# [【Visual Studio Code 】使用Visual Studio Code + Node.js搭建TypeScript开发环境](http://www.cnblogs.com/CrazyAnts/p/6489331.html)

**1、准备工作**

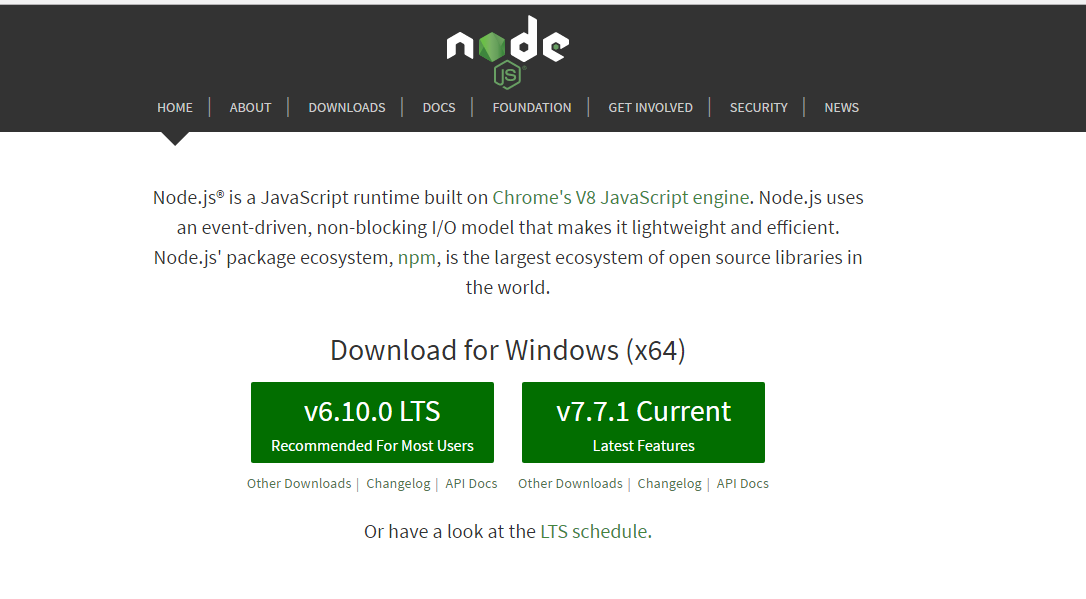
* Node.js [Node.js - Official Site](https://nodejs.org/en/)
* Visual Studio Code [Visual Studio Code - Official Site](https://code.visualstudio.com/)

**安装Node.js**

一方面提供一个开发的Runtime；另一方面提供的npm工具，我们可以利用这个工具来安装TypeScript。

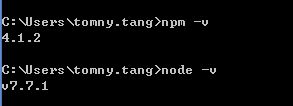
**下载Node.js安装包**

首先按照准备工作里面提供的链接下载对应平台的Node.js安装包



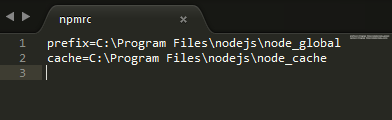
**安装Node.js并检测是否安装成功**

安装Node.js，安装过程基本上是下一步，即可完成，然后在CMD中运行如下命令：node -v 来查询当前node.js的版本号，然后输出如图所示的结果就表示node.js安装成功，接着可以输入命令：npm -v 来查询当前npm工具的版本号[可能需要时间稍长点]，便会输出如图所示的结果表示npm工具可用



**设置node.js的npm安装package的全局路径[非必须]**

由于npm安装工具默认会跑到C盘[因为npmrc文件中默认的设置为：prefix=${APPDATA}\npm]，这样有时可能会因为系统权限的问题，导致不能正常成功的安装某些工具，那么我们可以先将npm安装的全局路径自定设置一下，比如：我们可以在nodejs的目录下[即你的node.js安装后的根目录]新建两个目录：*node\_global*，*node\_cache*，然后找到*nodejs*目录下的*node\_modules/npm*目录下名为*npmrc* 或者 *.npmrc*文件，[为安全，我们可以先将该文件copy一个副本出来进行备份]使用文本编辑器打开，修改并新增如下：prefix 和 cache 分别对应之前新建的目录*node\_global* 和 *node\_cahce*



|  |  |
| --- | --- |
| 1  2 | prefix=C:\Program Files\nodejs\node\_global  cache=C:\Program Files\nodejs\node\_cache |

