jQuery

$   等价于    jQuery

jquery选择器  ：

基本选择器 ：   $("#id")   $(".class")  $("p")  $("p,.class")  $("\*")

层级选择器 :    空格    >      +     ~

1、简单选择器

:first    $("li:first")   查找第一个li元素

:last     查找最后一个元素

:odd      查找下标为奇数的行

:even     查找下标为偶数的行

:gt       $("li:gt(n)")   n>=0  查找下标大于某个数的行

:lt       查找下标小于某个数的行

:has      $("li:has(ul)")  查找包含ul的li元素

:not      $("li:not(.box)")   查找li中类名不是box的li元素

:eq       $("li:eq(n)")  n>=0   查找第几个li

2、查找过滤选择器（方法）

.first()    $("p").first()

.last()     $("li").last()

.eq()       $("li").eq(n)    查找第n个li元素

.next()     $("#box").next()    查找id为box的元素的下一个兄弟元素

.next()            $("#box").next(“p”)    查找id为box的元素的下一个兄弟元素p

.nextAll()  查找某个元素后面所有的兄弟元素

.nextAll("p")  查找某个元素后面所有的兄弟元素p

prev()   prevAll()   同next()   nextAll()

parent()    查找某个元素的父节点

find()      $("div").find("span")  查找div中所有的span元素     $("div span")

children()  $("div").children()  查找div中的所有孩子元素

            $("div").children("span")  查找div中所有的孩子元素span   $("div>span")

filter()    $("p").filter(".box")  查找p中带有box类名的p     $("p.box")

            $box =  $("p").filter(function(index){  //根据函数的返回值查找元素

                    //如果返回值为true  表示按照该条件查找到了某个元素 ，如果返回值为false，表示没有找到任何元素

                    参数 index 表示 p的索引下标

            })

siblings()  表示除了本身之外的所有兄弟元素

3、样式操作  css()

获取样式  ：

     $().css("样式名称")

     例如：     $("li").css("color")   获取li的字体颜色

设置样式 ：

     $().css("样式名","样式值")   只能操作一个样式   同js中的 .style......

     $().css({"样式名":"样式值","样式名":"样式值",.....})  一次可以操作多个样式   同js中的.style.cssText = ""

实例：　　页面加载　奇偶行颜色不同　，鼠标移入高亮显示　　鼠标离开颜色恢复

４、jquery的事件

$().事件(function(){

     .....

})

$(document).ready(function(){

})

5、jquery对象和js对象的互换

1、假设 obj  是一个js对象  ，转成jq对象：

   $( obj )

2、 假设 $(obj) 是一个jq对象 ，转成js对象:

   $(obj).get(下标)

   $(obj)[下标]

6、样式类操作

addClass()   为指定的元素添加一个或多个样式类   （类似：className）

removeClass()  为指定的元素删除一个或多个样式类

toggleClass()  为指定的元素切换某个样式类        判断某个元素是否含有某个样式类  如果有就删除  没有就添加

（扩展）hasClass()     判断某个元素是否含有某个样式类  返回一个boolean

7、jquery中获取下标方法

index()

     1、没有参数   表示获取该元素在同辈元素中的索引位置

     2、如果参数是一个对象  表示这个对象在原集合中首次出现的位置

     3、如果参数是一组对象  表示这组对象中的第一个元素在原集合中的索引位置

8、属性操作   attr()

获取属性值

     对象.attr("属性名称")

设置属性值

     对象.attr("属性名","值")  只能操作一个属性值

     对象.attr({"属性名":"值",....})

removeAttr() 删除某个元素的某个属性

如果要操作的属性值为布尔类型，则使用 prop （例如：checked ）

9、属性选择器

$("div[class]")     表示div中含有class属性的div元素

$("div[class=box]")  表示div中class名字为box的div元素

$("div[class^=box]") 表示div中class名字以box开始的div元素

$("div[class$=box]") 表示div中class名字以box结束的div元素

$("div[class\*=box]") 表示div中class名字包含box的div元素

$("div[class!=box]") 表示div中class名字不包含box的div元素

10、内容操作

html()

text()

val()     操作表单

获取内容：

     对象.html()    获取一个元素的内容

     对象.text()    获取被查找的所有元素内容

     对象.val()     获取表单内容

设置内容：

     对象.html("新内容")

     对象.text("新内容")

     对象.val("新内容")

11、文档操作

append()      父元素.append(子元素)

appendTo()    子元素.appendTo( 父元素 )     向后面追加   （内部操作）

prepend()     父元素.prepend(子元素)

prepend()     子元素.prependTo( 父元素 )    向前面 前置   （内部操作）

after()       元素1.after(元素2)    向元素1的后面添加元素2   （外部操作）

insertAfter()  元素2.insertAfter(元素1)  将元素1添加到元素2的后面

before()      元素1.before(元素2)   向元素1的前面添加元素2    （外部操作）

insertBefore()  元素2.insertBefore(元素1)  将元素1添加到元素2的前面

empty()     清空某元素

remove()    删除某元素

clone()     复制某元素

12、表单选择器

:text   找到所有的文本框

:password  找到所有的密码框  type="text/password/button/radio/checkbox/submit/reset... "   type中的值都可以通过  : 来查找

