

Angular01

自我介绍:

- 纪盈鑫
- 昵称: 小新老师
- 联系方式(微信): 18800108022
- 最近有点 气管炎..

课程总体安排回顾

阶段	讲师	课程内容	目的	你的目标
1	亮亮	HTML, JS, NodeJS, SQL	基础	
2	然然	CSS, Bootstrap, ajax..	web开发, 偏重页面的显示 -- 化妆	
3	东哥	JS高级, vue框架, DOM...	JS部分 -- 内在	找到工作 (2k)
4	华哥	vue扩展, 微信小程序, 公众号..	扩展 -- 就业中有用	高薪(4k)
5	小新	手机端开发(Angular, react) python vue项目	让你拿高薪, 所以内容比较多且 高端 东哥的JS高级部分必须很扎实!	高薪(8k)

详细描述本阶段课程

第五阶段共分4个部分:

- Angular (7天)
 - WEB 3大框架之一, Google公司开发维护. -- 与vue同级别
 - 扩展 ionic 用于开发手机端App
- React(7天)
 - WEB 3大框架之一. Facebook公司开发维护. --与vue同级别
 - ReactNative 扩展用于开发手机端的App
- Python (5天)
 - 目前编程语言排行榜前三
 - Python与JS非常相似: 学习过程犹如在复习JS
 - 使用python制作 接口服务器, 来搭配vue项目使用 --- 体验**全栈工程师**的工作
- vue (3天)
 - 前端三大框架之一, 并且是国内使用最多的框架

- 临近毕业前, 完成一个网站项目. 在实战中复习vue

本阶段教学特点

因为本阶段结束后, 大家就要面临就业 和 实际工作了,
所以老师的教学偏重于实际工作中的方式.

- 解决BUG
 - 学会阅读报错提示, 定位BUG位置 和 原因.
 - 知错能改, 善莫大焉
 - 面向百度编程: 程序员必须掌握的技能
 - 面向文档编程: 学会看官方文档 -- 使用说明

Angular

前端三大框架

框架名	维护者	相似度	手机端	就业
vue	开源框架, 社区维护, 阿里巴巴	vue 同 angular	mintUI, vantUI, uni-app	☆☆☆☆☆
angular	Google	angular 同 vue	ionic	☆☆☆
react	Facebook	js的原生 DOM	react-native	☆☆☆☆

官方网站: <https://angular.cn/>

环境搭建: 与vue相同, 开发时需要使用脚手架来创建项目包!



- 要求node版本 `>= 10.13.0` , 最好是最新的版本

检查node版本命令
`node -v`

```
C:\Users\web>node -v
v12.10.0
```

- 如果版本比要求的低, 则到 node 官方下载最新版本 直接点击安装, 自动覆盖低版本的node

<https://nodejs.org/zh-cn/>

目前最新: 12.18.3

- 下载脚手架 之前, 要确认 npm 是否为中国镜像

```
npm config get registry
```

```
C:\Users\web>npm config get registry
https://registry.npm.taobao.org/ ← 带有taobao字样, 中国镜像!
```

- 设置中国镜像的方式

执行下方命令行即可, 然后再次运行上方命令, 查看是否成功!

```
npx nrm use taobao
```

或

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

- 全局安装angular脚手架

查看自己电脑是否已经安装过ng. 如果没有装过会提示 ng 非内部命令
ng v

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

```
Angular CLI: 9.1.9
```

```
Node: 12.10.0
```

```
OS: win32 x64
```

