1. 下标——eq（）（**五星级**）

下标的表示方法和数组的表示方法是很相似的，也就是在eq（）方法中传入的参数是下标；



第二章

1. 节点的选择



1. first（）和last（）方法可以使用eq（）方法来代替，因此也就显得不这么重要。
2. Slice（）方法——用得少

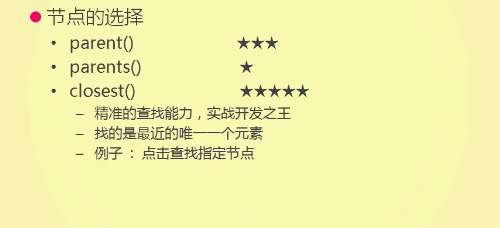
选取一组元素中的某一部分，这个方法和数组的方法是一样的。这里需要接受两个参数，一个是起始位置，另一个是结束位置（不包含结束位置）。

1. children（）——获取当前元素的子节点

只要是当前元素的子节点，都可以被选中，但是子节点以下（孙节点……）都不可以被选中。同时，也可以传入参数对要选择的内容进行筛选。

1. find（）——在当前元素中查找符合条件都的节点；

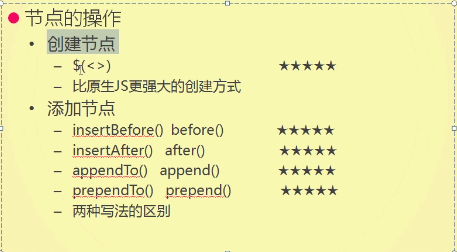
相比之下，find（）方法的查找范围比children（）更广，因此实际使用的时候就比较多。find（）方法可以查找到子节点的子节点。



1. parent（）——获取前元素获取父节点
2. parents（）——获取当前元素的所有祖先节点；
3. **closest（）**——查找最近的一个祖先元素（唯一）（**五星级**）

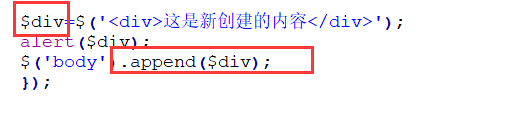
这个方法必须接受一个参数，这个参数就是筛选条件，然后选中的是符合条件的那个唯一的离当前元素最近的元素节点。但是需要注意的是，当前元素自己也算是所有祖先元素的一员，如果当前元素自身就符合查找条件的话，那么就会选中当前元素自身。

1. 节点的操作



1. 节点间的创建

创建节点的时候必须要使用尖括号，也就是说在正常的查找样式的引号内添加尖括号，这样就是创建元素节点了。jQuery中创建元素节点的方法相比原生js的创建方法更加强大，因为创建节点的时候是可以添加自己想要添加的内容的，包括属性，和元素内部包含的内容。



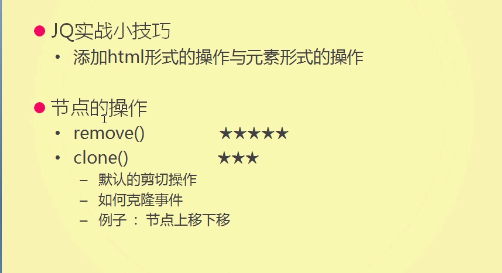
1. append（）——把元素添加到当前元素的里面的最后面；
2. prepend（）——把元素添加到当前元素的里面的最前面；
3. before（）——把元素添加到当前元素的外面的最前面；
4. after（）——把元素添加到当前元素的外面的最后面；
5. 上面的四个方法对应的xxxTo（）方法两者效果是一样的如下，两个都是追加操作，但是注意，两个参数的位置是不一样的。对于后续的操作来说，前者是“将新增的li标签追加到ul中，然后再将ul的背景颜色设置为红色”；后者是“将新增的li标签追加到ul中，并且将该li标签的背景颜色设置为红色”

注意，上面的两个操作的后续操作（css操作）针对的对象都是前面的哪一个，这一点需要注意。



**Tips：**

7、在创建新的元素的时候，如果是后续会对它有操作，那么最好的方式就是将这个新创建的节点存进一个变量里面。这样在后续的使用中更加方便。

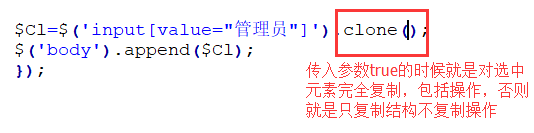


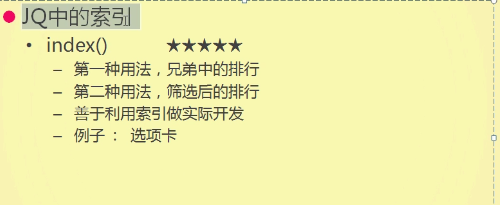
1. remove（）删除节点

如果在原生js中还需要先找到待删元素的父节点，然后在使用removeChild（）方法来说进行删除，但是在jQuery中就不用，可以直接进行删除，也可以传入参数对需要删除的内容进行筛选。

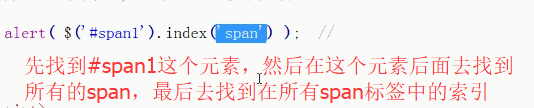
1. clone（）——克隆（复制）

在之前的那些节点操作中都是剪切式的，但是clone方法则是使用的是复制方法，也就是说，在使用这个方法的时候是将对应的内容复制一份，但是默认操作行为只会克隆结构，而不会克隆之前元素的操作行为，如果需要赋值行为的话，那就需要传入一个参数true。

