

# Angular01

两个推荐的网站:

文本解决问题: 百度

视频解决问题: B站 目前最好的免费技术分享网站: 哔哩哔哩

## 简介

讲师相关:

- 纪盈鑫
- 小新老师
- 联系方式: vx 18800108022

FTP地址:

地址: code.tarena.com.cn

账号: tarenacode

密码: code\_2013

## 课程介绍

### 大 前端开发工程师

- web前端工程师: 开发网页的
- 大前端: 开发网页web + 开发App: 手机端

前3个阶段: 学习之后 就可以开发网站, 可以称为 web开发工程师!

目前企业的要求: 复合型人才

阶段	讲师	内容
1	亮亮	web开发基础-HTML SQL JS
2	然然	web开发基础-CSS ajax
3	东东	web开发基础-JS高级 htm15 vue 正则
4	华哥	web高级+app: vuex htm15高级 小程序--App
5	小新	三大框架--React Vue Angular. 首尾体验 全栈编程: 服务器+前端+UI切图 一个人

### 详说第五阶段

课程	时长	目标
Angular + ionic	6天	学会angular框架 -- Google公司 ionic 手机端组件库-- 相当于mintUI
React + ReactNative	7天	学会react框架 -- Facebook公司 ReactNative 手机端组件库 -- 相当于mintUI
Python	4.5天	体验全栈 -- 学习服务器开发结合vue, 体验前后端分离项目的制作
VuePro	2.5	使用 vue 制作网站 --- 回顾所有vue知识点 git上传到新浪云 进行发布

## 本阶段教学特点

本阶段偏 **应用型开发**，大量使用提高开发效率的 插件，工具； 熟练使用 **面向百度编程**。

最不希望听到的问题：xxx是什么。

## Angular

Angular属于前端 三大框架之一！

- Angular：中大型项目。来自Google 公司维护
- Vue：中小型项目。来自开源社区
  - Vue的最初作者：Google的离职程序员 -- angular 与 vue 非常相似
- React：中小型项目。来自Facebook公司

Angular官方网站：<https://angular.cn/>

## 脚手架的安装

同vue，官方提供了脚手架，可以快速生成项目包，进行开发！

要求：

- node的版本必须 **>=10.9**，推荐使用最新版本

下载地址：<http://nodejs.cn/download/>

```
1 node -v
```

- 检查 npm 的中国镜像

npm默认从国外源下载数据，非常慢。

taobao 每隔几秒就从国外服务器下载所有内容到国内的服务器，这个国内服务器就是中国镜像

```
1 # 有 taobao 字样的结果 是正确的
2 npm config get registry
3
4 # 切换为中国镜像
5 npm config set registry https://registry.npm.taobao.org
```

- 安装 angular 的脚手架

带-g: 就是全局安装命令. -g是 global 全局

```
1 npm i -g @angular/cli
```

安装过程中, 有问题: 直接 回车 即可.

```
? Would you like to share anonymous usage data with the Angular Team at Google under Google's Privacy Policy at https://policies.google.com/privacy? For more details and how to change this setting, see http://angular.io/analytics. No
+ @angular/cli@11.0.5
added 254 packages from 201 contributors in 86.373s
```

