jQuery是一个js函数库

就绪函数：$(document).ready(function(){ }); $(function(){ });

js就绪函数：window.onload = function(){ }

两者的区别，jQuery是在html所有标签（dom）都加载之后，就回去执行。

Js等到所有内容，包括外部图片之类的文件加载完之后，才回去执行

选择器：所有的选择器都以$()开头

元素选择器：基于元素名选取元素 p标签：$(“p”)

Id选择器：id为test的p标签 $(“#test”)

.class选择器：class为test的p标签 $(“.test”)

属性选择器：所有href值等于‘#’的元素 $(“[href=’#’]”)

层次选择器：div里的所有span元素 $(“div span”)

$(“ul:eq(index)”)在匹配的集合中（ul）选择索引值为index的元素

$(“ul:gt(index)”)在匹配的集合中（ul）选择索引值大于index的元素

$(“ul:lt(index)”)在匹配的集合中（ul）选择索引值小于index的元素

更多实例：

$(“\*”) 选取所有元素

$(this) 选取当前html元素

$(“p.intro”) 选取class为intro的p元素

$(“p:first”) 选取第一个p元素

$(“ul li:first”) 选取第一个ul元素的第一个li元素

$(“ul li:first-child”) 选取每个ul元素的第一个li元素

$(“[href]”) 选取带有href属性的元素

$(“a[target=’\_blank’]”) 选取所有target属性值等于“\_blank”的<a>元素

$(“a[target!=’\_blank’]”) 选取所有target属性值不等于“\_blank”的<a>元素

$(“:button”) 选取所有type=“button”的input元素和button元素

$(“tr:even”) 选取偶数位置的<tr>元素

$(“tr:odd”) 选取奇数位置的<tr>元素

事件：

jQuery事件方法的语法：

$(“p”).click(); 单机后触发

$(“p”).dbIclick(); 双击后触发

$(“p”).mouseenter(); 当鼠标穿过元素触发

$(“p”).mouseleave(); 当鼠标离开元素时触发

$(“p”).mousedown(); 当鼠标指针移动到元素上方，并按下鼠标按键时触发

$(“p”).mouseup() 当在元素上松开鼠标按钮时触发

$(“p”).hover() 当鼠标移动到元素上时，触发

$(“p”).focus() 当元素获得焦点时候，触发

$(“p”).blur(); 当元素失去焦点时，触发

隐藏和显示

Hide()隐藏 show()显示

慢慢隐藏--hide

$(“div”).hide(1000,”linear”,function(){ }) 1000-隐藏的速度，“linear”-表示过渡使用那种函数（jQuery自身提供两种”linear-线性”,”swing-摆动”） function(){}-表示完成隐藏后调用的函数

jQuery toggle()通过jQuery可以使用Toggle方法来切换hide（）和show（）方法

显示被隐藏的元素 隐藏被显示的元素

$(selector).toggle(speed,callback) speed-”slow,fast,毫秒”

淡入淡出

Fadeln() fadeOut() fadeToggle() fadeTo()

1. 慢慢显示已经隐藏的元素 $(selector).fadeIn(speed,callback) speed:”slow,fast,毫秒”
2. 慢慢隐藏已经显示的元素
3. 来回切换
4. $(selector).fadeTo(speed,opacity,callback) speed:”slow,fast,毫秒” opacity:透明度

滑动：

slideDown() slideUp() slideToggle()

1. 向下滑动元素 $(selector).slideDown(speed,callback)speed:”slow,fast,毫秒”
2. 向上滑动元素 $(selector).slideUp(speed,callback)speed:”slow,fast,毫秒”
3. 两者之间切换

动画：

Animate()方法用于创建自定义动画

$(selector).animate({params},speed,callback);

Params参数定义形成动画的css属性

Speed “slow fast 毫秒”

callback返城之后的函数

停止滑动：

Stop()方法

$(selector).stop(stopAll,goToEnd)

stopAll参数规定是否应该清除动画队列--停止正在活动的动画

goToEnd参数规定是否立即完成当前动画

Callback方法：

方法后的callback,执行完方法后的调用。否则直接调用

链条：

可以把多个动作方法连接在一起

例：

$(“#p1”).css(“color”,”red”).slideUp(2000).slideDown(2000)

捕获与设置：

jQuery Dom操作

text:设置或返回元素的文本内容

Html:设置或返回元素的内容（html）  
 val():设置或返回表单字段的值--设置内容在括号上填写响应的值

attr获取属性的值.attr(“属性名”)--设置内容在括号上填写响应的值

上面的四个方法都有回调函数，回调函数有两个参数：被选元素中当前元素的下标，以及原始的值

Text(function(i,origText){

I:下标， origtext原始的值

})

添加元素：

.append(“追加文本”)

.prepend(“在开头追加文本”)

.after(“在元素之后插入内容”)

.before(“在元素之前插入内容”)

删除元素：

.remove(“删除被选元素以及子元素”)

$(“p”).remove(“.italic”)--删除class等于italic的所以p元素

.empty(“删除被选元素的子元素”)