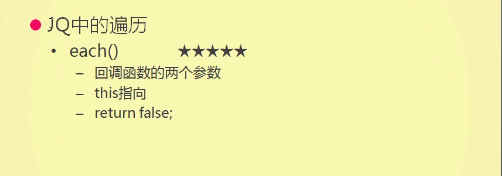




1. index（）
   1. 兄弟中的排行——索引值代表当前元素在所有兄弟节点中的排列顺序，默认是从0开始计算。这个用法和标签元素的种类没有直接关系，主要是由标签之间的关系来决定的。因为，index（）方法只查找存在兄弟关系的标签的索引值。
   2. 筛选后的排行——先找到当前元素，然后在筛选出来的所有元素中去查找当前元素在这一组元素中的索引值。



1. 遍历

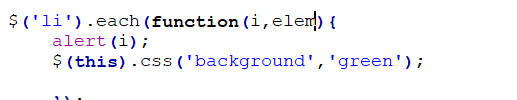


**$（）.each（I,elem）——专用于遍历对象**

1. 在jQuery中是可以省略循环的，大部分时间是用不着循环的，但是也不排除在一些特殊情况下需要用到循环操作。
2. each（）方法

这个方法的回调函数需要两个参数，第一个参数就是索引值，第二个就是对应索引值所对的元素。

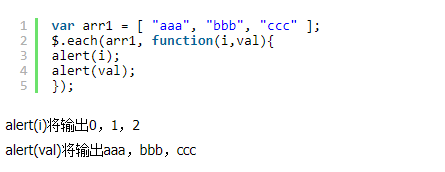
（3）在循环中的this和回调函数的第二个参数是一样的，也就是指的是当前操作的元素。



上面这些语句的意思就是：选出所有的li标签，然后遍历这些标签设置那些标签的背景颜色为绿色。

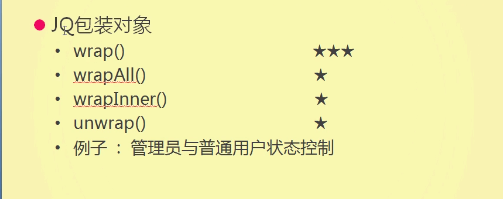
**$.each（）——可用于遍历任何集合，包括数组、Json**

1. 在这个方法中同样可以传入两个参数，一个是要遍历的集合名，另一个就是要进行的操作，也就是回调函数。

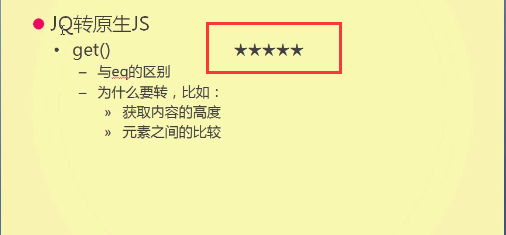


（5）在jQuery的循环中，如果需要break的效果的话就使用return false语句结束代码执行，这个是jQuery和原生js不一样的地方。

1. JQ包装对象



1. 包装对象用的很少，因此不必花太多时间。这个包装实际上就是动态创建一个标签添加到选中的元素外面，但是需要注意，这个wrap（）方法在使用的时候，添加标签的方法和创建标签是一样的，都需要使用尖括号包起来（wrap（’<div>’） ）；
2. wrapAll（）——这个方法和wrap的区别就是前者选择的是所有的符合条件的元素，然后将其包起来。但是，值得注意的是，如果是在一组元素中有不符合条件的元素标签的话，将会被剪切出来，然后将符合条件的包在一起。会破坏布局。
3. wrapInner（）——在当前元素内部添加一个包装标签。
4. unwrap（）——相当于删除父节，但是不能操作body，其他的标签都是可以的。
5. jQuery转原生js



（1）get（）——将jQuery转换为原生js的方法，默认情况获取的是一个原生的集合，那么就可以在get（）中传入参数（下标）来获取对应元素。

（2）get（）和eq（）的方法的区别就是，get之前是jQuery的写法，后面是原生js的写法；但是eq（）不管是前面后面都是jQuery的写法。

（3）jQuery转换为js原生语法的原因：在一些特殊的情况下，有时候某些操作无法通过jQuery直接完成，那就需要进行转换，比如获取内容的高度，元素之间的比较等。

练习：

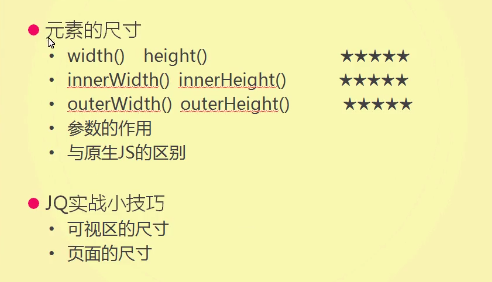
左右文字切换

练习2： 评分效果

实现思路：获得所有的评分元素，然后进行遍历，在遍历中对其进行操作（jQuery无需写出遍历，默认遍历该组选中元素）；在鼠标移入的时候现将所有的元素的北京设置为空，然后在进行上色；鼠标溢出的时候遍历全部元素，清空所有设置；在点击按钮（选中）开始评分的时候，首先需要清除鼠标移出事件，然后将当前元素之前的所有元素背景设置清空，再进行背景色设置。