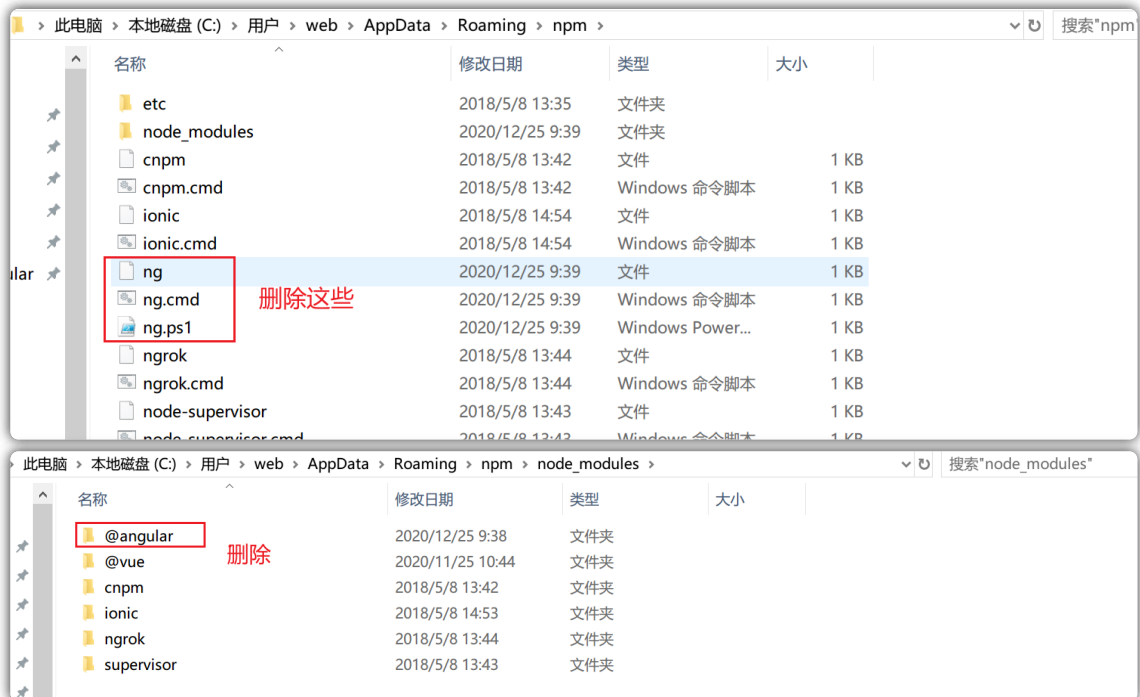
- 查看已安装的脚手架版本

```
1 ng v
```

- 常见报错: 已经安装过, 再次安装会因为电脑权限问题, 拒绝覆盖

报错关键词: File Exists

解决办法: 手动到 npm 目录下, 删除已安装的 ng 相关的程序即可!



## 生成项目包

因为网络问题, 时间可能会比较久. 生成不成功, 使用 其他人生成 的包也可以!

生成命令: `ng new 包名`

注意: cmd在哪个目录下打开, 就会生成在此目录下

问题: 过程中的所有询问都 回车 确认即可!

例如: `ng new ngpro`

```
CREATE ngpro/src/app/app.component.css (0 bytes)
CREATE ngpro/e2e/protractor.conf.js (904 bytes)
CREATE ngpro/e2e/tsconfig.json (274 bytes)
CREATE ngpro/e2e/src/app.e2e-spec.ts (656 bytes)
CREATE ngpro/e2e/src/app.po.ts (274 bytes)
✓ Packages installed successfully.
D:\webtn2008\04_Angular>
```

## 启动命令

在 **项目包根目录** 下方执行：

项目包的 `package.json` 所在目录下

```
1 ng s -o
```

报错：有大片爆红，关闭防火墙

(node:3260) UnhandledPromiseRejectionWarning: Error: spawn EPERM

使用其它人提供包的同学，执行命令时要添加 `npx` 前缀

`npx`：npm提供的一个命令，会优先执行项目包下方的 模块，如果没有才会执行全局模块

项目包中已经含有了所有需要的模块，就算不全局安装angular 一样能用！

命令：`npx ng s -o`

## 开发环境的配置

古话：工欲善其事必先利其器。

VSCode的可扩展性非常强，可以通过各种插件进行调整。

小新老师的外号：插件小王子



**Angular 10 Snippets - TypeScript, Html, Angula...** 10.0.3 1.7M ★ 4.5

242 Angular Snippets (TypeScript, Html, Angular Material, Flex Layout, ...

Mikael Morlund




**Angular Language Service** 0.1100.2
 1.9M ★ 3.5

Editor services for Angular templates  
 Angular

angular基础包, 提供基础的变色  
 报错等功能


**Chinese (Simplified) Language Pack for Visual Studio Code** 1.52.2
 7.6M ★ 5

中文(简体)  
 Microsoft

中文界面

此扩展已全局启用。


**Live Server** 5.6.1
 8.9M ★ 4.5

Launch a development local Server with live reload feature for static & dynamic pages  
 Ritwick Dey

