jQuery

JQuery基本兼容了所有主流浏览器（包括老版本IE）

JQuery可以使用链式编程

引入：

一种是生产环境版（mini版），代码丑化压缩了，体积小，但可读性差

一种是开发版，js可读性好，但体积大

这里引入压缩版：

<script src="http://code.jquery.com/jquery-1.11.3.min.js"></script>

cdn引入jq相关库时，若出现警告，可以给script标签设置跨域元属性,阻止发送cookie给cdn链接 crossorigin=”anonymous”

必须在js代码之前引入

（1）JQuery入口函数：

$(document).ready(function(){//jQuery代码})

可简写为：$(function(){//jQuery代码})

它的本质是DOM中的 onDOMContentLoaded 事件，即DOM（标签）加载完才执行js代码，使得js代码可以放到body上面。

（2）JQuery顶级对象$，相当于DOM的window对象

jQuery() 可简写为 $() 用来获取DOM的element对象，然后包装成jQuery对象，以此来使用$的属性和方法。

注意：JQuery对象和DOM对象是不同的，原生js获取的是DOM对象，$()获取的是jQuery，不能相互调用各自的属性和方法。只能调用自己有的属性/方法。

jQuery对象转成DOM对象：

有些DOM对象的属性/方法，jQuery没有封装，要使用这些属性/方法时，就需要把jQuery对象转成DOM对象。L两种方法，n为索引

let $(‘xxx’)[n]

或 let b=$(‘xxx’).get(n)

有时DOM对象也想转成jQuery对象：

let a=document.querySelector(‘#xxx’)

let b=$(a)

（3）基本使用：

let a=$(‘选择器’) 选择器与css一样

注意：不管获取过来的是一个，多个还是无，都是一个伪数组。但是如果用a[索引]的话，会转换成DOM对象。但有以下三种方式，就不要索引了

\*jQuery隐式迭代：如：

a.css() 会自动遍历a里面的所有元素，都调用css()

也可以自己遍历：

a.each(function(I,e){}) i为索引，e为各个DOM元素对象

$.each($(‘xxx’)或数组或对象,function(I,e){}) i索引，e是DOM元素对象

\*筛选选择器（索引从0开始）

$(‘ul li:first’) 选出ul的第一个li

$(‘ul li:last’) 选出ul的最后一个li

$(‘ul li:eq(x)’) 选出ul下的索引为x的li

$(‘ul li:odd’) 选出ul下的索引为奇数的li

$(‘ul li:even’) 选出ul下的索引为偶数的li

\*筛选父子兄

a.parent() 最近一级的父元素

a.children(‘选择器’) 最近一级的选择器选出的子元素，参数为空则找所有最近一级子元素，相当于子选择器

a.eq(n) 最近一级的索引为n的子元素

a.find(‘选择器’) 所有的选择器选出的后代，参数为空则是所有后代元素，相当于后代选择器

a.siblings(‘选择器’) 选出选择器对象的所有兄弟，参数为空则是所有兄弟，注意不包括自己

a.nextAll(‘选择器’) a.prevAll(‘选择器’) a的之后/之前的所有兄弟

a.hasClass(‘类名’) 有/无该类名返回true/false

获取/设置属性：

prop：只适用于内置属性

a.prop(‘属性名’) 获取

a.prop(‘属性名’,’属性值’) 设置

attr：只适用于自定义属性，规则与原生一样（如省略data-）

a.attr(‘属性名’) 获取

a.attr(‘属性’名,’属性值’) 设置

获取/设置文本：

a.text() 参数为空/字符串是获取/设置

a.html() 参数为空/字符串是获取/设置

a.val() 参数为空/不为空是获取/设置（表单）

创建/添加/删除元素：

let e=$(‘<tag>….</tag>’) 创建元素，但还需添加到body

$(‘xxx’).append(e) 内部添加，添加到xxx内部最后作为子元素

$(‘xxx’).prepend(e) 内部添加，添加到xxx内部最前作为子元素

$(‘xxx’).after(e) 外部添加，添加到xxx的后面作为兄弟元素

$(‘xxx’).before(e) 外部添加，添加到xxx的前面作为兄弟元素

$(‘xxx’).remove() 删除xxx自己

$(‘xxx’).empty() 删除xxx的所有后代，等同于$(‘xxx’).html()

