Jquery

一、javascript类库(JS库)

JS库就其实是一个js文件

1、定义：预定义了很多的对象(方法和属性)和函数

2、目的：为了简化Js的开发

3、四个方面：

(1)JQuery，针对PC端WEB

(2)JQuery UI，针对UI布局效果

(3)JQuery Mobile，针对移动端WEB

(4)QUnit，专门用于测试Js

4、$("选择器");

它是JQuery的【工厂函数】

参数是选择器的内容；返回值是JQuery对象(不是DOM)

作用：接受JQuery的选择器内容

比如var $user=$("username");

$user前面的$可以省略不写

5、使用JQuery的步骤

(1)在HTML页面引入JQuery文件

(2)使用JQuery的选择器定位元素

(3)使用JQuery的API完成需求

6、JQuery对象和DOM对象

(1)概念

DOM对象：通过DOM获取的HTML页面元素

JQuery对象：封装DOM对象后所产生的对象

JQuery对象的底层还是DOM对象

(2)转换

①JQuery对象转换为DOM对象

I JQuery对象是数组对象，写法为 JQuery对象[索引值];

II JQuery提供get(index)方法，index表示索引值

②DOM对象转换为JQuery对象

使用JQuery的工厂函数，写法为 $(DOM对象)

\*注：DOM对象与JQuery对象的属性或方法【不可以】相互调用

7、读取超链接内文字

$("a").text()

等效于DOM中的 var a=document.getElementsByTagName("a")[0];

console.log(a.innerHTML);

8、JQuery的特点

(1)具有相对完善的处理机制 -- 并不报错

(2)链式操作

(3)隐式迭代<-->显式迭代

二、JQuery选择器

JQuery选择器的用法类似于CSS中的选择器

1、选择器的用法：

(1)基本选择器

#id

.className

element

\*

selector1,selector2 -- 结果为【并集】

(2)层级选择器

sel1 sel2，祖先元素与后代元素

sel1>sel2，父元素与子元素

sel1+sel2，指定元素的下一个元素

sel1~sel2，指定元素后边所有的兄弟元素

(3)过滤选择器

用法：在选择器前增加“:”

①基本过滤选择器

:first，匹配第一个

等效于 first()方法

:last，匹配最后一个

等效于 last()方法

:even，匹配索引值为偶数

:odd，匹配索引值为奇数

:eq(index)，匹配索引值为index的元素

等效于 eq(index)方法

:gt(index)，匹配索引值大于index的元素

:lt(index)，匹配索引值小于index的元素

:not(selector)，匹配与selector选择器相反的元素

:header，匹配h1~h6所有的标题元素

:animated，匹配JQuery的动画效果

②子元素过滤选择器

用法：必须在其前面增加【空格】

:nth-child(index)，匹配第index个子元素，其中index从1开始

:first-child，匹配第一个子元素

:last-child，匹配最后一个子元素

:only-child，匹配唯一一个子元素

③内容过滤选择器

:contains(文本)，匹配包含指定文本内容的元素

:empty，匹配没有子元素也没有文本元素的空元素

:parent，匹配包含子元素或文本元素的不为空元素

:has(selector)，匹配包含selector的元素的父元素

④可见性过滤选择器

:visible，匹配可见的所有元素

:hidden，匹配不可见的所有元素

⑤属性过滤选择器

[attrName]，匹配包含属性attrName的元素

[attrName=value]，匹配包含属性attrName等于value的元素

[attrName!=value]，匹配包含属性attrName不等于value的元素(包含没有attrName属性的元素)

[attrName^=value]，匹配包含属性attrName以value开始的元素

[attrName$=value]，匹配包含属性attrName以value结束的元素

[attrName\*=value]，匹配包含属性attrName中含有value的元素

[attrName1][attrName2]，匹配包含属性attrName1且attrName2的元素，结果为【交集】

⑥表单对象属性过滤选择器

:enabled，可用输入框

:disabled，不可以输入框

:checked，被选中(单/多选框)

:selected，被选中(下拉列表)

(4)表单选择器

三、JQuery操作DOM

1、原生DOM

(1)document对象

属性：

documentElement属性--指向<html>

方法：

getElementById

getElementsByName

getElementsByTagName

querySelector

querySelectorAll

(2)element对象

元素树：

parentElement

children

操作属性：

getAttribute()

setAttribute()

removeAttribute()

