54课

jquery使用的三种方法:

1. $(function(){})
2. $(document).ready(function(){ })
3. $(window).on(‘load scroll’,function(){ console.log(‘1’)}) //加入绑定的事件

$ 这是函数 ，一个jquery函数 他内部自己定义的

1.在使用$（）时候这是个jq函数 2. 在使用$. 使用点的时候是一个jq对象

$是jQuery对象是一个包含所有匹配的任意多个dom元素的伪数组对象

//给绑定文档加载完成的监听

$(this).index()

返回的下标是根据整体的子节点来说的

jQuery(function(){

在click后面加的是个括号（） 括号里面可以写函数

jq内容 }) 也可以把jQuery变成$

1. jq 核心函数 简称jq函数

引入jq库后可以直接使用$即可

当函数用是:$(\*\*\*)——————》 $(function(){})

当对象用：$.\*\*\*()--------🡪 $.extend({ }) 使用jq对象的extend的方法，添加插件

1. jq核心对象 简称jq对象

得到jq对象：执行jq函数返回的就是jq对象

使用jq对象：$obj.\*\*\*()

this是发生事件的DOM元素

在jq中

$(function(){

$(‘#xxx’).click(function(){

alert(this.innerHTML); 等于同下

alert($(this).html() )

})

})

var $button=$(‘button’) 这个是下面使用的变量

1.$(‘<div class=”div”></div>’).appendTp(‘div’) appendTo把谁添加到谁里面

2.使用$.trim() 去除左右的空格 括号中可以写入字符串

3. size()/length 这两个包含的dom元素的个数 例如 console.log($button.size(),$button.length)

4.[index]/get(index) 得到对应位置的dom元素 例如：console.log($button[1],$button.get(1))

5.each() 遍历所有的dom元素 例如：

$button.each( function ( index, domEle){ console.log(index,domEle.innerHTML,this) })

在上面中index就是遍历的个数以下标的形式排列，domEle就是元素 也可以不写可以console.log(this)直接遍历输出内容

6. index() 得到所在兄弟元素中的下标 例如：console.log($(‘.button3’).index())

伪数组 具有的属性：1.Object对象 2.length属性 3.数值的下标属性 就是获得的多个元素

例如 <div></div> <div></div> <div></div> <div></div> var $weiArry=$(‘div’)

看见没有就是获得的多个元素

基本选择器 改变样式 例如：

$(‘#div1’).css(‘background’,’red’) 改变css样式 也可以$(‘#div1’).css(‘background’:’red’) 使用冒号

你自己改变多的属性和值 看看用的是不是用冒号

层次选择器

$(‘ ul>span’) 选择ul下的所有的子元素span 在给定的父元素下匹配所有的子元素 获取的子元素用大于号获取

$(‘ .box+li’) 选中class是box的下一个li

选中元素后面的指定元素或者所有的元素

(‘.div .class1~.class2’)

选中class1后面的指定的元素class2 例如：$(‘.div .class1~.class2’).css(‘background’,’red’) 这个是选中的是div后面的class1的后面的所有的class2的元素 注意那个~是个波浪号啊 波浪号

也可以是这样 $(‘.div .class1~\*’).css(‘background’,’red’) \*是全部的元素

过滤选择器

$(‘div:first’)

div:first 选中是第一个div的元素

div:last 选中的是最后一个div的元素

$(‘div:not(.box)’) 选中div里面不是clsss是box的元素 注意注意啊 这个not（）是指匹配元素的那个属性的内部 不是元素内部的元素 就是

<div class=”div”>11</div> <div class=”div” id=””div>11</div> <div class=”div”>11</div>

$(‘.div’).not(‘#div’).css 这个选中的是1和3元素

$(‘li:gt(0):lt(3)’) gt是大于 lt是小于 li是个伪数组 选择大于下标是0的元素 选出来那几个后 然后在选择小于下标是3的元素 是选择出来后再 从选择的那几个元素 在选择新的元素 依次执行