第三章

一、元素尺寸

1、width（）/height（）

这两种方法获取的值是仅仅是宽高相关的值，但是和padding之类的值是没有关系的。

2、innerWidth（）/innerHeight（）

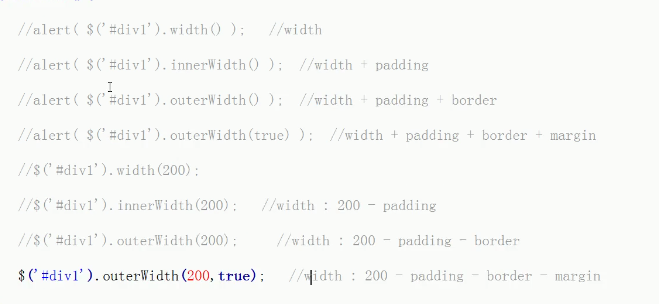
这两种方法和其前面两种方法有差别，这两种方法的返回值包括了border（不包含border以内）的尺寸。

3、outerWidth（）/outerHeight（）

这两种方法的返回值就是从padding一直到border的尺寸之和。

4、获取与margin相关的值——outerWidth（true）/outerHeight（true）

想要获取包含margin尺寸的话，那么就在参数中传入true就可以获取到了。



5、参数的作用

根据上面的几种用法，就可以知道在设置值的时候是按照其本身的规则来实现的。也就是说，我们传入的数值是包含了对应的哪几部分，是按照那种方法本身的意义来实现的。具体的内容可以见上面的截图。

6、与原生JS的区别

原生js不能获取到隐藏元素的尺寸的，但是jQuery是可以获取到隐藏元素的尺寸的。

7、

可视区的尺寸——$(window).width（）

页面的尺寸——$（document）.height（）

上面两个操作分别是获取可视区尺寸和获取页面的尺寸的操作，在开发过程中用的比较多，因此还是需要记住的。

8、滚动距离

用的比较多的是scrollTop，对应的是y轴的距离。scrollLfet用的比较少，对应的是x轴的滚动距离。

当页面滚动到底部的时候，那么此时滚动距离就是页面高度减去可视区的高度：$（document）.scrollTop=$（document）.height（）-$（window） .height（）

如果传入参数的话（数字），那么就是设置滚动距离了。



9、元素距离

要获取当前元素距离页面左边和页面顶部的距离的话，那就调用offset（）方法，然后分别调用left和top属性来获得相应的值。要注意的是，offset（）获取的值都是相对于整个页面的距离。

在原生JS中，offsetLeft在获取值的时候，当网上遇到第一个有定位的祖先节点就会结束查找，也就是说获得是当前元素到该祖先节点的距离。但是，jQuery中的offset（）方法就不会受到定位的影响，获得的值行还是我们需要的当前元素到页面的距离。

10、position——获取到有定位的祖先节点的距离（不认margin值）

这个方法也是有left和top属性，但是它无法识别margin值，但是能识别padding值。例如，用定位或者在父级上添加padding属性，可以设置出需要的距离。

11、实战技巧——利用计算原理，获得相应的值

offsetParent（）——获取最近的有定位的祖先节点；

$（选择器）.offsetParent（）

这个语句就是获得当前元素与有定位的父级元素之间的距离。

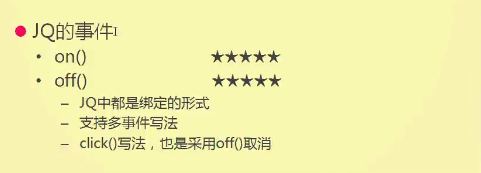
实例、懒加载页面中的图片

要实现这种图片懒加载的效果，就是监控图片与可视区之间的额距离关系。只要图片进入到可视区，那就将图片加载出来，没有进入可视区的就不加载。

有一种做法就是，先将图片的src值存进一个自定义的属性中，然后然后在需要的时候将这个属性值取出来，将其放到图片真正的src属性中，这样就可以将图片加载出来。

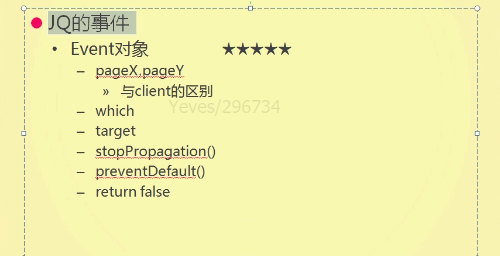
触发滚动事件，：$（window）.scroll（）

二、jQuery中的事件

1、在jQuery中的事件操作都是绑定的形式，在同一个元素或者同一个事件下可以绑定多个操作，多个事件可以使用空格隔开。

2、使用on的方法实际上和不使用on的方法效果是一样的，应为不使用on的写法在jQuery源码中，最终也是调用的是on的方法，只是on的方法相对来说功能更加丰富。