安装命令

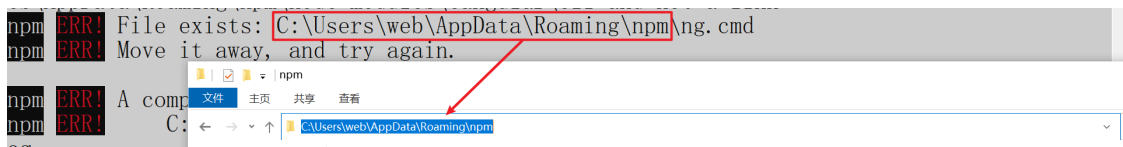
```
npm i -g @angular/cli
```

报错: 如果你安装过, 则此命令行会做两件事

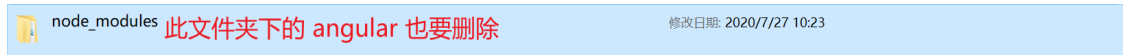
- 先删除旧的脚手架: **由于权限问题, 有的电脑cmd命令行无法自动删除 则会报错**
- 再安装新的脚手架

```
C:\Users\web>npm i -g @angular/cli
npm ERR! code EEXIST已存在 拒绝删除 xxx
npm ERR! path C:\Users\web\AppData\Roaming\npm\ng.cmd
npm ERR! Refusing to delete C:\Users\web\AppData\Roaming\npm\ng.cmd: is outside C:\Users\web\AppData\Roaming\npm\node_modules\@angular\cli and not a link
npm ERR! File exists: C:\Users\web\AppData\Roaming\npm\ng.cmd 此文件已存在
npm ERR! Move it away, and try again.
npm ERR! 移除它, 并且重新尝试!
npm ERR! A complete log of this run can be found in:
npm ERR! C:\Users\web\AppData\Roaming\npm-cache\_logs\2020-07-27T02_17_43_636Z-debug.1
og
```

找到已存在的文件所在目录



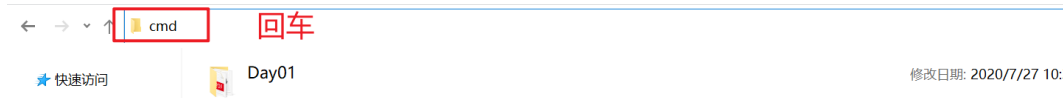
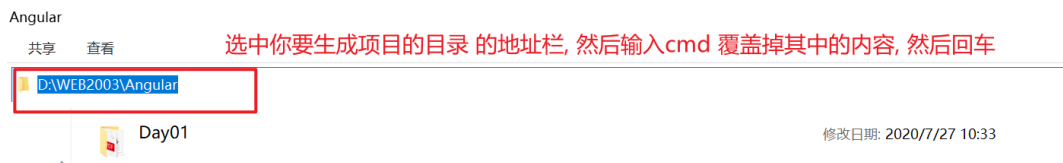
删除掉此文件夹下的 ng.cmd



