

# Angular01

---

第五阶段讲师: 纪盈鑫

昵称: 小新老师

联系方式(微信): 18800108022

| 几个时间点可能无法快速回复: 上课时间, 午睡:13~14, 骑车回家6-8

FTP地址

网址: `code.tarena.com.cn`

账号: `tarenacode`

密码: `code_2013`

笔记相关:

- 笔记中图片偏多, 所以采用 `md`, 并会导出 `pdf` 和 `html` 版本
- `html`格式: 可能会乱码, 用360浏览器打开, 在网页上右键选择 编码: 改为UTF8即可

## 本阶段注意

最好是win10, 重装可以找中心的mis老师;

| 也可以找马云 或者 百度重装系统

## 本阶段的课程安排与在整个课程体系中的位置

---

阶段	讲师	内容	目的
1	成亮	mysql, node.js, js	入门
2	然然	css, Bootstrap, html, ajax	web原生开发
3	东哥	js高级, vue, DOM, jquery	web实际开发场景 用第三方框架取代原生方式进行开发
4	吴华	微信小程序, vuex, html5, mintUI	提高薪资和就业竞争力 小程序, vue做手机端开发需要额外的框架 把开发场景从前3个阶段的面向浏览器开发 转为 面向App开发
5	小新	angular, vue, react, python	提高薪资和就业竞争力 angular+ionic: 目前主流App开发解决方案之一 react+ReactNative: 目前主流App开发解决方案之一 最后几天学习python制作服务器端, 使用vue制作网页端, 体验一次 <b>全栈开发</b> 的流程

学习难度: 只要在**东哥阶段的基础打得好**, 本阶段特别简单!

## Angular

- Vue
  - Google的离职人员开发, 后来由开源社区进行维护
- Angular
  - Google开发并维护
  - 由于 angular 和 vue 都是 google程序员开发, 所以整体思路非常相似
- React
  - Facebook开发并维护

官方: <https://angular.cn/>

安装脚手架: <https://angular.cn/guide/setup-local>

来自 linux 的习惯: 命令行前端用 \$ 代表这是个命令

查看当前计算机的node版本: 最低10.9.0

```
$ node -v
```

下载地址: <https://nodejs.org/zh-cn/>

当前稳定版: 12.18

安装方式: 一直next 到最后即可. 新版本会自动替换旧版本

检查中国镜像

npm自带的镜像是国外网站, 由于网络限制, 导致下载速度慢甚至无法下载

```
$ npm config get registry
```

中国镜像地址: <https://registry.npm.taobao.org/>

设置中国镜像

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

安装angular脚手架: 安装过程有问题都回车即可

```
$ npm install -g @angular/cli
```

-g: 代表是全局安装 global. 即在哪里执行都可以

检测angular版本号

你的电脑如果是中心的, 可能有安装过; 则再次安装会报错

```
$ ng v
```

\* 如果出现 ng 非内部或外部命令: 恭喜你, 正常执行上方的安装命令即可!

\* 最新版本是 10.0.7, 版本低的要升级

**报错:**

由于npm安装的内容在c盘上, 所以命令行进行升级操作时, 需要替换文件, 有时会因为权限不够, 导致不允许替换, 而报错:

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

解决方案: 手动删除之前的ng文件即可

删除位置: C:/Users/xxx/AppData/Roaming/npm

- \* Users 在C盘中的文件夹名可能叫 用户
- \* xxx: 不同电脑不同, 可以通过 win+R 打开cmd, 看到默认的用户名
- \* AppData: 文件夹可能不可见, 需要打开 查看隐藏文件 才可以 --- 具体百度

删除:

- \* ng 相关的文件
- \* 到 node\_modules 文件夹下, 删除 @angular 文件夹

重新执行安装命令即可!



## 生成项目包

本质: 下载300M的项目文件

ng new 包名

例如: ng new nopro  
生成过程中的询问选项, 都回车即可!