:checkbox:checked  表示找到被选中的复选框

:checked  找到所有被选中的单选或复选按钮

:selected 找到被选中的下拉列表

:input 找到所有的表单项 （包括textarea/select）               $("input")

1、基本动画

show()     显示

     第一个参数：动画执行时间   默认毫秒  （可省略）

     第二个参数：动画执行方式   swing   linear  （可省略）

     第三个参数： 回调函数

hide()     隐藏    参数同上

toggle()   显示/隐藏   参数同上

2、slideDown / slideUp    下拉 / 上滑

slideDown()    通过改变元素的高度 动态的 显示某个元素

slideUp()      通过改变元素的高度 动态的 隐藏某个元素

slideToggle()  切换  slideDown/slideUp

3、淡入/淡出

fadeIn    淡入  通过不透明度的变化动态的显示某个元素

fadeOut   淡出  通过不透明度的变化动态的隐藏某个元素

fadeToggle   切换

fadeTo()   将某个元素的透明度改为某个目标值

           两个参数　　 第一个参数 ：　时间（不可以省略）　　　　第二个参数：　透明度的目标值

４自定义动画

animate()

第一个参数：  {"属性":"目标值",.....   }

第二个参数:  动作完成时间   毫秒

第三个参数:  动作方式  swing  linear

第四个参数:  回调函数

stop()  停止动画

5、jquery的遍历

$().each(function(index,ele){

     index:  遍历对象的下标

     ele:    遍历的对象    dom对象

})

1、jquery的位置操作

offset()   获取距离文档顶部 和 左侧的距离   结果是一个对象     有两个属性  left    top

position()  获取距离具有定位的父元素的顶部 和  左侧的距离  结果是一个对象  有两个属性 left  和  top

end()   结束离他最近的前面的选择器

2、jquery的事件

事件三要素：  事件源    事件    事件处理程序

事件绑定 bind：

对象.bind("事件",function(){   为某个对象绑定一个事件，一个处理程序

})

对象.bind("事件1 事件2",function(){ 为某个对象绑定多个事件   一个处理程序

})

对象.bind({                        为某个对象绑定多个事件   多个处理程序

     "事件1":function(){},

     "事件2":function(){},

     .....

})

取消事件绑定  :  unbind()

 对象.unbind("事件");

事件委托 ：delegate()

父级对象.delegate("子对象","事件",function(){

     .....

})

取消事件委托 ： undelegate()

父级对象.undelegate("事件")

on()  既可以实现绑定  又可以实现委托

off()  取消绑定  或 委托

绑定：

对象.on({

     "事件1":function(){},

     "事件2":function(){},

     ....

})

委托：

    父级对象.on("事件","子对象",function(){

    })

one :

 对象.one("事件",function(){

 })

hover  表示事件的切换  自动切换mouseover和mouseout 两个事件

对象.hover(function(){

     响应mouseover 事件

},function(){

     响应mouseout  事件

})

3、jquery的滚动条位置

scrollTop()    获取滚动条垂直方向滚动的距离

scrollLeft()   回去滚动条水平方向滚动的距离

4、jquery的宽度和高度的获取

width() /  height()   获取内容实际的宽度和高度

innerWidth()  /  innerHeight()   获取的宽度和高度包括补白  不包括边框

outerWidth()  /  outerHeight()   获取的宽度和高度包括补白和边框

5、jquery的ajax异步加载 之 load

load() 方法有三个参数

第一个参数：  请求数据的url

第二个参数：  发送的数据 data  格式json格式    {  }

第三个参数：  回调函数

load()  方法是一个局部方法  ，通过jquery的选择器来调用的  选择器.load()

load()  方法是异步的，加载的信息中如果需要响应事件，必须通过回调函数来完成

load()  方法如果要请求不同结构的内容，可以把这些内容写到一个文件中，请求数据时  在url后面写一个选择器的名称即可（这里的选择器一般是基本选择器）

6、ajax的异步请求

$.get(请求路径,{data},function(msg){

     回调函数中的参数 就是请求服务器的数据

})

$.post()

$.getJson("url",function(msg){

     回调函数中的参数 就是请求服务器的数据

})

$.getScript("url",function(msg){

     立即执行ajax请求的js代码

})

$.ajax()

第一种方法：请求成功   通过success属性 的回调函数获取请求的数据

$.ajax({

          url:"http://localhost/jqueryAjax08/data.json",

          type:"get",

          data:{"uname":"lichune"},

          success:function(msg){

              alert( msg.name );

          },

          error:function(){

              alert("请求失败了");

          },

          complete:function(){

              alert("请求完成");

          }

     })

第二种用法：

    var deff = $.ajax({

          type:"get",

          url:"http://localhost/jqueryAjax08/data.json",

          data:{"uname":"lichune"}

     });

     deff.done(function(msg){  //请求成功  执行done

          alert(msg.name);

     })