

Angular01

讲师: 纪盈鑫

昵称: 小新老师

擅长: 服务器-java, php. 前端: iOS android vue angular react uniapp 小程序...

联系方式(微信): 18800108022

FTP 地址:

网址: `code.tarena.com.cn`

账号: `tarenacode`

密码: `code_2013`

笔记分3种格式, 内容相同

- html: 必须右键下载到本地 再打开 否则会乱码
- pdf
- md: 使用 typora软件打开

整体课程安排

- 以前的web开发: 开发网页, 针对电脑端
- 现在的web开发: web+app开发 -- 开发电脑上的网页 和 手机上的应用

现在的企业需求: 复合型人才 -- 服务器, UI切图, web开发, app开发..

阶段	讲师	内容	目的
1	亮亮	node.js, express, sql, mysql,html,js	熟悉开发的整个流程, 入门
2	然然	css, bootstrap, ajax..	化妆: 页面变好看
3	东哥	js高级, vue, 正则, jquery..	核心: 逻辑操作 -- 长脑子
4	吴华	小程序, vuex, mintUI, 微信公众号...	补充vue:开发手机端, 小程序App 你制作的手机端小程序 只能在微信上运行
5	小新	angular, react, python, vue项目	开发出能够直接安装在手机上的App 最终要体验全栈开发: python制作接口, 使用vue制作网站使用python的接口

本阶段授课的特色:

- 更加接近实际开发的场景
 - 遇到不会的如何解决: 问百度 -- 面向百度编程
 - bug的解决: 实际演示如何解决bug
- 听课时: 先听 后写
 - 代码中会非常多的注释
 - 边听边写: 写的时候没听到, 听的时候没写..

Angular

目前手机端软件开发的方向

- 原生开发: 程序体验好, 但是成本高
 - iOS: object-c, swift语言
 - Android: java 语言
- 混合开发: 使用前端html的技术开发手机App, 不区分客户端
 - 优势: 开发一个程序 就可以同时运行在 ios 和Android; 所以成本低!
 - 劣势: 程序的体验 没有原生流畅; 有一些功能无法实现.
 - 适合: 中小型企业, 资金紧张 但是 对质量要求不高.

混合开发的技术:

- vue
 - mintUI: 很多 手机端样式的 组件库: 典型的 H5 应用, 偏向web风格. --属于最差的选择
- uniapp
 - 基于vue语法, 引用了 微信小程序的组件和api: 制作的App可以打包成手机端软件
 - 体验较好
- angular + ionic
 - 更加优秀的 手机端框架, 程序可以打包成手机端软件安装;
- react+ ReactNative
 - 非常优秀的 手机端框架. 打包的软件就是原生的手机端代码: 是目前最主流的混合开发
- Flutter
 - 可以把开发的软件 转化成原生的应用, 但是要使用新的 Dart 语言

Angular语法类似于 vue, 所以 在东哥阶段 vue学的好, 则本阶段非常简单

- Angular: 谷歌公司出品
- Vue: 谷歌公司的离职程序员出品

环境搭建

与vue相同, 需要安装脚手架, 然后通过脚手架生成 项目包 来进行开发

需要node的版本高于 10.9

```
node -v
```

中国镜像:

默认node带有npm工具, 用于下载第三方模块: 默认的下载地址是 外网的, 速度慢 甚至无法下载

马云家出钱: 在国内搭建了服务器, 定时从外网下载所有的资源. 国内程序员到这个国内服务器下载就可以

检查当前镜像地址:
`npm config get registry`

```
C:\Users\web>npm config get registry
https://registry.npm.taobao.org/ ← 带有taobao字样
```

如果不是taobao镜像, 则切换到淘宝镜像即可!

```
npm config set registry https://registry.npm.taobao.org/
```

```
C:\Users\web>npm config set registry https://registry.npm.taobao.org/
C:\Users\web>npm config get registry ← 再次测试
https://registry.npm.taobao.org/
```

下载Angular的脚手架

全局安装 angular 脚手架
`npm install -g @angular/cli`

-g: 代表全局安装

```
C:\Users\web>ng v
```

```
Angular CLI: 10.1.2
Node: 12.10.0
OS: win32 x64
```