包会生成在 cmd 所在的文件夹下



下载完毕:

```
CREATE nopro/src/app/app.component.css (0 bytes)
CREATE nopro/e2e/protractor.conf.js (869 bytes)
CREATE nopro/e2e/tsconfig.json (299 bytes)
CREATE nopro/e2e/src/app.e2e-spec.ts (638 bytes)
CREATE nopro/e2e/src/app.po.ts (301 bytes)
✓ Packages installed successfully.
```

但是有的同学会报错: git 不是内部或外部命令, 也不是可运行的程序...

因为没有安装git软件, 但是并不重要! 不影响使用

安装了git的同学:

```
Successfully initialized git.
```

报错: Package install failed, see above. 网络环境差, 无法下载完成

## HelloWorld

启动之后, 访问 localhost:4200 可能会拒绝访问

此时 大概率是 系统防火墙拦截了你的请求: 关闭防火墙即可!

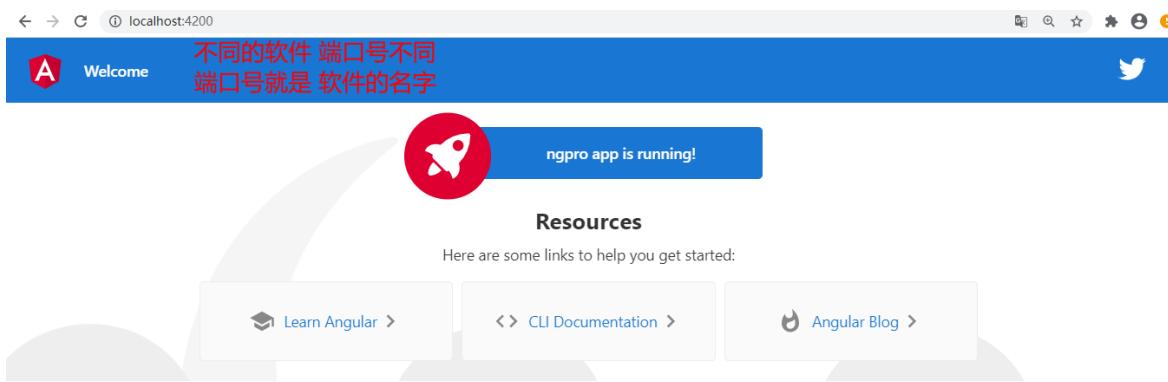
控制面板->防火墙->关闭即可!