**设置package的环境变量[有必要]**

通过步骤3设置后，后续在安装工具的时候，比如安装TypeScript，最终会安装到*node\_global*中，所以为了后续使用工具命令行，我们可以配置一个环境变量：  
首先新建一个NODE\_PATH -> NODE\_PATH=D:\EasBuilding\nodejs\node\_global，然后在Path下新增%NODE\_PATH%

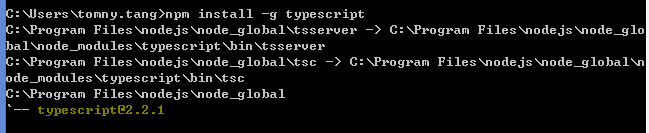
**查看typescript版本**

[这里只是为了说明问题，可不用进行这一步] 打开CMD，运行命令：tsc -v，这里不能正常像是版本号，这是由于这里还没有安装typescript，所以我们接下来就先安装TypeScript Compiler

**安装TypeScript**

**安装TypeScript Compiler**

在前面安装好Node.js后，我们可以直接使用npm工具来安装TypeScript，这个TypeScript的Package其实也是一个Compiler，我们可以通过这个Complier将typescript编译成javascript。打开命令提示符CMD(或其他终端Terminal)，输入指令：npm install -g typescript，稍等片刻即可完成TypeScript Compiler的安装



