疯狂系列 -- 关于 AngularJS 的胡思乱想

作者：JamieLee
原文链接：https://ld246.com/article/1593536945546
来源网站：链滴

<img src="https://ld246.com/images/img-loading.svg" alt="" data-src="https://b3logfile
com/bing/20200504.jpg?imageView2/1/w/960/h/540/interlace/1/q/100">
<h2 id="5-1-AngularJS-入门">5.1 AngularJS 入门</h2>

AngularJS 较之于 jQuery 的变化

[jQuery]：会将 HTML 元素包装成 jQuery 对象，开发者需要通过标准的 j
uery 方法来操作对象，完成动态修改 HTML 页面的功能。
[AngularJS]：使用指令，将 HTML 元素与 AngularJS 模型变量进行“双
绑定”，用 JS 脚本来修改模型变量的值，即可让 HTML 页面发生改变。

<table>
<thead>
<tr>
<th align="center">jQuery</th>
<th><img src="https://ld246.com/images/img-loading.svg" alt="image.png" data-src="https
//b3logfile.com/file/2020/06/image-6ed15990.png?imageView2/2/interlace/1/format/jpg"><
th>
</tr>
</thead>
<tbody>
<tr>
<td align="center">AngularJS</td>
<td><img src="https://ld246.com/images/img-loading.svg" alt="image.png" data-src="https
//b3logfile.com/file/2020/06/image-ae53f17f.png?imageView2/2/interlace/1/format/jpg"></
d>
</tr>
</tbody>
</table>
<h3 id="5-1-1-AngularJS-的基本设计">5.1.1 AngularJS 的基本设计</h3>
<h4 id="-AngularJS-的组成-">【AngularJS 的组成】</h4>

组件有哪些?

模块、控制器、过滤器、函数、指令、服务 等

组件间的依赖关系如何管理?

依赖注入

<img src="https://ld246.com/images/img-loading.svg" alt="image.png" data-src="https:
/b3logfile.com/file/2020/06/image-f83bbf94.png?imageView2/2/interlace/1/format/jpg"></
>
<h4 id="-AngularJS-效果举例-">【AngularJS 效果举例】</h4>

实现功能：商品价格 X 商品数量 = 总价

<table>
<thead>

<tr>
<th>页面效果</th>
<th><img src="https://ld246.com/images/img-loading.svg" alt="image.png" data-src="https
//b3logfile.com/file/2020/06/image-8eadc991.png?imageView2/2/interlace/1/format/jpg"><
th>
</tr>
</thead>
<tbody>
<tr>
<td>代码实现</td>
<td><img src="https://ld246.com/images/img-loading.svg" alt="image.png" data-src="https
//b3logfile.com/file/2020/06/image-e7d1c968.png?imageView2/2/interlace/1/format/jpg"><
td>
</tr>
<tr>
<td>代码解析</td>
<td>① 导入 AngularJS 的代码库；
② <body//
gt; 标签中添加的 “ng-app” 指令 -- 用于设置 AngularJS 应用范围（是一个未命名模块）；
<s
rong>③ 为两个 <input//> 分别双向绑定和命名，用 “ng-model” 指定对应的 An
ularJS 变量名为 price 和 num；（“ng-model” 指令，是视图(V)与控制器(C)之间的沟通桥梁）<b
>④ 通过 {{ AngularJS 表达式 }} -- {{ price * num }} 可以看到 HTML 元素 和
ngularJS 变量之间双向绑定的影响，当在 HTML 页面中改变 <input//> 标签的值时，对应的 A
gularJS 变量也会发生变化，表达式的结果也相应改变；</td>
</tr>
</tbody>
</table>
<h4 id="-AngularJS-架构模式-">【AngularJS 架构模式】</h4>

AngularJS 的 MVC（MV Vm）模式

<table>
<thead>
<tr>
<th><img src="https://ld246.com/images/img-loading.svg" alt="image.png" data-src="https
//b3logfile.com/file/2020/06/image-270a8b1c.png?imageView2/2/interlace/1/format/jpg"><
th>
</tr>
</thead>
</table>