简易服务器


**Vetur** 0.31.3
 6.3M ★ 4.5

Vue tooling for VS Code  
 Pine Wu

vue必备, 暂时可以禁用


**Vue 3 Snippets** 1.0.4
 1.5M ★ 5


A Vue.js 3 And Vue.js 2 Code Snippets Extension  
 hollowtree

vue代码块, 暂时可以禁用


**vscode-icons** 11.0.0
 6.7M ★ 5


Icons for Visual Studio Code  
 VSCode Icons Team

修改 图标


**Image preview** 0.26.2
 303K ★ 5

Shows image preview in the gutter and on hover  
 Kiss Tamás

侧边栏图片预览


**Path Intellisense** 2.3.0
 3.9M ★ 5

Visual Studio Code plugin that autocompletes filenames  
 Christian Kohler


**Prettier - Code formatter** 5.8.0
 9.8M ★ 4

Code formatter using prettier  
 Prettier

代码 一键美颜: 自动格式化

设置

save

找到 12 个设置

打开设置同步

常用设置 (1)

文本编辑器 (6)

正在格式化 (2)

文件 (3)

Editor: Format On Save

☒ 在保存时格式化文件。格式化程序必须可用, 延迟后文件不能保存, 并且编辑器不能关闭。

勾选后, ctrl+s 就能自动格式化代码

右键文件, 选择 格式化文档的方式


选择格式化程序

Prettier - Code formatter

JavaScript 和 TypeScript 的语言功能


配置默认格式化程序...


点击之后, 选Prettier




**TabOut** 0.2.0
 81K ★ 4.5


Tab out of quotes, brackets, etc  
 Albert Romkes **按tab键, 把光标移动到) "' 外**






**Auto Rename Tag** 0.1.5
 **自动 同步修改双标签**
4.2M ★ 3

Auto rename paired HTML/XML tag  
 Jun Han
 



**Auto Close Tag** 0.5.9
 **自动闭合 标签**
3.8M ★ 4

Automatically add HTML/XML close tag, same as Visual Studio IDE or Subli...  
 Jun Han
 

福利网站, 编程字体

<http://www.programmingfonts.org>

## 项目的启动

一定要在 **项目根目录** 下执行启动命令!!!

一定要在 **项目根目录** 下执行启动命令!!!

一定要在 **项目根目录** 下执行启动命令!!!

package.json 所在 就是根目录

启动命令:

随着学习: angular的作者非常贴心! 常见问题: 使用者英文太差, 准确率低!

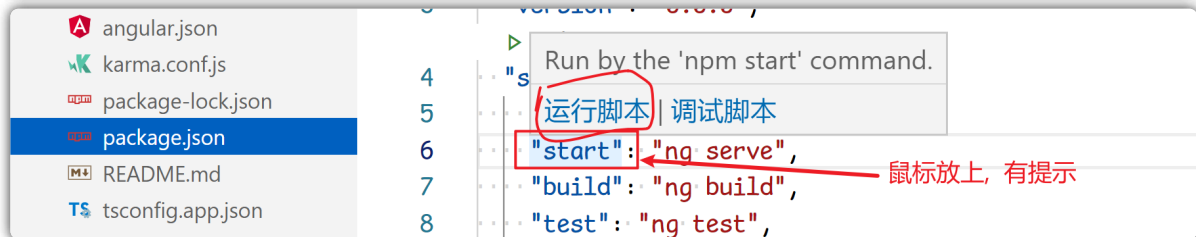
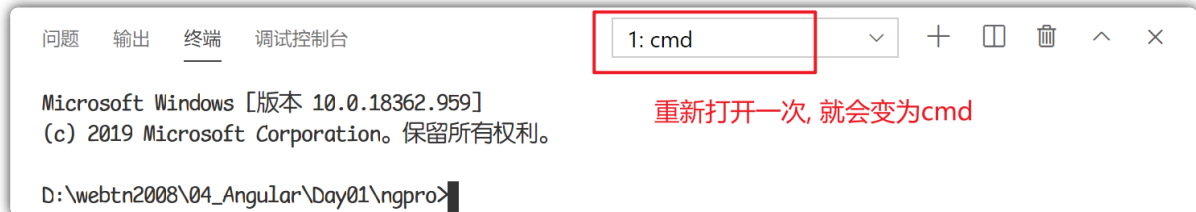
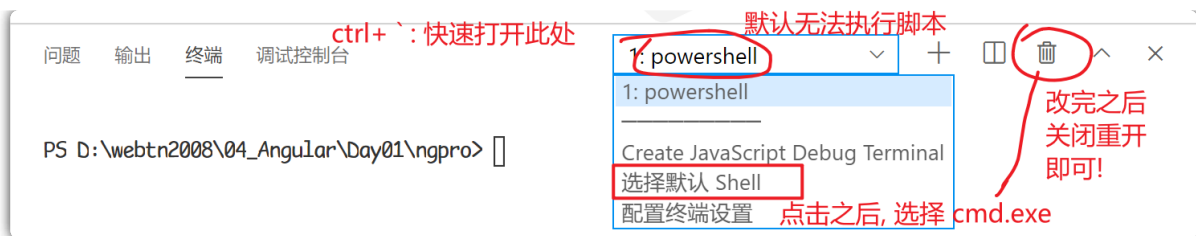
```

1 原始版: 启动之后需要手动到浏览器输入: localhost:4200 打开网站
2  ng serve
3
4 简化:
5  ng s
6
7 自动化: 自动在默认浏览器打开网站
8  ng s -o
9
10 -o: 是 --open 的缩写
  
