JQuery可以分为 J和Query 分别是Javascript和查询的意思

JQuery是一个JS的库

1. **JQuery本质是一个闭包**
2. **JQuery为什么要使用闭包来实现**

为了避免多个框架的冲突

1. **JQuery如何让外界访问内部定义的局部变量**

Window.xxx = xxx；

1. **JQuery为什么要给自己给自己传递一个window参数？**

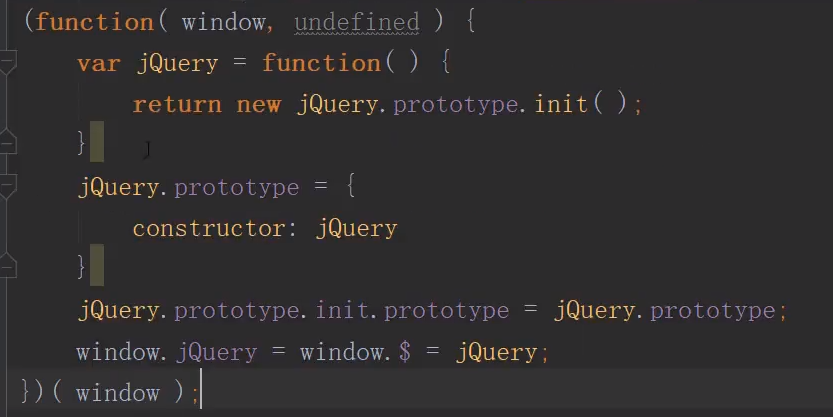
为了方便后期压缩代码

为了提高查找效率

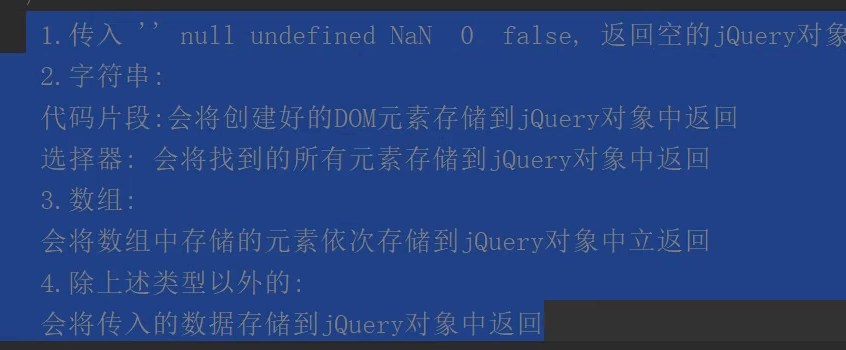
1. **JQuery为什么要给自己传递一个undefined参数？**

为了方笔后期压缩代码

传递一个正确的undefined，保证内部的undefined正常使用，因为IE9以下的浏览器undefined可以被修改



JQ入口函数传入不同类型数据的四种情况：



JQuery中的each和map静态方法都是两个参数，要遍历的数组或伪数组，和一个回调参数，他们不同的地方时map默认没有返回值，each会返回变量的对象 ，但是map方法可以通过回调函数内设置return

Trim方法会去掉一个字符串两端的空格，他的参数就是需要去掉空格的字符串，并且会返回一个新的字符串

IsWindow()，判断参数中的对象是否为window对象，返回布尔值 ， 同理还有 isArray() 、 isFunction()

另外还有判断空对象的isEmptyObject()

和判断是否为纯对象的isPlainObject()

IsNumeric（）判断是否为数字

立即执行函数，就是不用调用直接执行的函数

语法为(function 名称（）{})（）

$.holdReady（）参数为布尔值，为true就会暂停ready入口的加载，

False为开始加载

**内容选择器**

：contains(‘文本’) 【搜索包含这段文本的Dom元素】

：has（‘Dom元素’） 【搜索包含这个dom元素的Dom元素】

：empty 【搜索没有文本节点也没有元素节点的dom元素】

：parent 【搜索包含文本或元素节点，即不为空的DOm元素】

**属性和属性节点的区别**

所有对象都有属性，可以通过【对象.属性 = 值】来添加或修改一个对象的属性（JQuery用【.prop()方法，使用方法和下面的attr大同小异，通过此方法删除属性节点只会使属性的值变为undefined，并不会删除属性节点】，并且它也可以获取属性节点，例如checked这种的属性节点，他的返回值为true或false）