启动命令

必须在项目根目录下执行  
\$ ng serve --open

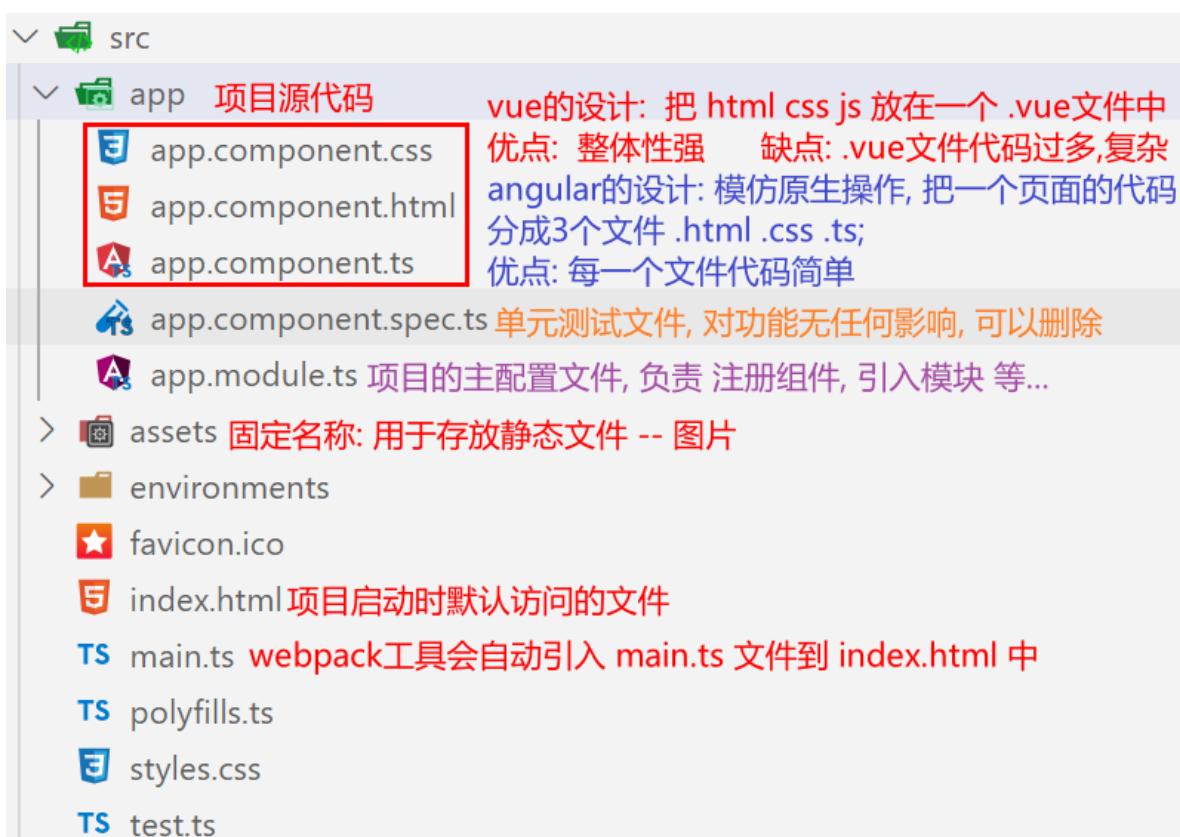
简写:  
\$ ng s -o

会自动在默认浏览器中打开 <http://localhost:4200/>  
也可以手动在你喜欢的浏览器中打开



## 解析 ng s -o 命令做了什么

- 启动了项目服务器，服务器的端口号(名字)是 4200
- 在浏览器上访问: `localhost:4200`
  - localhost: 域名；在网络上计算机的自称，相当于是 我
  - 4200: 端口号；计算机中程序的唯一名称
  - localhost:4200 代表访问 当前计算机 的 名字时 4200的程序
- 服务器有默认设定：如果不指定访问哪个文件，则默认访问 `index.html`