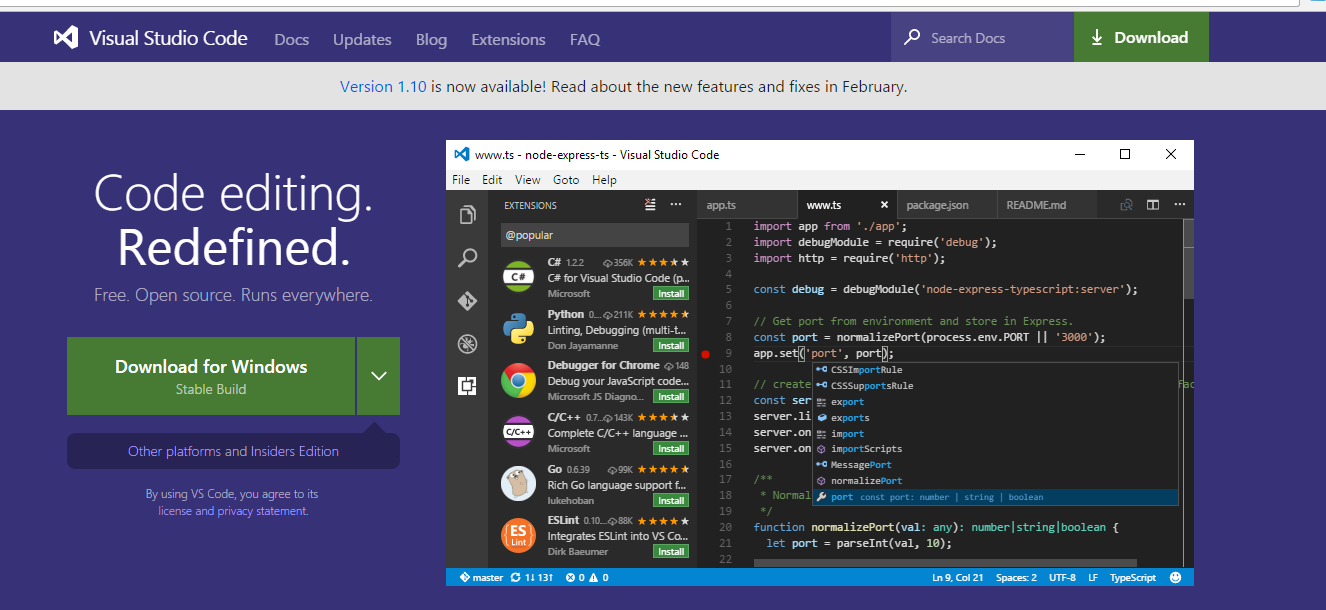
**更新TypeScript Compiler**

如上图所示，下载的TypeScript版本为1.8.10，在官网，该版本为最新稳定版，所以不需要更新，如果我们下载的版本小于这个版本，我们可以使用如下命令来进行更新：npm update -g typescript

