* 一、选择id：$(‘#div’)
* 二、选择标签：$(‘li’)
* 三、选择class：$(‘.box’)
* 四、选择所有li：$(‘li’)
* 五、选择第一个li：$(‘li:first’)
* 六、选择最后一个li：$(‘li:last’)
* 七、选择任意一个li：$(‘li:eq(i)’)或者$(‘li’).eq(i)
* 八、选择奇数行li：$(‘li:even’)
* 九、选择偶数行li：$(‘li:odd’)
* 十、选择li中class=‘box’的元素：
* ①$(‘li .box’)
* ②$(‘li’).filter(‘.box’)
* ③$(‘li’).filter(‘[class=box]’)
* 十一、选择ul下所有的元素子节点：$(‘ul’).children()
* 十二、内部含有：
* ①has() $(‘li’).has(‘span’)
* ②find() $(‘li’).find(‘span’)
* 十三、内部除了：not() $(‘li’).not(‘span’)
* 十四、选择下一个兄弟节点： $(‘li’).next()
* 十五、选择上一个兄弟节点： $(‘li’).prev()
* 十六、添加class：addClaa(‘box’)
* 十七、移除class：removeClass(‘box4’)
* 十八、A的父级：A. Parent()
* 十九、A的定位父级：A.offsetParent()
* 二十、祖先节点集合：A.parents()
* 二十一、最近的祖先节点：A.closet()
* 二十二、兄弟节点：A.siblings()
* 指定的兄弟节点：A.siblings(‘div’)
* 二十三、下面所有的兄弟节点：A.nextAll()
* 下面兄弟节点中指定的所有节点：A.nextAll(‘h2’)
* 二十四、上面所有的兄弟节点：A.prevAll()
* 上面兄弟节点中指定的所有节点：A.prevAll(‘h2’)
* 二十五、到某个祖先节点：A.prevUntil(‘body’)
* 二十六、所有祖先节点：A.parents() A.parentUntil() 无参
* 二十七、往上直到某个兄弟节点:A.prevUntil(‘h2)
* 二十八、往上所有兄弟节点:A.prevUntil()
* 二十九、往下直到某个兄弟节点:A.nextUntil(‘h2’)
* 三十、往下所有兄弟节点:A.nextUntil()
* 三十一、克隆节点:A.clone()
* 带操作克隆:A.clone(true)
* 三十二、包装
* ①每条外面加包装 A.wrap（B） 每条A的外面包装B
* ②整体外面加包装 A.wrapAll(B) 整体A的外面包装B
* 举例： <div>
* <span>span</span> <span>span</span>
* <p>p</p> 加包装后： <span>span</span>
* <span>span</span> <p>p</p>
* </div>
* ③内部加包装 A.wrapInner(B) A的 内部加包装B
* 举例：
* <span> <span>

span <div>

<p>p</p> 加包装后 span

* </span> <p>p</p>
* </div>
* </spam>
* ④删除包装（不删除内部包装，删除父级，且不包括body）： A.unwrap()
* 三十三、添加成为组合
* 模式：
* var elem=$(‘div’);
* var elem2=elem.add(‘span’)
* 三十四、指定节点范围 $(‘li’).slice(1,5) 含1，不含5
* 三十五、数据串联化
* 一、样式
* width（） 样式宽
* innerWidth（）可见宽
* outerWidth（）占位宽
* outerWidth（true）占位宽+margin
* 二、滚动条的滚动举例 scrollTop（）
* 模式：$(window).scrollTop();这个就是滚动条的滚动举例
* 三、到文档的距离和margin到文档的举例
* ①到文档的距离：
* A.offset().left A.offset().top
* ②margin到文档的距离：
* A.position().left A.position().top
* 四、值：val() 赋值与取值
* 五、文本：text()获取所有元素的内容，且只获取文本的内容

赋值会让所有元素的内容变为赋值的值

* + html()获取所有元素的第一个内容
* 一、用on定义事件
* 模式一：
* $(‘div’).on(‘click’,function(){

alert(1);

