# 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 html5<br>vue 正则 |
| 4 | 华哥 | web高级+app: vuex html5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<br>ionic 手机端组件库-- 相当于mintUI |
| React + ReactNative | 7天 | 学会react框架 -- Facebook公司<br>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,<br>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<br>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!

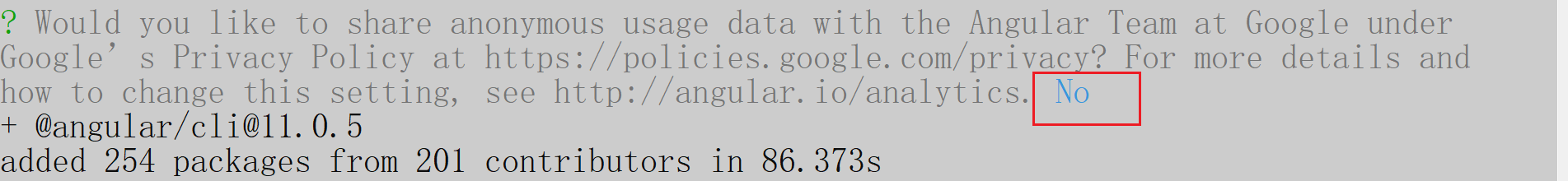
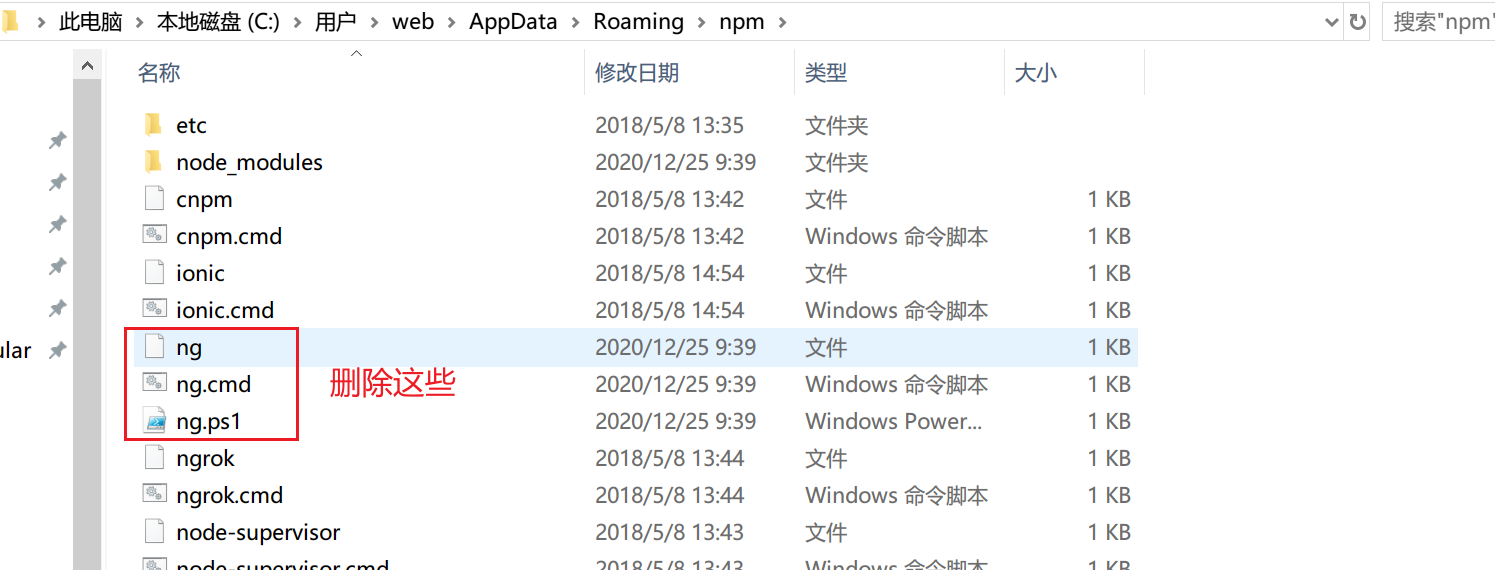
* Angular: 中大型项目. 来自Google 公司维护
* Vue: 中小型项目. 来自开源社区
  + Vue的最初作者: Google的离职程序员 -- angular 与 vue 非常相似
* React: 中小型项目. 来自Facebook公司