为什么说 AngularJS 是 MVVM 模式？？？

<p><img src="https://ld246.com/images/img-loading.svg" alt="image.png" data-src="https:
/b3logfile.com/file/2020/06/image-82c98faa.png?imageView2/2/interlace/1/format/jpg"></
>
<h3 id="5-1-2-AngularJS-下载和安装">5.1.2 AngularJS 下载和安装</h3>

官方站点

<a href="https://ld246.com/forward?goto=https%3A%2F%2Fangularjs.org" title="Angula
JS 是一个纯粹的 JS 库，与其他 JS 库的下载和安装一致" target="_blank" rel="nofollow ugc">Ang
larJS 最新版本下载

版本

AngularJS 1.x（jQuery 版）

真 轻量级 JS 框架，使用 JavaScript

Angular 2.x

开发高度依赖 Node.js，使用 TypeScript 脚本

GitHub 的托管站点

<h4 id="-AngularJS-库-">【AngularJS 库】</h4>
<table>
<thead>
<tr>
<th align="left">angular.min.js</th>
<th align="left">无注释（体积小），适合实际开发</th>
</tr>
</thead>
<tbody>
<tr>
<td align="left">angular.js</td>
<td align="left">带注释（没压缩），适合源码阅读</td>
</tr>
<tr>
<td align="left">angular-xxx.min.js</td>
<td align="left">xxx 表示特定功能。
 如 angular-animate.min.js 是动画支持库；
 如 an
ular-cookies.min.js 是 Cookie 访问支持库；</td>
</tr>
<tr>
<td align="left">angular-xxx.js</td>
<td align="left">xxx 表意同上，带注释没压缩。</td>
</tr>
</tbody>
</table>
<h4 id="-AngularJS-API-文档-">【AngularJS API 文档】</h4>
<table>
<thead>
<tr>
<th>访问地址</th>
<th><a href="https://ld246.com/forward?goto=https%3A%2F%2Fdocs.angularjs.org" target
"_blank" rel="nofollow ugc">在线文档浏览</th>
</tr>
</thead>
<tbody>

<tr>
<td>组成部分</td>
<td>功能函数 function</td>
</tr>
<tr>
<td></td>
<td>指令 directive</td>
</tr>
<tr>
<td></td>
<td>服务 service</td>
</tr>
<tr>
<td></td>
<td>过滤器 filter</td>
</tr>
</tbody>
</table>
<h4 id="-AngularJS-安装方法-">【AngularJS 安装方法】</h4>
<table>
<thead>
<tr>
<th>安装 AngularJS 库</th>
<th align="left">在 HTML 页面中引入 AngularJS 的 JS 文件</th>
</tr>
</thead>
<tbody>
<tr>
<td>代码位置</td>
<td align="left">HTML 页面的开始位置</td>
</tr>
<tr>
<td>代码功能</td>
<td align="left">导入 AngularJS 类库后，可在自己的脚本中使用 AngularJS 的特定功能；</td>
</tr>
<tr>
<td>代码内容</td>
<td align="left">[注：src 属性的值，要对应具体版本的文件名称]
<img src="https://ld246.com/images/img-loading.svg" alt="image.png" data-src="https://b3logfile.com/file/2020/06/image-799278f4.png?imageView2/2/interlace/1/format/jpg"></td>
</tr>
</tbody>
</table>
<h2 id="5-2-表达式">5.2 表达式</h2>
<h3 id="5-2-1-AngularJS-表达式">5.2.1 AngularJS 表达式</h3>
<table>
<thead>
<tr>
<th>功能</th>
<th align="left">在 HTML 页面上生成输出</th>
</tr>
</thead>
<tbody>
<tr>