**安装Visual Studio Code**

**下载Visual Studio Code安装包并安装**

首先按照准备工作里面提供的链接下载对应平台的vscode安装包

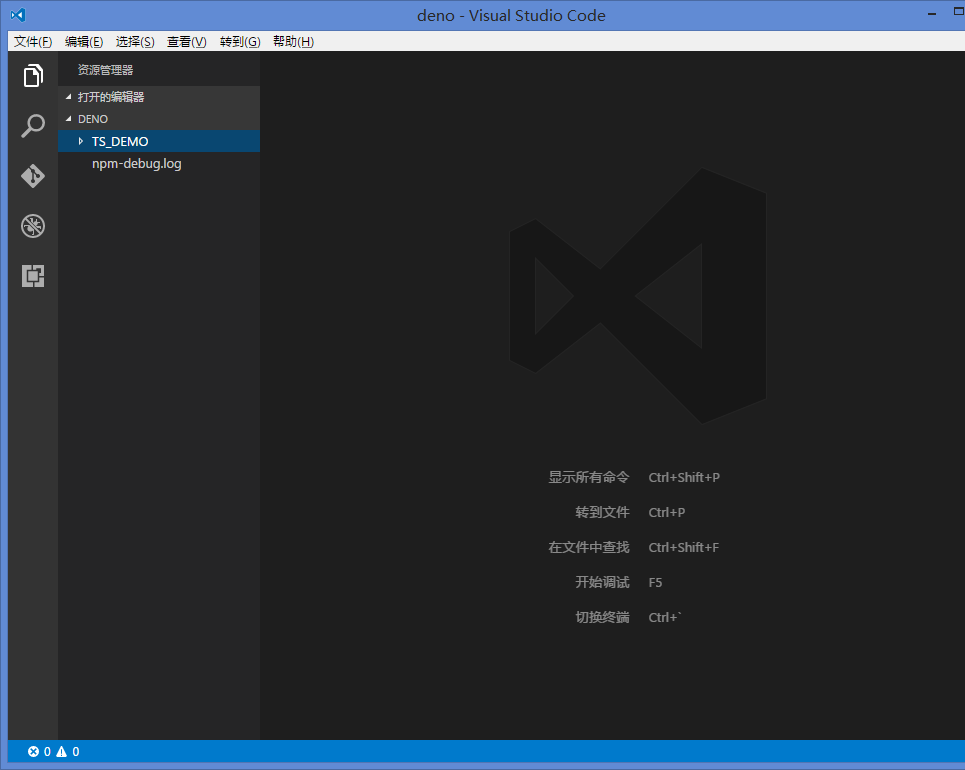


**我的第一个TypeScript程序**

**新建一个workspace工作目录**

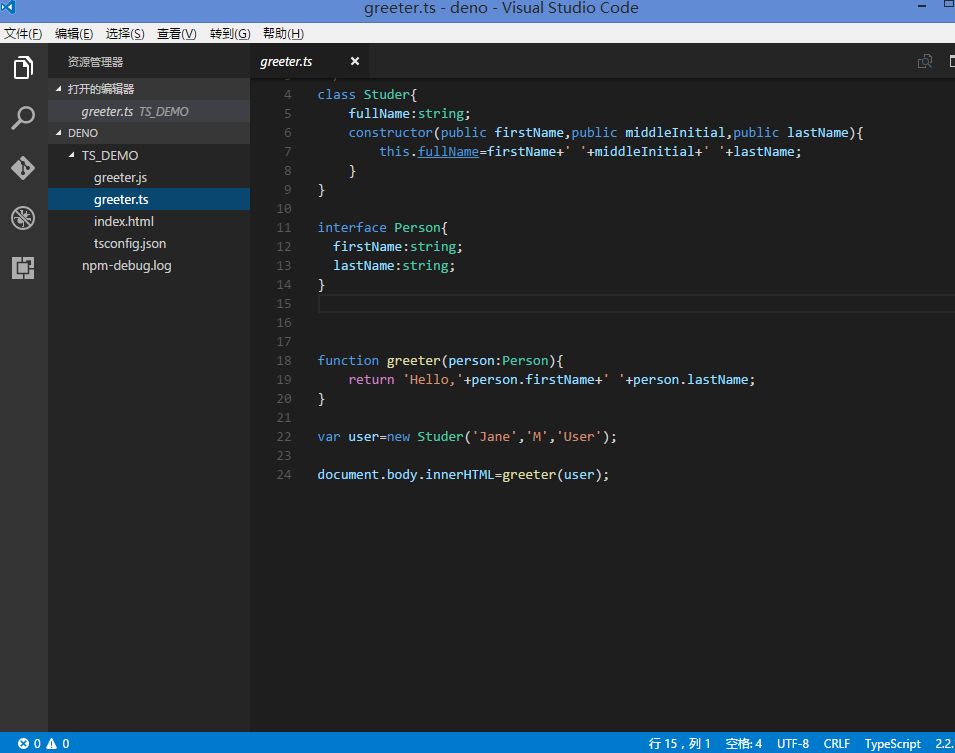
这里我先在桌面上新建一个DemoModules的目录来存放一些我的Demo Projects，然后选中DemoModules右键使用 vscode 打开或者打开CMD，切换到DemoModules下，然后输入命令：code . 既可以使用 vscode 打开DemoModules目录。

接着，在该目录下新建一个名为TS\_DEMO的目录：



**创建我的第一个typescript文件**

在TS\_DEMO下新建一个名为greeter.ts的文件，并输入如下代码：



[复制代码](javascript:void(0);)

class Studer{

fullName:string;

constructor(public firstName,public middleInitial,public lastName){

this.fullName=firstName+' '+middleInitial+' '+lastName;

}

}

interface Person{

firstName:string;

lastName:string;

}

function greeter(person:Person){

return 'Hello,'+person.firstName+' '+person.lastName;

