jQueryMoble

Day1 ················································1

Day2 ················································1

Day3 ················································1

Day4 ················································1

**Day1**

jQuery是一个JS函数库，适用于几乎所有项目；

jQueryUI是一个HTML UI组件库，适用于PC页面项目；

Bootstrap是一个CSS样式库+HTML UI组件库，适用于响应式Web项目；

AngularJS是一个JS框架，适用于数据操作为主的SPA应用；

jQueryMobile是一个HTML UI组件库，适用于移动Web应用。

1.jQueryMobile概述

jQuery Mobile is a HTML5-based user interface system designed to make responsive web sites and apps that are accessible on all smartphone, tablet and desktop devices.

jQM V1.3之前提供了六种默认的主题样式：A/B/.../F

jQM V1.4之后提供了二种默认的主题样式：A/B，若需要定制主题，可以使用jqm官方提供的ThemeRoller

官网： http://www.jquerymobile.com

API手册官网： http://api.jquerymobile.com

jQM的内容分为四个部分：

1. Page&Navigation——重点
2. CSS Framework
3. Widgets
4. Form Widgets

练习：自学Introduction和Pages

提示：

(1)jQM中通过data-\*属性为当前元素甚至子元素添加事件监听函数以及class样式！—— 学习jQM必须记忆所有的data-\*属性！！

(2)jQM自带所有的class都以 .ui- 开头！

2.jQM四部分之一——Page

jQM属于典型的SPA框架！每个HTML文件中可以声明多个data-role="page"的元素，每个Page必须占满整屏，故某一时刻，只能看到一个Page。

一个HTML文件必需至少一个page，若用户未提供，则jqm会自定添加；若用户提供了一个page则jqm不再添加！

每个Page都是全屏绝对定位，Page之外不要放任何内容！

练习：学习Navigation

2.jQM四部分之一——Navigation

Navigation: 指在多个Page之间跳转/切换

**(1)使用超链接**

<a href="#PageID">跳转到同一个HTML中另一个Page</a>

<a href="xx.html">跳转到另一个HTML中第一个Page</a>

**(2)使用JS跳转**

<button onclick="$.mobile.changePage('4.html')" type="button">跳转到PAGE40</button>

<button onclick="$.mobile.changePage('#p2')" type="button">跳转到PAGE2</button>

//提示：只有起始HTML中的元素才能使用on/bind/click之类的事件绑定方法

//其它模板页面上的元素事件监听函数都必须使用如下方法：

$(document).delegate('#bt4', 'click', function(){

$.mobile.changePage('5.html');

})

|  |
| --- |
| jQM中的页面跳转都是SPA应用中的页面跳转！被跳转的模板页面中只需要一个Page即可。 |

3.jQM四部分之一——Transition(过场动画)

只有SPA应用中的页面切换才有过场动画！

jQM提供的过场动画：

(1)fade

(2)pop

(3)flip

(4)turn

(5)slide

(6)slideup

(7)slidedown

(8)slidefade

(9)flow

(10)none

使用方法：

<a href="目标地址" data-transition="动画效果"></a>

<button onclick="$.mobile.changePage('目标地址'， {transition: '动画效果'})"></button>

练习：十种过场动画效果

3.jQM四部分之二——Button

三种形式的按钮：

(1)超链接按钮

<a href="#" data-role="button"></a>

(2)BTN按钮

<button type="button"></button>

1. INPUT按钮

<input type="button" value="">

行内按钮： data-inline="true"

mini按钮： data-mini="true"

带图标的按钮： data-icon=""

修改图标位置： data-iconpos="left/right/top/bottom/notext"

练习：创建带图标的按钮：主页、用户、配置、前进、后退、警告、定位、打分、赞、关闭、加号、减号、垃圾桶....