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/
* 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
* # 有 taobao 字样的结果 是正确的  
  npm config get registry  
    
  # 切换为中国镜像  
  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
* npm i -g @angular/cli
* 安装过程中, 有问题: 直接 回车 即可.
* 
* 查看已安装的脚手架版本
* 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

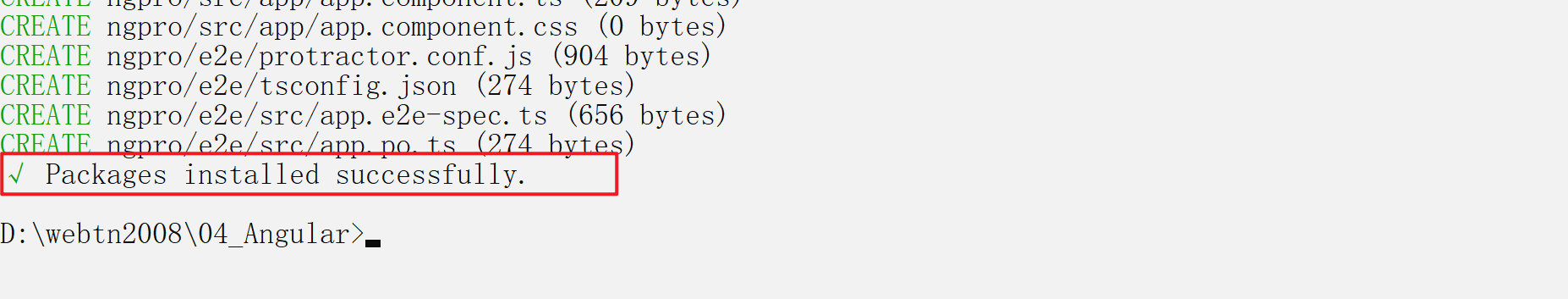
因为网络问题, 时间可能会比较久. 生成不成功, 使用其他人生成的包也可以!