获取/设置尺寸，位置

\*尺寸

a.width() a.height() 不包含内外边距和边框

a.width(数值) a.height(数值) 设置宽高，不用写单位

innerWidth() 包含内边距 outerWidth() 包含内边距和边框

outerWidth(true) 包含边框，内外边距

\*位置

Offset，scroll参数空为获取，不空为设置

a.offset() 中心点距离window的left和top距离，与父元素无关

a.position() 中心点到带有定位的父元素边缘的距离，无父就window（只能获取）

a.scroll() 被卷去的left，top，与原生的scroll一样

（4）修改样式：

css(‘css属性’) 返回带单位的字符串的属性值

css(‘css属性’,’属性值’) 修改样式属性值，属性值为’’则是去掉该样式，若属性值是数值，可不带引号不带单位

css({ 修改多个样式，csss属性可不带引号，值是数值可不带

css属性 : ‘值’ , 引号和单位，css属性为多个单词组成时小驼峰

…

})

addClass(‘类名’) 增加类，与原生className的最大区别是不覆盖原有类

removeClass(‘类名’) 去掉类，只去掉该类

toggleClass(‘类名’) 有/无该类，则去掉/增加该类，又叫切换类，不覆盖原有

（5）事件：

$(‘xxx’).事件类型(function(){})

事件函数里面，用 $(this) 代表事件源，也是一个jQuery对象

事件注册方式：分为单个注册和用on事件注册

\*$(‘xxx’).click(function(){}) 注册单个事件

\*$(‘xxx’).on( { click:function(){} , … } ) 注册多个事件

\*$(‘ul’).on(‘click’,’li’,function(){}) 事件委派形式注册，第二个参数必须是事件源的

子选择器；给ul注册click事件，当li触发时，通过事件冒泡到ul，总体来看，只注册了一个事件，就能让若干个li使用

Ul li用单个事件注册的形式，后来添加的li是没有被注册事件的；而on事件可以让后来添加的li也使用事件。

\*$(‘xxx’).one(‘click’,function(){}) 该click只触发一次，触发后就失效

解绑事件：只能解绑on绑定的事件

$(‘xxx’).off(‘click’) 解绑click事件，若参数为空，则解绑全部事件

自动触发事件：

若a已经绑定click事件，则会执行事件函数，若未绑定，相当于注册空事件函数

\*a.click()

\*a.trigger(‘click’)

\*a.triggerHandler(‘click’)

事件类型：基本上是DOM事件去掉on，新的事件类型：

hover(经过回调,离开回调) 参数若只写一个，则经过离开都会回调，两个参数则当鼠标经过/离开时分别回调，相当于mouseenter和mouseleave

change() 单选/复选发生改变

事件对象：与原生DOM一样

（6）jQuery效果：

以下动画一般都有以下参数：

参数：

\*speed：可以是字符层如’fast’和’slow’，可以是不带单位数值（毫秒）

\*匀速或变速

\*回调函数：显示/隐藏完成之后执行的函数

①显示和隐藏碎

toggle() 显示/隐藏则隐藏/显示，有个默认动画

hide() 隐藏元素（display:none）

show() 显示元素

一般都不加参数（简单的显示隐藏）

②滑动：

slideUp() 上滑

slideDown() 下滑

slideToggle() 上滑/下滑就下滑/上滑

③淡入淡出

fadeIn() 淡入

fadeout() 淡出

fadeToggle() 淡入/淡出则淡出/淡入

fadeTo(ms,tmd) 变成一个透明度，除了基本的三个参数，还有一个必须要的参数透明度（0-1）

④自定义动画：animate

animate()

参数：

\*对象，里面为要变成的css样式（键值对），不可省

\*后三个参数就是基本的ms，匀速/变速，回调函数，可以省略

⑤动画队列

当动画触发后，进入动画队列，是一定会执行的，所以就会出现一个情况：当短时间内多次触发动画，则会出现旧动画未完成就有新动画的情况。

stop() 停止上一个动画

（7）jQuery其他

\*拷贝对象

$.extend(false,obj1,obj2) obj2拷贝给obj1

若obj1本来有obj2没有的属性，则保留

若obj1有obj2相同的属性，则被obj2覆盖

第一个参数可省，省后默认false，true/false为深拷贝/浅拷贝

浅拷贝：引用数据类型属性的拷贝只是拷贝了地址，所以最终obj1和obj2的引用数据类型的属性指向同一个对象

深拷贝：引用数据类型属性会重新开辟一个新的内存空间进行拷贝数据，所以最终obj1和obj2的引用数据类型属性指向的是各自的对象

\*多库共存

当其他库有定义$时：

\*使用jQuery

\*let xxx=$.noConflict() 将$换成xxx

jQuery moboile

jquery mobile：用来布局触屏智能手机和平板的页面，它是用H5,C3构建的最小脚本。

引入：

<link rel="stylesheet" href="http://code.jquery.com/mobile/1.4.5/jquery.mobile-1.4.5.min.css">