1. jQM四部分之二——Grids(网格布局系统)

jQM类似于Bootstrap，使用div+css模拟了表格的“行”和“列”的布局概念。列中可以继续嵌套行。

**jQM中“行(grid)”分为4种：**

(1).ui-grid-a 一行中只有两列，每列宽50%， .ui-block-a/b

(2).ui-grid-b 一行中只有三列，每列宽33%, .ui-block-a/b/c

(3).ui-grid-c 一行中只有四列，每列宽25%, .ui-block-a/b/c/d

(4).ui-grid-d 一行中只有五列，每列宽20%, .ui-block-a~e

**jQM中的“列(block)”分为5种：**

(1).ui-block-a 一行中的第1列

(2).ui-block-b 一行中的第2列

(3).ui-block-c 一行中的第3列

(4).ui-block-d 一行中的第4列

(5).ui-block-e 一行中的第5列

提示：

(1)jQM的布局系统中，没有默认的槽宽——即行和列都没有margin / padding。

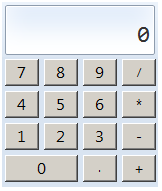
(2)在任何行中，.ui-block-a都必须重启一行——即block-a只能出现在一行中的最左边。

(3)行中可继续嵌套列。

练习：

(1)在一行中呈现四张图片（图片间有30px的槽宽）

(2)使用Grid布局系统实现下列界面（网页版计算器）



作业：

(1)调查当前市场中还有哪些UI组件库。

(2)调查当前市场中还有哪些JS框架。

(3)整理出今天学过的data-\*及其含义。

**Day2**

复习：

jQueryMobile是一个HTML UI组件库，适用于移动App应用。

组件主要分为四部分：

1. Page & Navigation

Page data-role="page"

Navigation

<a href="#id/xx.html">

$.mobile.changePage()

Transition

Loader

1. CSS Framework

Button

Icon

Grids

.ui-grid-a/b/c/d

.ui-block-a/b/c/d/e

Body & Bar

1. Widgets
2. Form Widgets
3. jQM组件第二部分CSSFramework——Body&Bar

.ui-bar.ui-bar-a 带背景颜色的条状块

.ui-body.ui-body-a 一个带边框的块

2.jQM组件第三部分Widgets——Toolbar

两种工具条：

顶部工具条： data-role="header"

底部工具条： data-role="footer"

可以添加的属性：

data-position="fixed"

data-fullscreen="true"

data-theme="a/b"

顶部工具条中最多可以放置两个超链接——呈现出按钮效果

休息：学习Navbar的使用

2.jQM组件第三部分Widgets——Navbar

导航条的基本结构：

<div data-role="navbar">

<ul>

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

</ul>

</div>

注意：

(1)导航条中默认使用Grid系统来布置，一行中最多只能放置5个导航项，若超过5项会变为多行两列；

(2)导航条可以放在工具条中，一般用于footer工具条；

(3)导航条项可以添加图标，但默认情况下导航条中自带的必有文字。

3.jQM组件第三部分Widgets——Tabs

<div data-role="tabs">

<div data-role="navbar">ul>li>a</div>

<div id="content1"></div>

<div id="content2"></div>

</div>

1. jQM组件第三部分Widgets——Panel

注意：jQM中的Panel才是真正的Panel！——Bootstrap中的.panel算是特例。

Panel高度应该是100%，宽度是固定，可以从屏幕左侧/右侧出现，jQM提供了三种出现动画样式：

1. overlay: 覆盖式
2. reveal: 揭开式（默认值）
3. push: 推动式

当前版本的jQM要求Panel必须放在Page内：

<div data-role="page">

<!-- header -->

<!-- content -->

<!-- footer -->

<div data-role="panel" id="mypanel">

<!-- panel content goes here -->

</div><!-- /panel -->

</div><!-- page -->

休息：学习两种table

5.jQM组件第三部分Widgets——Table

jQM提供了两种形式的响应式表格：

(1)Table Reflow：回流式表格(默认值)

(2)Table Column Toggle：列切换式表格，若某列未执行“优先级”，默认为永久显示；使用data-priority="1~6"属性指定列的优先级，6为最低。

data-role="table"

data-mode="reflow/columntoggle"

午间自学： Collapsible、 Popup、Listview

1. jQM组件第三部分Widgets——Popup

Bootstrap中的弹出组件： tooltip、popover、modal

jQM中Popup可以实现上述三种组件的效果。

<div data-role="popup" id="">弹出框内容</div>

<a href="#id" data-rel="popup" data-position-to1111

dataPositionTo:

·origin(默认值)

·window(窗口正中央)

·#id(指定元素的正上方)