* })
* 模式二：
* $(‘div’).on(‘click.a’,function(){
* alert(1);
* })
* 给click事件加个命名空间，方便取消此时间
* 模式三：
* $(‘div’).on(‘click mouseover’,function(){
* alert(1);
* })
* 二、关闭事件
* 模式一：
* $(‘div’).off() 取消了div的所有事件
* 模式二：
* $(‘div’).off(‘mouseover’) 取消了div的mouseover事件
* 模式三：
* $(‘div’).off(‘mouseover’，fn1) 取消了div的mouseover事件的fn1
* 模式四：
* $(‘div’).off(‘mouseover.a’) 取消了div的命名空间为mouseover.a的事件
* 特别注意：
* $(‘div’).off(‘mouseover.a.c’) 取消了div的命名空间为mouseover.a.c的事件
* $(‘div’).off(‘mouseover.a’) 取消了div的命名空间为mouseover.a的事件,包括上面的mouseover.a.c的事件也取消了
* 三、event对象
* 鼠标位置：ev.pageX ev.pageY 相对于文档的
* 阻止事件：
* ①e.preventDefault 阻止默认事件
* ②e.stopPropagation 阻止冒泡 这是兼容的
* ③return false 阻止默认事件和冒泡
* 原生的事件对象：
* e.originalEvent 原生的事件对象，e.originalEvent.clientX 就相当于原生的e.clientX
* 四、事件只执行一次：one（）
* 举例：
* $(‘div’).one(‘click’,function(){

alert(1);

* })
* 九、事件委托与取消事件委托
* ①事件委托
* $(‘ul’).delegate（‘li’,’click’，function(){

$(this).css(…);

* }）;
* 这里的this指的就是li，这就是事件委托
* ②取消事件委托
* $(‘ul’).undelegate(); 取消了对ul的事件委托
* 十、主动触发：一打开就触发

$(‘’#div’).strigger(‘click’); 主动触发#div的click事件，当然了，得现有click的事件函数

* 十一、对象传参：ev.data、 ev.target与ev.type
* 举例：
* $(‘#div’).on(‘click’,{name:’hello’},function(ev){

alert(ev.data.name);//显示hello

* })
* 而这里的ev.target指的就是事件源，即被click的#div
* 而这里的ev.type指的就是事件类型，即click

1. animate()有哪些参数？

四个参数：

①、{} 属性：值

②、时间

③、运动形式 默认swing(慢快慢) 自身提供了swing 和 linear

④、callback

注意一：做完一个动作后再做，就再写.animate 这个是队列操作，不是堆栈

注意二：做完一个动作后再做，也可以写成

$(‘div’).animation

* $(‘div’).animation

1. 动画停止
2. stop(参数一,参数二);

参数一：是否清空队列；false：不清空，true：清空

参数二：是否立马跳到最后结果：

四种类型

1. 、A.stop() 停止当前运动，不停止后续运动
2. 、A.stop(true) 停止所有运动
3. 、A.stop(true,true) 当前运动马上到达目的地，后续运动停止
4. 、A.finish() 当即到达所有目的地
5. 延迟

delay(1000) 延迟1000ms

* 四、高亮hover（）
* 语法：
* hover（function(){…},function(){…}）;
* 五、显示与隐藏
* 显示：show(); A.show(1000); 1000ms后显示
* 隐藏：hide(); A.hide(1000); 1000ms后隐藏
* 六、淡入与淡出
* 淡入：fideIn() 可加时间，可加callback
* 淡出：fideOut() 可加时间，可加callback
* 淡入淡出到固定透明度 fideTo（time，opacity） 时间和透明度为参数
* 其、向下展开与向上卷曲
* 向下展开：slideDown（）可加时间，可加callback
* 向上卷曲：slideUp（）可加时间，可加callback
* 一、插入
* (1)
* A.insertBefore(B) AB
* A. Before(B) BA
* A.insertAfter(B) BA
* A.After(B) AB
* A.appendTo(B) B(A) A含在B里面，且在B的最后
* A.append (B) A(B) B含在A里面，且在A的最后
* A.propendTo(B) B(A) A含在B里面，且在B的最前面
* A.propend (B) A(B) B含在A里面，且在A的最前面
* 二、元素长度：$(‘li’).size()
* 三、循环：each（）
* 模式一：从第0个开始循环
* $(‘li’).each(function(i,elem){

$(elem).html(i);

* })
* 模式二：从第j个开始循环
* $(‘li：gt(j-1)’).each(function(i,elem){

$(elem).html(i);

* })
* 四、转集合并可用下标:$(‘div’).get(0) div的第0个
* 一、判断类型：$.type(a);
* 二、去掉前后空格：$.trim(str);
* 三、找数组中的位置
* 四、改变this指向的方法
* 五、防止冲突：
* 六、把字符串解析成json
* 七、把类数组转成真正的数组
* 八、调用插件工具
* $.contains();检查一个元素是否是另一个元素的后代 包含关系
* $.each({left:100,top:200},function(index,value){ 循环任何可以遍历的对象
  + console.log(value);
* })
* $.extend(要被扩展的对象,function(){}) 扩展作用，将扩展的内容接到被扩展的对象后面
* $.isQQ();