版本号查看

angular的命令是 ng:
`ng v`

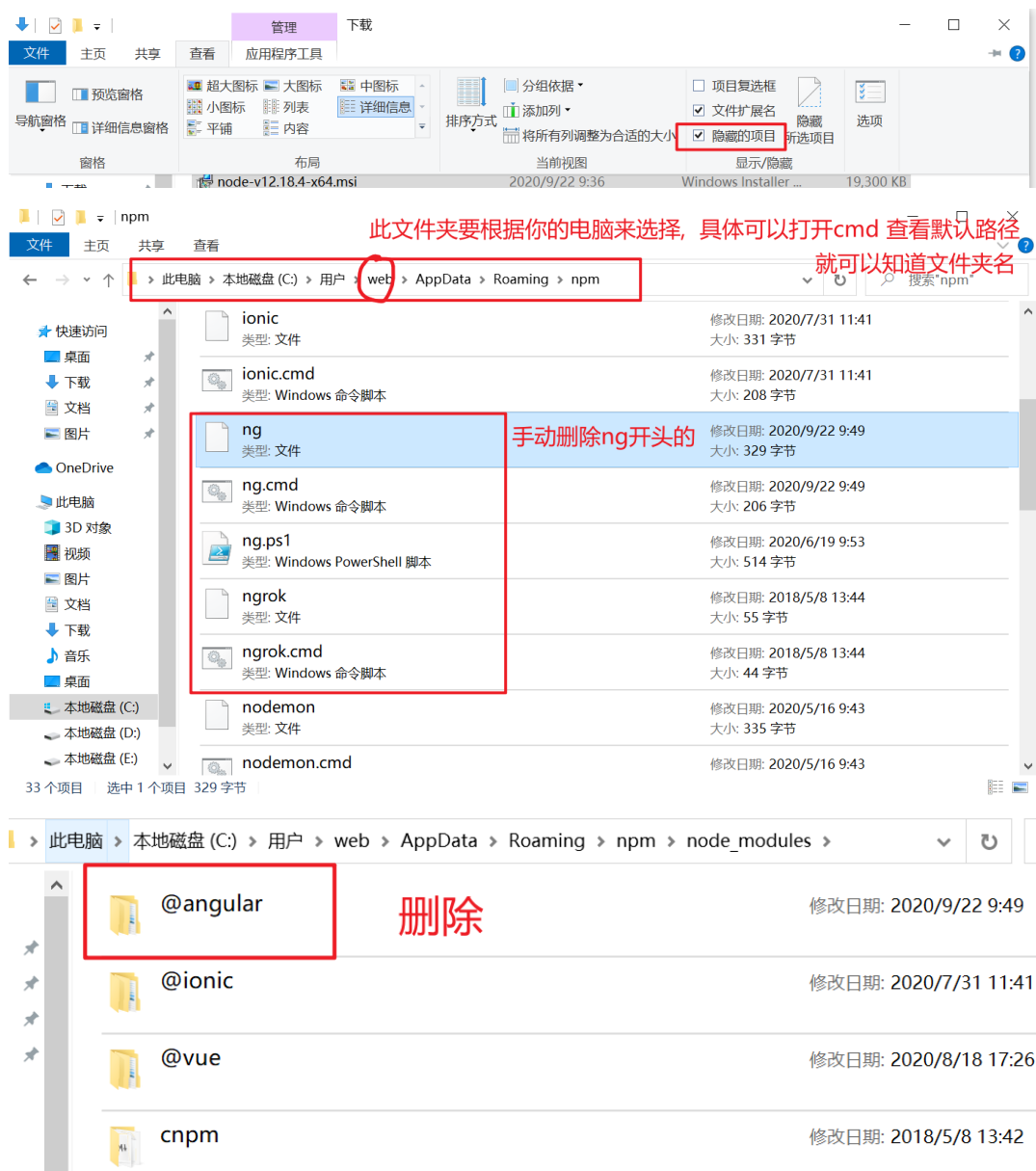
```
+ @angular/cli@10.1.2
updated 1 package in 12.272s
C:\Users\web>
```

常见报错有两种:

- 镜像不正确, 导致下载时 无法找到angular的安装包, 所以下载失败
- 你的电脑上已经安装过
 - 测试: 输入 `ng v` 看是否已经安装过, 如果不是 10 版本, 则可以重装
 - 报错原因: 你的电脑权限比较高, 无法通过cmd 来替换. 把新的替换旧的 会报错!

```
errno: -4048
file exists at xxxx
```

- 解决办法: 手动删除之前的旧版本, 然后再次安装即可!