7.jQM组件第三部分Widgets——Listview

jQM中的Listview仿了Anddroid中的ListView组件，在Bootstrap对应于list-group组件。

8.jQM组件第四部分Form Widgets

**Day3**

jQuery是一个JS函数库，用于简化DOM操作；

jQueryUI是一个UI组件库，用于编写PC页面；

Bootstrap是一个CSS样式库+UI组件库，用于编写响应式网页；

AngularJS是一个JS框架，用于操作数据；

jQM是一个UI组件库，用于编写移动App页面；

ExtJS是一个重量级JS框架。

jQ+jQUI：可以

jQ+Boot：可以

jQ+NG：可以

jQ+jQM：可以

jQUI+Boot：很少，偏UI的组合

jQUI+NG：可以

jQUI+jQM：很少

Boot+NG：可以，功能全面

Boot+jQM：很少

NG+jQM：可以，功能全面

ngRoute <=> Navigation

ngAnimate <=> Transitions

AngularJS + AngularUI：

1.补充：AngularJS四大特性之四——模块化设计

Module(模块)：为了避免声明全局变量函数，把功能相关（如商品管理相关、用户管理相关）的组件，封装在一个“盒子”中——体现的设计思想“高聚合低耦合”。一个模块中可以声明下列组件：

**(1)控制器(controller)——操作数据**

module.controller('控制器名', function($scope){ })

**(2)过滤器(filter)——对数据进行输出过滤/格式化**

{{ 数据 | 过滤器 }}

**(3)服务(service)——供其他组件调用的功能对象**

$http、$interval、$rootScope

**(4)指令(directive)——增强HTML标签的功能**

ngSrc ngRepeat ngIf ...

(5)函数(function)

(6)对象(object)

(7)提供者(provider)

(8)类型(type)

(9)工厂(factory)

(10)全局值(value)

...

**自定义过滤器(filter)：**

module.filter('过滤器名', function(){

return function(input){

var output = null;

//....

return output;

}

})

**自定义服务对象(service)：**

module.service('服务名', function(){

//return { aa:bb , xx:yy} 有的服务是一个对象

//return function(){ } 有的服务是一个函数

})

**自定义指令(directive)：**

module.directive('指令名(小驼峰法则)', function(){

return {

restrict: 'E', //E:element C:class M：comment A：attribute

replace: true, //替换原始标签

template: '<p>COPYRIGHT ® TARENA.COM 2001~{{year}}</p>'

/\*templateUrl: 'xx.html'\*/

//compile: function(){}, //NG底层在编译当前指令时会调用此方法

//link: function(){} //NG底层在将当前指令编译好后，将模板内容与作用域对象进行链接

};

})

**注意：指令的本质是一段HTML模板，此模板必须经过编译(compile)放到DOM树，然后再经过链接(link)找到相关联的作用域对象，绑定其中的数据**。

ECMAScript Europe Computer Manufacturing Association

1. WebApp阶段项目——开发啦项目 下次讲解：16:30

所用工具：AngularJS + jQueryMobile

项目进度安排：

day01:

am: 解决NG和jQM整合中的最大冲突

pm: 使用jQM编写静态网页

day02:

am: 编写服务器端动态PHP页面

pm: 使用$http获取动态数据，替换静态内容

1. NG和jQM整合中的最大冲突

NG用于SPA应用——ngRoute

http://127.0.0.1/index.html#/main

jQM用于SPA应用——Navigation

http://127.0.0.1/main.html

二者都提供了单页应用中的页面跳转解决方案，相比较而言，jQM中的页面跳转方案比较简单。

**NG和jQM整合中的最大冲突**：jQM中使用AJAX动态请求的模板页面，加载到DOM树上，不会被AngularJS处理！——AngularJS在启动时已经运行完毕！——动态创建的HTML元素即使包含指令，不会被重新的编译并链接！

var **element** = angular.element('新添的DOM元素');

var scope = **element**.scope();

**element**.injector().invoke(function($compile){

var linkFn = $compile( **element** ); //编译新的DOM节点

linkFn( 作用域对象 ); //把作用域和DOM节点进行链接

})