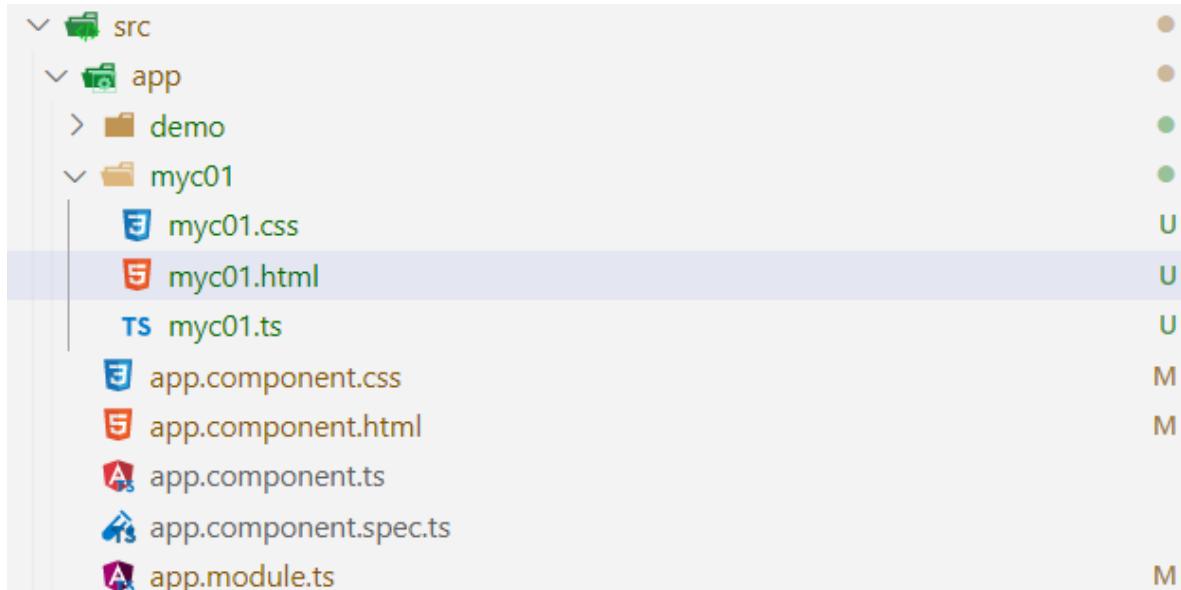


**ts:** angular使用的不是 `js` 语言, 而是 `ts` 语言, 所以文件结尾都是 ts

- JavaScript: 想当年 是一个程序员为了应付 KPI , 用了7天时间模仿java 和 c 制作的
  - 由于是赶工的作品, 所有有先天缺陷: 没有面向对象 使用的是原型进行扩展...
- TypeScript: 微软公司在JavaScript 基础上进行了完善; 整体与js语法完全一致 并且添加了很多 令人兴奋的新特性

# 组件

- vue
  - 组件是 .vue 文件；把 html js css 都写在这个vue文件中
- angular
  - 组件是由3个文件组成: .html, .css, .js



<!-- 使用myc01 --> ctrl+左键 放在标签上, 可以看到声明方式

```
<app-myc01></app-myc01>
@Component({
  // selector: 代表当前组件使用时的名字，为了防止与系统
  // 标签重名，习惯上添加 app- 前缀； 使用时 <app-myc01></
  app-myc01>
  selector: 'app-myc01',
  // 运行的是 ts 文件， ts中引入 html 和 css
  templateUrl: './myc01.html',
  styleUrls: ['./myc01.css'],
})
(component) AppModule.Myc01: typeof Myc01
```

myc01.ts

```
// 区别于 原生html开发：
// 原生的HTML：运行的是html文件，然后通过 link 来引入css， 通过script 引入js

// angular:
// 与原生相同点--分成三个文件；
// 不同点：主文件为 .ts 文件； 运行.ts文件，此文件中加载了 html 和 css

// 组件生成完毕之后，必须注册为全局组件 才能全局使用!!
```

```
// 全局配置文件: src/app/app.module.ts

// 快捷代码块: ng-component
import { Component, OnInit } from '@angular/core';

@Component({
  // selector: 代表当前组件使用时的名字, 为了防止与系统标签重名, 习惯上添加 app- 前缀; 使用时 <app-myc01></app-myc01>
  selector: 'app-myc01',
  // 运行的是 ts 文件, ts中引入 html 和 css
  templateUrl: './myc01.html',
  // css文件可以有多个, 所以是数组类型
  styleUrls: ['./myc01.css'],
})
// 面向对象的类
// Myc01 如果有爆红, 不影响使用, 但是在设置中搜索 experimentalDecorators; 勾选即可
export class Myc01 implements OnInit {
  constructor() {}

  ngOnInit(): void {}
}
```

### myc01.html

```
<p>Hello, myc01</p>
```

### 根组件: app.component.html

```
<h1>Hello world!</h1>

<!-- 使用myc01 -->
<!-- 运行: ng s -o -->
<app-myc01></app-myc01>
```

### 全局配置文件: app.module.ts

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

// 默认的引入: app.component.ts 即根组件
import { AppComponent } from './app.component';
// 引入myc01
// {}中的名字 就是在 myc01.ts 中自定义的 class 名
import { Myc01 } from './myc01/myc01';

