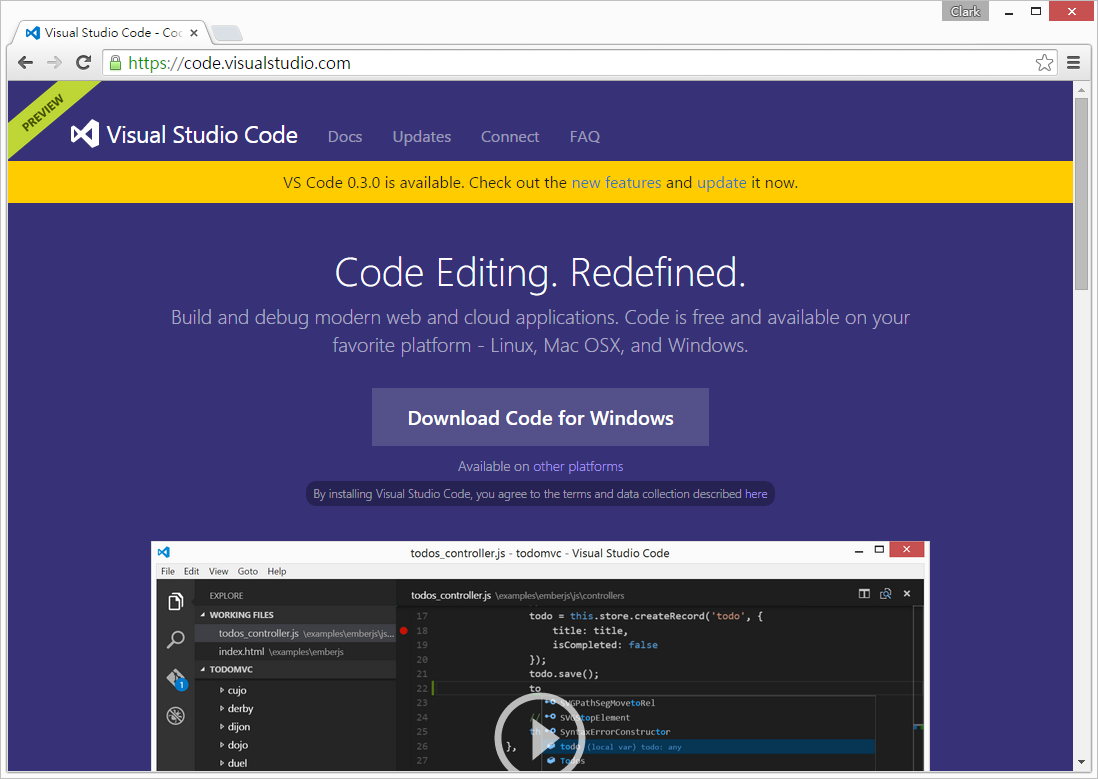
**前言**

为了解决JavaScript：缺少面向对象语法、缺少编译期间错误检查...等等问题。微软提供了一个开源的TypeScript语言，让开发人员能够使用面向对象撰写TypeScript程序代码，接着再透过TypeScript编译程序将程序代码编译成为JavaScript程序代码，就能够建立经过编译检查的JavaScript程序代码来提供平台使用。