4.面试题：AngularJS的启动过程

**一、启动阶段**

        大家应该都知道，当浏览器加载一个HTML页面时，它会将HMTL页面先解析成静态DOM树，然后逐个加载DOM树中的每一个元素节点。处理时，会遇到<script src="js/angular.js">这个元素，当浏览器解析到这个节点时，发现它是一个js文件，那么浏览器会停止解析剩余的DOM节点，开始执行这个js（即angular.js），同时Angular会设置一个事件监听器来监听浏览器的DOMContentLoaded事件。当Angular监听到这个事件时，就会启动Angular应用。

**二、初始化阶段(initialize)**

        Angular开始启动后，它会查找ng-app指令，然后初始化一系列必要的组件（即$injector、$compile服务以及$rootScope），接着重新开始解析ng-app 指令所对应的DOM树片段。

**三、编译(compile)、链接(link)**

        $compile服务通过遍历DOM树片段的方式查找有声明指令的DOM元素。当碰到带有一个或多个指令的DOM元素时，它会排序这些指令（基于指令的priority优先级），然后使用$injector服务查找和收集指令的compile函数并执行它。

        每个节点的编译方法运行之后，$compile服务就会调用链接函数。这个链接函数为绑定了作用域的指令设置监控。这一行为会创建实时视图。

简而言之：DOM树上的每个指令都要转换为浏览器可以识别的内容(如样式、元素、绑定特定的JS监听函数)——称为“编译”；编译完成后，DOM树必须与某个scope对象关联——称为“链接”

**var link = $compile( DOM片段 )     ;**

**link( $scope );**

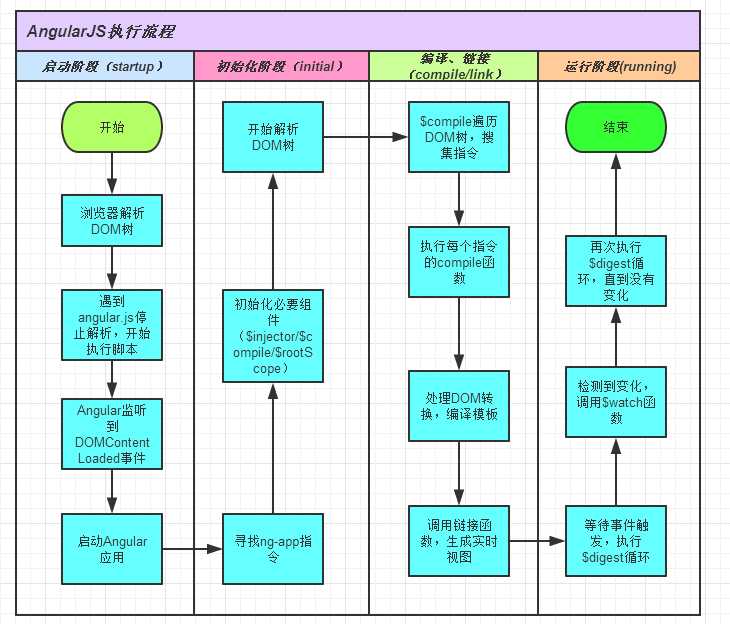
 最后，在$compile服务完成后，AngularJS运行时就准备好了。

**四、运行阶段(run)**

        Angular提供了自己的事件循环。指令自身会注册事件监听器，因此当事件被触发时，指令函数就会运行在AngularJS的$digest循环中。$digest循环会等待$watch表达式列表，当检测到模型变化后，就会调用$watch函数，然后再次查看$watch列表以确保没有模型被改变。

        一旦$digest循环稳定下来，并且检测到没有潜在的变化了，执行过程就会离开Angular上下文并且通常会回到浏览器中，DOM将会被渲染到这里。

        将以上过程的关键步骤绘制成一张图，如下：

[](http://static.oschina.net/uploads/space/2015/0525/153837_tPbY_257704.png)

5.jQueryMobile中的事件——重点

jQM中基于原生的BOM事件，扩展了一些新的事件，分为如下几类：

**(1)Page相关事件**

pagebeforecreate： Page被客户端加载下来，尚未挂载到DOM树，其中的数据也尚未处理

pagecreate：Page被客户端加载下来，已经挂载到DOM树上，开始处理内部元素

**(2)触摸相关事件**

swipe：滑动事件(大于30px)

swipeleft：左滑动

swiperight：右滑动

tap：轻敲屏幕

taphold：轻敲并保持一段时间

**(3)内容滚动相关事件**

scrollstart：开始滚动

scrollstop：滚动结束

**(4)屏幕方向改变事件**

orientationchange

**(5)虚拟鼠标事件**

vmouseover

vmousemove

vmouseout

....