3、off——取消事件。一旦调用了取消事件，那么该元素上的事件就会被取消了。括号中掺入的是过滤条件。

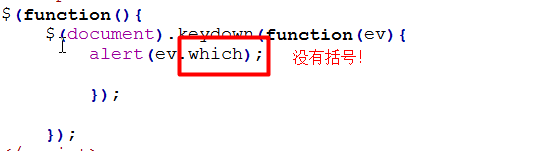
4、在event对象中有两个属性是获取当前点击的位置相对于整个页面的距离：event.pageY、event.pageX分别就是获得当前点击的位置和与整个页面顶部和页面左边的距离。

**键盘操作**

5、event.clientX/Y就是当前位置相对于可视区的顶部和可视区的距离。

6、which——指的是键盘的键值

event.which——获取的是当前键盘按键的编码值。其中event.keycode也是指的是 键盘键值。



实际上，原生的js和jQuery之间并不会冲突，只要按照相关规则进行书写，js的语法也是可以在jQuery的代码块中正常执行的。

7、target——指的是目标元素

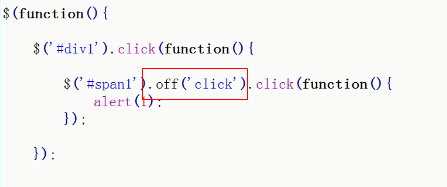
在一个代码块中，哪一个对象调用了一个事件，那么this就是指的哪一个。event.target就是指的是事件源，例如，点中一个div的时候触发一个事件，那么就会返回这个div对象。（更详细的事件源可以仔细去了解）

8、event.stopPropagation（）的作用就是阻止事件冒泡。

9、event.preventDefault（）阻止默认事件。

10、return false——阻止默认事件和冒泡，是上面的两个事件的集合。

实战技巧——处理事件的多次添加



如上图所示，如果没有红框中的那一部分的话，当多次点击div1的时候，那么span1事件就会被多次绑定，这并不是我们需要的。如果在页面结构复杂，来不及调整结构的时候，那就可以借助off（）来解决这个问题，就是在每次绑定新的事件之前，先将之前的事件都解除掉，然后才进行行的事件绑定。

11、事件委托

（1）事件委托的原理就是利用事件冒泡，好处就是提升性能，可以简化很多添加的操作，还有就是可以让后续添加到被委托元素的内部的新元素也具有和之前就存在的元素一样具有同样的效果。这种效果会经常用到，所以还是需要好好掌握的。

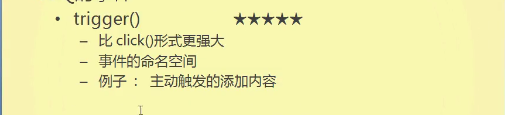
（2）事件委托的注意事项：

在事件委托中，this指的就是被委托的事件，例如下图代码中的事件委托中，li就是this所致的那一个元素。

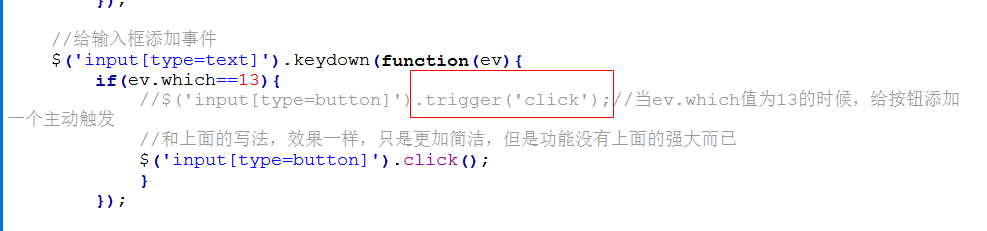
（3）ev.delegateTarget——指的就是当前委托元素的事件源。如上图，在event对象中的delegateTarget属性就是查找当前元素的事件源，也就是说可以找到ul。

（4）事件委托也是一种事件绑定的行为，因此也是具有事件委托取消的方法，也就是$（委托对象）undelegate（），这里的委托对象一定不是当前的那个元素，而是事件委托的事件源，也就是如上图所示的ul。

12、trigger（）——主动触发



主动触发trigger（‘事件名’）可以在任意的时候触发一个事件，可以节省一些代码和操作。

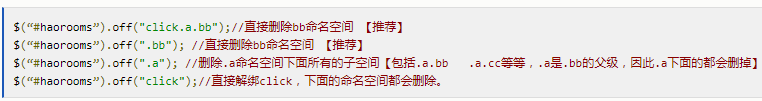
13、命名空间

所谓命名空间就是在事件类型后面添加一个点号，再附上一个别名，以便于引用事件，如“click.a”。

（1）如果是有单独的事件名和添加了命名空间的事件，如果只想执行那个单独的事件名的那个事件，那就在事件名后面添加感叹号，其他的有命名空间的事件就是事件名点上命名空间就可以调用了。

（2）删除命名空间

删除命名空间可以使用off（‘事件名（包括命名空间）’）



（3）事实上，如果要绑定事件，一是可以使用on来进行事件绑定，另一个就是使用bind（）进行事件绑定；要是要删除bind进行操作的事件绑定的话，那就使用unbind（）来删除事件绑定。

（4）命名空间的用途

三、工具方法



jQuery的方法分为两种，一种是针对jQuery对象的，必须要获取jQuery对象才能调用；另一种就是工具方法，既可以给jQuery用，也可以给原生的JS用，就是不用先获取对象直接先写$符号。