@NgModule({
  // 注册为全局组件
```

```
declarations: [AppComponent, Myc01],  
imports: [BrowserModule],  
providers: [],  
bootstrap: [AppComponent],  
}  
export class AppModule {}
```

## 练习: 自制 myc02 组件

- 创建目录结构

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

- 在html中书写: <h2>你好, myc02</h2>
- 在ts中: 使用 `ng-component` 代码快速生成基础结构
  - 修改默认生成的模板中的各处位置

`selector` 为 app-myc02      类名: Myc02  
`html` 和 `css` 的引入

- 到 `app.module.ts` 中注册 myc02 组件为全局组件
- 使用: 到 `app.component.html` 中使用 myc02 组件

## 作者提供的快捷命令

背景: 作者认为 组件制作的流程是固定的

生成3个文件 并且 注册到全局

程序员的座右铭: 懒!!

```
$ ng generate component 组件名
```

`generate:` 生成  
`component:` 组件

简写:

```
$ ng g c 组件名
```

## 实测

在项目根目录下执行

```
$ ng g c myc03
```

```
D:\WEBTN2004\17_Angular\Day01\ngpro>ng g c myc03
CREATE src/app/myc03/myc03.component.html (20 bytes)
CREATE src/app/myc03/myc03.component.spec.ts (621 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 (621 bytes)
```

自动生成组件需要的  
3个文件 + 1个没用的  
spec.ts 是单元测试文件  
对开发没有用

自动注册为全局组件

## html 与 ts 文件的合作

生成组件: myc04 并且 在根组件中加载使用

```
$ ng g c myc04
```

myc04.component.ts

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

@Component({
  selector: 'app-myc04',
  templateUrl: './myc04.component.html',
  styleUrls: ['./myc04.component.css'],
})
export class Myc04Component implements OnInit {
  // 此处的属性，相当于在 vue 的 data(){ return {} } 书写的
  // 可以直接在 html 中使用

  name = 'dongdong';
  age = 18;
  married = true; //结婚

  names = ['东东', '然然', '亮亮'];

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

  constructor() {}

  ngOnInit(): void {}
}
```

myc04.component.html

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

<ul>
  <li>{{ name }}</li>
  <li>{{ age }}</li>
  <li>{{ married }}</li>
  <li>{{ names }}</li>
  <li>{{ boss }}</li>
```

```

<li>{{ names[0] }}</li>
<li>{{ boss.name }}</li>
</ul>

<!-- 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 > 19 }}</li>
  <li>{{ age >= 19 }}</li>
  <li>{{ age < 19 }}</li>
  <li>{{ age <= 19 }}</li>
  <li>{{ age == 19 }}</li>
  <li>{{ age != 19 }}</li>
</ul>

<!-- 逻辑运算符 -->
<!--
  逻辑与: && 读并且
  逻辑或: || 读或者
  逻辑非: ! 读不是
-->
<ul>
  <li>{{ true && true }}</li>
  <li>{{ true && false }}</li>
  <li>{{ true || true }}</li>
  <li>{{ false || false }}</li>
  <li>{{ !false }}</li>
  <li>{{ !true }}</li>
</ul>

<!-- 三元/目运算符 -->
<ul>
  <li>{{ age > 18 ? "成年" : "未成年" }}</li>
</ul>

```

## 属性 和 事件

新建组件: myc05 并展示到根组件中

```
$ ng g c myc05
```

myc05.component.ts

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

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

  img =
    'https://tse2-mm.cn.bing.net/th/id/OIP.k_j23KT_a1tpBE9KwQ1nagHaLH?
pid=Api&rs=1';

  // 函数写在类中：称为 方法
  show() {
    alert('事件触发');
  }

  constructor() {}

  ngOnInit(): void {}
}
```

myc05.component.html

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

<h3 title="斗罗大陆">唐三的作品</h3>

<!--
  vue中：
  v-bind:属性名="变量"
  :属性名="变量"
-->

<!--
  [属性名]="变量"
-->
<h3 [title]="title">唐三的作品</h3>

<img [src]="img" alt="" height="100" />

<!--
  vue中：
  v-on:事件名="函数名()"
  @事件名="函数名()"
-->
<!--
```

```
angular中:  
  (事件名)="函数名()"  
-->  
<button (click)="show()">点击</button>
```

## 练习: 计数器

组件: myc06

- 一个属性代表按钮上的数字: num=1;
- 点击时, 把属性num的值 +1

每次点击, 按钮上的数字会自动+1



myc06.component.ts