最新版本: **10.0.4**

+ @angular/cli@10.0.4
added 280 packages from 206 contributors in 43.359s

- 使用脚手架生成项目包
 - 确保命令行在你希望项目生成的目录下打开

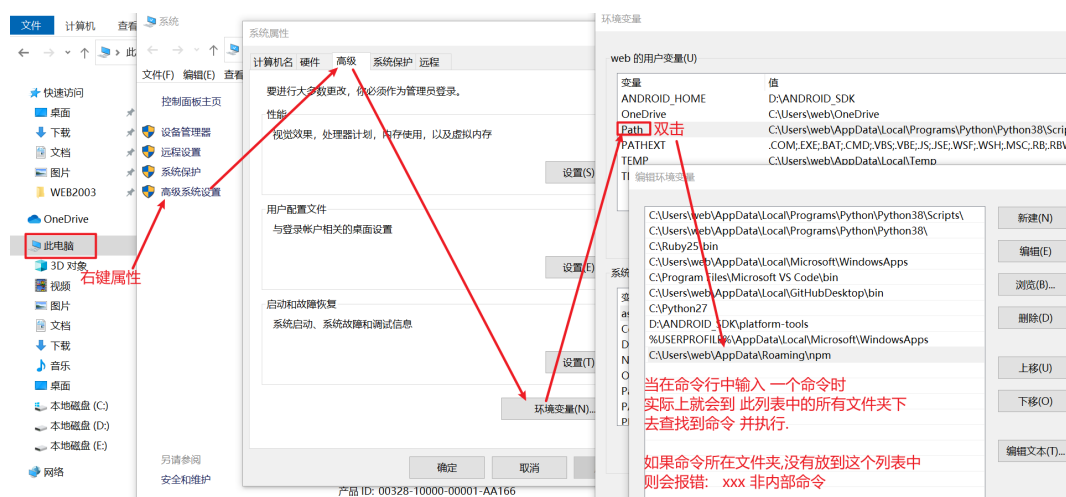


- 生成命令, 过程中的询问都 **回车确认** 使用默认项

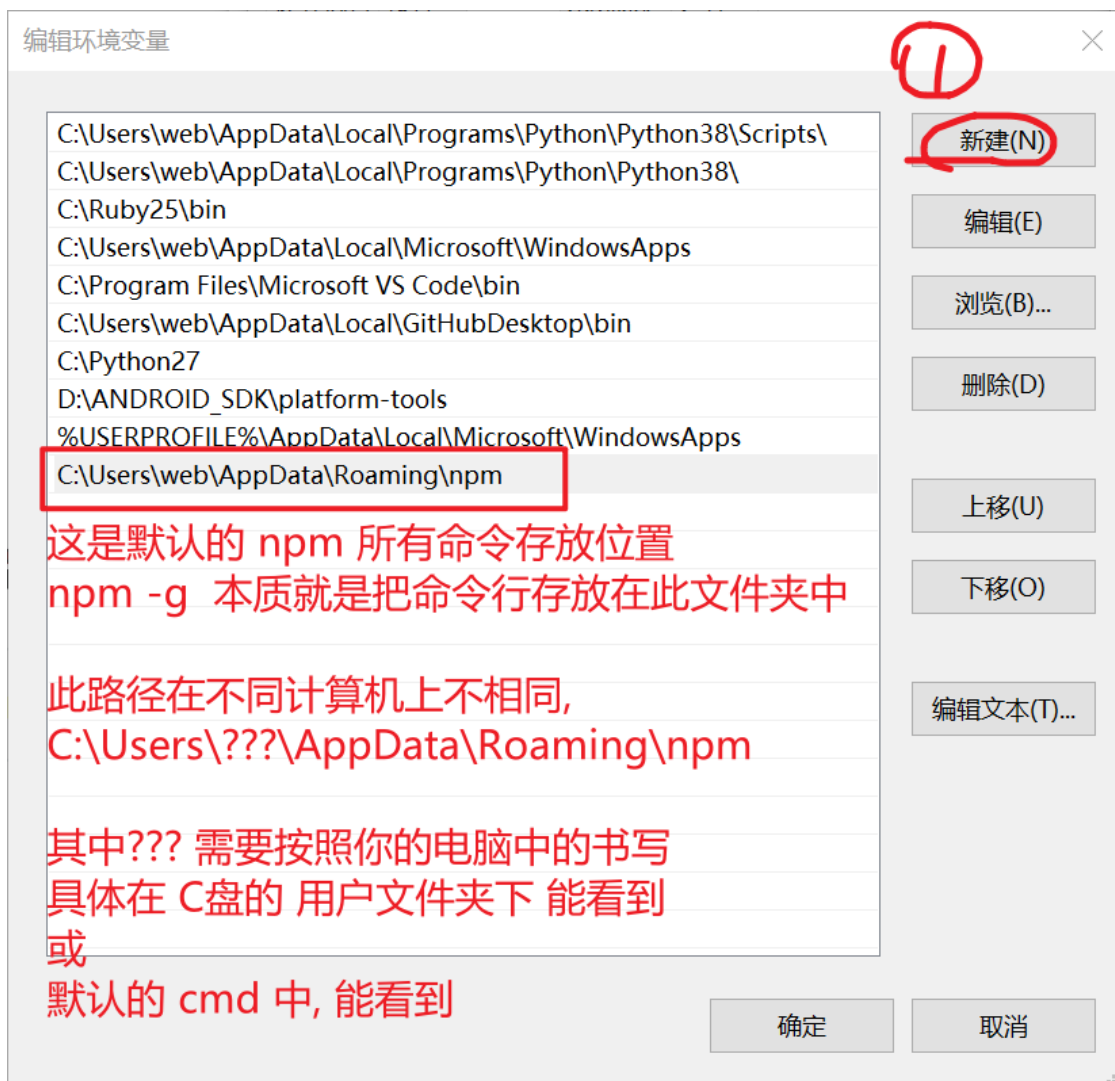
ng new 项目名
例如: ng new ngpro

报错: **ng 非内部命令**

原因: 你电脑的环境变量被你删除过!