(3)node对象

nodeName、nodeValue、nodeType

节点树：

parentNode

childNodes

替换节点

删除节点

插入节点

复制节点

(4)this用法

this用于指代DOM对象(必须要具有上下文环境)

在全局域使用this指代window对象

Js的构造器中

2、JQuery中操作DOM

(1)基本操作

html()，作用类似于innerHTML属性

获取：html()

设置：html(html代码)

text()，作用类似于textContent属性

获取：text()

设置：text(文本内容)

val()，作用类似于value属性

获取：val()

设置：val(value)

attr()，作用类似于getAttribute和setAttribute

获取：attr(name)

设置：attr(name,value)

删除：removeAttr(name)

(2)样式操作

css()

获取(指定CSS的属性名的值)

css(attrName)

设置

css(attrName,attrValue)

css(options)

options指{key:value}

例如 $("#d1").css({

"background":"pink",

"width":400

});

attr("class",className)

设置样式 -- 覆盖原有的所有的样式

addClass("...")，追加样式

基于原有的样式，增加一个样式

removeClass()，移除样式

如果不传递参数，则删除所有样式

如果传递参数，则删除指定样式

也可传递多个参数，中间用空格隔开

toggleClass("...")，切换样式

只能传一个参数，表示在无样式与指定样式之间切换

hasClass(className)，判断样式

判断是否【包含】某个【指定】样式，并不是判断是否含有任意样式

(3)遍历节点

①父元素

parent()

不传递参数：获取指定元素的父元素

传递参数：获取指定元素的指定(符合表达式的)父元素

例如 <div>

<p>Hello</p>

</div>

<div class="selected">

<p>Hello World</p>

</div>

$("p").parent(".selected");

得到的结果是<div class="selected"><p>Hello World</p></div>

parents()

不传递参数：获取指定元素的祖先元素(从父元素一直取到根元素<html>)

②子元素

children()，所有子元素

children(表达式)，指定子元素

比如children(":eq(1)")

③兄弟元素

prev()，上一个兄弟元素

next()，下一个兄弟元素

siblings()，所有兄弟元素

④find()，在指定元素的后代元素中查找指定元素

(4)创建节点

文本节点 -- text()

属性节点 -- attr()

元素节点 -- $(HTML代码)

固定套路，比如：

var $li=$("<li title='香蕉'>香蕉</li>");

(5)插入节点

①内部插入 -- 插入在指定元素内(子元素)

append()

prepend()

appendTo()

prependTo()

②外部插入 -- 插入在指定元素外(兄弟元素)

before()

after()

insertBefore()

insertAfter()

(6)删除节点

remove()，删除自身节点及后代节点

empty()，删除后代节点，保留自身节点

(7)替换节点

replaceWith，前面的元素是被后面的元素替换

replaceAll，后面的元素是被前面的元素替换

(8)复制节点

①DOM -- cloneNode(boolean)

boolean，表示是否复制【后代节点】

默认为false，表示不复制

②JQuery -- clone(boolean)

boolean，表示是否复制【事件】

默认为false，表示不复制

四、事件

1、ready()

作用类似于DOM中的window.onload

(1)写法：

$(document).ready(function(){...});

$().ready(function(){...});

$(function(){...});

(2)ready与onload的区别

ready

有简写方法

在HTML页面中能编写多个

必须等待HTML页面所有DOM元素都加载完毕后再执行

执行的速度快

onload

无简写方法

在HTML页面中只能存在一个

必须等待HTML页面所有内容都加载完毕后再执行

执行的速度慢

2、事件绑定

bind(type,data,callback)

type，设置绑定的事件名称

指定多个事件名称时，中间用空格隔开

data，作为event.data属性值传递给事件对象的额外数据对象

可省略

callback，事件的处理函数

3、解绑事件

unbind(type,callback)

它可以解绑bind()绑定的事件，也可以解绑常规的例如click()绑定的事件

4、事件冒泡

三个阶段：捕获、触发、冒泡

取消冒泡

event事件对象的stopPropagation()

addEventListener( , ,bool)

5、事件对象event

target，绑定事件的目标元素

PageX，clientX，offsetX，x -- 鼠标坐标值x

PageY，clientY，offsetY，y -- 鼠标坐标值y

阻止默认行为

event.preventDefault();

return false;

6、事件模拟

trigger()

例如 $("#btn").click(function(){

$("#test").text("xxxxxx");

}).trigger("click");

//不点击就可以实现click事件

五、Js库的冲突问题

$符号的使用

1、先引入其他Js库，后引入jQuery

(1)jQuery中不再使用$符号，直接用jQuery

例如 jQuery(document).ready(function(){

console.log("xxxxxx");