只有DOM元素才有属性节点，可以通过setAtrribute来修改(JQuery用 【.attr（）】获取和修改(传一个参数为获取，传两个参数，如果有，为修改 || 没有，为添加) removeAttr（）是删除，同时删除多个属性可在参数字符串中用空格隔开 例如checked这种属性节点，它会返回该属性名称或undefined

**CLass类操作**

AddClass()：添加一个类，同时添加多个可在字符串中用空格隔开

removeClass()：删除一个类，删除多个空格隔开

toggleClass()：切换，有就删除，没有就添加，操作多个与上同理

**HTML值操作**

.html() 和原生JS中的innerHTML大同小异，传入字符串参数会想目标DOM元素中添加文本或节点，无参数会返回获取值

.text()和原生JS的innerTEXT大同小异,和.html()语法一样

.value() 获取表单标签的value值，语法和.html()一样

**操作样式**

**1.** .css(‘width’，’100px’);最繁琐的方法

**2.** .css().css().css();链式方法，阅读性差

**3.** .css({width:’100px’,……}) JSON添加法，推荐

**尺寸与定位**

.width() 不包括padding、border

.innerWidth 包括padding，但不包括border

.outerWidth 包括padding和border的宽度 (高度同理)

.offset()获取与窗口的位置，与原生JS中的offsetXXX大同小异可通过.left等获取具体某个的位置(默认返回伪数组)

修改位置可在参数中以JSON的方式修改

.position()获取以定位基点的位置,获取方法和上方的.offset()类似，但是注意：此方法只能获取，不可修改，可通.css（）来修改

.scrollTop() 和原生JS中的scrollTop大同小异，获取一个DOM元素可视区到总高度的距离 ,设置为参数传整型数字

(因兼容问题,获取写为$(‘html’).scrollTop+$(‘body’).scrollTop

获取的时候写为(‘html，body’) )

**事件绑定**

1. .事件名称(funcition(){})
2. .on(‘事件名称’,function(){})
3. One（‘事件名称’,fn）此方法绑定的事件只会执行一次

相同点：事件名称都不带on，并且注册多个相同事件不会覆盖

不同点：方法1编码效率略高，但是此方法有些事件不能注册；方法2编码效率略低，但是包含原生中所有的事件，且功能强大(可自定义事件和进行事件命名空间操作)

**事件解绑**

.off()谁需要解绑就给谁添加，如果没有参数，代表解除所有事件；一个参数可以输入一种类型事件的字符串，代表解除这一类所有的事件；两个参数可以输入一种类型事件的字符串，并且可以指定一个函数名，代表只删除这一个事件

**阻止事件冒泡和默认事件**

阻止事件冒泡：event.stopPropagation();

阻止默认事件：event.preventDefault();

PS：return false会同时将两者全部停止

**自动触发事件**

.trigger()传入触发事件类型，会触发事件冒泡和默认事件

.triggerHandler()语法同上，不会触发冒泡和默认事件

PS:绑定于a标签上的事件无论哪种方法自动触发都不会发生冒泡和默认事件，想要触发冒泡默认事件可以在a标签内添加子元素，将事件注册在其身上

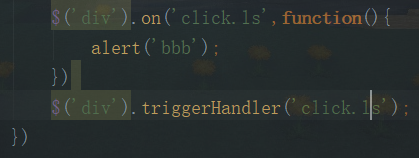
**自定义事件**

两个条件：1.事件必须通过on方法注册，2.由trigger或triggerHandler触发



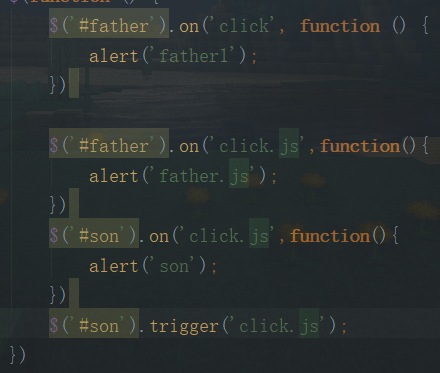
**事件命名空间**

1.事件必须通过on方法注册，2.可以通过trigger或triggerHandler触发某一个事件



**事件命名空间小细节**

如果用trigger(点击和triggerHandler无效)自动触发一个子元素的带有命名空间事件的话，那么这个子元素它的父级中只有命名空间和它一样的才会被冒泡，如果子元素没有带命名空间，则会触发父元素的所有相同类型事件