<td>分类</td>
<td align="left">① 简单表达式；② 复合对象表达式；</td>
</tr>
<tr>
<td>定义</td>
<td align="left">必须在符号 -- 两对花括号 {{ }} 之内</td>
</tr>
</tbody>
</table>
<h3 id="5-2-2-简单表达式">5.2.2 简单表达式</h3>
<table>
<thead>
<tr>
<th>举例</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>元素</td>
<td>① 直接量（如 数字、字符、布尔值）；② 变量；</td>
</tr>
<tr>
<td></td>
<td>① 直接量：<img src="https://ld246.com/images/img-loading.svg" alt="image.png" data
src="https://b3logfile.com/file/2020/06/image-b6209795.png?imageView2/2/interlace/1/for
at/jpg">
[注：标签 <p//> 相当于换行了]</td>
</tr>
<tr>
<td></td>
<td>② 变量：<img src="https://ld246.com/images/img-loading.svg" alt="image.png" data-s
c="https://b3logfile.com/file/2020/06/image-cad65a00.png?imageView2/2/interlace/1/forma
/jpg"></td>
</tr>
</tbody>
</table>
<h3 id="5-2-3-复合对象表达式">5.2.3 复合对象表达式</h3>
<table>
<thead>
<tr>
<th align="left">元素</th>
<th align="left">① 对象；② 数组；③ 简单表达式？</th>
</tr>
</thead>
<tbody>
<tr>
<td align="left">① 对象：</td>
<td align="left"><img src="https://ld246.com/images/img-loading.svg" alt="image.png" dat
-src="https://b3logfile.com/file/2020/06/image-31294ebe.png?imageView2/2/interlace/1/fo
mat/jpg">
【解析】
 一、用“ng-init”初始化了一个对象 fkit 和对象属性 name、doma
n；
 二、在表达式内直接使用对象的属性 fkit.name 和 fkit.domain ；
[注：中文字符用
引号 ' ' 引起]</td>
</tr>
<tr>

<td align="left">② 数组：</td>
<td align="left"><img src="https://ld246.com/images/img-loading.svg" alt="image.png" dat
-src="https://b3logfile.com/file/2020/06/image-ca7f20c6.png?imageView2/2/interlace/1/for
at/jpg">
【解析】
 一、也是用“ng-init”初始化了一个数组 users[] 和数组元素 -- 3 个
象 {name:'...',age:'...'}；
 二、在表达式内直接使用了数组元素的两个属性值 users[i].name 和 u
ers[i].age；</td>
</tr>
</tbody>
</table>
<h3 id="5-2-4-表达式自动处理空指针异常">5.2.4 表达式自动处理空指针异常</h3>

AngularJS 对于空对象的处理

<table>
<thead>
<tr>
<th>容错性</th>
<th>AngularJS 自动判断某个变量是否存在，不存在则解析为空（不是 null）</th>
</tr>
</thead>
<tbody>
<tr>
<td>代码</td>
<td><img src="https://ld246.com/images/img-loading.svg" alt="image.png" data-src="https
//b3logfile.com/file/2020/06/image-2436276e.png?imageView2/2/interlace/1/format/jpg"><
r>① 和 ② 的变量未定义，所以都将显示为空</td>
</tr>
<tr>
<td>显示</td>
<td><img src="https://ld246.com/images/img-loading.svg" alt="image.png" data-src="https
//b3logfile.com/file/2020/06/image-0b05bec9.png?imageView2/2/interlace/1/format/jpg"><
td>
</tr>
</tbody>
</table>

AngularJS 和 JS 处理空对象时的区别

<table>
<thead>
<tr>
<th>AngularJS</th>
<th>JS</th>
</tr>
</thead>
<tbody>
<tr>
<td>处理 null 和 undefined 值时不抛异常</td>
<td>遇见空，会报错</td>
</tr>
<tr>
<td>{{ a.b.c }}</td>
<td>((a 或 {}) .b 或 { }) .c</td>
</tr>

</tbody>
</table>
<h3 id="5-2-5-AngularJS-与-JS-表达式的区别">5.2.5 AngularJS 与 JS 表达式的区别</h3>
<table>
<thead>
<tr>
<th>区别</th>
<th>AngularJS 表达式</th>
<th>JS 表达式</th>
</tr>
</thead>
<tbody>
<tr>
<td>能否在 HTML 页面中出现</td>
<td>放在 {{ }} 中即可</td>
<td>不能</td>
</tr>
<tr>
<td>是否会出现 NullPointerException</td>
<td>自动处理表达式中的 null 和 undefined</td>
<td>会引起空指针错误</td>
</tr>
<tr>
<td>是否有分支、循环和异常处理</td>
<td>表达式不支持</td>
<td>表达式支持</td>
</tr>
<tr>
<td>是否支持过滤器的筛选功能</td>
<td>支持</td>
<td>不支持</td>
</tr>
<tr>
<td>表达式变量</td>
<td>基于 $scope ，是局部变量</td>
<td>基于 window 对象，是全局变量</td>
</tr>
</tbody>
</table>
<h4 id="-访问-JS-变量-">【访问 JS 变量】</h4>