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



### 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

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

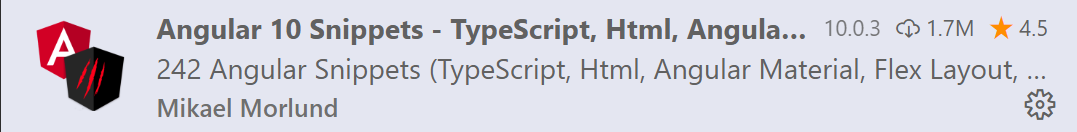


image-20201225141856698



image-20201225141947036



image-20201225104238789



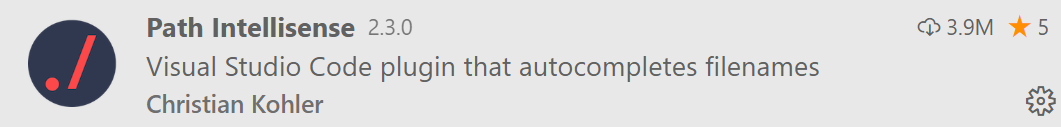


image-20201225104357498



image-20201225104456228

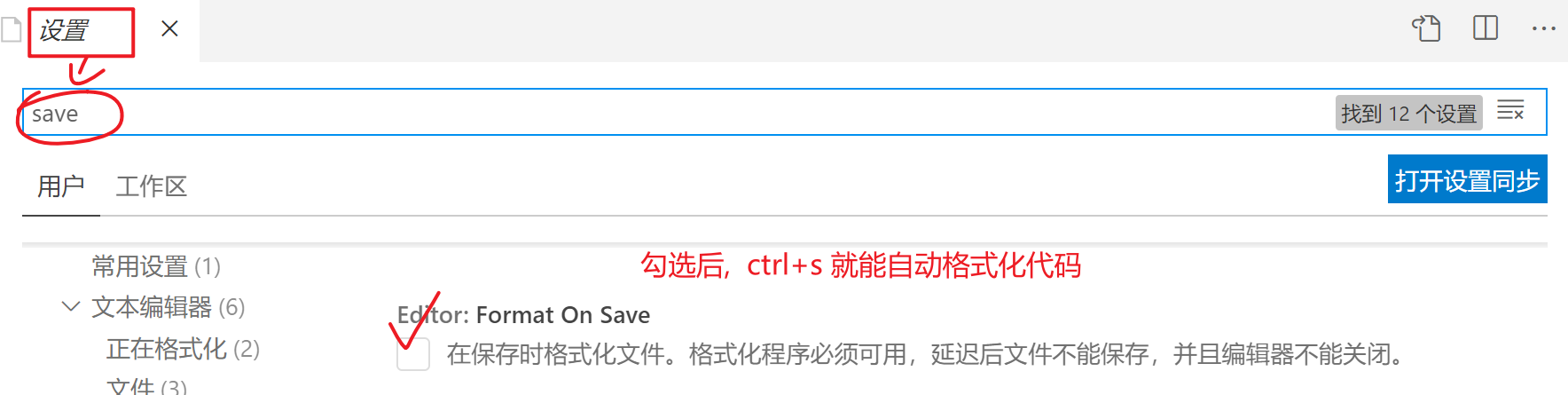


image-20201225104705378

右键文件, 选择 格式化文档的方式

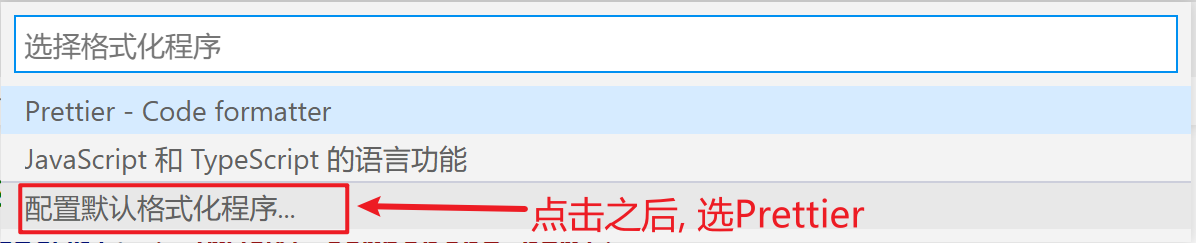




image-20201225104809480



image-20201225105104676

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:

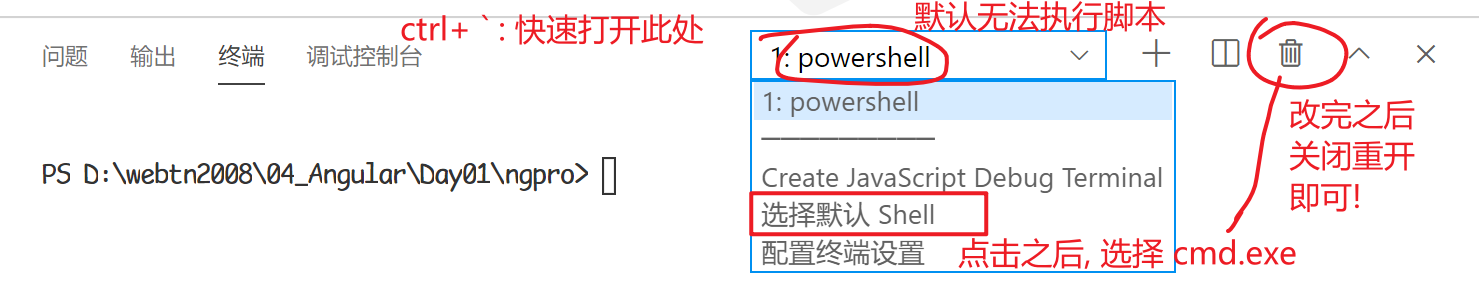
随着学习: angular的作者非常贴心! 常见问题: 使用者英文太差, 准确率低!