脚手架: 就是一个记录, 记录了某个软件包 需要下载哪些扩展等信息 还会有一些专业的命令行;
利用脚手架 进而 下载一个完整的项目包

利用脚手架下载项目包

脚手架就是用来下载项目包。所以可以不安装脚手架 直接使用项目包。

生成项目包命令，在命令执行的目录下生成：

`ng new 包名`

例如：`ng new ng-app`

快速打开指定目录 到 cmd 的方式



```
D:\WEBTN2005\18_Angular\Day01>ng new ng-app
```

```
? Would you like to add Angular routing? (y/N)
```

回车使用默认项
要不要路由?

大写的是默认的 N 不用

选项2: css

安装完毕后, 有两种表现, 与 电脑是否安装了git软件有关:

- 安装了git

```
Successfully initialized git.
```

```
D:\WEBTN2005\18_Angular\Day01>
```

- 没有安装git, 并且会有提示: 说没有安装git. 不影响使用!

```
CREATE ng-app/e2e/src/app.po.ts (301 bytes)
```

```
✓ Packages installed successfully.
```

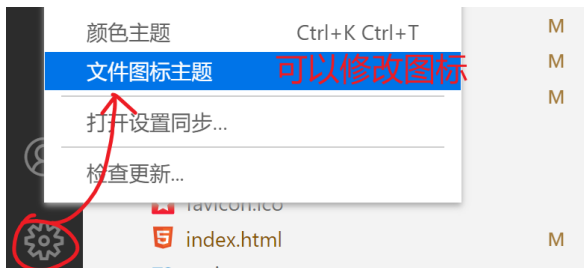
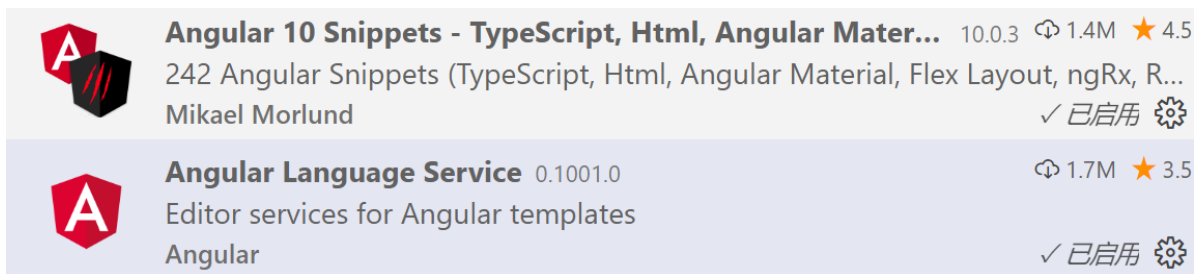
出现红色文字报错, 基本就是 网速不够, 网络不行, 无法下载成功导致的

生成的安装包:

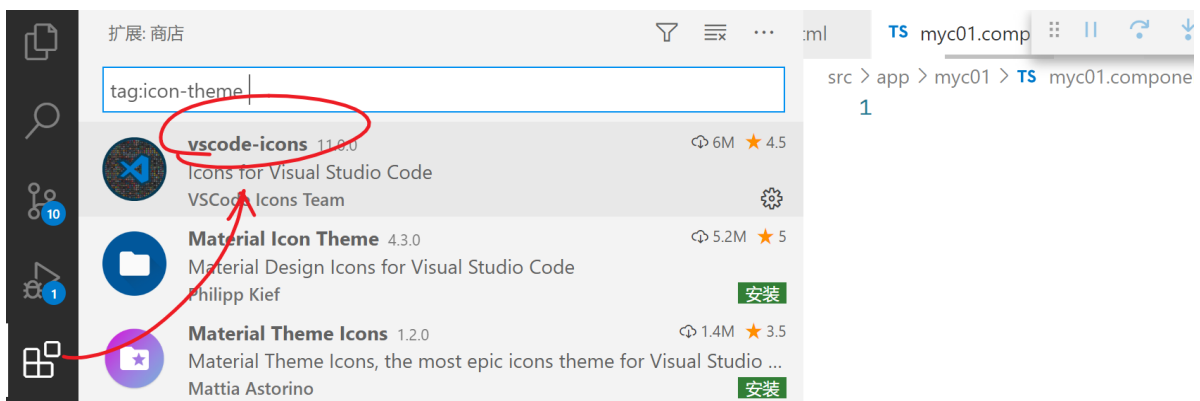
```
16_angular/ng-app.7z
```

Angular插件

让vscode在书写 angular 代码时 有代码提示, 报错提示, 快捷代码块



安装其他xx主题



项目开启

在vue中, 开启项目需要到项目的根目录下执行: `npm run serve`

angular项目启动, 到项目目录下执行

执行的命令分两个版本:

全写格式: `ng serve --open` 开启服务器 并 在默认浏览器中打开

简写格式: `ng s -o` 与上方同含义

简单打开: `ng serve` 或 `ng s` 省略`--open` 或 `-o`: 则不会自动打开浏览器

脚手架未安装成功 && 使用其他人的包: 则执行的命令行前必须带有 `npx` 前缀

例如: `npx ng serve --open`

`npx`: 代表使用项目包中的`ng`命令

不写`npx`: 代表使用全局的`ng`命令

```
ms
** Angular Live Development Server is listening on localhost:4200, open your browser on http://localhost:4200/ **
: Compiled successfully.

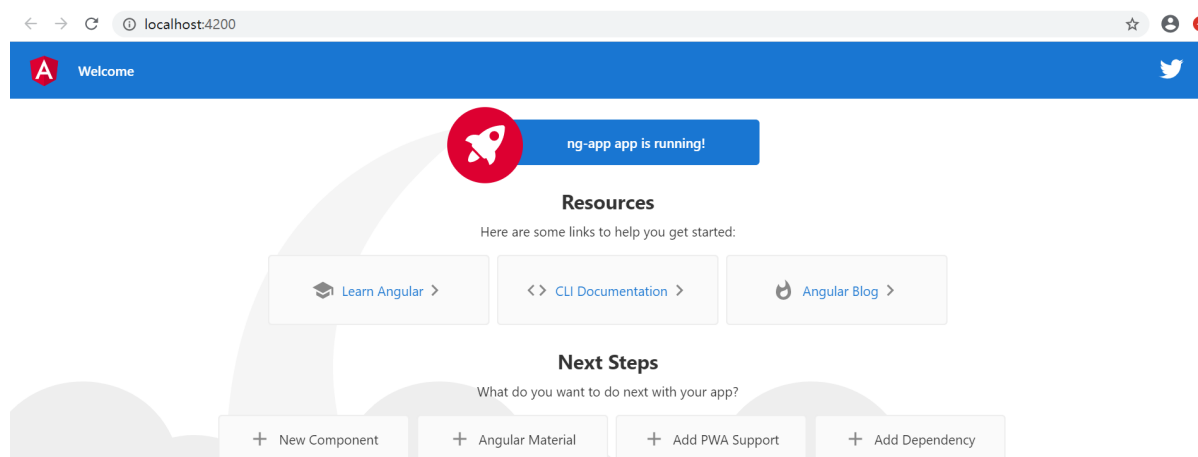
Date:                               - Hash:
5 unchanged chunks

Time:    ms
: Compiled successfully.
```

如果无法启动 或者 启动之后无法访问, 考虑是系统防火墙阻止. 关闭防火墙即可!