在jQM整合NG的项目中为元素绑定滑动事件监听函数：

**(1)使用jQM提供的方法：**

$(document).delegate('目标元素', 'swipeleft', function(){

$.mobile.changePage('main.html', {transition: 'slide'});

})

**(2)使用NG（ngTouch模块）提供的方法：**

提示：页面必须引入angular-touch.js，且自定义模块必须声明依赖ngTouch模块。

<div ng-swipe-left="jump( 'main.html' )">

.controller('控制器名', function($scope){

$scope.jump = function(url){

$.mobile.changePage(url, {});

}

})

面试题：

**如何让元素水平居中？**

行内/行内块元素在父容器中水平居中： .parent{ text-align:center; }

块级(宽度不足100%)在父容器中水平居中： .block { margin-left: auto; margin-right: auto;}

**如何让元素竖直居中——多种方法？**

1. 单行文本/图片垂直居中 .parent {line-height: 容器高}
2. 不适用于老IE：多行文本/图片竖直居中

.outer { display: table; }

.outer .inner { display: table-cell; vertical-align: middle;}

1. 绝对定位+负margin——子容器必须有height

**Day4**

1.WebApp阶段项目——开饭啦项目

所用工具：AngularJS + jQueryMobile

项目进度安排：

day01:

am: 解决NG和jQM整合中的最大冲突

pm: 使用jQM编写静态网页

day02:

am: 编写服务器端动态PHP页面

pm: 使用$http获取动态数据，替换静态内容

jQuery.get(url, function(data){

$scope.xx = data;

$scope.$apply(); //轮询$watch队列，应用Model的修改

}, 'text/plain')

如何使用客户端技术解决多个页面中相同的头部/尾部问题：

<div id="header"></div>

<div id="content">

......

</div>

<div id="footer"></div>

<script>

$(function(){

$('#header').load('tpl/header.html');

$('#footer').load('tpl/footer.html');

})

</script>

<div ng-include=" 'tpl/header.html' "></div>

2.提示：jQM应用中，location.href不再是当前页面的URL，而是上一个页面的URL

1. NG和jQM整合冲突点：

(1)SPA解决方案的选择

(2)jQM动态加载的Page/DOM元素无法绑定Model数据，必需重新编译并链接新添加的DOM节点

(3)在main.html实现：点击菜品列表中的某个菜的小图，可以弹出其对应的大图——popup

提示：jQM中的popup组件不能和NG框架的ngRepeat整合

结论：放弃jQM中popup，自己编写需要的效果

(4)两个Page间如何传递参数

·没有使用ngRoute就没有路由参数。

·<a href="detail.html?did=3">无效——下一个页面使用location对象无法获得当前页面的URL

结论：使用H5本地存储 localStorage/sessionStorage

WebView wv=new WebView(this);

this.setContentView(wv);

wv.getSettings().setJavaScriptEnabled(true);

wv.loadUrl( "http://kaifanla.applinzi.com/index.html" );

4.完整的购物车功能车功能

需求：一个订单可以购买多样商品，每个商品的数量可能各不相同。

数据库设计：

kf\_user表：

uid

uname

upwd

...

kf\_dish表：

did

name

price

img

...

kf\_order表：

oid 订单编号

user\_name 接收人姓名

sex

phone 接收人电话

addr 接收人地址

order\_time 下单时间

uid 用户编号

kf\_order\_detail表

odid 自增编号

order\_id 订单编号

dish\_id 菜品编号

dish\_count 购买数量

1 10 3 1

2 10 5 3

3 10 6 2

4 11 3 2

5 11 6 1

6 12 3 2

找不到网页

http://kaifanla.applinzi.com/处的网页不能载入为：

net::ERR\_CACHE\_MISS

**可能原因：**

(1)当前App未申请网络使用权限，修改AndroidManifest.xml：

<uses-permission android:name="android.permission.INTERNET"/>

(2)打开3G/WIFI网络访问

(3)Web服务器上指定的路径没有index.html、index.php，请求URL中又没有指名具体访问哪一个页面

**Day1**