AngularJS 怎么访问 JS 变量或者表达式？？？

通过 $window 服务

<h2 id="5-3-模块与控制器">5.3 模块与控制器</h2>
<table>
<thead>
<tr>
<th>模块的组成部分</th>
<th>module & controller</th>
</tr>

```
</thead>
<tbody>
<tr>
<td><img src="https://ld246.com/images/img-loading.svg" alt="image.png" data-src="https
//b3logfile.com/file/2020/06/image-6f52cb12.png?imageView2/2/interlace/1/format/jpg"></
d>
<td>一、控制器是模块的下级程序单元；<br>二、控制器是模块内的重要作用域；<img src="https
//ld246.com/images/img-loading.svg" alt="image.png" data-src="https://b3logfile.com/file/
020/06/image-b89c7ad0.png?imageView2/2/interlace/1/format/jpg"></td>
</tr>
</tbody>
</table>
<h4 id="-AngularJS-食用方式正解-">【AngularJS 食用方式正解】</h4>
<ul>
<li>先用 ng-app 、ng-module、ng-controller 这些指令划定使用范围，也可同时命名；</li>
<li>再调用 app.module()、app.controller() 这些方法来创建内容和使用；</li>
</ul>
<h3 id="5-3-1-模块的-2-种加载方式">5.3.1 模块的 2 种加载方式</h3>
<ul>
<li>使用“ng-app”指令来加载模块
<ul>
<li>匿名模块：ng-app</li>
<li>命名模块：ng-app = "moduleName"</li>
</ul>
</li>
</ul>
<table>
<thead>
<tr>
<th>加载方式</th>
<th>定义</th>
<th>说明</th>
</tr>
</thead>
<tbody>
<tr>
<td>匿名模块</td>
<td>ng-app 指定空属性值或者不指定值</td>
<td>AngularJS 会自动加载并创建 AngularJS 模块</td>
</tr>
<tr>
<td>命名模块</td>
<td>ng-app 指定属性值，即给新模块命名</td>
<td>需要再调用 angular 对象的 module() 方法</td>
</tr>
</tbody>
</table>
<h3 id="5-3-2-module---对象方法创建模块">5.3.2 module() 对象方法创建模块</h3>
<table>
<thead>
<tr>
<th align="left">语法格式</th>
<th align="left">angular.module( name, [requires], [configFunction])<br>&lt; 参数 &gt;<br>
 模块名 -- “字符串”；<br>② 依赖模块 -- 数组；<br>③ 配置函数 -- 此参数可略，因为 module()
```

的返回值是它创建的 angular.Module 对象，而对象中包含 config() 方法，调用对象的 config() 方也能传入 configFunction 参数；</th>
</tr>
</thead>
<tbody>
<tr>
<td align="left">创建命名模块</td>
<td align="left"><img src="https://ld246.com/images/img-loading.svg" alt="image.png" dat-src="https://b3logfile.com/file/2020/06/image-a1e9a30b.png?imageView2/2/interlace/1/format/jpg">
① 用 ng-app 指令创建命名模块“fkApp”;
② 调用 module() 时，模块名作一参传入；
③ 依赖数组为空，作为二参传入 -- 表明不依赖其他模块；</td>
</tr>
</tbody>
</table>
[总结]：

先用 ng-app 指令命名 & 绑定标签；
调用 angular.module() 方法创建模块；

<h4 id="-模块对象的-2-个全局属性方法-">【模块对象的 2 个全局属性方法】</h4>
<table>
<thead>
<tr>
<th>方法</th>
<th>定义</th>
<th>（模块内）作用范围</th>
</tr>
</thead>
<tbody>
<tr>
<td>moduleObject.value(globalVariableName,value)</td>
<td>添加全局变量</td>
<td>一处定义，到处作用</td>
</tr>
<tr>
<td>moduleObject.constant(globalConstantName,value)</td>
<td>添加全局常量</td>
<td>一处定义，到处作用</td>
</tr>
</tbody>
</table>
<h4 id="-controller---对象方法创建控制器-">【controller() 对象方法创建控制器】</h4>