```

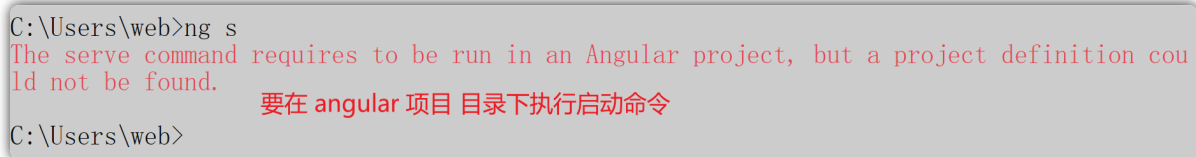
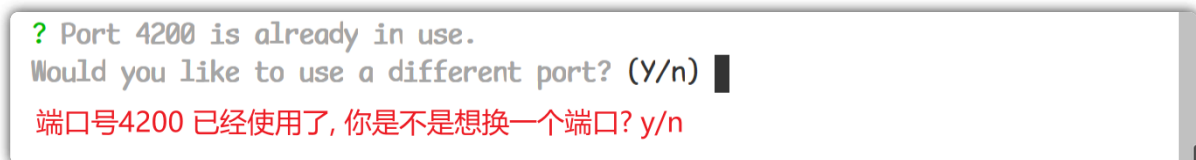
注意事项:

windows提供了两款执行命令行的工具:

- cmd: 旧版
- powershell: 新版, 特点: 默认不支持脚本的执行!!
  - 所以 所有的命令行 千万不要在 powershell 执行



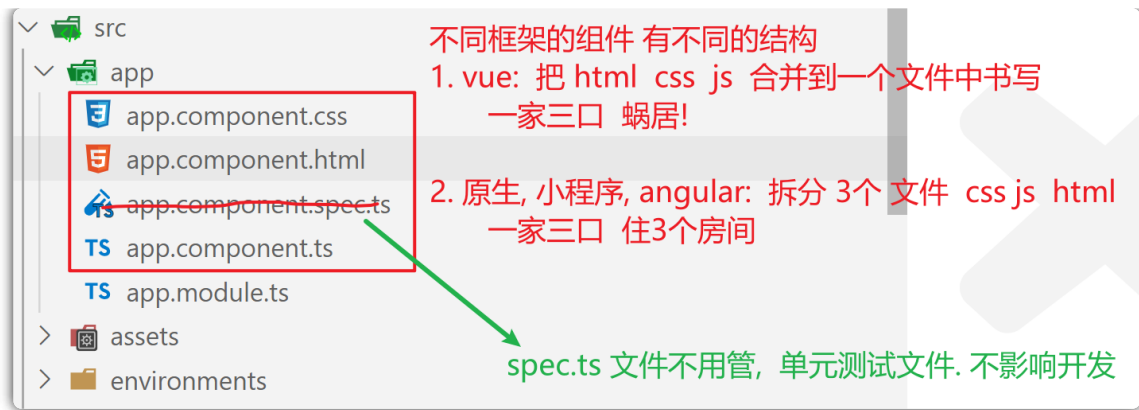
注意重复启动的问题!



## 项目的启动流程

当在浏览器中输入 <http://localhost:4200/> 发生了什么?

- **localhost**: 域名. 计算机在网络中的名称. 此处相当于 **我** 代表当前计算机
- **4200**: 端口号. 计算机中的程序的唯一名称. 此处4200 就是 angular服务器的名称.
- **localhost:4200**: 访问当前计算机上 唯一标识是4200的的程序
- angular服务器的处理
  - 类似于公司: 有一个岗位--前台. 负责接待来访人员
  - angular服务器的前台: 是 `index.html` 文件
- `index.html` 中有一个标签: `<app-root></app-root>`
  - 这个标签就是 自定义组件
- webpack工具: 自动打包工具
  - angular默认集成此工具, 会自动打包 `main.ts` 文件到 `index.html` 中
- `main.ts` 中加载了 `app/app.module.ts`
- `app.module.ts`: 是全局配置文件
- 根组件: `app.module.ts`



angular的逻辑语言: 是 TypeScript 是 JavaScript 的升级版 由 微软公司维护.