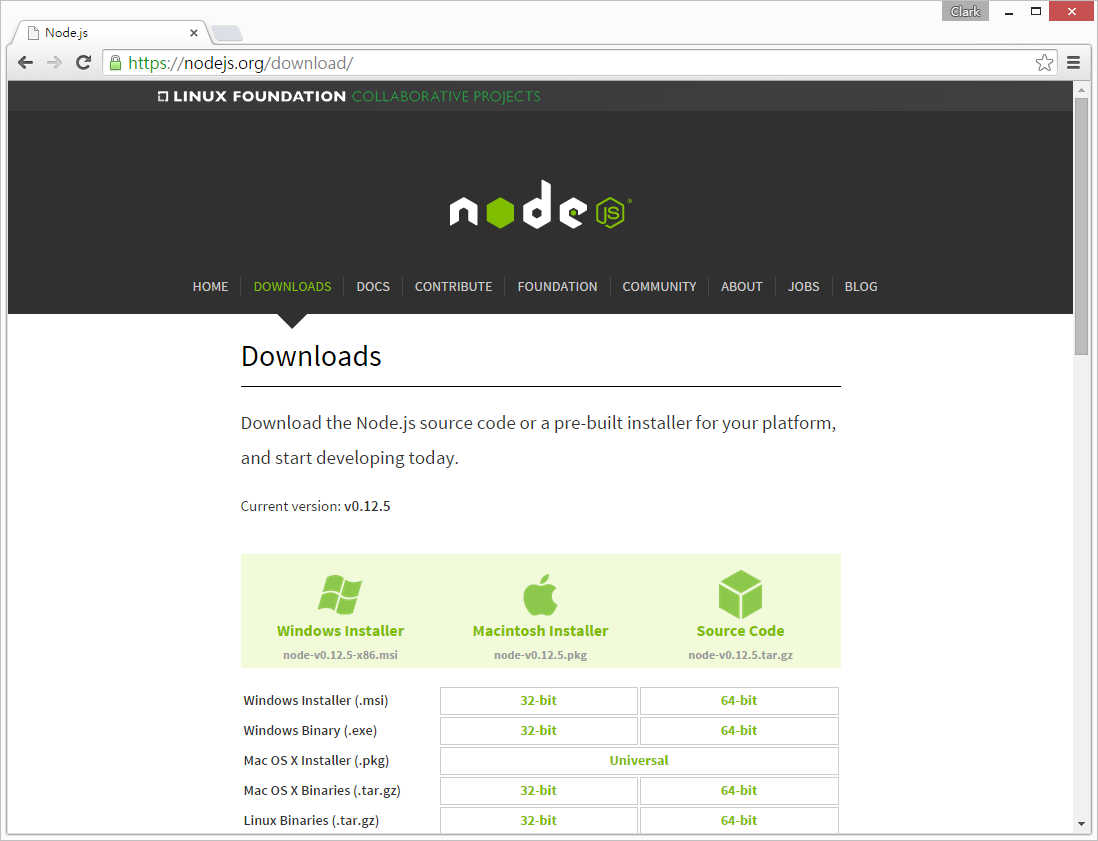
本篇文章介绍如何在「**Windows**」、「**OS X**」操作系统中，透过Visual Studio Code这个工具开发TypeScript，让没有预算添购相关工具的开发人员，也能够学习TypeScript的语法。主要为自己留个纪录，也希望能帮助到有需要的开发人员。

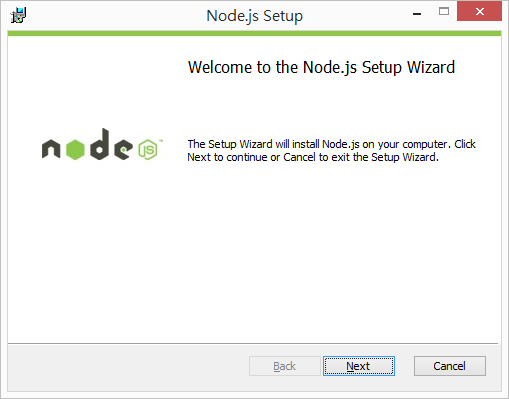


**安装Node.js**

首先要安装Node.js，后续就可以使用NPM这个工具来安装TypeScript Compiler。而Node.js的安装程序，可以从Node.js官网下载。

* [**https://nodejs.org/download/**](https://nodejs.org/download/)



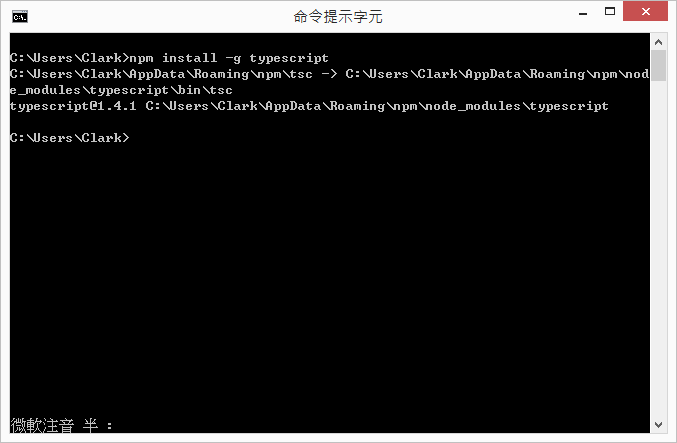


**安装TypeScript Compiler**

**安装TypeScript Compiler**

装完Node.js，接着就可以使用NPM来安装TypeScript Compiler，之后就能透过这个Compiler来将TypeScript编译成为JavaScript。开发人员使用命令提示字符(or终端机)，输入下列指令即可完成TypeScript Compiler的安装。