```
import { Component, OnInit } from '@angular/core';  
  
@Component({  
  selector: 'app-myc06',  
  templateUrl: './myc06.component.html',  
  styleUrls: ['./myc06.component.css'],  
})  
export class Myc06Component implements OnInit {  
  num = 1;  
  
  addOne() {  
    this.num++;  
  }  
  
  constructor() {}  
  
  ngOnInit(): void {}  
}
```

myc05.component.html

```
<p>myc06 works!</p>  
  
<button (click)="addOne()">{{ num }}</button>
```

## 一些特殊的属性

组件: myc07

```
$ ng g c myc07
```

.ts

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

@Component({
  selector: 'app-myc07',
  templateUrl: './myc07.component.html',
  styleUrls: ['./myc07.component.css'],
})
export class Myc07Component implements OnInit {
  html = '<h1>ng g c 组件名</h1>';

  enable = true;

  // 双向数据绑定
  word = '默认值';

  constructor() {}

  ngOnInit(): void {}
}
```

.html

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

<!--
  vue 和 ng 中的{{}}
  相当于原生DOM 的 innerText
  原样输出

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

<!-- disabled 属性 -->
<!-- 可以通过 boolean 布尔值来切换是否生效 -->
<button [disabled]!="enable">不可用的按钮</button>
<!-- 点击事件的本质：就是执行""里的代码 -->
<button (click)="enable = !enable">切换不可用状态</button>

<br />
<!-- 双向数据绑定 -->
<!--
  vue: v-model
  ng: [(ngModel)]
  快捷提示: ng-model

angular默认不加载 Form 模块，导致双向绑定写法不可用！
```

```

手动加载即可： 全局配置文件 app.module.ts 中
-->
<input type="text" [(ngModel)]="word" />
<p>{{ word }}</p>
<!--