控制器 -- 依旧是先定义（名称），后注册；

控制器变量 -- 先作为函数形参传入，后操作；

<table>
<thead>
<tr>
<th><img src="https://ld246.com/images/img-loading.svg" alt="image.png" data-src="https//b3logfile.com/file/2020/06/image-fcc049b5.png?imageView2/2/interlace/1/format/jpg"></

```
h>
</tr>
</thead>
</table>
<h3 id="5-3-3-多个模块怎么加载">5.3.3 多个模块怎么加载</h3>
<ul>
<li>RULE：默认仅仅启动第一个模块
<ul>
<li>其余模块可用 angular.bootstrap(...); 启动</li>
</ul>
</li>
</ul>
<h4 id="-bootstrap--指定加载模块-">【bootstrap（）指定加载模块】</h4>
<table>
<thead>
<tr>
<th align="center">bootstrap()<br>方法<br>示例代码</th>
<th><img src="https://ld246.com/images/img-loading.svg" alt="image.png" data-src="https
//b3logfile.com/file/2020/06/image-0eb4d56e.png?imageView2/2/interlace/1/format/jpg"><
th>
</tr>
</thead>
<tbody>
<tr>
<td align="center"><strong>代码 ② 解析</strong></td>
<td>1、方法参数说明：bootstrap( DOM 对象，[某数组] )；<br>2、一参：DOM 对象需要对应 H
ML 元素；二参：要启动的模块列表；<br>3、bootstrap() 启动模块代码需要在最后；</td>
</tr>
</tbody>
</table>
<h3 id="5-3-4-作用域--scope-对象">5.3.4 作用域 $scope 对象</h3>
<ul>
<li>以下没看懂</li>
</ul>
<blockquote>
<p>调用模块的 controller（）方法注册控制器时，传入的第二个参数</p>
<p>是一个函数，在该函数中可声明一个 $scope 形参，该形参就表示了</p>
<p>AngularJS 控制器<strong>对应的作用域</strong>。①</p>
<p>~~从 AngularJS 的设计架构来看，$scope 充当了 <strong>AngularJS 应用的模</strong><b
>
~~</p>
<p><strong><del>型</del></strong>，因此通过 $scope 作用域指定的属性，和 HTML 中 <str
ng>ng-model</strong> 对应</p>
<p>的变量是双向绑定的，也可自动显示在与指定 <strong>ng-bind</strong> 绑定的 HTML 元<
p>
<p>素中。②</p>
</blockquote>
<table>
<thead>
<tr>
<th>糊涂点</th>
<th></th>
</tr>
</thead>
```

<tbody>
<tr>
<td>①</td>
<td>意思是 $scope 在哪，控制器就只能在哪起作用呗。（那 scope 是一个黑帮老大，它罩着控制，在哪儿坐镇，控制器就可以在哪里威风）</td>
</tr>
<tr>
<td>②</td>
<td>$scope 作用域指定的“属性”和 ng-xxx 这两条指令绑定的“HTML 元素”勾勾搭搭的，可能什么亲戚关系。: P
---- 就是说 scope 的左膀右臂就是 ng-model 和 ng-bind ， scope 通过这个指令小弟绑架“HTML 元素”做这做那的。</td>
</tr>
</tbody>
</table>
注：ng-model 指令作用是，给“HTML 元素”起个 AngularJS 的家名；
ng-bind 指令作用是，将“表达式的值”与“HTML 元素”进行绑定；
<h4 id="--scope-的用法-">【$scope 的用法】</h4>
<table>
<thead>
<tr>
<th>说明</th>
<th>$scope 有点类似 C 中的命名空间，通过点 . 来访问属性</th>
</tr>
</thead>
<tbody>
<tr>
<td>代码</td>
<td><img src="https://ld246.com/images/img-loading.svg" alt="image.png" data-src="https//b3logfile.com/file/2020/07/image-8b6a3087.png?imageView2/2/interlace/1/format/jpg"><td>
</tr>
<tr>
<td>解析</td>
<td>---- 只需要在控制器的 function() 中传入 $scope 象，就能点调所有与 ng-model 和 ng-bind 有关的“HTML 元素”让这些元素做这做那的。</td>
</tr>
<tr>
<td>页面效果</td>
<td><img src="https://ld246.com/images/img-loading.svg" alt="image.png" data-src="https//b3logfile.com/file/2020/07/image-fc0b733a.png?imageView2/2/interlace/1/format/jpg"></d>
</tr>
</tbody>
</table>
<h4 id="--scope-的第三个指令小弟-ng-click-">【$scope 的第三个指令小弟 ng-click】</h4>

