



黑马程序员™
www.itheima.com

传智播客旗下
高端IT教育品牌

TypeScript 初体验

目录

Contents

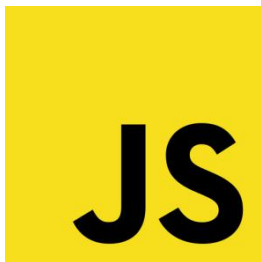
- ◆ TypeScript 概述
- ◆ 开发工具准备
- ◆ 第一个 TS 文件
- ◆ 注释和输出语句

■ 1. TypeScript 概述

- JavaScript 是什么
- TypeScript 是什么
- TypeScript 相比 JavaScript 的优势

1. TypeScript 概述

1.1 JavaScript 是什么



JavaScript (缩写: JS) 是一种运行在客户端 (比如: 浏览器) 中的编程语言。

当应用于浏览器时, 为网站提供动态交互特性, 让网页“动”起来。



1. TypeScript 概述

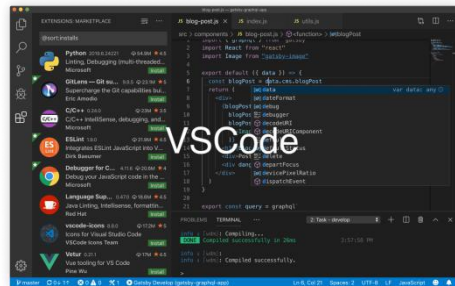
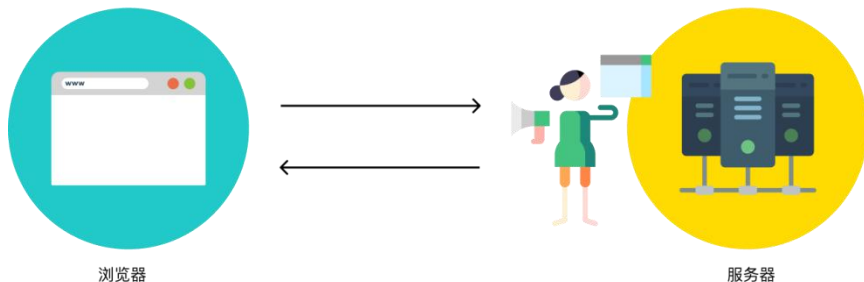
1.1 JavaScript 是什么

JavaScript 的运行环境：1 浏览器 2 Node.js。

Node.js 让 JavaScript 摆脱了浏览器的束缚，可以实现服务端/桌面端编程等。

其他：微信小程序、微信小游戏 等等。

总结：JavaScript 既能运行在 浏览器 中，也可以运行在 Node.js 中，前后端通吃，无所不能。



1. TypeScript 概述

1.2 TypeScript 是什么



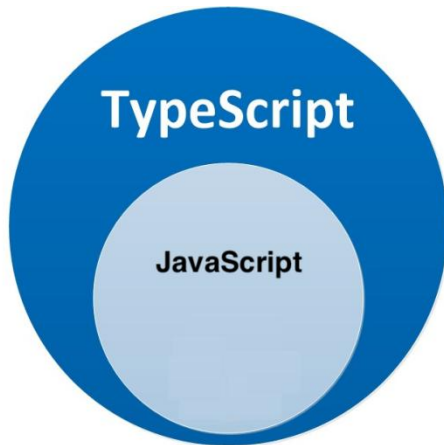
TypeScript (简称: TS) 是 JavaScript 的**超集** (JS 有的 TS 都有)。

TypeScript = **Type** + JavaScript (为 JS 添加了**类型系统**)。

TypeScript 是**微软**开发的开源编程语言, 设计目标是**开发大型应用**。
可以在任何浏览器、任何计算机、任何操作系统上运行。

```
// TypeScript 代码: 有明确的类型, 即 : number (数值类型)
let age: number = 18

// JavaScript 代码: 无明确的类型
let age = 18
```