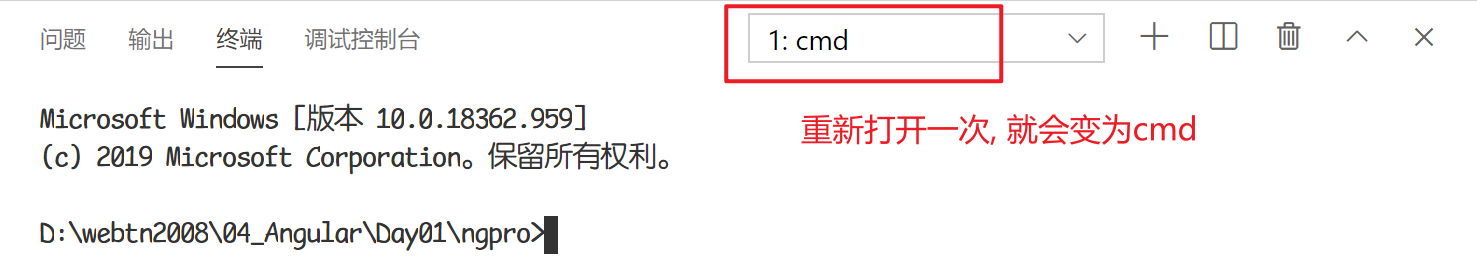
原始版: 启动之后需要手动到浏览器输入: localhost:4200 打开网站  
ng serve  
  
简化:   
ng s  
  
自动化: 自动在默认浏览器打开网站  
ng s -o  
  
-o: 是 --open 的缩写

注意事项:

windows提供了两款执行命令行的工具:

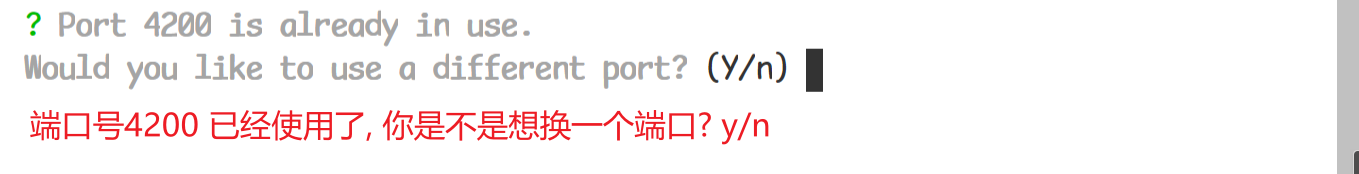
* cmd: 旧版
* powershell: 新版, 特点:默认不支持脚本的执行!!
  + 所以 所有的命令行 千万不要在 powershell 执行







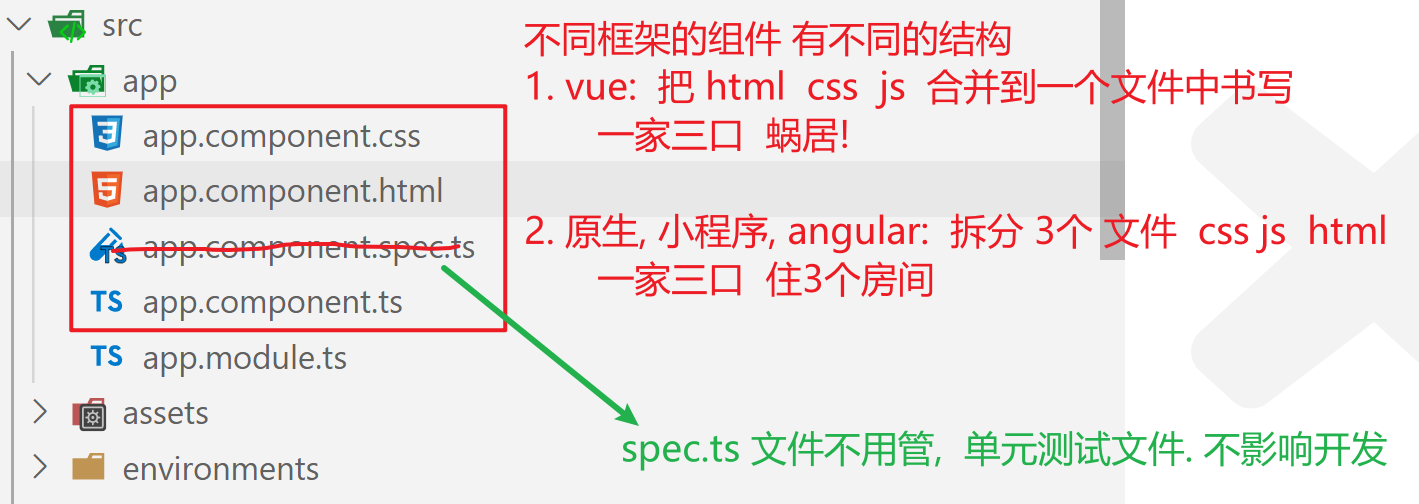
注意重复启动的问题!