});

(2)在ready()内部定义形参$

在ready()内部$表示jQuery；在ready()外部$表示其他Js库

(3)使用自调函数接收jQuery

在自调函数内部$表示jQuery；在自调函数外部$表示其他Js库

(4)移交$符号的使用权

jQuery.noConflict();

2、先引入jQuery，后引入其他Js库

移交$符号的使用权

jQuery.noConflict();

六、动画效果

1、显示与隐藏

show()

无参数，表示无动画效果

有参数，表示有动画效果

show(speed,callback)

speed，设置动画执行的速度

预定义速度 - slow/normal/fast

设置时间 - 单位为毫秒

callback，回调函数，在动画执行完毕后的回调函数

hide()，与show一样

2、滑动式动画

slideUp()，向上滑动

无参数，有滑动效果，有默认的speed值

有参数，有滑动效果，值可设置

slideUp(speed,callback)

speed、callback与show、hide一样

slideDown()，向下滑动

与slideUp一样

3、淡入淡出效果

fadeIn()

fadeOut()

与滑动式效果用法一样

4、动画切换

toggle(speed,callback)，功能可以替换show()和hide()

slideToggle(speed,callback)，功能可以替换slideUp()和slideDown()

5、自定义动画

animate(param,duration,easing,callback)

param，参数，设置自定义动画的参数(直接使用CSS属性)

格式：{key:value}

\*注：不能使用CSS中所有的背景颜色

duration，可省略，设置动画执行的时长(毫秒)

easing，可省略，要使用的擦除效果的名称(需要插件支持)

callback，可省略，在动画完成时执行的函数

例如 $("#panel").animate({

left:200

},3000);

animate(params,options)

params，与param一样

\*注：不能使用CSS中所有的背景颜色

options，选项，格式：{key:value}

duration，与上面一样

easing，与上面一样

complete，与callback一样

queue，boolean值，默认值为true

例如 $(this).animate({

left:200

},{

duration:3000

}).fadeOut(3000);

6、并发和排队效果

(1)并发效果

设置的多个动画同时执行

①只需要一个animate，将param里面的参数全部写入即可

例如 $(this).animate({

width:300,

height:300

},3000);

②需要animate().animate()链式操作，需要设置queue

例如 $(this).animate({

width:300,

},{

duration:3000,

}).animate({

height:300

},{

duration:3000,

queue:false

});

(2)排队效果

设置的多个动画按照先后顺序依次执行

实现方式：多个animate()的连续调用

七、类数组操作

1、类数组对象：Object，存储方式类似于数组的结构

Arguments对象，接收函数的参数

jQuery对象，本身就是类数组对象，底层是DOM对象

2、属性

length，数组的长度/元素的个数

3、方法

get(index)，根据index返回对应的DOM对象

eq(index)，根据index返回对应的内容

index(obj)，根据obj返回对应的索引值

4、隐式迭代 - 显示迭代

隐式迭代：要遍历谁；得到遍历后的结果

$().each(callback)，jQuery对象方法

回调函数：function(index,domEle){...}

index，索引值

domEle，DOM对象

用于jQuery遍历

$.each(object,callback)，jQuery全局函数

object，当前需要遍历的对象或数组

回调函数跟上面一样

显式迭代：要遍历谁；如何遍历的；得到遍历后的结果

while

do...while

for

forin (for的加强用法)

八、jQuery UI

1、基本内容

jQuery UI，必须基于jQuery使用

使用jQuery UI的步骤：(引入顺序不能改变)

引入jQuery文件、引入jQuery UI的Js文件、引入jQuery UI的CSS文件

2、Effect(效果)

(1)animate()方法

扩展了jQuery中animate()方法，允许操作CSS中所有有关背景的属性

\*注：背景颜色必须要指定具体的属性名称

例如backgroundColor、borderBottomColor

(2)effect()方法

执行jQuery UI默认提供的动画效果的方法

3、Interactions(交互组件)

(1)draggable(options)，鼠标拖动指定元素

options选项，格式为{key:value}

①axis，设置当前元素只能向x轴或y轴拖动

例如 draggable({

axis:"x"

});

②containment，设置当前元素在指定区域中拖动

例如 draggable({

containment:"parent"

});

指可以在当前元素的父元素中拖动

"document"/"window"，指当前页面中拖动

数组，格式为[x1,y1,x2,y2]