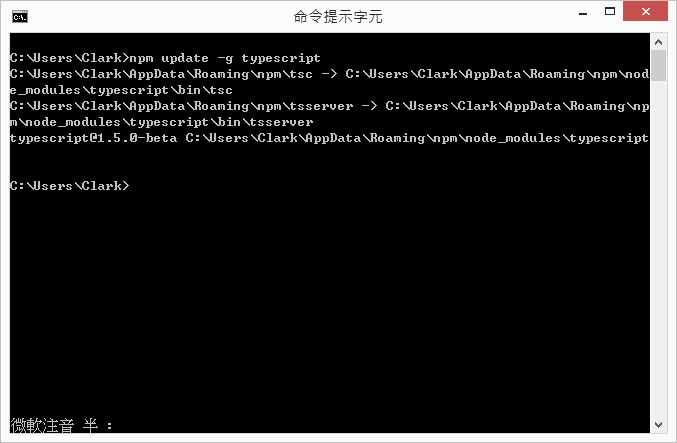
npm install -g typescript



**更新TypeScript Compiler**

检视上一个步骤所安装的TypeScript Compiler，会发现安装的版本为1.4.1。但是因为后续步骤，需要使用到1.5.0版新加入的功能，所以开发人员同样使用命令提示字符(or终端机)，输入下列指令来更新TypeScript Compiler到1.5.0版以上。

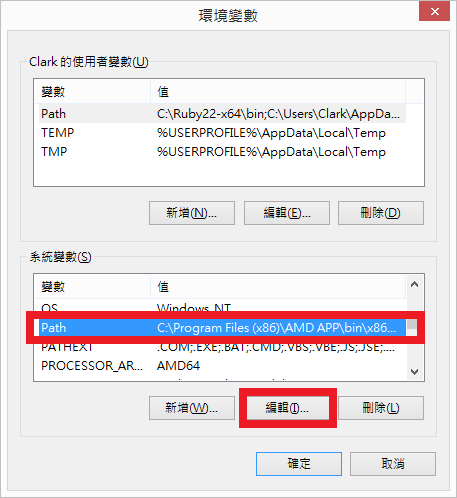
npm update -g typescript



**移除环境变量(Windows only)**

有些开发人员的计算机，先前可能已经安装过TypeScript相关工具，这些工具会在Windows环境变量中加入TypeScript Compiler的安装路径。为了统一使用NPM来管理TypeScript Compiler的版本，开发人员需要手动从环境变量中移除TypeScript Compiler的安装路径：

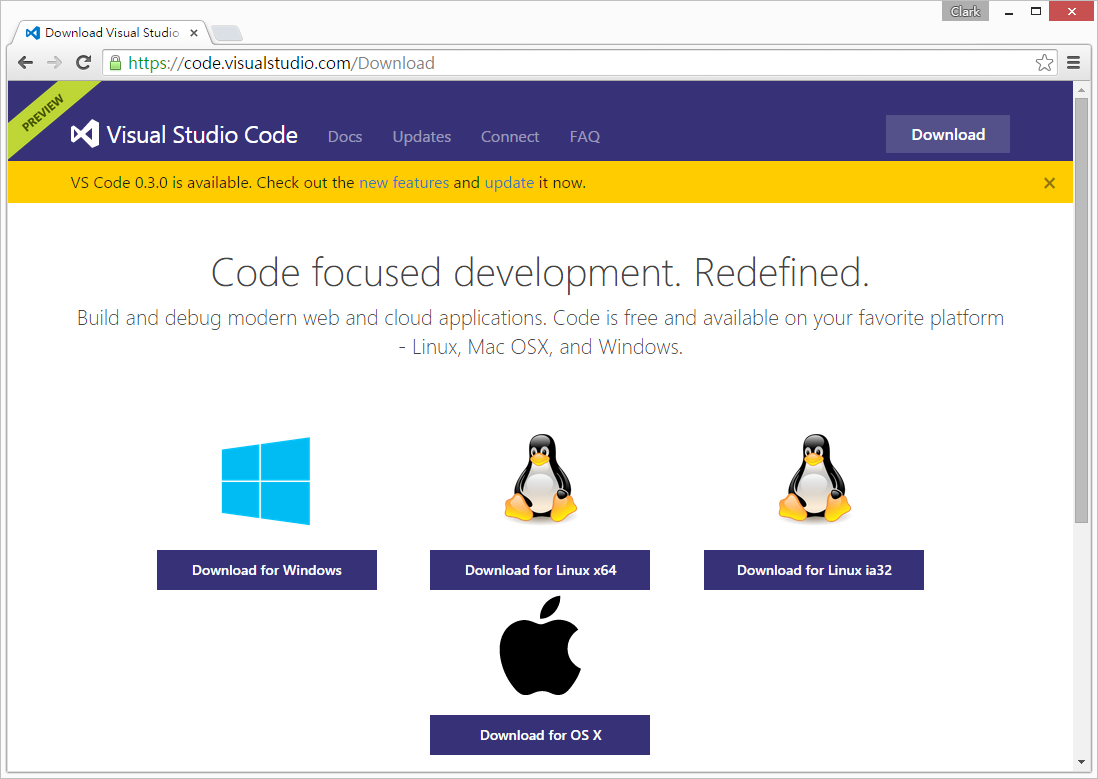
C:\Program Files (x86)\Microsoft SDKs\TypeScript\1.0\