1. TypeScript 概述

1.3 TypeScript 相比 JS 的优势

会的人少

工作好找

工资又高

待遇也好

大型项目必备

用了TS就不想
再用JS



TS

JS



我信你个鬼

1. TypeScript 概述

1.3 TypeScript 相比 JS 的优势

悲伤的故事：男程序员都是好男人，因为他们总会在电脑前问：我到底又错在哪了？

JS 的类型系统存在“先天缺陷”，绝大部分错误都是类型错误（Uncaught **Type**Error）。

- 优势一：类型化思维方式，使得开发更加严谨，提前发现错误，减少改 Bug 时间。
- 优势二：类型系统提高了代码可读性，并使维护和重构代码更加容易。
- 优势三：补充了接口、枚举等开发大型应用时 JS 缺失的功能。
- [Vue 3](#) 源码使用 TS 重写，释放出重要信号：TS 是趋势。
- [Angular](#) 默认支持 TS；[React](#) 与 TS 完美配合，是很多大型项目的首选。
- 我呼吁：准程序员们，为了发量、为了对象，让我们手拉手一起用 TS 吧！

为啥出来这么个bug?



目录

Contents

- ◆ TypeScript 概述
- ◆ 开发工具准备
- ◆ 第一个 TS 文件
- ◆ 注释和输出语句

2. 开发工具准备

2.1 开发工具介绍

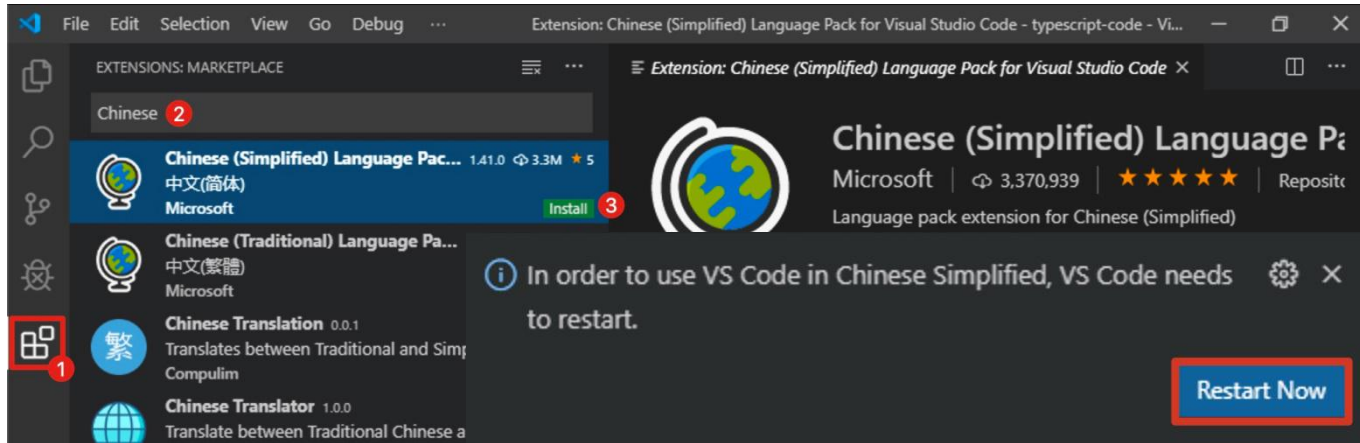
1. 写代码的工具：Visual Studio Code（简称：VSCode），微软开发的代码编辑工具。
 2. 运行 TS / JS 代码的工具：Node.js，一个基于 Chrome 浏览器 V8 解析引擎的 JavaScript 运行环境。
- 说明：本课程中，TypeScript 基础知识，使用 Node.js 来运行。



2. 开发工具准备

2.2 安装 VSCode

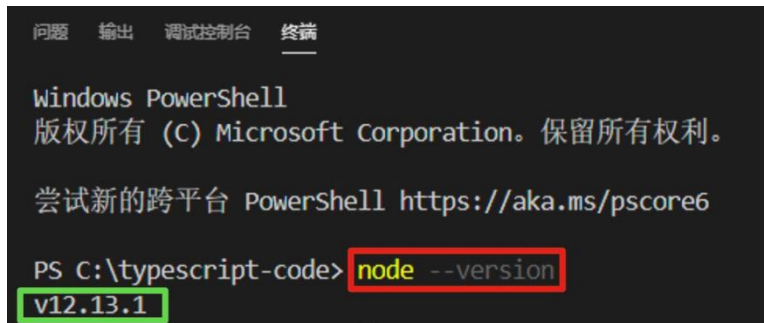
1. 按照步骤引导安装即可。
2. 安装中文汉化插件：Chinese (Simplified) Language Pack for Visual Studio Code
3. 点击右下角弹出的对话框，重启（Restart）VSCode。