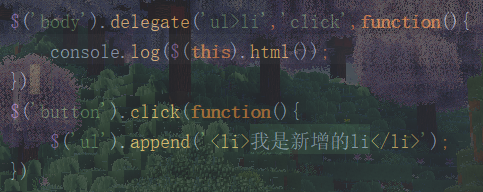


图中只有父级的click.js事件会被冒泡，如果son的事件没有命名空间，那么父级的click和click.js都会触发【PS：就算是son设置的是click.js事件，但是trigger依然可以调用click，效果和正常click一样会触发父级全部冒泡】

**事件委托**

被委托人.delegate(委托人,事件,fn)

事件委托，就像大人给小孩子起名字一样，小孩子没有出生，无法自己起名字，名字都是由大人起的，所以小孩子一出生就带有名字，因为他的名字已经由大人帮忙起了，在程序中，动态生成的元素可以通过页面加载完毕时就存在的DOM元素帮忙为自己绑定事件



这里li让body帮忙给自己绑定事件，所以下面无论新增多少li，他们的身上都有click事件

**移入移出事件**

因为原生JS提供的mouseover和out事件会因子元素导致不正常触发(移入移出子元素也会触发事件)，所以JQ提供了mouseenter和leave事件，他们解决了这一问题，而且JQ还为他们做了简化，即.hover()方法，这个方法最多可以传两个函数参数，传一个表示移入移出都会执行该函数；传两个则会移入执行第一个参数，移出执行第二个

.Siblings() 返回其他的兄弟 .index()返回索引值

.eq()利用索引选取其中一个元素 .get()同前

两者的不同是eq返回JQ对象，get返回dom元素

**动画**

PS：JQ的动画具有动画序列，快速的触发动画会导致堵塞，会出现即使你已经没有在进行触发操作了，但系统还是会继续讲动画通道中堵塞的动画慢慢执行完)



因为动画队列的原因，链式动画可以直接这样写

显示 .show()

隐藏 .hide()

切换 .toggle() 三个方法都可将一个整型数字作为参数，起到控制动画执行时间的作用(动画的元素包含宽高、透明度和display显示)，还可传入第二个参数，是一个链式函数

拉下 .slideDown()

拉上 .slideUp()

切换 .slideToggle() 和上面的显示动画差不多，但是动画包含元素中没有宽这一个可动项

淡入 .fadeIn()

淡出 .fadeOut()

切换 .fadeToggle()

淡入到 .fadeTo()

自定义动画

.animate() 有四个参数，JSON格式的属性操作、动画执行时间、回调函数和动画速度(linear和swing)，并且在进行属性操作时可接受字符串，如width:’+=100’,就是每次操作都会在原基础上增加100，width:’hide’会使width隐藏，‘toggle’切换

.delay() 延迟 .finish()与stop类似，它会使全部动画立即完成

.stop() 停止正在运行的动画,通过该方法可以解决动画通道堵塞的现象 stop有四种传参情况：

1. 立即停止当前的，继续后面的 stop(false)
2. 立即停止所有的 stop(true)
3. 立即完成当前的，继续后面的 stop(false,true)
4. 立即完成当前的，停止后面的 stop(true,true)

$.fx.off 默认为false，如果为true，将关闭JQ所有动画的过程，直接显示动画最终效果

$.fx.interval 动画的帧数，默认13，值越大动画越卡，越小越流畅并且会消耗更大的浏览器性能

**查找筛选**

.children() 可通过传入类名或ID从父级中筛选到子元素

.find() 比上面的好用，这个可以无限向下查找

**节点操作**

.append() 参数为被添加的内容，默认添加到调用者内部最后

.appendTo() 和append()只是语法不同，参数为传入的元素，调用者为要被传入的元素(一下的,prependTo,insertAfter，insertBefore均为如此)

.prepend() 将内容添加到内部的最前位 .prependTo()

.after() 将内容添加到元素外的下面 .insertAfter()

.before() 将内容添加到元素外的上面 .insertBefore

**删除节点**：

.remove() 将调用者从文档中移出，不过可以用一个参数接收着他

.empty() 不会讲调用者直接移出，但是会清空它所有的子节点

.detach() 此方法和remove上述功能一样，他们的不同是，remove()方法会将调用者自己包括他身上绑定的所有方法统统删除，而detach()则会保留这些方法(可以通过变量接收被删除的元素，再重新将其添加到文档中进行测试，会发现被remove掉的元素身上的方法已经消失了，而被detach掉的元素身上仍保留着这些方法)

**替换节点**：

.replaceWith() 将调用者全部替换为参数里的元素

.replaceAll() 同上面的appendTo一样，与replaceWith调用者参数相反