1、$.type（）——查看变量的类型

原生的JS中的typeof的功能不是很够用，很多时候判断的结果也是很准确，因此jQuery定义了一种更加强大的判断变量类型的方法$.type（）。

这个方法就是讲需要判断的对象当做参数传进括号中，进行我们需要的判断。

2、$.isFunction（）

使用的方法和上面的类型判断是一样的，都是把需要判断的对象当做参数传入，然后返回的是一个布尔值。如果是函数，那就返回一个true，否则就返回一个false。

3、$.isNumeric（val）

判断传入的内容是不是数字类型，但是要注意的是，如果传入的是一个数字字符串的话，这个方法也会返回true。所以，人如果需要更加精确的判断，那么还是需要结合type的判断来进行更加精确的判断。

4、$.isArray()

判断是否是数组，返回值就是两个布尔值。

5、$.isWindow()

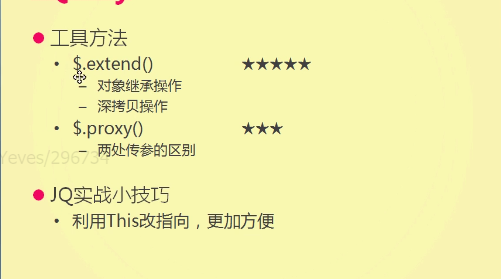
判断是不是window。

6、$.isEmptyObject()

判断是不是空的对象。例如，秀饿了一个JSON，但是里面没有内容，那么使用这个方法判断就会返回true，否则就会返回false。

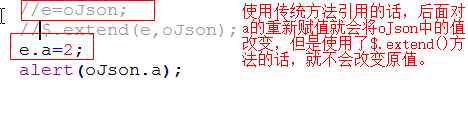
7、$.isPlainObject()

判断是不是对象自变量。

8、$.extend()——对象拷贝和深度继承

根据原生的js学习我们已经知道，对象的引用会相互影响，虽然引用之后两者是相互独立的空间，但是由于二者是同步的，那么不管是引用的对象还是被引用的对象发生变化都会影响到对方，将值同步。

使用$.extend（）方法的话，就可以实现屏蔽这种影响，如果对两个对象建立了引用，就算是将引用的对象进行了改变，也不会影响到被引用的而对象。



这个方法可以传入任意多个参数，从第二个以后的那些内容都会都将会拷贝到第一个参数的后面（从后添加），这是属于浅拷贝，此方法默认是深拷贝的，在传入的内容存在多层引用的时候还是会像传统方法一样受到影响。如果需要进行深拷贝的话，那就在第一个参数之前传入true进行深拷贝，也就是说不管传入的内容有多少层都不会影响拷贝。

9、proxy（）（代理）

属于改变this指向的方法，但是由于相对比较麻烦，所以用得也不算太多，但是用得也不少。

这个方法需要传入两个参数，一个是调用的函数，另一个是需要指向的对象。如果需要调用参数中的那个函数的话，还需要在方法后面再添加一个加括号，如果不加括号就只是指向改变了而已，但是实际上函数并没有被调用；如果调用的函数需要传参数的话，既可以在前面的括号传入，也可以在后面的括号传入。两种传参方式的区别就是：前面传参是在事件触发的时候才会真正将参数传进去，放在后面传入的话，那就是执行到这个方法的时候，不管事件是否触发，都会把参数传进去并且立即执行触发事件（调用函数）。

10、改指向小技巧

很多时候如果需要改指向的话，那么比较方便的方法就是存一个变量来改指向，这样比较简单明了，而且还便于修改。

**第四章**

一、运动



1、时间的概念

在运动中，实际上时间是可以决定一个物体运动的快慢的。

2、常见的简单运动方法

（1）show（）/hide（）/toggle（）方法

一个显示和隐藏的效果，以及一个两者智能切换的效果。方法中可以传入几个参数，一个是运动效果的快慢，另一个就是回调函数，就是在执行完显示隐藏效果后接下来将会继续执行的动作。

时间参数可以使关键字也可以是具体的数值。Fast（200ms）、normal（400ms）、slow（600ms）。

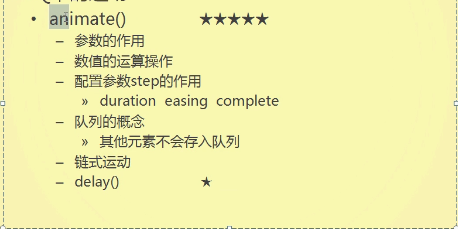
（2）fadeIn（）/fadeOut（）/fadeToggle（）淡入淡出

控制淡入淡出的，可以不写任何参数，就是有一个默认的normal参数，但是也可以写具体的时间数值来控制变化的速度快慢。

（3）slideUp（）/slideDown（）/slideToggle（）方法

上下滑动，也可以看作是上下显示隐藏的效果。其中也是可以传入事件参数和回调函数。

3、复杂的运动效果



（1）参数

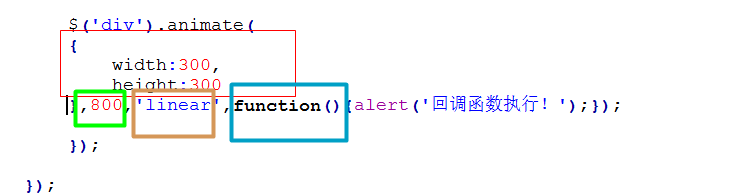
Animate（）方法可以接受多个参数，首先是将要操作的对象选择出来，然后就去设置参数：