绘制矩形，左上角x1,y1，右下角x2,y2

选择器直接使用

③cursor，设置拖动当前元素时，鼠标的样式

参考CSS中cursor的属性值

④revert，当拖动结束时，当前元素是否回到初始位置

默认值为false，不回到初始位置

⑤snap，设置拖动的当前元素是否吸附到其他元素上

Boolean，如果为true则表示吸附

选择器直接使用

⑥snapMode，设置吸附的方式

inner，吸附到指定元素的内部

outer，吸附到指定元素的外部

\*注：该选项必须配合snap使用

例如 draggable({

snap:"#div1",

snapMode:"inner"

});

⑦handle和helper，实现复制当前元素，拖动的是复制后的元素

例如 draggable({

handle:"p",

helper:"clone"

});

⑧grid，设置当前元素每次拖动的距离

数组，[x,y]

例如 draggable({

grid:[100,100]

});

事件

①start，鼠标开始拖动当前元素时触发

②drag，鼠标拖动当前元素，一直触发

③stop，鼠标结束拖动当前元素时触发

例如 $("#d1").draggable({

start:function(){

start++;

$("#d2>span").text(start);

},

drag:function(){

drag++;

$("#d3>span").text(drag);

},

stop:function(){

stop++;

$("#d4>span").text(stop);

}

});

(2)droppable(accept,drop)，鼠标拖放指定元素

accept选项，设置当前元素允许接收拖放的元素

drop事件，拖放事件(必须要有的事件)

例如 $("#d1").droppable({

accept:"#d2",

drop:function(event,ui){

... ...

}

});

(3)resizable()，鼠标拉动缩放指定元素

alsoResize选项，设置缩放当前元素时，哪个元素同时进行

例如 $(".p\_Lst").resizable({

alsoResize:".p\_Img",

});

$(".p\_Img").resizable();

(4)sortable()，鼠标拖动排序指定元素

4、Widget(部件)

(1)accordion()，手风琴效果

HTML运用格式：

<div>

<h3>标题</h3>

<div>内容</div>

<h3>标题</h3>

<div>内容</div>

<h3>标题</h3>

<div>内容</div>

</div>

(2)autocomplete()，自动提示

source选项，设置提示内容的备选项

(3)datepicker()，日期拉选框

(4)dialog()，对话框

选项

modal，设置为模式对话框，boolean类型

true，对话框弹出后，页面其他元素、背景不可操作

autoOpen，设置是否自动打开对话框

false，自动关闭对话框

事件

open，打开对话框

close，关闭对话框

例如 $("#dialog").dialog({ //设置样式为对话框

autoOpen:false,

buttons:{

"确认":function(){

$("#dialog").dialog("close");

},

"取消":function(){

$("#dialog").dialog("close");

}

}

});

$("#opener").button().click(function(){

$("#dialog").dialog("open"); //打开对话框

});

(5)tabs()，多页签样式

HTML运用格式：

<div>

<ul>

菜单

<li><a href="#id"></a></li>

<li><a href="#id"></a></li>

</ul>

上面有几个li元素，就要定义几个div元素

并且定义id属性值，与a元素的href一一对应

<div id=""></div>

<div id=""></div>

</div>

(6)tooltip()，工具提示框

5、contextmenu()，禁止鼠标右键功能

document/window对象绑定contextmenu事件

例如 $(document).contextmenu(function(event){

event.preventDefault();

});

\*详情见案例32--自定义右键菜单

九、使用插件

1、基本内容、概念

组件/插件/组件化开发

优点：

开发更灵活

低耦合 - 代码之间的关系越来越小

jQuery插件的作用

扩展jQuery的功能；呈现组件化特点

使用jQuery插件的步骤

与jQuery UI一致，都是基于jQuery实现

先引入jQuery文件，再引入jQuery插件文件

2、layDate，日期插件

将layDate文件夹直接拷贝到工程目录中，laydate目录中的结构不能改变

如何使用：

(1)方法1：在input输入框中增加对应的内容即可

指定class的值为laydate-icon

绑定onclick事件，指定laydate()方法

(2)方法2：定义input输入框

指定class的值为laydate-icon

通过Js代码绑定click/focus事件，指定laydate()方法

laydate(options)方法的选项：

options格式：{key:value}

elem，指定日期控件显示的元素

istime，是否开启时间选择，默认值为false

isclear，是否开启【清空】按钮，默认值为true

istoday，是否开启【今天】按钮，默认值为true

issure，是否开启【确认】按钮，默认值为true

format，设置日期格式，默认值为yyyy-MM-dd hh:mm:ss(通用格式)