双向绑定：
方向1：属性显示在输入框上
方向2：输入框变化时，属性也会随之变化-- p标签的值变了
-->

```

```

src > app > A app.module.ts > ⚙ AppModule
17 // 引入Form模块
18 import { FormsModule } from '@angular/forms';
19
20 @NgModule({
21   // 注册为全局组件
22   declarations: [
23     AppComponent,
24     Myc01,
25     Myc02,
26     Myc03Component,
27     Myc04Component,
28     Myc05Component,
29     Myc06Component,
30     Myc07Component,
31   ],
32   // 引入模块的属性 : imports
33   // 引入模块之后，必须重启服务器才可以!!
34   imports: [BrowserModule, FormsModule],
35   providers: [],
36   bootstrap: [AppComponent],

```

## 插件

### 必装: angular插件

	<b>Angular Language Service</b> 0.1000.7 Editor services for Angular templates Angular	1.7M ★ 3.5
	<b>Angular 8 Snippets - TypeScript, Html, Angular Material, ngRx...</b> 8.1.2 242 Angular Snippets (TypeScript, Html, Angular Material, Flex Layout, ngRx, RxJS, P... Mikael Morlund	1.4M ★ 4.5 <b>安装</b>

### 选装:

- 让导航栏的图片变好看

	<b>vscode-icons</b> 10.2.0 Icons for Visual Studio Code VSCode Icons Team	5.8M ★ 4.5
---	---	------------

- 通过tab 按钮移动光标

**TabOut 0.2.0** 62K ★ 4.5



Tab out of quotes, brackets, etc  
Albert Romkes 安装

- 代码格式化插件: 在代码中右键 选择 格式化文档

**Prettier - Code formatter 5.1.3** 7.9M ★ 4



Code formatter using prettier  
Prettier 安装

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。  
电脑配置高的同学, 可以在设置中配置此栏目

- 在最常用的代码提示上 加星提示

**Visual Studio IntelliCode 1.2.10** 6.6M ★ 4



AI-assisted development  
Microsoft 安装

- 路径自动提示

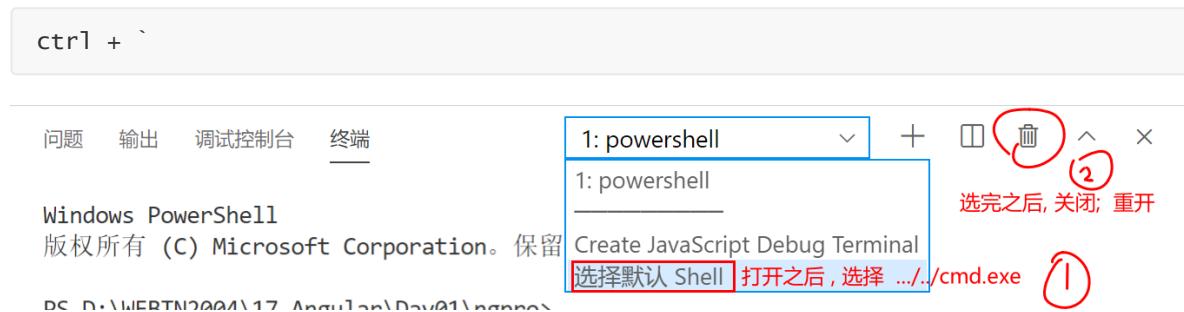
**Path Intellisense 2.2.1** 3.3M ★ 5



Visual Studio Code plugin that autocompletes f...  
Christian Kohler 齿轮图标

VSCode自带快捷命令行工具, 快捷键:

ctrl + `



问题 输出 调试控制台 终端

Windows PowerShell  
版权所有 (C) Microsoft Corporation。保留

1: powershell  
1: powershell  
Create JavaScript Debug Terminal  
选择默认 Shell 打开之后, 选择 .../cmd.exe

experimentalDecorators 找到 1 个设置

用户 工作区 打开设置同步

扩展 (1) 类中有个小报错: 非代码错误, 可以通过此处取消报错

TypeScript (1)

JavaScript > Implicit Project Config: Experimental Decorators

Enable/disable **experimentalDecorators** for JavaScript files that are not part of a project. Existing jsconfig.json or tsconfig.json files override this setting.

## 作业

最好生成全新的项目包执行

练习1: 加减计数器

- 点击 加 和 减 按钮, 中间的数字+1 和 -1
- 当数字达到20的时候, 则加号按钮变为不可用状态
- 当数字达到1的时候, 则减号可以点, 但是会提示: **最小为1** 数字不会变化

-	5	+
---	---	---

练习2: 西西的爱好

西西晚上喜欢:	<input type="text"/>	在输入框中输入西西的爱好, 点击确定按钮, 让文字出现在黑色框中 并删除 输入框中的内容
西西的爱好	<input type="button" value="确定"/>	

练习3: 商品信息

商品名: iPhone

商品价格:	<input type="text" value="7999"/>
商品数量:	<input type="button" value="-"/> <input type="text" value="5"/> <input type="button" value="+"/>
总价:	<input type="text" value="XXXX"/>

商品价格是输入框, 可以修改  
商品数量可以通过按钮修改, 最小为1  
数量为1时, 减按钮不可点击  
总价格会自动 随数量和单价而变化

练习4: 轮播图

- 需要一个数组属性, 存储多个网络图片的网址; (图片地址自行到百度图片查找)
- 需要一个 `curp` 属性来代表当前页数, 默认值可以为1
- html中, 图片的地址 需要通过 `curp` 从 数组中读取
- 点击上一页 和 下一页按钮, 变化 `curp`的值, 图片就会随之变化
- 注意循环的极限值问题: 当最后一页时, 点击下一页变为第一页. 当第一页时点击上一页变为第一页