第一个参数设置样式属性和性对应的值；

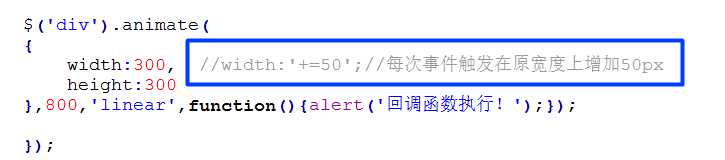
第二个参数就是时间参数，控制运动的快慢；

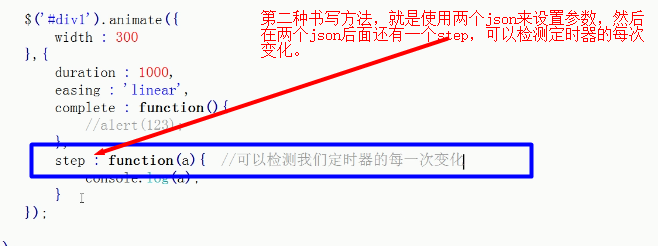
第三个参数知名运动形式，只有两种，一种是缓冲（swing，默认形式，是“慢快慢”形式），另一种就是匀速的变化（linear）；

第四个参数是运行结束的回调函数

（2）数值的运算操作

如果时间每次触发，某些数值操作需要在上一次的基础上继续变化，那么可以通过擦混入字符串的形式来实现，例如每次事件触发的时候width都要在上一次的基础上增加100，那么就可以写作**’+=100’**这样的形式。



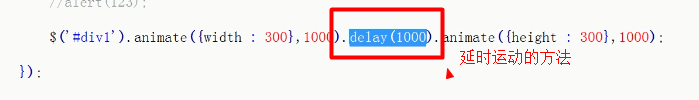


Step还可以接受一个参数吧，是一个对象tween，也就是说可以接受两个参数。其中b.pos是运动过程中变化的比例值。有时候可能会用到

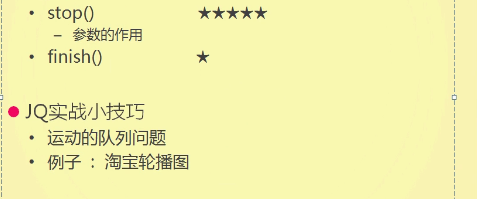
4、运动中的队列的概念

首先需要注意的是，定时器是异步的，所以定时器执行的时候不会影响其他代码的执行。但是在jQuery中，如果同时有几个animate的方法调用的话，jQuery会将几个调用的参数获取到然后存进一个数组中，然后按照向后顺序执行代码，也就是说现在的执行方式不再是异步的而是同步的方法在执行了，并且其他的非运动方法不会存进这个队列中。

由于上面的是同步运动的效果，因此完全可以使用链式运动的方式来进行等效调用；也就是使用点号链接起来就可以实现链式操作了。

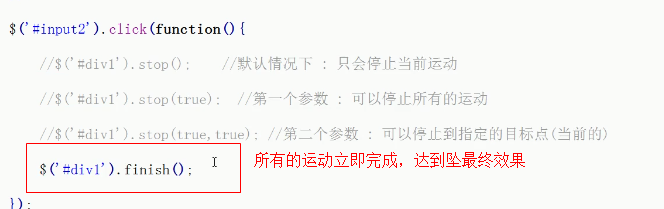


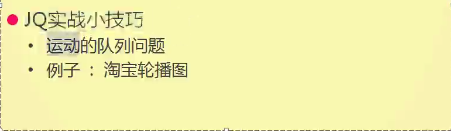
5、停止运动



使用stop（）在默认情况下，只会停止当前运动，如果需要停止所有的运动没那么就传入一个参数stop（true），那就可以实现这个效果，这是第一个参数。

第二个参数stop（true，true），停止到指定的目点（但是只会对当前运动有效），如果需要全部的运动都到达指定的目标点，那就是用finish（）

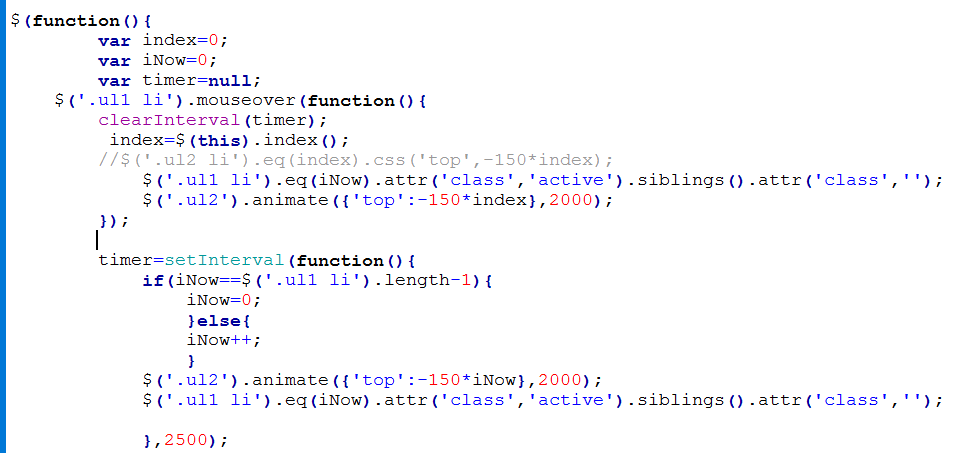


（1）运动的队列问题

Stop()还有清空运动的队列的行为，比仅仅是停止运动。因此，在某些情况下，为了避免出现运动的时候队列过长的情况出现，这时候就需要将stop（）加入到我们写的链式运动中或者是一些运动的语法中，避免这个问题。