<script src="http://code.jquery.com/jquery-1.11.3.min.js"></script>

<script src="http://code.jquery.com/mobile/1.4.5/jquery.mobile-1.4.5.min.js"></script>

</head>

使用：

1、基本布局：

<div data-role=”xxx”></div>

xxx为角色，有page，header，main，footer，navbar，button，controlgroup，popup

（1）整体页面结构：

<div data-role=”page” id=”page1”> 一般要制定id，方便跳转页面

<div data-role=”header” data-position=”fixed”></div> 头部定位到顶部

<div data-role=”main” class=”ui-content”></div> 主体给内外边距

<div data-role=”footer” data-position=”fixed”></div> 底部定位到底部

</div>

data-posistion=”fixed” data-position=”inline” 都是顶部底部定位到顶部底部

data.theme="" a-z 主题颜色

data-position=”left” 定位到左边

data-fullscreen=”true” 顶部/底部定位等级是否最高

常用样式class：

ui-content内外边距 ui-corner-all圆角边框 ui-shadow阴影

（2）导航条选项卡：这种写法才会均分宽度，header，main，footer都可用

<div data-role=”navbar”>

<ul>

<li></li>

>/ul?

</div>

（3）网格布局：

ui-grid-a/b/c/d 两/三/四/五列布局

ui-block-a b 两列

ui-block-a b c 三列

……

<div class="ui-grid-a">

<div class="ui-block-a">

第一列

</div>

<div class="ui-block-b">

第二列

</div>

</div>

（4）弹出面板；

<a href=”panelid”></a> hearer，main，footer里，点击弹出面板

<div data-role="panel" id=".."></div> 与header，main，footer同级

data-swipe-close=”true” 是否允许滑动退出面板

（5）列表

<ul data-role="listview">

<li></li>

</ul>

data.inset="true" 给列表边距和圆角

（6）可折叠列表：一个列表显示后，其他列表都折叠

<div data-role=”collapsibleset”>

<div data-role="collapsible" data-collapsed-icon="carat-r" data-mini="true">

<h3>name</h3>

  （5）的列表

</div>

</div>

（7）过滤（搜索）：

<form class="ui-filterable">

<input data-type="searchs" id="s">

</form>

<ul data-input="#s" data-role="listview" data-filter="true" data-autodividers="true" data-inset="true">

<li></li>

</ul>

（7）表单：

<form>

其他与以前一样

<div class="ui-field-contain"> //表单获得

<fieldset data-role="fieldcontain">

<select>

<option></option>

</select>

</fieldset>

</div>

</form>

2、a标签，按钮：

//跳转页面，跳转动画为slide（滑动），动画不指定默认为fade（淡入淡出）

<a href=”pageid” data-transition=”slide”>…</a>

按钮：

（1）<input type=”button” value=”…”></input>

（2）<a href=”#” data-role=”button”></a>

（3）<a href=”#” class=”ui-btn”></a>

（4）data-type设置按钮排列：horizontal(水平组合按钮) vertical（垂直组合按钮）

两端/中间的按钮是直角/圆角边框

<div data-role=”controlgroup” data-type=”…”>

若干按钮

</div>

class; ui-btn-inline 一行显示按钮

data.ajax="false" a跳转html时必用，否则无法跳转

data-direction=”reverse” 将切换效果翻转

3、icon：只能用于header，footer，不能用于main

顶部的icon在文本左边，底部的icon在文本的上面

（1）属性指定icon

a标签或按钮的属性 data-icon=”…”

常用：home，info，back

（2）class指定icon: class=”ui-icon-home”

（3）class指定icon位置： class=”ui-btn-icon-left”

4、弹窗：

定义一个弹窗，并给id，未触发时隐藏

<div data-role="popup" id="pop"></div>

点击显示对应id的弹窗，弹窗出现后点击页面任意位置弹窗隐藏  
<a href="#pop" data-rel="popup"></a>

data-position-to=”window” 弹窗定位到页面正中间

data-rel=”back” 点击关闭弹窗

5、事件：on绑定

新增事件：

pagexxx 页面事件，页面创建，切换等时触发

scrollxxx 开始/结束滚动时触发

swipexxx 滑动时触发

tagxxx 触屏点击事件

orientationchange 定位事件，当屏幕变为水平/垂直时触发

举例：

//当切换到xxx页面时回调，当左滑p标签时，隐藏p

  $(document).on('pagecreate','xxx',function(){

    $('p').on('swipeleft',function(){

      $(this).hide()

    })

  })

//...绑定向左滑动事件，跳转新页面

  $('...').bind('swipeleft',function(){

    $.mobile.changePage('xxx' //后面参数可不加

    ,{

      transition:'',//切换效果

      reverse:true //将切换效果翻转

    },true,true)

  })