环境变量修改之后, 必须确定保存, 然后重启你的CMD命令行工具 才会有效!

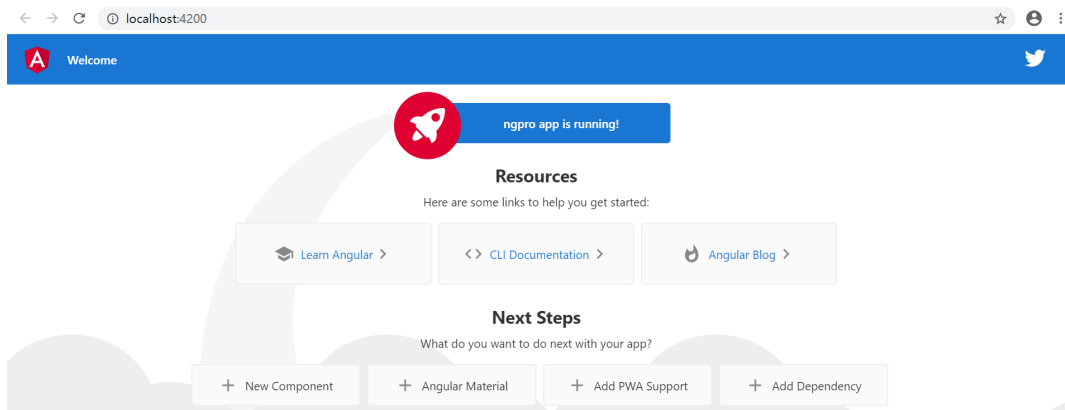


```
C:\Windows\System32\cmd.exe
CREATE ngpro/src/app/app.component.ts (209 bytes)
CREATE ngpro/src/app/app.component.css (0 bytes)
CREATE ngpro/e2e/protractor.conf.js (869 bytes)
CREATE ngpro/e2e/tsconfig.json (299 bytes)
CREATE ngpro/e2e/src/app.e2e-spec.ts (638 bytes)
CREATE ngpro/e2e/src/app.po.ts (301 bytes)
√ Packages installed successfully.
warning: LF will be replaced by CRLF in .editorconfig.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in .editorconfig.
The file will have its original line endings in your working directory
```

- 启动项目: 到生成的 ngpro 文件夹下, 运行cmd

```
ng serve --open
```

项目的地址: localhost:4200

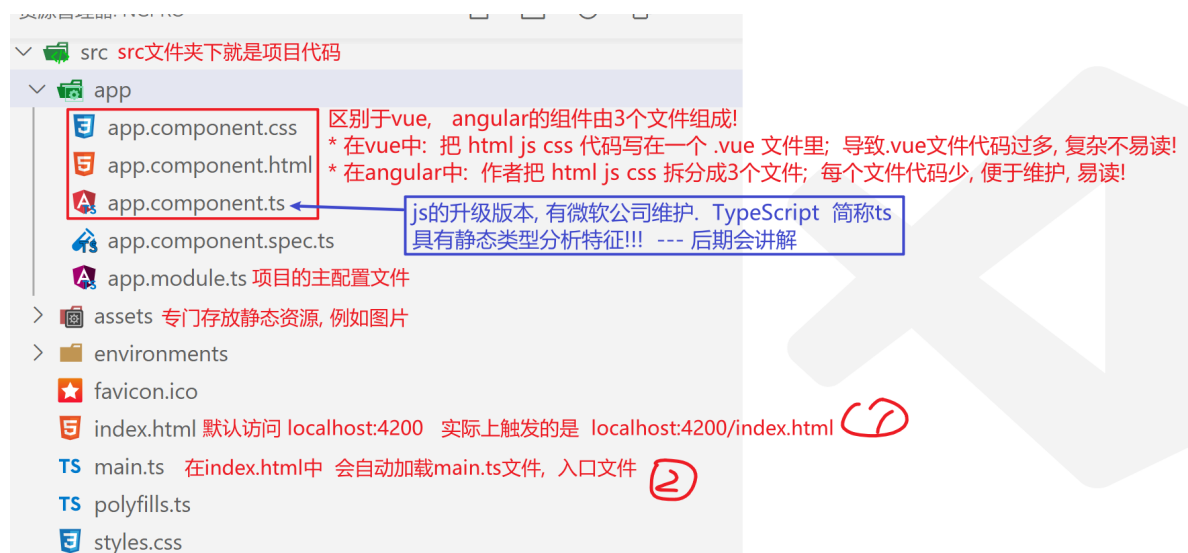


下载不成功的同学

可以直接使用其他同学生成的项目包即可!