把一些java的特征 加入到 JavaScript 里.

与JavaScript使用 没有什么区别. -- 新特性后期会讲

## 组件结构

angular组件由三个文件组成:

- html
- css
- ts: 主文件, 负责合并 html 和 css

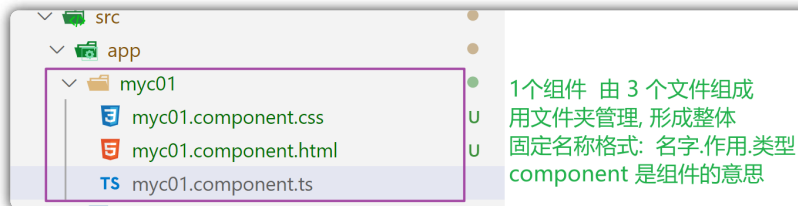
```
1 // 东哥: 长脑子... JS逻辑部分
2 /**
3  * TS语言: 推荐 面向对象开发方式. 所以会有大量的 类 class
4  *
5  * 关于主体问题:
6  * 原生开发中: 主体是 html. 在 html 中引入 JS 和 CSS
7  * angular中: 主体 ts. 在 ts 文件中引入 html 和 css
8  */
9 import { Component } from '@angular/core';
10
11 @Component({
12   selector: 'app-root',
13   // 引入html
14   templateUrl: './app.component.html',
15   // 引入css: 可以多个css
16   styleUrls: ['./app.component.css']
17 })
18 export class AppComponent {
19   title = 'ngpro';
20 }
21
```

## 组件的制作与使用

组件的制作:



- angular的组件 由 3 个文件组成: `html css ts`
- 文件的命名有固定格式: `名称.作用.类型`
  - 例如: `app.component.css` 文件名是app, 作用是组件, 类型是css
  - 例如: `app.module.ts` 文件名是app, 作用是 模块, 类型是ts
  - 例如: `然然.渣.男人` 名字然然, 作用:渣, 类型 男人



- 注册组件: `app.module.ts`

```

6 // 引入组件
7 import { Myc01Component } from './myc01/myc01.component';
8
9 @NgModule({
10   // 注册全局组件
11   declarations: [AppComponent, Myc01Component],
12   imports: [BrowserModule],
13   providers: [],
14   bootstrap: [AppComponent],
15 })

```

制作 myc02 组件

- 制作文件夹: `src/app/myc02`
- 在此文件夹中, 生成3个文件:
  - `myc02.component.html`: 用 `h3` 标签, 内容 `你好, 我的myc02`
  - `myc02.component.css`: 绿色文字, 蓝色边框
  - `myc02.component.ts`:
    - 使用 `ng-component` 生成基本结构. 然后进行修改, 适配当前环境
- 注册全局组件: `app.module.ts`;
- 在根组件中使用: `app.component.html`

## 快捷命令

自动化命令:

`ng generate component 组件名`

- `generate` 生成
- `component` 组件

简化:

`ng g c 组件名`

例如: 生成myc03组件: `ng g c myc03`

```

CREATE src/app/myc03/myc03.component.html (20 bytes)
CREATE src/app/myc03/myc03.component.spec.ts (619 bytes)
CREATE src/app/myc03/myc03.component.ts (271 bytes)
CREATE src/app/myc03/myc03.component.css (0 bytes)
UPDATE src/app/app.module.ts (583 bytes) 自动更新