$(‘li:contains(“qqqq”)’) 选择li元素里文本内容是qqqq的元素 contains是包含

$(‘li:hidden’) 隐藏的元素 $(‘li:hidden’)[0]是第一个隐藏的元素

$(‘li[title]’) 选择li中有title的属性的元素

$(‘li[title=hello]’) 也可以是$(‘li[title=”hello”]’) 选择属性是title的属性 属性值是hello 符号可带可不带

:odd 下标是奇数行的

:even 下表是偶数行的

任务十 jq表单选择器

:text 选择所有的单行文本框

.html() 选择的是元素的文本内容

:option 是选中option的标签 或者也是 :selected属性值

任务11 jq工具方法

$.isArray(obj) 判断是否是数组 obj写入的是 看他是不是一个数组 返回true 和false

不知道什么是 JSON 和JSON.parse 不知道

任务12 jq常见效果2 多tab点击切换

$(this).index() 返回的是根据this对象的整个节点来算的 就是兄弟节点的个数

dom是dom jq是jq 在一行里不能混合使用

$( 这里可以写dom创建的变量 ) 就是封装成jq的对象 其实我也不懂

然后用jq的语法进行操作

任务13+jq的属性 、

$(‘div’).attr(‘title’) 设置或返回被选元素的属性值 div里面的寻找属性是title的属性值

$(‘div’).attr(‘name’,’wqrwerewr’) 向div里面加入name的属性 属性值是那一大堆 后台能看见

$(‘div’).removeAttr(‘title’) 删除div里面的title、的属性

$(‘div’).attr(‘class’,’class1’) 给div里面添加类class class名字是class1 如果元素之前有class的话那么设置的class会替换之前的class 会替换

$(‘div’).addClass(‘abc’) 给元素添加class=‘abc’ 就是在元素里面<div class=”abc”></div

但是他不会替换掉之前的class值 都会加载一起得

$(‘div’).removeClass(‘class1’) 删除掉class值是class1的那个值

$(‘div’).html() 得到div的文本的内容

$(‘div’).html(‘<h1>wer wetwet</h1>’) 向div里面加入内容

$(‘:text’).val() <input type=”text” value=”wetwett”>. ，冒号获取文本文档的类型获取type是

tex的属性值 .val是获取vlaue的内容值

$(‘:text’).val(‘asetasetest’) 设置元素的value的值是那个一大堆 val()是读写合一的

$(‘:button:first’).attr(‘checked’,false) 获取type是button的元素 添加属性checked是false 是不选button的的按钮

$(‘:button:first’).attr(‘checked’,ture) 获取type是button的元素 添加属性checked是ture 是全选button的的按钮

attr() 操作属性值为非布尔值的属性

prop() 专门操作属性值为布尔值的属性

任务14 jq的复习

$(‘: ’) 括号的：冒号 就是匹配type是xxx的元素 ：是type 除了input

$(‘:checked’) 找到选中的选框

$(‘div:eq(1)’) 找到所有div的第二个div元素 eq(元素的下标) eq（）找到给定索引的元素

$(‘div:not(p)’) 找到所有的div元素中不是p元素的元素

$(‘:hidden’) 找到所有的隐藏元素

$(‘:visible’) 找到所有的显示元素

$(‘[type]’) 找到所有的属性是 type的属性

$(‘[type=text]’) 找到属性值是type 属性值是text 的元素

任务15 16 17jq css

$(‘p’)..css( { color:”red”,font-size:”10”,width:”100”} ) 其中可以省略ps不写 用{}吧css封装成对象

$(‘div’).offset().left 元素相对于窗口左边的距离 上边的就是top

$(‘div’).position().left 元素相对于父元素左边的距离 上边就是top

注意 注意 这个是距离他父元素是多少px 这个是相对于他父元素的偏移量而定的

$(‘div’).scrollTop() 获取元素滚动条的坐标

$(‘body’).scrollTop() 获取网页的滚动条的坐标

$(‘html’).scrollTop()+$(‘body’).scrollTop() 兼容不同的浏览器

和上面的一样$(document.document.Element’).scrollTop()+$(‘document.body’).scrollTop()

最简洁的方法就是$(‘html.body’).scrollTop()

这是获取的滚动条的垂直的坐标

$(‘html.body’).scrollTop(300) 设置滚动条到达的位置 $(‘html.body’).scrollTop() 滚动条距离的位置

获取的水平滚动条的坐标是$(‘html.body’).scrollLeft()

任务18常见的效果

跳过

任务19

1. 内容尺寸 innerHeight() innerWidth() wdith和height
2. 内部尺寸 innerHeight() height+padding
3. 外部尺寸 outerHeight(false/ture): height+padding+border 如果是ture 就加上margin

任务20

$div.filter(‘[title=hello]’) 获取div中属性是title 属性值是hello filter是过滤

$div.not(‘[title=hello]’) 和上面相反

$div.first() 也是获取的第一个元素

$(‘div’).has(‘span’) 获取div中只含有span的元素 去掉那些不含有指定后代的元素

任务21 jq对象的查找

$(‘div’).parent() 找到div的父元素 可以打多个parent

$(‘div’).prevAll(‘li’) 找到div之前元素的li元素 和div同辈元素的上面的元素都是 是同辈元素

$(‘div’).siblings(‘li’) 找到div中所有的li的兄弟元素

任务22 全选和全不选 不懂

$(‘:checkbox’).prop(‘checked’,true) 找到所有的复选按钮 用prop专门操作布尔类型的值 用来全部选中复选按钮 如果是(‘checked’,’false’) 就是全部不选按钮 prop专门操作布尔类型的值