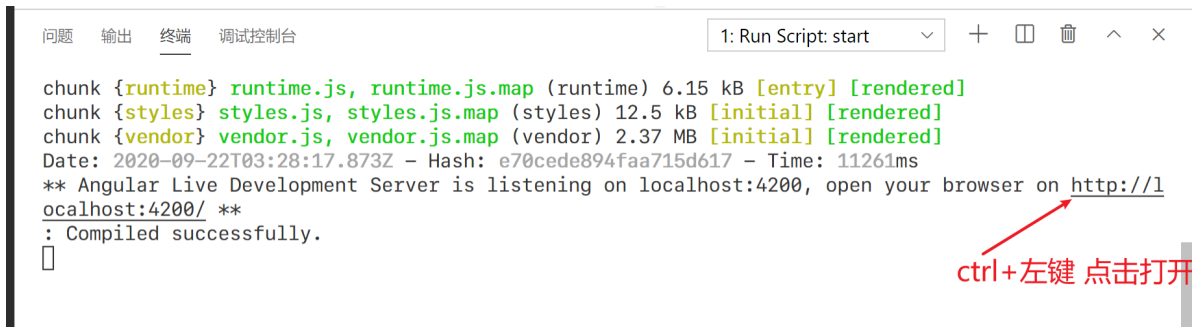
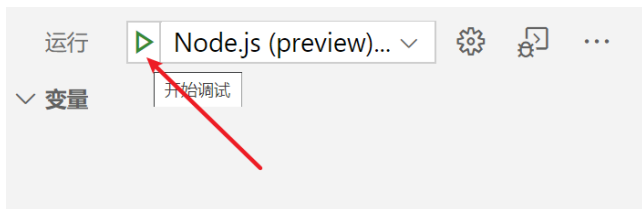
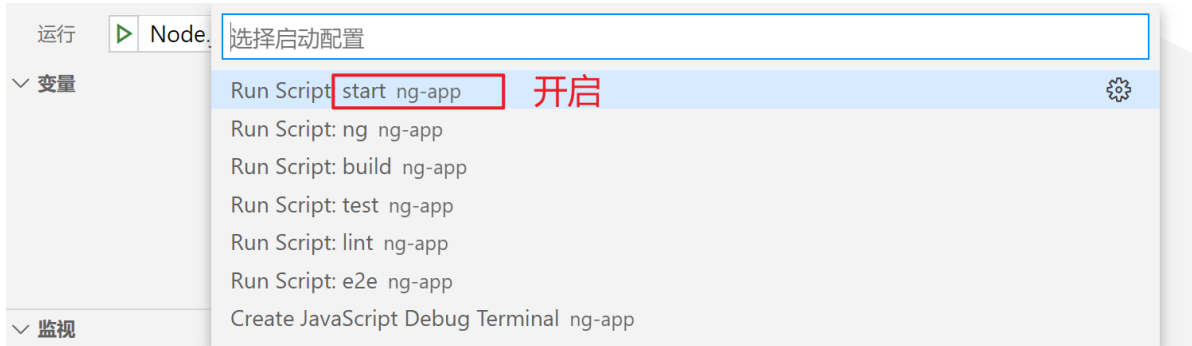
ng项目的端口号: 4200

端口号 是程序的名字, 在计算机中名字不能重复.



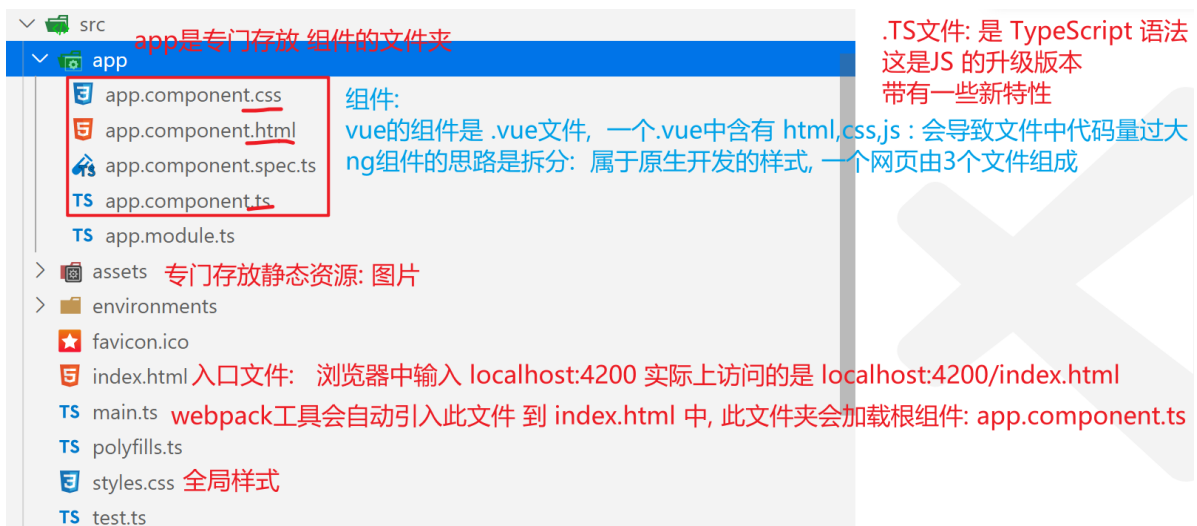
使用vscode 取代 命令行打开项目





项目文件结构

app.module.ts: 是项目的主配置文件, 在main.ts 中进行了加载



index.html



angular的文件命名格式:

名称.作用.类型

详情页: detail.page.ts 详情.页.ts

详情组件: detail.component.ts 详情.组件.ts

详情管道: detail.pipe.ts 详情.管道.ts

详情指令: detail.directive.ts 详情.指令.ts

单元测试文件: xxx.component.spec.ts 对项目的实际效果没有任何作用, 可以删除

```
import { Component } from '@angular/core';

/**
 * 原生的HTML 如果分成3个文件: 则主文件是 html
 * 在html中通过 link方式引入css, 通过script 引入js
 *
 * 在ng中, 也是3个文件, 但是主文件是: ts
 * 在ts中引入了 css 和 html
 */

@Component({
  // selector: 用于设定组件名, 查看index.html中就是 写的 <app-root></app-root>
  selector: 'app-hello',
  // templateUrl: 用来指定 对应html
  templateUrl: './app.component.html',
  // styleUrls: 引入css, 数组类型. 所以可以有多个css
  styleUrls: ['./app.component.css'],
})
export class AppComponent {
  title = 'ng-app';
}
```

手工制作组件

一共分3步

- 制作组件的3个文件: .html .css .ts
- 注册新增组件 到 配置中: 这样才能全局使用
- 使用

练习: 制作并使用 myc02 组件

- 创建文件夹: `src/app/myc02`
- 生成文件: `myc02.component.html`, `myc02.component.ts`, `myc02.component.css`
- 页面上书写: 红色文字, 内容为: 我是myc02组件
- 在 `myc02.component.ts` 文件中 生成基础代码结构, 并进行对应修改
- 在 `app.module.ts` 中注册 myc02 组件
- 在 `app.component.html` 中使用 myc02组件

练习: 制作出myc03 组件, 加载之后, 页面显示 我是myc03 蓝色文字

自动生成组件

组件的生成过程是固定的, 文件夹->3个文件->注册到配置中

作者提供了快捷命令行, 可以一步生成所有:

全写: `ng generate component` 组件名

简写: `ng g c` 组件名

提示: 未安装脚手架的同学 要带有 `npx` 前缀

例如生成 myc04

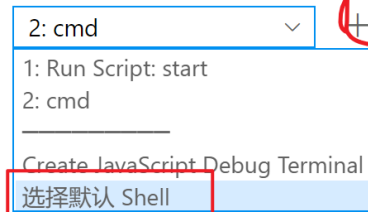


② 选择之后, 必须重新开启一个新的, 才是cmd

10.0.18363.900]
ration。保留所有权利。

Day01\ng-app>

① 打开之后, 选择 cmd.exe



D:\WEBTN2005\18_Angular\Day01\ng-app>ng g c myc04

CREATE src/app/myc04/myc04.component.html (20 bytes)

CREATE src/app/myc04/myc04.component.spec.ts (619 bytes)

CREATE src/app/myc04/myc04.component.ts (271 bytes)

CREATE src/app/myc04/myc04.component.css (0 bytes)

UPDATE src/app/app.module.ts (665 bytes) → 自动更新注册到 配置中

spec.ts
是单元测试文件
没有实际作用

数据交互

如何把 ts 文件中的变量 显示到 html 中

小程序: 把变量写在 data: {} 属性中, 使用 {{ 属性名 }}

vue: 把变量写在 data(){ return {}}, 使用 {{ 属性名 }}

angular做法与之前的框架非常相似

```
<p>myc04 works!</p>
```

```
<!-- ts的属性 可以使用 {{}} 在html中展示 -->
```

```
<ul>
```

```
<li>{{ name }}</li>
```

```
<li>{{ age }}</li>
```

```
<li>{{ married }}</li>
```

```
<li>{{ teachers }}</li>
```

```
<li>{{ boss }}</li>
```

```
<!-- 数组取值 依赖 下标 -->
```

```
<li>{{ teachers[0] }}</li>
```

```
<!-- 对象取值 依赖 .属性名 -->
```

```
<li>{{ boss.name }}</li>
```

```
</ul>
```

```
<!-- {{}}中支持一些简单的运算符-->
```

```
<ul>
```

```
<!-- 数学运算符 -->
```

```
<li>{{ 5 + 2 }}</li>
```

```
<li>{{ 5 - 2 }}</li>
```

```
<li>{{ 5 * 2 }}</li>
```

```
<li>{{ 5 / 2 }}</li>
```

```
<li>{{ 5 % 2 }}</li>
```

```
<!-- 比较运算符 -->
```

```
<li>{{ age > 18 }}</li>
```

```
<li>{{ age >= 18 }}</li>
```

```
<li>{{ age < 18 }}</li>
```

```
<li>{{ age <= 18 }}</li>
```

```

<li>{{ age == 18 }}</li>
<li>{{ age != 18 }}</li>

<!-- 三元运算符 -->
<!-- 条件?真值:假值 -->
<li>{{ married ? "已婚" : "未婚" }}</li>

<!-- 逻辑运算符 -->
<!--
    逻辑或 || 读或者, 松散逻辑 有真则真,全假为假
    逻辑与 && 读并且, 严格逻辑 全真则真,有假为假
    逻辑非 ! 读不是, 非真为假 非假为真
-->
<li>{{ true && true }}</li>
<li>{{ true && false }}</li>
<li>{{ false || true }}</li>
<li>{{ false && false }}</li>
<li>{{ !false }}</li>
<li>{{ !true }}</li>
</ul>