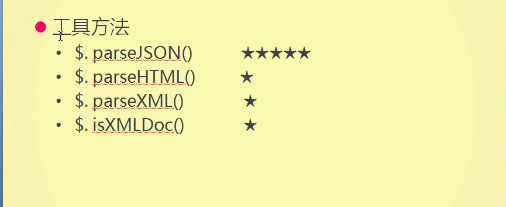
（2）实例——淘宝轮播图

实现轮播图主要有两个部分，一部分是图片展示区，另一部分是索引页。这里两部分实际上是需要联合在一起变化的，正好，二者的数目是相同的，因此也就可以索引所引来建立联系。



如上图代码所示，这是最简单的轮播图代码，变化主要包括两部分，一部分是图片的变化，另一部分是图片索引值变化，然后要实现自动播放就是调用一个定时器，然后在播放到最后一张的时候自动转换到第一张来。

6、工具方法



（1）$.parseJSON()：把JSON类型的字符串转换为真正的JSON数据。

使用方法就是直接将需要转换的字符串当参数传入就可以了。但是这个方法的缺点就是：只能解析JSON类型的字符串，其他的类型就无法解析，这样就让其安全性变得更好了。还有需要注意的是，这个方法针对的是严格的JSON，否则就法法解析。

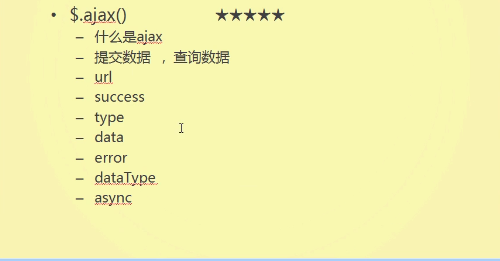
（2）$.parseHTML()：转化HTML格式的字符串，是起转化为DOM节点，放进一个数组中。

使用方法也是将字符串一参数形式传入，然后的到的就是各个DOM节点，然后可以用数组的方法来访问到。

（3）$.parseXML()：解析XML格式的字符串，将其转换为真正的XML节点。

（4）isXMLDoc（）：判断是否是XML的格式。

7、**ajax**

（1）什么是ajax？

就是可以不刷新页面（异步）实现前后端数据交换，提升用户体验。

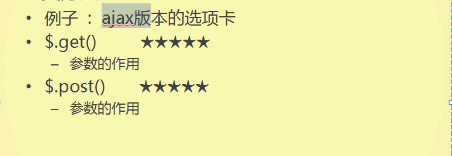
（2）error的作用：当页面请求出错的时候执行的回调函数，接受一个参数。

（3）dataType

前端指定ajax接受的数据类型，然后jQuery自动帮助我们完成这个转化的过程；如果不指定这个dataTye的话，那jQuery就无法帮我们进行转换，也就需要我们手动转换，使用$.parseJSON()方法。

dataType的值有XML、JSON、HTML。

（4）async——ajax是否异步，默认是异步。比较常见的使用同步的情况就是：初始化数据等后面的代码执行需要用到前面的数据的时候，如果前面的没有执行会直接影响程序的正确运行的话，那就需要使用同步的操作。



（5）$.get()方法——提交get方式的数据



其中，数据格式就是 dataType的三种数据格式。在这里指定之后就可以不用手动去转换数据格式。



其中用到error的时候还是比较少的，有时候使用get和post方法的话，可能无法实现异步操作的设定，这种情况下就需要使用ajax来写。

（6）$.post()方法——提交post方式的数据

Post方法和get方法是一样的用法，然后也是4个参数，写法上没有什么不一样，主要是两种方法自身船体数据的方法不一样而已。

第五章

一、jQuery的插件和组件

1、插件的概念就是基于jQuery的一些已经写好的程序，直接可以用来使用，而不用自己去知道怎么写，只要知道这个插件怎么用就可以了。



2、组件与插件的区别

总的来说，组件与插件的内涵式一样的，只是组件的功能比插件更加强大，可以将组件看作是插件的集合。组件实现的是一组功能的集合，但是插件是实现单一的某一种功能。

16-jQuery接口和后端交互

第六章

一、清空元素的方法

1、html（‘’）——将元素置空；

2、empty（）；

二、删除节点

1、remove（）

2、detach（）

区别：remove方法删除之后发挥的结果是被删除的这个元素，如果删除之后还需要使用这个元素的话还是可以继续使用这个结构，但是钙元素上面的操作全部会被删除；但是，detach（）方法删除元素，但是会（可以）保留元素之前的行为。