FTP上有传递, 但是较慢~~

HelloWorld



启动流程

- 启动服务器之后, 访问 `localhost:4200` 实际上访问的是 `src/index.html`
- `index.html`: 其中有一个标签 `<app-root></app-root>`
- **webpack**: 默认集成的打包工具, 会自动把 `main.ts` 打包引入到 `index.html` 中
- **main.ts**: 使用了 `app.module.ts` 文件
- `app.module.ts`: 此文件中加载了默认的根组件: `app.component.ts`

组件被分成3个文件:

- `html`: 负责页面上有什么
- `css`: 负责样式
- `ts`: 负责逻辑操作

我们的第一个组件

组件制作分3步骤:

- 3个文件: html css 和 ts
- 注册: app.module.ts 中注册
 - 配置文件有修改, 要重启服务器!
 - 服务器不能多开, 必须关闭之前的才能开启新的!

```
ng serve --open
简写
ng s -o
```

- 使用: 到根组件即可 `app.component.html`

← → ↺ ⓘ localhost:4200

Hello World!

我的第一个组件

myc01.component.ts

```
// 之前安装了angular插件

// ng-component 会有提示, 回车选择 自动生成!
// 生成的基础模板, 需要自己改造:
// 文件名结构: name.component.xxx
// 把 name 替换到生成的模板代码中, 共4个位置
import { Component, OnInit } from '@angular/core';

@Component({
  // 按alt键 然后左键选择 就是多光标. 按ESC退出此模式
  selector: 'app-myc01', //组件名, 使用时 <app-myc01></app-myc01>
  templateUrl: './myc01.component.html',
  // 默认是 .scss后缀, 此处修改为css
  styleUrls: ['./myc01.component.css'],
})
// 晚上时间 最好是 看一下 东哥的面向对象部分
// 类名必须大驼峰, 即首字母大写
export class Myc01Component implements OnInit {
  constructor() {}

  ngOnInit(): void {}
}
```

myc01.component.html

```
<h2>我的第一个组件</h2>
```

myc01.component.css

```
h2 {  
  color: orange;  
}
```

app.module.ts

```
import { BrowserModule } from '@angular/platform-browser';  
import { NgModule } from '@angular/core';  
  
import { AppComponent } from './app.component';  
// 1. 引入组件  
import { Myc01Component } from './myc01/myc01.component';  
  
// 当配置文件发生修改的时候，往往需要重启服务器才能正常使用  
// 在cmd中；按两次 ctrl+C，然后输入启动命令：  
// ng serve --open 作者怕大家写错，有简写格式：ng s -o  
// 服务器不能多开，同一个端口只能用一次。重复开启会出现端口占用报错!!!  
@NgModule({  
  // 2. 注册组件  
  declarations: [AppComponent, Myc01Component],  
  imports: [BrowserModule],  
  providers: [],  
  bootstrap: [AppComponent],  
})  
export class AppModule {}
```

app.component.html

```
<h1>Hello world!</h1>  
  
<!-- 组件必须要注册之后，才能被全局使用 -->  
<!-- 在 app.module.ts 中 -->  
<app-myc01></app-myc01>
```

练习

制作第二个组件: myc02

- 项目启动, 项目的根目录下执行


```
ng serve --open
或
ng s -o
```

正常会自动在默认浏览器弹出项目页面
也可以手动在你喜欢的浏览器中，访问 `localhost:4200`

- 创建新的组件myc02

```
src
--app
----myc02
-----myc02.component.html
-----myc02.component.css
-----myc02.component.ts
```