## 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. 计算器在网络中的名称. 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

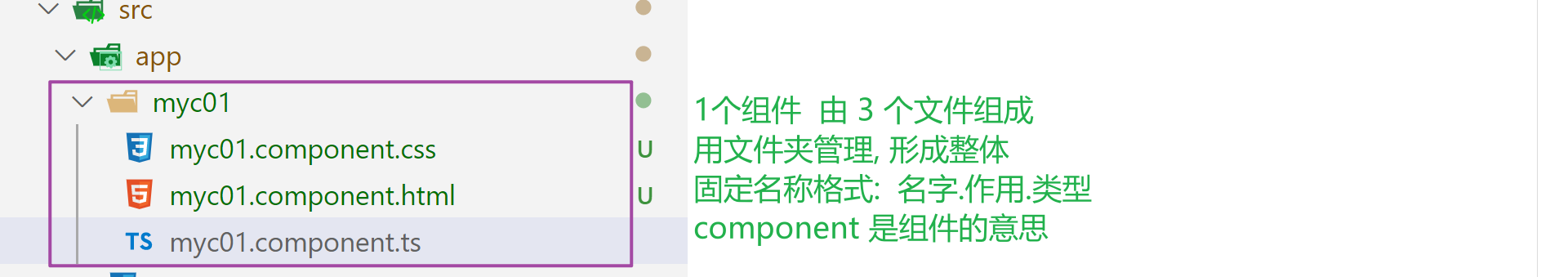
angular组件由三个文件组成:

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

// 东哥: 长脑子... JS逻辑部分  
/\*\*  
 \* TS语言: 推荐 面向对象开发方式. 所以会有大量的 类 class  
 \*   
 \* 关于主体问题:  
 \* 原生开发中: 主体是 html. 在 html 中引入 JS 和 CSS  
 \* angular中: 主体 ts. 在 ts 文件中引入 html 和 css  
 \*/  
import { Component } from '@angular/core';  
  
@Component({  
 selector: 'app-root',  
 // 引入html  
 templateUrl: './app.component.html',  
 // 引入css: 可以多个css  
 styleUrls: ['./app.component.css']  
})  
export class AppComponent {  
 title = 'ngpro';  
}

## 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
* 

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:

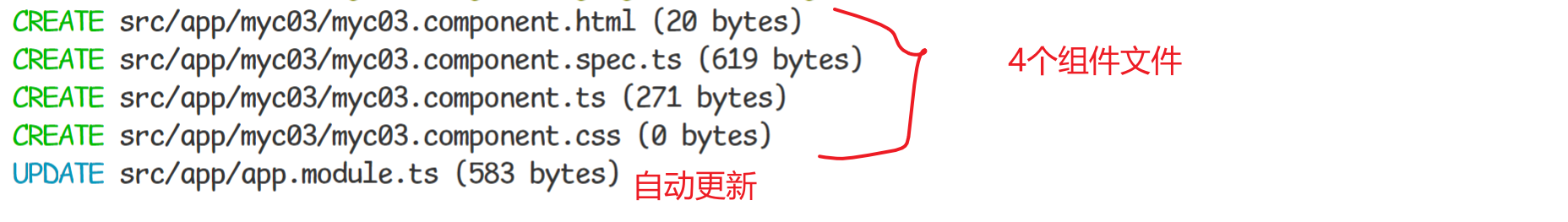
ng generate component 组件名

* 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



## 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!

import { Component, OnInit } from '@angular/core';  
  