2. 开发工具准备

2.3 安装 Node.js

1. 按照步骤引导安装即可。
2. 在终端（黑窗口）中输入：`node --version`，验证是否安装成功（注意：需要重启VSCode）。
3. 如果打印出版本信息说明安装成功。



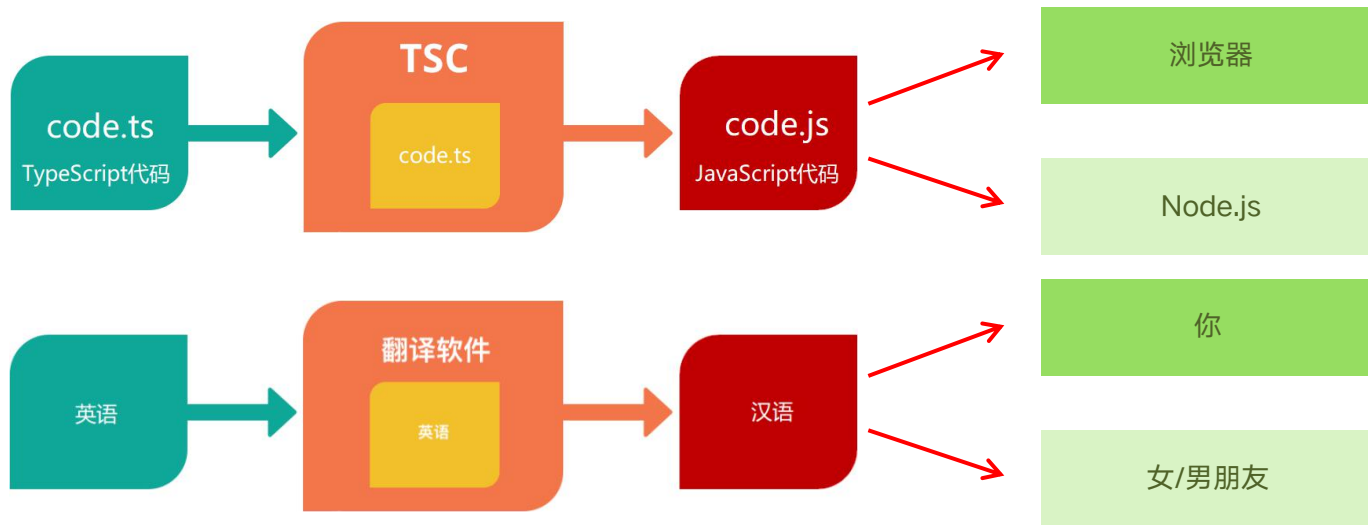
2. 开发工具准备

2.4 安装解析 TS 的工具包

问题：为什么要安装这个工具包？

答案：

- Node.js/浏览器，只认识 JS 代码，不认识 TS 代码。
- 需要先将 TS 代码转化为 JS，然后就可以在 Node.js/浏览器中运行了。



2. 开发工具准备

2.4 安装解析 TS 的工具包

安装步骤：

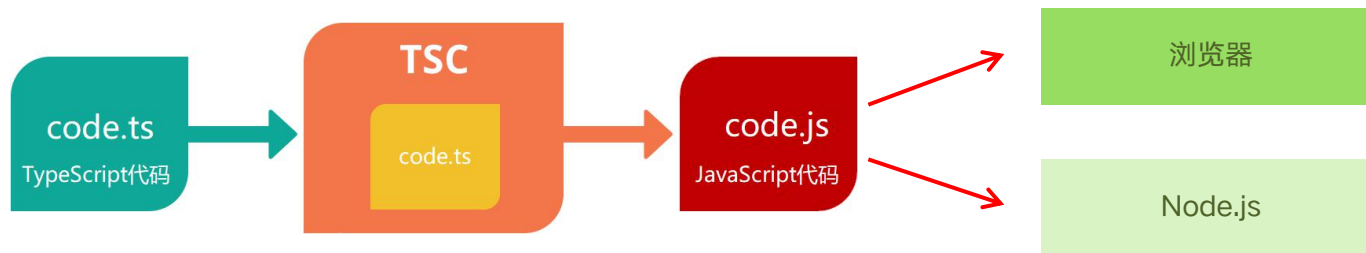
1. 打开 VSCode 终端。
2. 输入安装命令：`npm i -g typescript` 敲回车，来安装（注意：需要联网）。

typescript：就是用来解析 TS 的工具包。提供了 `tsc` 命令，实现了 TS -> JS 的转化。

npm：用来安装前端开发中用到的包，是安装 Node.js 时自动安装的。

i (install)：表示安装。

-g (--global)：全局标识，可以在任意目录中使用该工具。



目录

Contents

- ◆ TypeScript 概述
- ◆ 开发工具准备
- ◆ 第一个 TS 文件
- ◆ 注释和输出语句

3. 第一个 TS 文件

3.1 步骤

1 创建ts文件



2 写代码



3 执行代码

1. 创建ts文件:

- ① 在桌面中创建文件夹: code。
- ② 在文件夹上点击鼠标右键, 然后点击 Open With Code (用VSCode打开文件夹)。
- ③ 在 VSCode 中新建ts文件: hello.ts (注意: 文件后缀名为 .ts)。

2. 写代码: 在 hello.ts 文件中, 写入以下代码, 并保存。

```
console.log('Hello TS')
```


3. 第一个 TS 文件

3.1 步骤

1 创建ts文件



2 写代码



3 执行代码

问题1：TS 代码能直接在 Node.js 里面运行吗？不能

问题2：该如何处理呢？ 1 TS代码 -> JS代码 2 执行

3. 执行代码，分两步：^{JS}

① TS代码 -> JS代码：在当前目录打开终端，输入命令 `tsc hello.ts` 敲回车。

② 执行JS：输入命令 `node hello.js`（注意：后缀为 `.js`）。

解释：

- `tsc hello.ts` 会生成一个 `hello.js` 文件。
- `node hello.js` 表示执行这个 JS 文件中的代码。

3. 第一个 TS 文件

3.2 简化执行TS的步骤

问题：每次修改代码后，都要重复执行两个命令才能执行 TS 代码，太繁琐。

执行 TS 代码的两个步骤：

```
1 tsc hello.ts
2 node hello.js
```

简化方式：使用 **ts-node** 包，“直接”在 Node.js 中执行 TS 代码。

- 安装命令：npm i -g ts-node。
- 使用方式：**ts-node hello.ts**。

解释：

- ts-node 包内部偷偷的将 TS -> JS，然后，执行 JS 代码。
- ts-node 包提供了命令 **ts-node**，用来执行 TS 代码。

现在只需一步：

```
ts-node hello.ts
```

目录

Contents

- ◆ TypeScript 概述
- ◆ 开发工具准备
- ◆ 第一个 TS 文件
- ◆ 注释和输出语句

■ 4. 注释和输出语句

4.1 注释

注释是对代码的解释说明，用来帮助阅读和理解代码。

注意：注释的内容是不会执行的。

推荐在写代码时，添加代码注释，增加代码的可读性。

北冥有鱼，其名为鲲¹。

注1：鲲(kūn)，本指鱼卵，此处借用为表大鱼之名。



4. 注释和输出语句

4.1 注释

注释有两种形式：1 单行注释 2 多行注释。

形式一：单行注释

```
// 两个斜线，表示单行注释，只能注释这一行内容  
// 这是第二行注释  
// 快捷键：ctrl + /
```

形式二：多行注释

```
/*  
这是多行注释  
可以注释多行内容  
快捷键：shift + alt + a  
*/
```

■ 4. 注释和输出语句

4.2 输出语句

作用：在终端（黑窗口）中打印信息。

Node.js 会执行我们写的代码，为了能够知道代码执行的结果，就需要使用输出语句，将结果打印出来。

```
console.log('Hello TS')
```

解释：

- console 表示控制台，在 Node.js 中，指的是终端（黑窗口）。
- 小括号中的内容，表示要打印的信息。



总结

TypeScript初体验

1. TypeScript 是 JS 的**超集**，为 JS 添加了**类型系统**。相比 JS，开发体验更友好，**提前发现错误**，Bug更少，增加开发的幸福度。
2. JavaScript 的两个运行环境是什么？ 1 浏览器 2 Node.js
3. TypeScript 能直接在浏览器或 Node.js 中执行吗？ 不能
4. 如何将 ts 编译为 js？ `tsc hello.ts -> hello.js`
5. 如何简化执行 ts 代码？ `ts-node hello.ts`
6. 代码注释有哪两种？ 1 单行注释 2 多行注释
7. 在控制台打印信息的代码是什么？ `console.log()`



传智播客旗下高端IT教育品牌