- 书写组件的内容
 - html: `<h2>我是组件2</h2>`
 - css: 把h2标签颜色变为 `blue`
 - ts: 使用 `ng-component` 快捷提示生成代码, 并修改 4个 name 为 myc02, 注意类名要大驼峰 `Myc02`
- 注册组件: `app.module.ts`, 注册之后可能要**重启服务器**才可以
- 使用组件: `app.component.html`

练习: 完成组件3 myc03

- 创建组件文件
- 书写组件内容
- 注册组件
- 使用组件

生成组件命令

组件制作的过程是固定的, 官方为使用者提供了快捷创建命令.

```
ng generate component 组件名
```

```
- generate 生成
- component 组件
```

简化:

```
ng g c 组件名
```

必须在项目根目录下执行, 生成 myc04

```
ng g c myc04
```

```
D:\WEB2003\Angular\Day01\ngpro>ng g c myc04
CREATE src/app/myc04/myc04.component.html (20 bytes)
CREATE src/app/myc04/myc04.component.spec.ts (621 bytes) → 单元测试文件, 没有用删除都可以!
CREATE src/app/myc04/myc04.component.ts (271 bytes)
CREATE src/app/myc04/myc04.component.css (0 bytes)
UPDATE src/app/app.module.ts (952 bytes) ← 自动注册组件
```

数据绑定

vue中: 把 js 中的变量显示到 template 中

ng中: 把 ts 中的变量 显示到 html 中

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

<!-- 基础数据类型 -->
<ul>
  <li>{{ name }}</li>
  <li>{{ age }}</li>
  <!-- ng中 bool类型 转为文本的 true 和 false显示 -->
  <li>{{ married }}</li>
</ul>

<!-- 数组类型 -->
<ul>
  <li>{{ names[1] }}</li>
</ul>

<!-- 对象类型 -->
<ul>
  <li>{{ leader.name }}</li>
</ul>

<!-- {} 能够书写的运算符 -->

<!-- 数学运算符 -->
<ul>
  <li>{{ 5 + 2 }}</li>
  <li>{{ 5 - 2 }}</li>
  <li>{{ 5 * 2 }}</li>
  <li>{{ 5 / 2 }}</li>
  <li>{{ 5 % 2 }}</li>
</ul>

<!-- 比较运算符 -->
<ul>
  <li>{{ age > 18 }}</li>
  <li>{{ age >= 18 }}</li>
  <li>{{ age < 18 }}</li>
  <li>{{ age <= 18 }}</li>
  <li>{{ age == 18 }}</li>
  <li>{{ age != 18 }}</li>
</ul>

<!-- 逻辑运算符 -->
```

```

<ul>
  <!-- 逻辑与： 并且关系 && 全真为真 有假为假 -->
  <li>{{ true && true }}</li>
  <li>{{ true && false }}</li>

  <!-- 逻辑或： 或者关系 || 有真为真 全假则假 -->
  <li>{{ false || false }}</li>
  <li>{{ true || false }}</li>

  <!-- 逻辑非： 否,不是 ！ 非真为假 非假为真 -->
  <li>{{ !true }}</li>
  <li>{{ !false }}</li>
</ul>

<!-- 三元/三目 运算符 -->
<!-- 条件 ? 真值 : 假值 -->
<ul>
  <li>{{ married ? "已婚" : "未婚" }}</li>
</ul>

<!-- 对象类型的方法 -->
<ul>
  <li>{{ "my name is xiaoxin".toUpperCase() }}</li>
</ul>

<!-- ng中不能 进行实例化操作 -->
<!-- <p>{{ new Date() }}</p> -->

```

```

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

@Component({
  selector: 'app-myc04',
  templateUrl: './myc04.component.html',
  styleUrls: ['./myc04.component.css'],
})
export class Myc04Component implements OnInit {
  //面向对象的写法： 属性名=值；
  name = 'dongdong';
  age = 18;
  married = true;

  names = ['亮亮', '东东', '然然'];

  leader = {
    name: '文华',
    age: 33,
  };

  constructor() {}

  ngOnInit(): void {}
}

```

事件的绑定

生成新的组件: myc05

```
ng g c myc05
```

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

<!--
  在vue中: @click    v-on:click
  在ng中: (click)   ()代替了@
-->

<!-- 方法必须带()结尾 -->
<button (click)="showMsg()">点击事件</button>
```