yyyy/MM/dd

YYYY年MM月DD日

\*注：格式建议使用默认，以免向服务器端发送时引起乱码

min，设置最小日期，默认为1900年

max，设置最大日期

start，设置开始日期

3、表单验证插件

需要引入的文件：

validation插件的CSS文件

jQuery文件

validation插件的Js文件

validation插件提供的中文文件

validate(options)，核心方法，用于表单验证

(1)选项：

①rules，设置元素的验证规则

格式为{key:value}

key，指定验证元素的name属性值

value，设置元素的验证规则

如果是一种验证规则，则直接定义验证规则名称

例如 $("#myform").validate({

rules:{

user:"required"

},

messages:{

user:"请输入用户名"

}

});

如果是多种验证规则，则格式为{key:value}

key，验证规则名称

value，格则内容

②messages，自定义错误提示信息

value指定的是元素的name属性值，格式为{key:value}

key，指定验证元素的name属性值

value，设置错误提示信息

如果是一种验证规则，则直接定义验证规则名称

如果是多种验证规则，则格式为{key:value}

key，验证规则名称

value，错误提示信息

它会覆盖validate()方法底层提供的错误提示信息

(2)设置错误提示信息：(3种)

使用validate()方法的默认提示

利用messages选项设置自定义提示

在HTML元素中使用<label>元素设置自定义提示

(3)错误提示信息的优先级别

<label>元素→messages→默认提示

使用<label>元素设置自定义错误提示信息时：

class="error"，validate默认的错误样式

for="验证元素的name属性值"

(4)遇到单选框或多选框时：

错误提示信息，默认出现在第一个验证控件后

解决：在<HTML>元素对应正确的位置上定义<label>

label的class="error"

label的for=验证元素的name属性值

label元素内不能包含任何内容

验证规则只需要使用"required"

例如 <input type="radio" id="male" name="gender">男

<input type="radio" id="female" name="gender">女

<label class="error" for="gender"></label>

(5)遇到下拉列表时：

保证下拉列表中第一个option选项的value值为空

使用minlength属性，设置至少选择几个选项

使用maxlength属性，设置最多选择几个选项

(6)自定义验证方法：

$.validator.addMethod(name,method,message)

name，设置自定义验证方法的名称

method，是一个函数，编写具体的验证逻辑

function(value,element,params){...}

value，验证元素的value属性值

element，验证的元素，是DOM对象

params，调用自定义验证规则的value值

message，自定义错误提示信息

4、瀑布流插件--Masonry

(1)如何使用

先引入jQuery文件

再引入Masonry文件

(2)masonry()核心方法

默认调用--实现瀑布流效果

(3)使用Masonry插件的方式

HTML页面

div元素作为瀑布流布局的容器

所有显示为瀑布流的内容，放置在容器中

调用Masonry插件

作为容器的div调用masonry()

(4)选项

itemSelector，指定使用哪些页面元素进行布局

columnWidth，设置每列的宽度(默认单位为px，但是不需要写)

percentPosition，设置是否使用百分值

boolean值，true表示使用百分值

stamp，标记元素

固定在某个具体的位置，布局元素在固定元素下进行布局

isFitWidth，设置作为容器的宽度，以适应布局元素的总宽度

(5)方法

appended，向瀑布流尾部添加元素

prepended，向瀑布流首部添加元素

remove，从瀑布流删除指定的元素

(6)事件

使用jQuery方式进行绑定

bind(type,callback)

on(type,callback)

layoutComplete，当完成布局时触发

指的是添加元素后，完成布局时的触发，并不是初始化布局完成

\*必须配合appended和prepended方法使用

removeComplete，当删除元素时触发

\*必须配合remove方法使用

(7)通过对象方式创建瀑布流布局

new Masonry(容器div,选项)

十、开发插件

1、插件的分类

(1)全局函数插件 -- $.each()

和HTML页面元素无关

(2)对象方法插件 -- $().each()

和HTML页面元素有关

(3)选择器插件(目前几乎不适用)

XPath插件，用于解析XML文件

2、插件的机制

全局函数

①jQuery.extend(object)或$.extend(object)

object指Js对象

格式为 {

属性名:属性值,

方法名:function(){}

}

②$.方法名=function(){}

对象方法

①jQuery.fn.extend(object)或$.fn.extend(object)

②$.fn.方法名=function(){}