```

4个组件文件

## 变量的绑定

在vue中: 使用 {{ }} 把 JS 的变量 绑定到 html 中

{{}} 还可以进行很多 JS 的代码操作

angular中没有差异!

```
1 import { Component, OnInit } from '@angular/core';
2
3 @Component({
4   selector: 'app-myc03',
5   templateUrl: './myc03.component.html',
6   styleUrls: ['./myc03.component.css'],
7 })
8 export class Myc03Component implements OnInit {
9   // 面向对象中属性的写法: 属性名 = 值;
10  name = 'dongdong';
11  age = 33;
12  married = true;
13
14  names = ['东东', '亮亮', '然然'];
15
16  boss = { name: '文华', age: 40 };
17
18  constructor() {}
19
20  ngOnInit(): void {}
21 }
22
23 /**
24  * ts文件 就相当于 vue 的 script 范围, 用来书写 JS 代码
25  */
26
27 // 关注 面向对象 和 对象 在写法上的不同
28 let obj = {
29   name: '东东',
30   age: 33,
31 };
32
33 class Demo {
34   name = '东东';
35   age = 33;
36 }
37
```

```
1 <p>myc03 works!</p>
2
3 <!-- 属性的绑定 -->
4 <ul>
5   <li>{{ name }}</li>
6   <li>{{ age }}</li>
7   <li>{{ married }}</li>
8   <li>{{ names }}</li>

```

```

9      <li>{{ boss }}</li>
10     <!-- 数组下标取值 -->
11     <li>{{ names[0] }}</li>
12     <!-- 对象 属性取值 -->
13     <li>{{ boss.name }}</li>
14 </ul>
15
16 <!-- 数学运算符 -->
17 <ul>
18     <li>{{ 3 + 6 }}</li>
19     <li>{{ 3 - 6 }}</li>
20     <li>{{ 3 * 6 }}</li>
21     <li>{{ 3 / 6 }}</li>
22     <li>{{ 3 % 6 }}</li>
23 </ul>
24
25 <!-- 比较运算符 -->
26 <ul>
27     <li>{{ age > 20 }}</li>
28     <li>{{ age >= 20 }}</li>
29     <li>{{ age < 20 }}</li>
30     <li>{{ age <= 20 }}</li>
31     <li>{{ age == 20 }}</li>
32     <li>{{ age != 20 }}</li>
33 </ul>
34
35 <!-- 逻辑运算符 -->
36 <!--
37     逻辑或 || 或者 松散 有真则真 全假为假
38     逻辑与 && 并且 严格 全真则真 有假为假
39     逻辑非 ! 不是 非真为假 非假为真
40 -->
41 <ul>
42     <li>{{ true || false }}</li>
43     <li>{{ false || false }}</li>
44     <li>{{ true && true }}</li>
45     <li>{{ true && false }}</li>
46     <li>{{ !true }}</li>
47     <li>{{ !false }}</li>
48 </ul>
49
50 <!-- 三目运算符: 条件 ? 真值 : 假值 -->
51 <ul>
52     <li>{{ married ? "已婚" : "未婚" }}</li>
53 </ul>
54
55 <!-- 对象方法使用 -->
56 <ul>
57     <li>{{ name.toUpperCase() }}</li>
58 </ul>
59

```

## 属性的绑定

在 vue 中, 对应 v-bind, 简写:

:src="路径"

生成组件: `ng g c myc04`

```
1 <p>myc04 works!</p>
2
3 <!--
4   在 vue 中:
5     :src="imgUrl"
6   -->
7 <!--
8   在 ng 中:
9     [src]="
10  -->
11 <img [src]="imgUrl" alt="" />
12
13 <!-- 另一种写法 -->
14 
15
```

## 事件的绑定

在 vue 中:

@事件名="方法名" 或 v-on:事件名="方法名"

例如: @click=""

生成组件: `ng g c myc05`

```
1 <p>myc05 works!</p>
2
3 <!-- 事件 -->
4
5 <!-- 在 vue 中: @事件名="方法名" -->
6 <!-- 在 ng 中: (事件名)="方法名" -->
7
8 <button (click)="show()">点我</button>
9
10 <!--
11   与 vue 对比的小差异:
12   在 vue 中, 方法如果没有参数, 可以省略()
13   但在 ng 中, 不允许省略()
14   -->
15
```