**安装Visual Studio Code**

装完TypeScript Compiler，接着安装Visual Studio Code，之后就能透过Visual Studio Code来开发TypeScript程序代码。而Visual Studio Code的安装程序，可以从Visual Studio Code官网下载。

* [**https://code.visualstudio.com/Download**](https://code.visualstudio.com/Download)

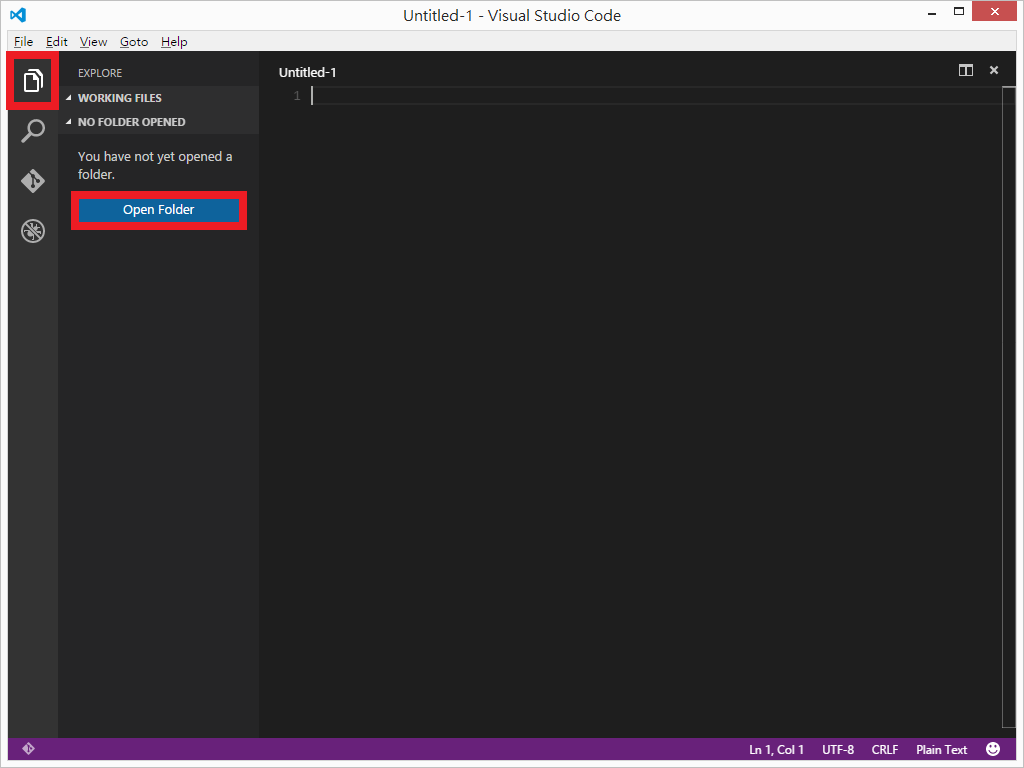