```

```

import { Component, OnInit } from '@angular/core';

@Component({
  selector: 'app-myc04',
  templateUrl: './myc04.component.html',
  styleUrls: ['./myc04.component.css'],
})
export class Myc04Component implements OnInit {
  // 属性, 面向对象写法: 属性名=值;
  name = 'dongdong'; //字符串类型 string
  age = 33; //数字类型 number

  // 布尔类型 boolean
  married = true; //已婚

  // 数组类型 array
  teachers = ['东东', '亮亮', '然然', '小新', '华哥'];

  // 对象类型 object
  boss = {
    name: '文华',
    age: 35,
    phone: '10086',
  };

  constructor() {}

  ngOnInit(): void {}
}

// 注意面向对象写法 和 JS对象写法的差异

let obj = {

```

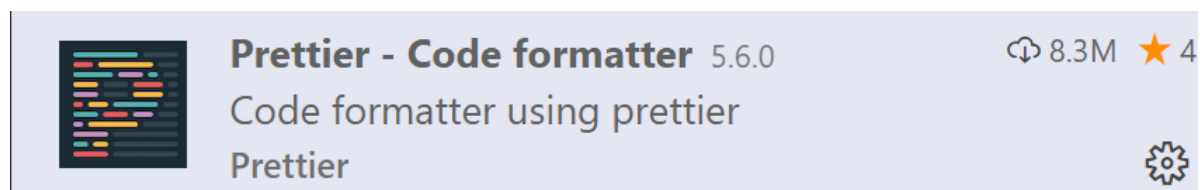
```
name: '东东',
age: 33,
};

class Obj {
  name = '东东';
  age = 33;
}
```

格式化插件

自动完善代码结构, 弥补代码的不足 并 格式化

对电脑性能要求较高, 性能太低可能会卡



右键代码区域
点击之后选择 prettier

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... 这是临时配置, 只是当前文件改动
点击之后, 选择 prettier

save

用户 工作区

常用设置 (2)

- 文本编辑器 (6)
 - 正在格式化 (2)
 - 文件 (3)
- 功能 (1)

Editor: Format On Save 但是电脑配置差, 会卡!

☒ 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。

保存代码时, 会自动格式化, 补全不足

指令

在vue中存在一系列指令, 这些指令在 angular中也有;

生成组件: myc05

```
ng g c myc05
```

```
import { Component, OnInit } from '@angular/core';

@Component({
  selector: 'app-myc05',
  templateUrl: './myc05.component.html',
  styleUrls: ['./myc05.component.css'],
})
export class Myc05Component implements OnInit {
  image =

  'https://ss1.bdstatic.com/70cFuxSh_Q1YnxGkpowK1HF6hhy/it/u=2066796227,20090082&
fm=26&gp=0.jpg';

  show() {
    alert('事件触发!');
  }

  data = '<h1>Hello world!</h1>';

  name = 'dongdong';

  constructor() {}

  ngOnInit(): void {}
}
```

```
<p>myc05 works!</p>

<!--
  属性的绑定指令
  vue: v-bind:属性名='值'   简写 :属性名=""  例如 :src="js代码"
  ng: [属性名]="""
  -->
<img [src]="image" alt="" />

<!--
  事件:
  vue: v-on:事件名="方法名()" @事件名=""  例如 @click=""
  ng: (事件名)="方法名()"  例如 (click)="""
  -->

<!--
  强调:
  vue中 如果方法没有参数, 则可以不写()结尾
  ng中: 方法必须带有()结尾
```

```

-->
<button (click)="show()">点击</button>

<!--
  html 文本解析
  vue中: v-html
  ng中: [innerHTML]
-->
<div>{{ data }}</div>
<div [innerHTML]="data"></div>

<!--
  双向数据绑定
  vue中: v-model=""
  ng中: [(ngModel)]    []代表属性->UI, ()事件触发会影响数据
-->
<!-- 单向绑定: 属性 传递到 UI -->
<!-- 但是用户更改UI时, 并不会影响到数据 -->
<input type="text" [value]="name" />
<p>{{ name }}</p>

<!-- 双向写法, 快捷 ng-model -->
<!--
  双向绑定功能 默认不是必备模块, 所以没有加载;
  若要使用, 必须手动加载:  app.module.ts 主配置文件中
-->
<input type="text" [(ngModel)]="name" />

<!--
  UI: user interface  用户界面
-->