@Component({  
 selector: 'app-myc03',  
 templateUrl: './myc03.component.html',  
 styleUrls: ['./myc03.component.css'],  
})  
export class Myc03Component implements OnInit {  
 // 面向对象中属性的写法: 属性名 = 值;  
 name = 'dongdong';  
 age = 33;  
 married = true;  
  
 names = ['东东', '亮亮', '然然'];  
  
 boss = { name: '文华', age: 40 };  
  
 constructor() {}  
  
 ngOnInit(): void {}  
}  
  
/\*\*  
 \* ts文件 就相当于是 vue 的 script 范围, 用来书写 JS 代码  
 \*/  
  
// 关注 面向对象 和 对象 在写法上的不同  
let obj = {  
 name: '东东',  
 age: 33,  
};  
  
class Demo {  
 name = '东东';  
 age = 33;  
}

<p>myc03 works!</p>  
  
<!-- 属性的绑定 -->  
<ul>  
 <li>{{ name }}</li>  
 <li>{{ age }}</li>  
 <li>{{ married }}</li>  
 <li>{{ names }}</li>  
 <li>{{ boss }}</li>  
 <!-- 数组下标取值 -->  
 <li>{{ names[0] }}</li>  
 <!-- 对象 属性取值 -->  
 <li>{{ boss.name }}</li>  
</ul>  
  
<!-- 数学运算符 -->  
<ul>  
 <li>{{ 3 + 6 }}</li>  
 <li>{{ 3 - 6 }}</li>  
 <li>{{ 3 \* 6 }}</li>  
 <li>{{ 3 / 6 }}</li>  
 <li>{{ 3 % 6 }}</li>  
</ul>  
  
<!-- 比较运算符 -->  
<ul>  
 <li>{{ age > 20 }}</li>  
 <li>{{ age >= 20 }}</li>  
 <li>{{ age < 20 }}</li>  
 <li>{{ age <= 20 }}</li>  
 <li>{{ age == 20 }}</li>  
 <li>{{ age != 20 }}</li>  
</ul>  
  
<!-- 逻辑运算符 -->  
<!--   
 逻辑或 || 或者 松散 有真则真 全假为假  
 逻辑与 && 并且 严格 全真则真 有假为假  
 逻辑非 ! 不是 非真为假 非假为真  
 -->  
<ul>  
 <li>{{ true || false }}</li>  
 <li>{{ false || false }}</li>  
 <li>{{ true && true }}</li>  
 <li>{{ true && false }}</li>  
 <li>{{ !true }}</li>  
 <li>{{ !false }}</li>  
</ul>  
  
<!-- 三目运算符: 条件 ? 真值 : 假值 -->  
<ul>  
 <li>{{ married ? "已婚" : "未婚" }}</li>  
</ul>  
  
<!-- 对象方法使用 -->  
<ul>  
 <li>{{ name.toUpperCase() }}</li>  
</ul>

## 指令

### 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

<p>myc04 works!</p>
  
   
<!--
  
 在 vue 中:
  
 :src="imgUrl"
  
 -->
  
<!--
  
 在 ng 中:
  
 [src]=""
  
 -->
  
<img [src]="imgUrl" alt="" />
  
   
<!-- 另一种写法 -->
  
<img src="{{ imgUrl }}" alt="" />

### 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

<p>myc05 works!</p>
  
   
<!-- 事件 -->
  
   
<!-- 在 vue 中: @事件名="方法名" -->
  
<!-- 在 ng 中: (事件名)="方法名" -->
  
   
<button (click)="show()">点我</button>
  
   
<!--
  
 与 vue 对比的小差异:
  
 在 vue 中, 方法如果没有参数, 可以省略()
  