前两个指令小弟 ng-model 和 ng-bind 太好使了，$scope 大佬又收了一个指令小弟 ng-click 这回这个小弟不绑定名称、表达式的了(太小儿科)，会给“HTML 元素”绑定一些方法啥的(显得高级点)。
这回 $scope 大佬也不点调名称和表达式了，点调方法，牛鼻不牛鼻

<table>
<thead>
<tr>

<th>演示一手点调</th>
<th>点一次方法，执行一下 count++（原本是 0 嘛），顺带输出 log 记录</th>
</tr>
</thead>
<tbody>
<tr>
<td>代码</td>
<td><img src="https://ld246.com/images/img-loading.svg" alt="image.png" data-src="https
//b3logfile.com/file/2020/07/image-f27af0b4.png?imageView2/2/interlace/1/format/jpg"></
d>
</tr>
</tbody>
</table>
<h4 id="-模型的作用-">【模型的作用】</h4>

定义传递给视图的数据内容，承载部分数据逻辑
---- 也就是说 $scope 对象可以定义啥内容能传到 HTML 页面上，且能够处理一些数据逻辑

<h4 id="--scope-能控制的指令有哪些-">【$scope 能控制的指令有哪些】</h4>

那么问题来了，$scope 大佬最后究竟收了几个指令小弟呢?

不到 :joy:
可能 AngularJS 家的指令（得是控制器级别以下的？）都已经是 $scope 大佬的小弟了吧 ：P</l
>

<h3 id="5-3-5-作用域--rootScope-对象">5.3.5 作用域 $rootScope 对象</h3>

就 $scope 能罩控制器还不算啥，这个 rootScope 对象才是老大中的老大，就叫它“教父”吧
教父能罩整个 AngularJS 应用，当然手下可以有很多个 scope 。

这个 rootScope 点调的那些 属性、方法，可以在各个 scope 控制范围内通用的。

<table>
<thead>
<tr>
<th>作用域</th>
<th>指定内容</th>
<th>使用范围</th>
<th>是否可用</th>
</tr>
</thead>
<tbody>
<tr>
<td><a href="https://ld246.com/forward?goto=mailto%3Ascope%40fkCtrl" target="_blank"
el="nofollow ugc">scope@fkCtrl</td>
<td>$scope.bookName</td>
<td>@fkCtrl</td>
<td>本控制器，可用</td>
</tr>

<tr>
<td>$rootScope@fkCtrl</td>
<td>$rootScope.theName</td>
<td>@myCtrl</td>
<td>跨控制器，仍可用</td>
</tr>
</tbody>
</table>
<table>
<thead>
<tr>
<th>代码</th>
<th><img src="https://ld246.com/images/img-loading.svg" alt="image.png" data-src="https
//b3logfile.com/file/2020/07/image-de052be1.png?imageView2/2/interlace/1/format/jpg"><
th>
</tr>
</thead>
</table>
<h3 id="5-3-6-黑帮老大和教父的优先级">5.3.6 黑帮老大和教父的优先级</h3>

如果黑帮老大管的属性啥的和教父管的属性重名了，AngularJS 会先访问哪个?

先访问 scope.zhangSan 还是 rootScope.zhangSan ？

答：当然是访问 scope 的啊，你是不是傻，你想啊，教父的是你想见就能见的吗 ：P

你都不一定会知道教父身边有这么一个人存在
scope 属性会覆盖 rootScope 中的同名属性