三、获取文本

1、text（）方法

是用html方法的话，会获取元素内部的所有元素节点；但是使用text方法的话就是只获取所有的文本节点。如果选中的是一个集合，包括了多个元素的话，他将会获得该集合内的所有的文本节点。在设置内容的时候，text（）内的所有内容都会被当文本进行设置。

四、替换节点（用得少）

1、替换对象.replaceWith（‘替换的内容’）

2、替换对象.replaceAll（‘替换的内容’）——去哪不替换

五、事件扩展

1、hover（）

接受两个参数，都是函数，第一个是鼠标移入的时候执行的函数，第二个是鼠标移出的时候执行的代码。（效果和over和out一样，但是调用的不是这两个而是enter和leave）

2、mouseenter和mouseleave的特点就是：子元素不会影响父元素。有时候由于一些特殊的效果，由于子元素会冒泡，然后父元素会收到影响，因此mouseenter和mouseleave方法就显得很有用了，他们不会影响让子元素影响到父级元素。

3、focusin（）和focusout与focus和blur区别

Focus元素对不存在冒泡特性，因此在需要使用冒泡功能的时候就会显得很不足。但是，focusin方法对支持冒泡，这个功能大大简化了代码的书写。

4、one（）——只能触发一次

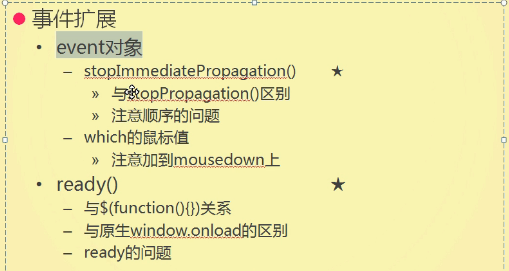
这个方法只能触发一次

5、on（）

这个方法总共用五个参数，其中一个是内部使用的，但是用得最多的是第一个参数和第四个参数。第一个参数是事件，第四个是事件触发以后执行的函数。然后第二个，第三个参数分别是事件委托对象和一个数据集合。

在这个方法中，有些特殊情况下树妖获取事件对象，但是ev是jQuery自带的专属于jQuery的事件对象，有时候需要使用原生event事件对象的时候，就需要使用originnalEvent来讲jQuery专用的ev事件对象转换为js用的原生事件对象。

6、triggerHandler（）方法和trigger（）方法大致一样的。但是前者这种主动触发不会触发事件的默认行为，但是后者会触发事件的默认行为，这就是二者的却别

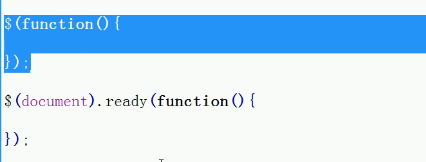


（1）stopPropagation（）是阻止冒泡的方法，这个阻止冒泡的方法的效果就是只阻止向父级冒泡，不去触发父级元素的事件，不会阻止这个元素本身的事件；但是stopImmediatePropagation（）阻止的就是父级元素的事件和当前元素自身的事件操作，但是需要注意的是，阻止的事件必须是写在阻止语句的后面。

（2）which的鼠标值

鼠标的键值必须是加在mouseup或者是mousedown上面，鼠标键的左中右分别是1,2,3.这个鼠标的事件不能加在click上，因为键盘也可以进行触发，因此不能使用click事件。

（3）ready（）事件

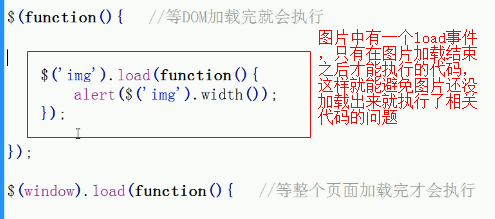


上面的两种写法是等效的，但是在代码执行的时候，调用的还是第一种方法。

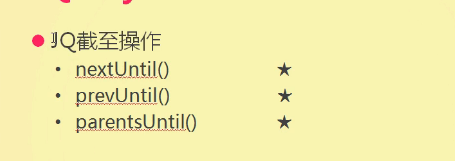
关于执行时间：



第一种加载速度回更快一点，但是在有些情况下必须要等全部加载结束之后才能执行一些代码，所以这个时候就需要注意这个使用的方法，否则就会出现问题。



6、截止操作



这几个方法就是截止到某个元素为止，参数就是那个截止的元素。

7、数据缓存



（1）Data和attr的区别就是：attr是真正直接在元素身上添加属性的，但是data是间接将数据存进了这个元素中，实际上是先将数据存进一个大的集合中，然后存在jQuery自身的一个库中，与当前元素建立间接的联系。这个还是和attr有很大的不同的。

Data的方法比较容易避免引起内存泄露。因此缓存大量数据使用data，设置自身属性则使用attr。

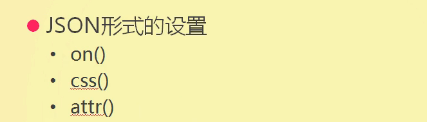
（2）prop（）方法

这个方法和attr很相似，只是attr在设置和获取属性的时候分别是使用的setattribute和getattribute方法；prop（）则是使用的是点号或者是方括号的方法来设置或者是获取属性值的。

（3）删除数据的方法

如上图的三个remove方法。

8、JSON形式的设置



使用回调的设置将会使我们能够实现的功能更加复杂，所以还是很有用处的。