Css类：

.addClass(“blue”)向被选元素添加一个或多个类

.removeClass(“”)从被选元素删除一个或多个类

.toggleClass()对被选元素进行添加/删除类的切换操作

.css()设置或返回样式属性

Css方法：

$(“p”).css(“background-color”)返回属性的值

$(“p”).css(“background-color”,”yellow”)设置属性的值

$(“p”).css(“{“background-color”:”yellow”,”font-size”:”200%”}”)设置多个css属性

尺寸：

Width():获取宽度

Height():获取高度

innerWidth():返回元素的宽度（包括内边距）

innerHeight():返回元素的高度（包括内边距）

outerWidth():返回元素的宽度（包括内边距和边框）

outerHeight():返回元素的高度（包括内边距和边框）

遍历：

利用其他元素的关系来查找（或选取）html元素

祖先：

Parent():方法返回被选元素的直接父元素

Parents():方法返回被选元素的所有祖先元素，一路向上直到文档的根元素（html）

parentsUntil():方法返回介于两个给定元素之间的所有祖先元素  
后代：

Children():方法返回被选元素的所有直接子元素

Find():方法返回被选元素的后代元素，一路向下直到最后一个后代  
同胞：

Siblings():返回被选元素的所有同胞（同级别）元素  
 next():返回被选元素的下一个同胞元素

nextAll():返回被选元素的所有跟随的同胞元素

nextUntil():返回介于两个给定参数之间的所有跟随的同胞元素

Prev():返回被选元素上一个同胞元素

prevAll():返回被选元素前面的所有同胞元素

prevUntil():也是中间的同胞元素，只不过是从下往上查找

过滤：

Fitst()：返回被选元素的首个元素

Last():返回被选元素的最后一个元素

Eq():$(“p”).eq(1);返回被选元素中电带有指定索引号的元素

Filter():允许定义一个标准，不匹配这个标准的元素会被从集合中删除，匹配的元素会被返回

Not():返回不匹配标准的所有元素

Ajax:

简介：局部刷新 与后台交互

Load():

Load（）：方法是简单但强大的ajax方法

此方法从服务器加载数据，并把返回的数据放入被选元素中

$(selector).load(url,data,callback)

Url:路径 data：参数 callback完成之后执行的函数

Get()/post():

Get：发送请求，获取数据

Post：发送请求，提交数据

noConflict():

$就是jQuery 如果其他框架也用$ 那么可以用jQuery代替

但是之前要先释放对$的控制 $.noConflict()

或者定义自己的符号或名称：var jq = $.noConflict();

JSONP:

$.getJSON(“网址”,function(data){

})

练习：<https://www.runoob.com/jquery/jquery-examples.html>

插件：<https://www.runoob.com/jquery/jquery-plugin-validate.html>

jQuery选择器：元素选择器 id选择器 class选择器 eq选择器 属性选择器 层次选择器 基本过滤选择器 内容过滤选择器

基本选择器：

所有元素$("\*")

当前元素$(this)

所有p元素$("p")

id选择器

id等于id的元素$("#id")

class选择器

class等于class的元素$(".class")

eq选择器

ul中第一个li元素$("ul li:eq(0)")

ul中下标大于1的li元素$("ul li:gt(1)")

ul中下标小于1的li元素$("ul li:lt(1)")

属性选择器:

带有href属性的元素$("[href]")

带有href属性且等于132的元素$("[href='132']")

带有href属性且不等于132的元素$("[href!='132']")

带有href属性且后缀为.jpg的元素$("[href$='.jpg']")

带有href属性且值等于或以它开头的元素且后面以连接符-相关联$("[href|='tom']")

带有href属性 且值以tom开头的元素或等于tom的元素$("[href^='tom']")

带有title属性 且值包含hello（只包含不等于）$("[href~='tom']")

带有title属性，且值包含hello（包含也等于）$("[href\*='tom']")

带有id属性且name属性的值以man结尾的input（多条件）$("input[id][name$='man']")

层次选择器：

$("p:last-child")被父元素包含且是最后一个元素的p元素（父元素不可以是body）

$("p:first=child")被父元素包含且是第一个元素的p元素（父元素不可以是body）

$("p:first-of-type")被父元素包含且是第一个p元素，不必是第一个元素（父元素可以使body）

$("p:last-of-type")被父元素包含且是最后一个p元素，不必是最后一个元素（父元素可以是body）

$("div p")div下的所有是p元素的子元素

$("div>p")div的儿子元素是p元素

$("div+p")div离着最近的兄弟元素是p元素

$("div~p")div的所有兄弟元素

基本过滤选择器：

第一个p元素$("p:first")

最后一个p元素$("p:last")

索引为偶数的tr元素$("tr:even")

索引为奇数的tr元素$("tr:odd")

所有的input元素$(":input")

所有启用的元素$(":enabled")

所有禁用的元素$(":disabled")

所有选中的复选框$(":checked")

所有被选中的下拉框$(":selected")

$(":file,:text,:button,:image,:reset,:checkbox,:radio,:password")

所有type等于这些的input元素

内容过滤选择器

所有不为空的input元素$("input:not:(:empty)")

所有class不为abc的input元素$("input:not:(.abc)")

所有标题元素 h1 h2 h3 等$(":header")

所有动画元素$(":adimated")

当前具有焦点的元素$(":focus")

所有包含文本hello的元素$(":container('Hello')")

所有空元素$(":empty")

所有含有子元素或者文本的父元素$(":parent")

所有隐藏的p元素$("p:hidden")

所有可见的表格$("table:visible")

文档的根元素（html标签 也就是全部的元素了）$(":root")

所有包含p元素在内的div元素$("div:has(p)")

选择class1下的class2的元素$(".class1 .class2")

选择同时含有class1和class2的元素$(".class1.class2")

选择class1或者class2的元素$(".class1,.class2")

z-set

添加

修改

删除

删除setid集合中的最高分

删除setid集合中的最低分

删除setid集合中指定的value

删除setid集合中下标从1到2的数据

删除setid集合中分数从fromscore到toscore的集合数据

查询

通过setid获取value和score

正着

倒着

正着（带分数）

通过setid和value获取下标

正着

倒着

通过排名（下标）的区间获取value和score

(从高到低)

(从高到低)score

（从低到高）

（从低到高）score

通过score区间获取value和score

(从高到低)

(从高到低)score

（从低到高）

（从低到高）score

判断是否存在

总数量