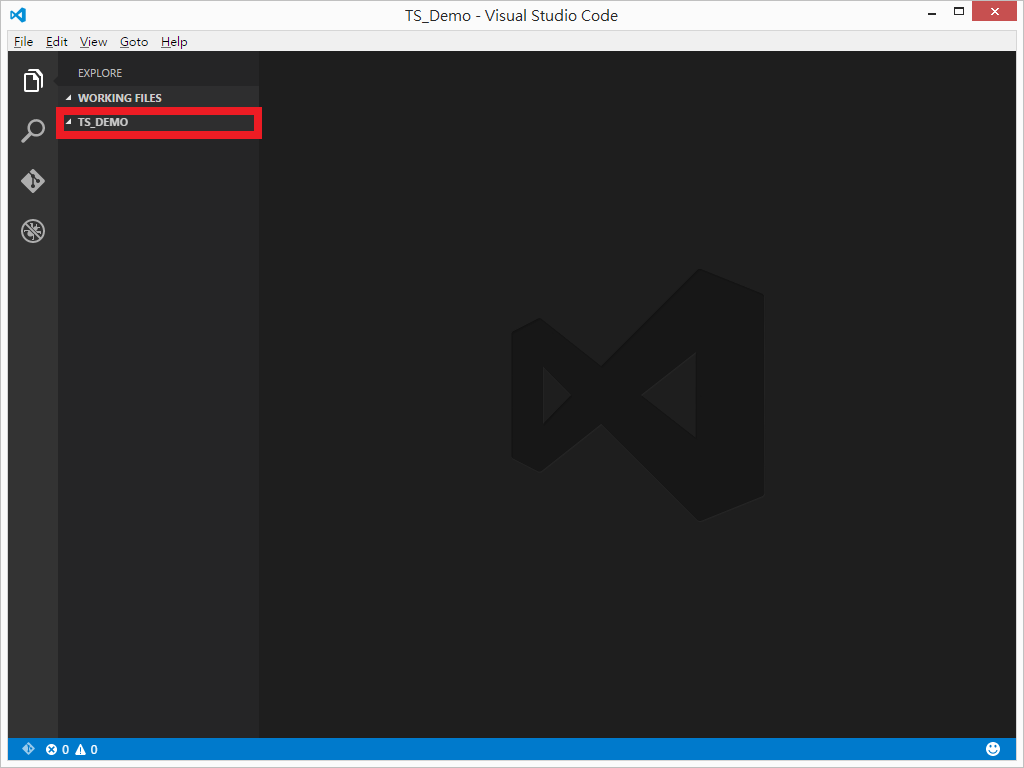


**开发TypeScript**

**建立Workspace**

完成安装步骤后，开启Visual Studio Code,并且选取一个文件夹做为开发TypeScript的工作文件夹(Workspace)。





**建立tsconfig.json**

接着在Workspace加入一个新档案「tsconfig.json」，并且输入下列JSON设定参数。

{

"compilerOptions": {

"target": "es5",

"noImplicitAny": false,