实现反选 $(‘:checkbox’).each(function(){this.checkes=!this.checked})

任务23 jq删改

$(‘div’).children(‘ul’) 获取所有div中子元素（不包含后代元素）是ul的元素

$(‘.div’).before(‘<p>你好</p>’) 在div元素的前面添加元素p before是在什么的前面

$(‘.div’).after(‘<p>你好</p>’) 在元素的后面添加p元素 -

$(‘.div’).replaceWith(‘<p>nihao</p>’) 使用replaceWith 进行内容替换

$(‘.div’).replaceAll(‘<p>nihao</p>’) 替换元素的 注意是前者替换后者

$(‘.div’).empty.empty() 清空div元素中的元素

$(‘.div’).remove() 清除div元素

任务24 添加删除记录

不懂

任务25 jq事件处理

on(‘mouseenter’,function(){}) 括号中前者执行的事件 后者执行的函数

mouseenter 鼠标移入事件

mouseleave 鼠标移出事件

hover(function(){},function(){}) hover有两个参数 前者是鼠标移入的事件 后者是鼠标移出的事件

off(‘mouseenter’) 解除鼠标移入的监听

off() 解除所有的监听

offsetX 原点为事件元素左上角

clientX 原点为窗口的左上角

pageX 原点为网页面的左上角

任务26面试题

mouseover:在移入子元素的时候也会触发 对应的mouseout

mouseenter 只在移入当前元素时触发 对应的mouseleave

前者如果没有子元素 那么两者是没有区别

任务27 事件委托 不懂

$(‘ul’).delegate(‘li’,’click’,function(){}) delegate是事件代理元素 注意是获取的是父元素去操作的子元素

$(‘,div’).undelegate(‘click’) 移出事件的委托 就是移出本事件的点击事件

if(confirm(‘qweqwe’)) {} 这个是确认框

* 任务28 复习
* 任务29 tab栏平滑翻页
* 任务30循环翻页

任务31自动翻页功能

* 任务32翻页时同步更新圆点
* 任务33点击圆点翻页
* 任务34解决快速点击翻页bug
* 任务35内置动画
* 任务30

淡入淡出 效果3个 默认带动画的

$(‘div’).fadeout(3000,function(){ alert(“ert”)}) 3秒让一个元素逐渐消失,动画结束后执行回调函数

$(‘div’).fadeIn(3000,function(){ alert(“ret”)}) 3秒让元素逐渐显示,上面类同

$(‘div’).fadeToggle(3000) 这是个开关 点击3秒显示在点击3秒消失

滑动效果 默认带动画的

$(‘div’).slideup(3000) 内容收缩时间是3秒 元素向上滑动 3秒后会消失

$(‘div’).slideDown(3000) 内容展开时间是3秒

$(‘div’).slideToggle(3000) 开关 可以展开也可以收缩

瞬间显示或者消失 默认不带动画的

$(“div”).hide(1000) 用1秒时间隐藏 默认瞬间完成

$(“div”).show(1000) 用1秒时间显示 他会慢慢的改变元素的透明度

$(“div”).toggle(1000) 进行切换的显示和隐藏的动画

任务36自定义动画的练习

$(‘div’).animate({width:1000 如果要加px就这么写“200px” },3000) 意思就是执行动画animate

在3秒内 执行{ }这个里面的样式

$(‘div’).animate({ },1000).animate({ },3000) 用一秒的时间执行第一个对象内部的内容，用三秒的时间执行第二个对象的内容

$(‘div’).animate({ left: ‘+=100’},1000) 在原来的元素的基础上进行操作

$(‘div’).stop() 停止div元素在执行的动画 可以加入点击事件

任务37 多库共存

任务38 window 和$(document)区别

window.onload=function(){}

包括页面的图片加载完成后才会回调 比较晚一点

这个只能加一个监听回调 不能使用多个window.onload=function(){}

window.onload=function(){} window.onload=function(){}

这样浏览器会使用第一个把下面的都给覆盖了

而$(function(){}) 也就是$(document).ready(function(){ })

页面加载完就回调 比较早

可以加入多个监听回调 可以写入多个$(document).ready(function(){ }) $(document).ready(function(){ })

任务39

区别$和$()的问题 $是jq文件的内置对象 $( )

还有正则表达式

在jq对象中可以向其中添加方法就是用 $.extend({ }) 向其中添加

在调用直接 $.方法的名臣 来使用