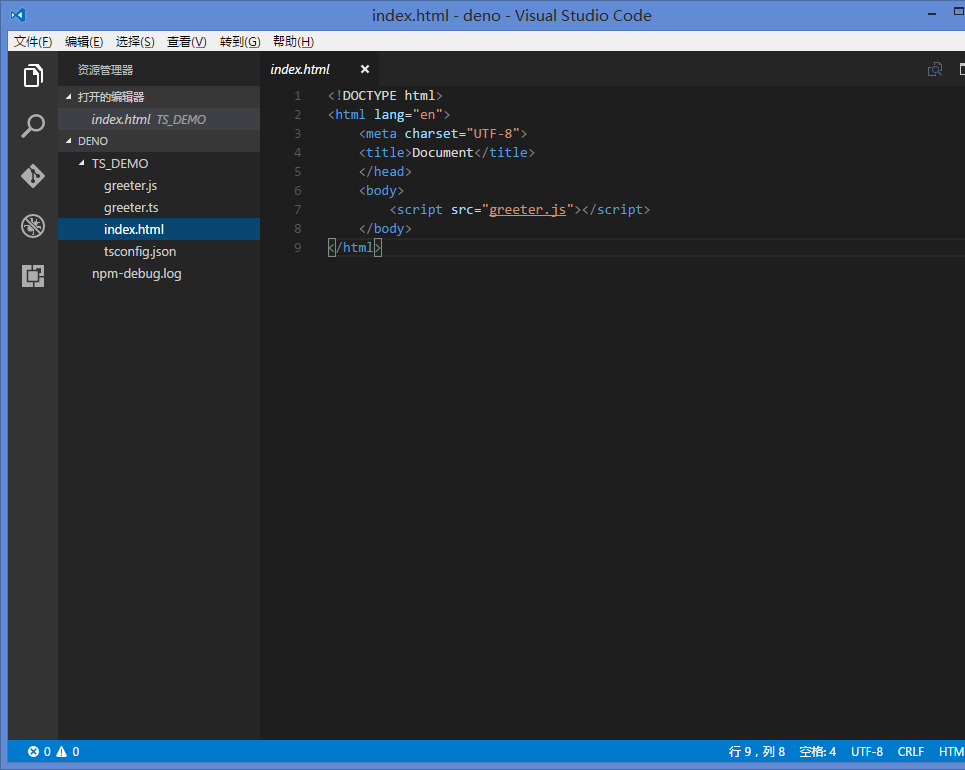
}

var user=new Studer('Jane','M','User');

document.body.innerHTML=greeter(user);

[复制代码](javascript:void(0);)

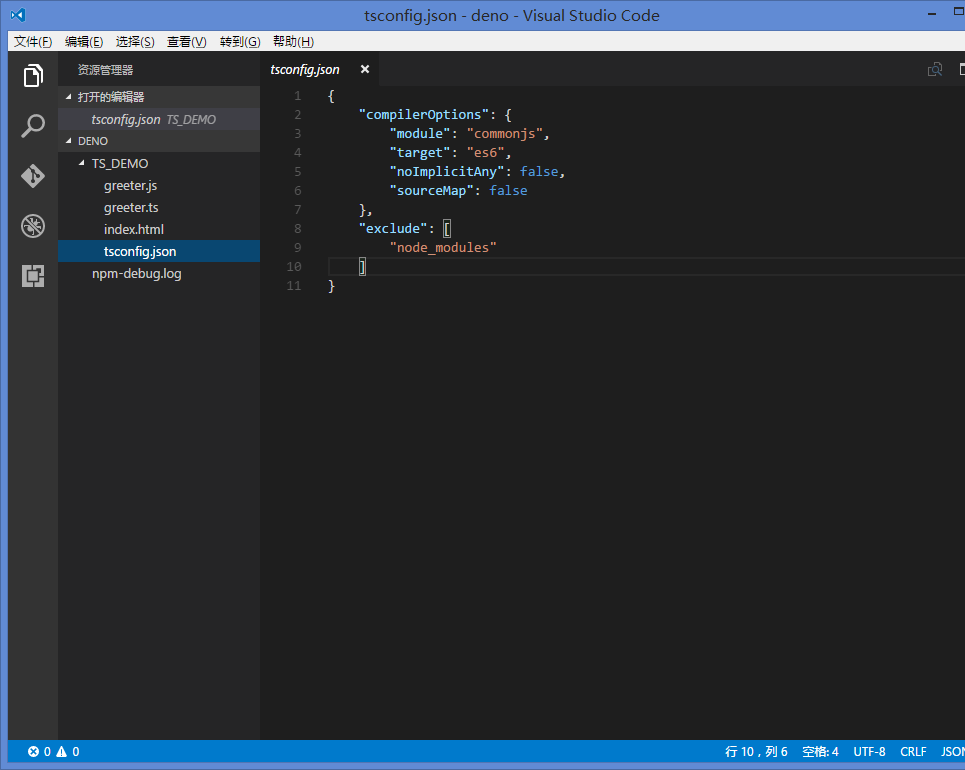
继续在该目录下新建一个index.html的文件，并在body中引入greeter.js脚本：



|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9 | <!DOCTYPE html>  <html lang="en">      <meta charset="UTF-8">      <title>Document</title>      </head>      <body>          <script src="greeter.js"></script>      </body>  </html> |