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

@Component({
  selector: 'app-myc05',
  templateUrl: './myc05.component.html',
  styleUrls: ['./myc05.component.css'],
})
export class Myc05Component implements OnInit {
  // 方法
  showMsg() {
    alert('hello!');
  }

  constructor() {}

  ngOnInit(): void {}
}
```

属性的绑定与使用

生成组件: myc06

```
ng g c myc06
```

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

@Component({
  selector: 'app-myc06',
  templateUrl: './myc06.component.html',
  styleUrls: ['./myc06.component.css'],
})
```

```

export class Myc06Component implements OnInit {
  title = '我是小新老师';

  // 百度图片 必应图片 找一张网图地址
  // 右键网页中的图片，复制图像地址 即可
  logo = 'https://www.baidu.com/img/flexible/logo/pc/result.png';

  html = '<h1>我是HTML代码</h1>';

  // 双向数据绑定
  // 方向1: ts -> html
  // 方向2: html -> ts ??
  word = '默认123';

  constructor() {}

  ngOnInit(): void {}
}

```

```

<p>myc06 works!</p>

<h3 title="xxx">5阶段讲师</h3>

<!--
  在vue中:
  v-bind:title=""
  :title=""

  在ng中
  [属性名]="""
-->
<h3 [title]="title">5阶段讲师</h3>

<img [src]="logo" alt="" />

<!-- 按钮属性 -->
<br />
<!-- 通过 true 和 false 可以切换属性是否可用 -->
<button [disabled]="false">可用</button>
<button [disabled]="true">不可用</button>

<!--
  html
  {{{}}相当于 DOM原生的 innerText

  显示html 在vue中 v-html=""
  在ng中: [innerHTML]="""
-->
<div>{{ html }}</div>
<div [innerHTML]="html"></div>

<hr />
<!-- 双向数据绑定 -->
<!-- vue中 v-model="" -->

```

```

<!-- ng中 ng-model可以快速提示: [(ngModel)]="" -->
<!-- ng默认不加载 form 模块, 不具备ngModel属性, 必须到 app.module.ts 配置文件中
进行 Form 模块加载才可以! -->
<input type="text" [(ngModel)]="word" />
<div>{{ word }}</div>

```

app.module.ts

```

import { BrowserModule } from '@angular/platform-browser';
import { NgModule } from '@angular/core';

// 引入Forms模块
import { FormsModule } from '@angular/forms';

import { AppComponent } from './app.component';
// 1. 引入组件
import { Myc01Component } from './myc01/myc01.component';
import { Myc02Component } from './myc02/myc02.component';
import { Myc03Component } from './myc03/myc03.component';
import { Myc04Component } from './myc04/myc04.component';
import { Myc05Component } from './myc05/myc05.component';
import { Myc06Component } from './myc06/myc06.component';

// 当配置文件发生修改的时候, 往往需要重启服务器才能正常使用
// 在cmd中; 按两次 ctrl+C, 然后输入启动命令:
// ng serve --open      作者怕大家写错, 有简写格式: ng s -o
// 服务器不能多开, 同一个端口只能用一次. 重复开启会出现端口占用报错!!!
@NgModule({
  // 2. 注册组件
  declarations: [
    AppComponent,
    Myc01Component,
    Myc02Component,
    Myc03Component,
    Myc04Component,
    Myc05Component,
    Myc06Component,
  ],
  // 注入Forms模块: 修改配置文件, 特别是注入模块 必须重启服务器!!
  // 即:cmd中 按两次 ctrl+c 然后 ng s -o
  imports: [BrowserModule, FormsModule],
  providers: [],
  bootstrap: [AppComponent],
})
export class AppModule {}

```

计数器练习

组件: myc07

按钮

要求

- * 按钮每次点击时, 数字+1
- * 当数字达到20时, 则按钮变为不可用状态

输入框

进阶:

- * 输入框的文字 与 按钮中的文字联动
- * 输入框中输入文字, 按钮上的文字也会变化

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

@Component({
  selector: 'app-myc07',
  templateUrl: './myc07.component.html',
  styleUrls: ['./myc07.component.css'],
})
export class Myc07Component implements OnInit {
  // 变量: 代码运行期间可以变化的量
  num = 1;

  doAdd() {
    this.num++;
  }

  constructor() {}

  ngOnInit(): void {}
}
```

```
<p>myc07 works!</p>

<!--
  'false' 和 false 是一样的么??
  答: 不同, 字符串 和 bool 类型
-->
<!-- <button (click)="doAdd()" disabled="false">{{ num }}</button> -->

<!-- [disabled]="false" 这个false才是bool类型 -->
<button (click)="doAdd()" [disabled]="num == 20">{{ num }}</button>
<!--
  html中不支持: 自增自减 +=
-->
<button (click)="num = num + 1" [disabled]="num == 20">{{ num }}</button>

<!--
  (click)="js代码"
  点击时 就会执行 js代码; js代码本身并不是必须是函数
-->

<hr />
<input type="text" [(ngModel)]="num" />
```


练习2

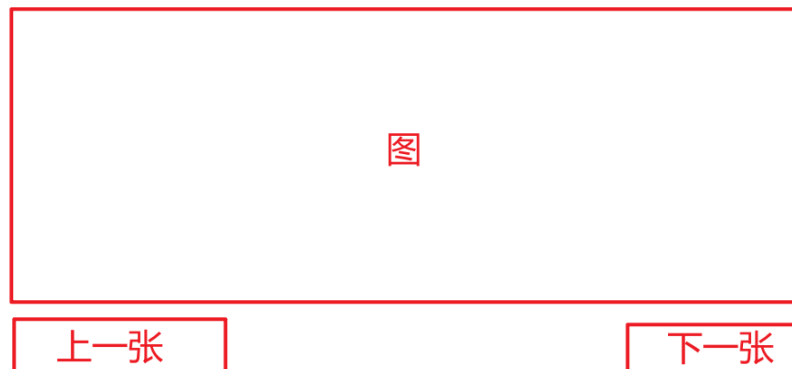
组件: myc08

商品名: iPhone
商品价格: 此输入框是可以改动的
商品数量: - + - 按钮 减少到1 不可用, 即不能减少到0
总价格: xxxxx 总价格会随着 单价和数量发生变化

练习3





组件: myc09

- 需要一个数组属性, 其中存放5张网络图片地址: 可以从百度图片或者必应图片找图
- 需要一个 `curP` 属性来记录当前页数, 例如 `curP=0` 代表当前是序号0的图片
- 前端页面通过 `curP` 属性来读取图片地址, 放到 标签的src属性中
- 点击上一页和下一页按钮, 变化`curP`的值即可
- 注意循环, 当`curP`到达极限值时, 需要变化. 例如 5张图, 下一页到第5张时, 再次点击则变第一张图

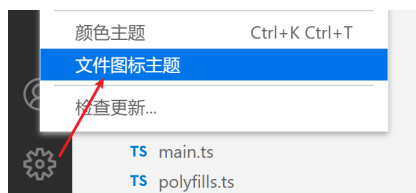


插件

安装之后, 代码中有丰富的代码提示

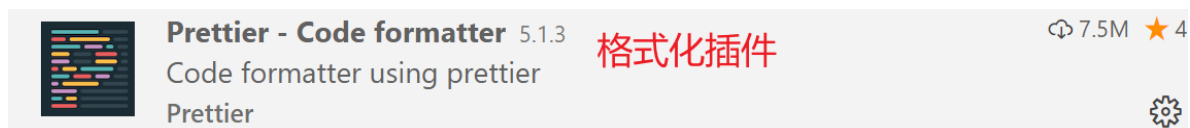
	Angular 8 Snippets - TypeScript, Html, Angular Material... 8.1.2 1.4M ★ 4.5 242 Angular Snippets (TypeScript, Html, Angular Material, Flex Layout, ngRx, R... Mikael Morlund 
	Angular Language Service 0.1000.4 1.6M ★ 3.5 Editor services for Angular templates Angular 

自定义文件图标



安装之后, 代码中右键 -> 文档格式设置方式.. 选择 prettier

使用时: 右键代码 -> 格式化文档 即可



电脑配置好的同学, 可以开启**保存时自动格式化**