 但 在ng 中, 不允许省略()
  
 -->

### 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



<p>myc06 works!</p>
  
   
<!-- 双向数据绑定 -->
  
   
<!-- 在 vue 中, 使用 v-model -->
  
<!--
  
 在 ng 中, 快捷提示 ng-model
  
 实际写法: [(ngModel)]
  
 -->
  
<!-- 实时监听输入框的变化, 事件 ngModelChange -->
  
<input type="text" [(ngModel)]="word" (ngModelChange)="\_doChange()" />
  
<p>{{ word }}</p>
  
<!--
  
 注意!!!
  
 angular默认不加载 Form 模块, 不支持双向数据绑定
  
 必须手动到 app.module.ts 中手动加载模块, 并重启
  
 -->
  
   
<!--
  
 双向数据绑定:
  
 方向1: 数据 绑定到 UI
  
 方向2: UI变化时, 会同步更新到数据 -- Form表单
  
 -->

import { Component, OnInit } from '@angular/core';
  
   
@Component({
  
 selector: 'app-myc06',
  
 templateUrl: './myc06.component.html',
  
 styleUrls: ['./myc06.component.css'],
  
})
  
export class Myc06Component implements OnInit {
  
 constructor() {}
  
   
 word = '123546';
  
   
 \_doChange() {
  
 console.log(this.word);
  
 }
  
   
 ngOnInit(): void {}
  
}

### html展示

在 vue 中: v-html

生成组件: ng g c myc07

<p>myc07 works!</p>
  
   
<!--
  
 在 vue 中: v-html="data"
  
 -->
  
   
<!-- {{}} 相当于 v-text 原生的 innerText -->
  
<p>{{ data }}</p>
  
   
<!-- 在 ng 中: [innerHtml]="" -->
  
<p [innerHTML]="data"></p>

import { Component, OnInit } from '@angular/core';
  
   
@Component({
  
 selector: 'app-myc07',
  
 templateUrl: './myc07.component.html',
  
 styleUrls: ['./myc07.component.css'],
  
})
  
export class Myc07Component implements OnInit {
  
 data = '<h1>Hello World!</h1>';
  
   
 constructor() {}
  
   
 ngOnInit(): void {}
  
}

### 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

import { Component, OnInit } from '@angular/core';
  
   
@Component({
  
 selector: 'app-myc08',
  
 templateUrl: './myc08.component.html',
  
 styleUrls: ['./myc08.component.css'],
  
})
  
export class Myc08Component implements OnInit {
  
 // 变量: 运行期间可以变化的量
  
 size = 17;
  
   
 color = 'black';
  
   
 constructor() {}
  
   
 ngOnInit(): void {}
  
}

<p>myc08 works!</p>
  
   
<!-- 静态样式: 运行期间不会变化 -->
  
<p style="font-size: 20px; color: green">Hello World!</p>
  
   
<!-- 动态样式: 运行期间可以变化 -->
  
<!-- 快捷: ng-style -->
  
<!-- 对象类型 {属性名:值} 属性名的命名规范: 只允许 数字 字母 \_
  
font-size: 不能做属性名, 因为 - 是非法字符.
  
解决方案: fontSize 或 'font-size'
  
-->
  
<button (click)="size = size + 1">变大</button>
  
<br />
  
<button (click)="color = 'green'">绿green</button>
  
<button (click)="color = 'red'">红red</button>
  
<button (click)="color = 'blue'">蓝blue</button>
  
<p [ngStyle]="{ fontSize: size + 'px', color: color }">Hello World!</p>
  
<!--
  
 vue中相同:
  
 (click)="js代码"
  
 本质就是点击后 执行 "" 里的代码
  
 -->
  
   
<hr />
  
   
<!-- 动态的 class -->
  
<!-- 需求: size 20~30 success 30~ danger -->
  
<!-- 快捷 ng-class -->
  
<span
  
 class="btn"
  
 (click)="size = size + 1"
  
 [ngClass]="{ success: size >= 20 && size < 30, danger: size >= 30 }"
  
 >{{ size }}</span
  
>

## 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