## 双向数据绑定

在 vue 中使用 `v-model`

生成组件: `ng g c myc06`

### 注册Forms模块

```
13 // 引入: Form 模块
14 import { FormsModule } from '@angular/forms';
15
16 @NgModule({
17   // 注册全局组件
18   declarations: [AppComponent, Myc01Component, Myc02Component, Myc03Component,
19     Myc04Component, Myc05Component, Myc06Component],
20   // 注册模块
21   imports: [BrowserModule, FormsModule],
22   providers: [],
23   bootstrap: [AppComponent],
24 })
```

```
1 <p>myc06 works!</p>
2
3 <!-- 双向数据绑定 -->
4
5 <!-- 在 vue 中, 使用 v-model -->
6 <!--
7   在 ng 中, 快捷提示 ng-model
8   实际写法: [(ngModel)]
9 -->
10 <!-- 实时监听输入框的变化, 事件 ngModelChange -->
11 <input type="text" [(ngModel)]="word" (ngModelChange)="_doChange()" />
12 <p>{{ word }}</p>
13 <!--
14   注意!!!
15   angular默认不加载 Form 模块, 不支持双向数据绑定
16   必须手动到 app.module.ts 中手动加载模块, 并重启
17 -->
18
19 <!--
20   双向数据绑定:
21   方向1: 数据 绑定到 UI
22   方向2: UI变化时, 会同步更新到数据 -- Form表单
23 -->
24
```

```
1 import { Component, OnInit } from '@angular/core';
2
3 @Component({
4   selector: 'app-myc06',
5   templateUrl: './myc06.component.html',
6   styleUrls: ['./myc06.component.css'],
7 })
8 export class Myc06Component implements OnInit {
9   constructor() {}
10
11   word = '123546';
12 }
```

```

12
13   _doChange() {
14     console.log(this.word);
15   }
16
17   ngOnInit(): void {}
18 }
19

```

## html展示

在 vue 中: v-html

生成组件: `ng g c myc07`

```

1 <p>myc07 works!</p>
2
3 <!--
4   在 vue 中: v-html="data"
5   -->
6
7 <!-- {{{ 相当于 v-text 原生的 innerText -->
8 <p>{{{ data }}}</p>
9
10 <!-- 在 ng 中: [innerHTML]=" " -->
11 <p [innerHTML]="data"></p>
12

```

```

1 import { Component, OnInit } from '@angular/core';
2
3 @Component({
4   selector: 'app-myc07',
5   templateUrl: './myc07.component.html',
6   styleUrls: ['./myc07.component.css'],
7 })
8 export class Myc07Component implements OnInit {
9   data = '<h1>Hello World!</h1>';
10
11   constructor() {}
12
13   ngOnInit(): void {}
14 }
15

```

## 样式

在 vue 中:

- `:style="{样式名: 值...}"`
- `:class="{样式类: true/false}"` 根据true/false 来决定是否生效

新建组件: `ng g c myc08`

```

1 import { Component, OnInit } from '@angular/core';
2
3 @Component({
4   selector: 'app-myc08',
5   templateUrl: './myc08.component.html',
6   styleUrls: ['./myc08.component.css'],
7 })
8 export class Myc08Component implements OnInit {
9   // 变量: 运行期间可以变化的量
10   size = 17;
11
12   color = 'black';
13
14   constructor() {}
15
16   ngOnInit(): void {}
17 }
18

```

```

1 <p>myc08 works!</p>
2
3 <!-- 静态样式: 运行期间不会变化 -->
4 <p style="font-size: 20px; color: green">Hello World!</p>
5
6 <!-- 动态样式: 运行期间可以变化 -->
7 <!-- 快捷: ng-style -->
8 <!-- 对象类型 {属性名:值} 属性名的命名规范: 只允许 数字 字母 _
9 font-size: 不能做属性名, 因为 - 是非法字符.
10 解决方案: fontSize 或 'font-size'
11 -->
12 <button (click)="size = size + 1">变大</button>
13 <br />
14 <button (click)="color = 'green'">绿green</button>
15 <button (click)="color = 'red'">红red</button>
16 <button (click)="color = 'blue'">蓝blue</button>
17 <p [ngStyle]="{ fontSize: size + 'px', color: color }">Hello World!</p>
18 <!--
19 vue中相同:
20 (click)="js代码"
21 本质就是点击后 执行 "" 里的代码
22 -->
23
24 <hr />
25
26 <!-- 动态的 class -->
27 <!-- 需求: size 20~30 success 30~ danger -->
28 <!-- 快捷 ng-class -->
29 <span
30   class="btn"
31   (click)="size = size + 1"
32   [ngClass]="{ success: size >= 20 && size < 30, danger: size >= 30 }"
33   >{{ size }}</span>
34 >

```

## 作业

## 作业

- 单价大于 10000 之后, 变为 橙色 orange
- 总价小于 1w5, 显示为绿色

iPhone ¥8999 - 4 + ¥35886

涨价 点击可以把价格 +100

iPhone 修改输入框 可以修改上方的名称