1. $() jq选择符，
2. css() css样式 写一个参数是获取 ，写两个参数是赋值
3. html() html内容 不写参数是获取 ，写参数是赋值
4. click(funciton(){} ) 点击事件
5. attr() 操作属性的方法 写一个参数是获取，写两个参数是赋值
6. val() 不写参数是获取value值 写参数是赋值
7. 多元素取值（一个集合） 只能取到第一个
8. Windown.onload=funciton(){} == $( funciton(){} ) 并不完全一样 功能类似
9. 属性选择器：
10. $(‘ input[value]’) 选择有value值的来操作
11. $(‘ input[value=123]’) 选择value=123值的来操作
12. $(‘ input[value ^=123]’) 选择value为123开头的值来操作
13. $(‘ input[value $=123]’) 选择value为123结尾的值来操作
14. $(‘ input[value \*=123]’) 选择value有123值来操作

引号的问题 可以加也可以不加

如果值中间有空号 要加引号 例如：$(‘div’[class=”box box2”])

1. 链式操作针对的是设置操作

例如：$(‘div’).html(‘123’).css(‘color’,’red’).click(funciton(){})

获取操作不能使用链式操作

1. $(‘div’).size() 或者 $(‘div’).length 集合的长度 length

class的操作

1. addClass() 添加class
2. removeClass 删除class
3. toggleClass 只能判断增加或者删除class 有就删除，没有增加

显示隐藏：

1. hide() 隐藏
2. Show() 显示

节点的选择：

1. prev() 获取上一个兄弟节点 可以写参数 有筛选的功能
2. Next() 获取下一个兄弟节点 可以写参数 有筛选的功能
3. prevAll() 获取上一个所有兄弟节点 可以写参数 有筛选的功能
4. nextAll() 获取下一个所有兄弟节点 可以写参数 有筛选的功能
5. sibling() 所有的兄弟节点 可以写参数 有筛选的功能
6. first() 集合中的第一项
7. last() 集合中的最后一项
8. slice() 截取集合中的某项 两个参数 起始位置 和结束位置 不包 含结束位置
9. children() 获取子元素 参数是可以筛选和过滤 子节点里面的节点是选择不到的
10. find() 查找的范围更广泛

例如：$(‘ul’).find(‘p’).css() == $(‘ul p’).css() 尽量用第一种方法

1. parent() 找到父级元素
2. parents() 找到所有的的祖先级元素 是一个集合 参数具有筛选的功能
3. closest() 特别重要 实战之王 找当前元素指定的最近的祖先元素（包括自身） 必须要写一个参数 只能选择到一个唯一的元素

下标：

1. eq()

添加节点：

1. $(‘<div>’) 创建标签
2. append() 把元素添加到指定的节点的里面的最后
3. prepend() 把元素添加到指定的节点的里面的最前面
4. before() 把元素添加到指定节点的前面
5. after() 把元素添加到指定节点的后面
6. appendT0 () 把把元素添加到指定的节点的里面的最后
7. prependTo() 把元素添加到指定的节点的里面的最前面
8. insertBfore() 把元素添加到指定节点的前面
9. insertAfter() 把元素添加到指定节点的后面
10. remove() 删除节点
11. clone() 克隆 复制 参数为true 是可以克隆元素的操作行为
12. Index() 索引值代表的就是当前元素在所有兄弟节点中排第几

里面的参数是 筛选过滤

JQ中的遍历：

1. each 类似于 原生js的for循环

方法里接受一个函数 里面有两个参数 e I

跳出循环 return false

JQ包装对象：

1. wrap() 包装
2. wrapAll() 整体包装 会改变整体布局 用的比较少
3. wrapInner() 内部包装
4. unwrap() 删除包装 相当于删除父节点 body是不能被删除的

Jq转原生js 的方法

1. get() 默认情况获取的是一个集合，那么可以通过参数来指定集合中的指定项； 后面跟原生js

需要用到原生的js 获取内容的高度

元素之间的比较

获取元素的尺寸：

1. widht() height() 获取元素宽高 不包含padding 和 边框 margin

参数可以赋值

1. innerWidth() 包含padding 的宽度 不包含边框 margin

参数可以赋值

1. outWidth() 包含padding和边框的宽度 参数为true 时包含margin

参数可以赋值 outWidth(‘200’,true)

原生js是获取不到隐藏元素的宽高

Jq是可以获取到隐藏元素的宽高

1. 获取可视区的尺寸：

$(window).height()

1. 页面的尺寸：

$(document).height()

1. 滚动距离：

scrollTop() 参数可以设置滚动距离 可以获取也可以设置

scrollLeft()

滚动距离==页面高度-可视区高度

$(document).scrollTop() == $(document).height()-$(window).height()

元素距离：

1. offset() top left 元素到页面(不是可视区)的距离

例如 $(‘div’).offset().top

1. position().left 或者 top 不认margin值 获取到定位父级的距离

如果没有定位就是到页面的距离

1. 通过计算原理，得到相应值：

offsetParent() 找到定位父级

例如：

$(‘#div2’).offset().left-$(‘#div1’).offsetParent().offset().left

Jq的事件：

在jq中的事件操作全是绑定的形式

1. on() 比直接click 实现的功能更丰富 可实现多个事件的操作
2. Off() 取消事件 不写参数是全部删除 写参数是筛选出要取消的事件
3. event对象 ev.pageX ev.pageY 相对于页面
4. ev.cientX ev.cientY 相对于可视区
5. ev.which 键盘值
6. ev.target 目标元素 事件源
7. ev.stopPropagation() 阻止冒泡
8. ev.preventDefault() 阻止默认行为
9. return false

多次添加on的处理方式： 可以点击之前添加off() 清除事件

例如：$(‘#div1’).click(function(){

$(‘span’).off(‘click’).click(function(){

alert(1)

})

}）

Jq的事件委托：

1. delegate()

例如：$(‘ul’).delegate(‘li’,’click’,function(){])

事件委托的this 不是ul 是li

ev.delegateTarget 可以找到ul

undelegate() 取消委托 通过ul来取消