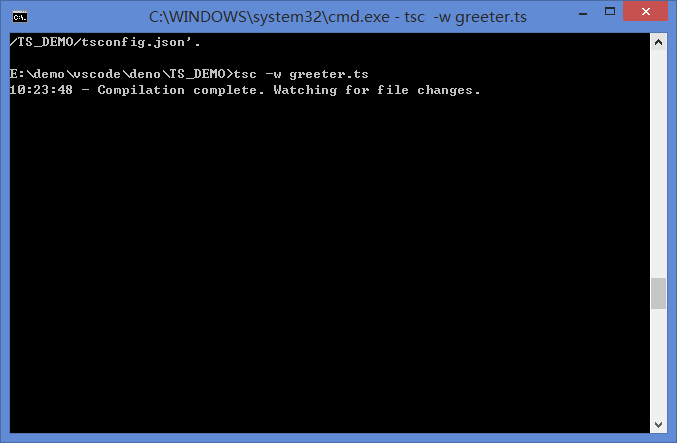
**编译typescript文件**

首先在TS\_DEMO目录下创建一个名为tsconfig.json的文件，可以手动创建该文件，并输入如下代码：

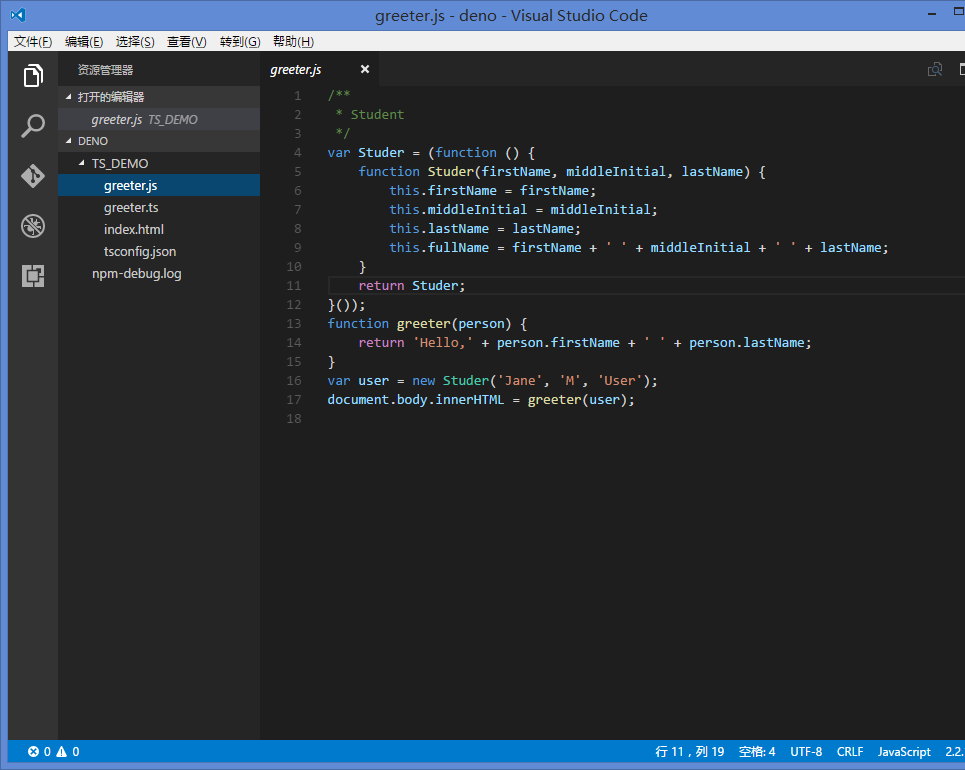


也可以使用命令自动创建这个配置文件，CMD中切换到TS\_DEMO目录，然后输入命令：tsc -init 即可自动创建

接下来我们继续输入编译命令，编译greeter.ts；输入指令：tsc -w greeter.ts, 其中-w是watch监控的意思，当typescript文件内容发生改变时会自动进行编译。



我们可以看到，TS\_DEMO目录下多了一个*.js*后缀的同名文件，这就是typescript编译后的javascript文件。



最后我们将index.html用Chrome或者Firefox打开看看效果：



[转]https://segmentfault.com/a/1190000006124164