```

```

// 引入 form 模块, 才能进行双向绑定操作
import { FormsModule } from '@angular/forms';

// 修改配置文件之后, 最好重启服务器

@NgModule({
  // declarations: 此属性用于 注册组件
  declarations: [
    AppComponent,
    Myc01Component,
    Myc02Component,
    Myc03Component,
    Myc04Component,
    Myc05Component,
  ],
  // imports: 此属性用于 注册模块
  imports: [BrowserModule, FormsModule],

```



练习

计数器练习, 生成组件 myc06 完成此练习

- 中间是输入框, 其中有数字, 默认值是5 -- 双向绑定
- 点击 加号, 可以让数字 +1 -- 事件
- 点击减号, 可以让数字 -1 -- 事件
- 也可以直接修改输入框的值 -- 双向绑定
- 当数字达到20时, 则 + 不可点击, 数字到1时 减号不可点击
 - 依赖 disabled 属性, 值为true 则不可用, 值为false 则可用



```
<p>myc06 works!</p>

<div>
  <!--
    (click)=""" 代表点击时 执行"" 中的代码
  -->
  <button (click)="count = count - 1" [disabled]="count <= 1"></button>
  <input type="text" [(ngModel)]="count" />
  <button (click)="doAdd()" [disabled]="count >= 20">+</button>
</div>

<button disabled>不可用</button>
```

```
import { Component, OnInit } from '@angular/core';

@Component({
  selector: 'app-myc06',
  templateUrl: './myc06.component.html',
  styleUrls: ['./myc06.component.css'],
})
export class Myc06Component implements OnInit {
  count = 5;

  doAdd() {
    this.count++;

    // 面向对象中的this: 本质就是一个标识
    // this.xxx 就代表 xxx 是成员属性
    // xxx 就代表 xxx 是局部变量
  }

  show() {
    alert(123);
  }

  constructor() {}

  ngOnInit(): void {}
}
```

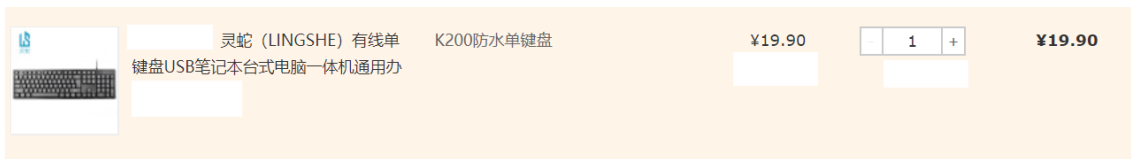


```
}
```

作业

- 购物车 Cell
 - 布局可以使用 flex 弹性盒子进行: 手机端部分通常都是弹性盒子布局
 - 点击 加 和 减 按钮, 可以修改数量, 数量修改时, 总价格会随着自动变化
 - 数量为1之后, 减号按钮不可点击
 - 数据存放在对象类型中

```
goods = {  
  pic:  
  'https://img10.360buyimg.com/cms/s80x80_jfs/t1/45249/15/4307/121563/5d1f1033E53e2293a/3a3c6921fbfe170a.jpg',  
  title : '灵蛇 (LINGSHE) 有线单键盘USB笔记本台式电脑一体机通用办',  
  desc: 'K200防水单键盘',  
  price : 19.9,  
  count: 1  
}
```



- 轮播图
 - 准备一个图片数组, 其中存放4个图片地址(网图地址即可)
 - 需要一个属性, curP 代表当前页数, 默认值为0
 - 点击两侧按钮时, 可以让中间的图片发生变化
 - 本质上: 图片在数组里, 通过curP获取对应图片地址 填写到 img 标签中, 点击按钮变化 curP的值即可
 - 实现循环滚动: 点击到最后一张 再点则回到第一张; 点击到第一张 再前一页到最后一张