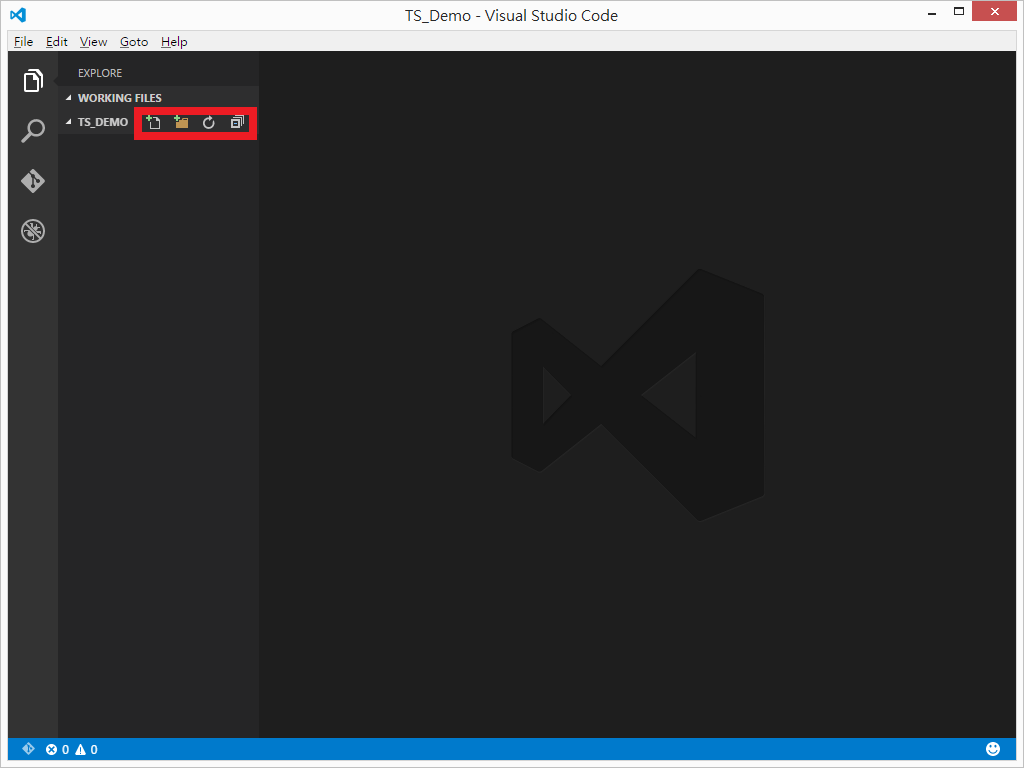
"module": "amd",

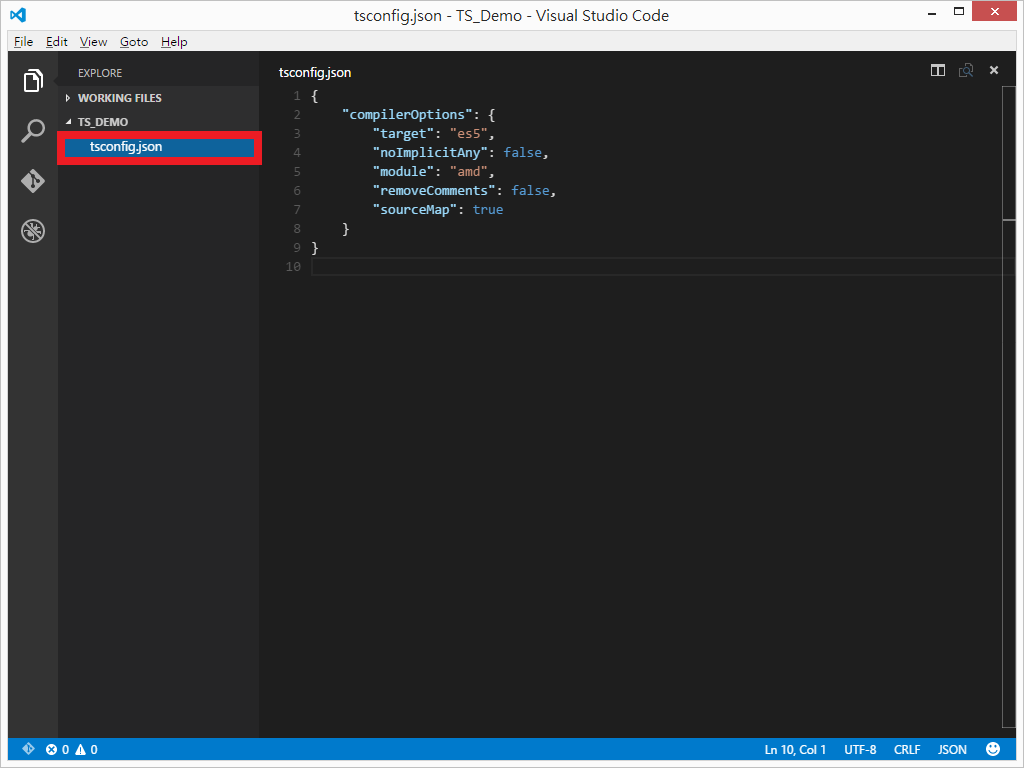
"removeComments": false,

"sourceMap": true

}

}





**建立.settings\tasks.json**

再来同样在Workspace加入一个新文件夹「.settings」，并且在这个文件夹中加入一个新档案「tasks.json」，接着输入下列JSON设定参数。

{

"version": "0.1.0",

"command": "tsc",

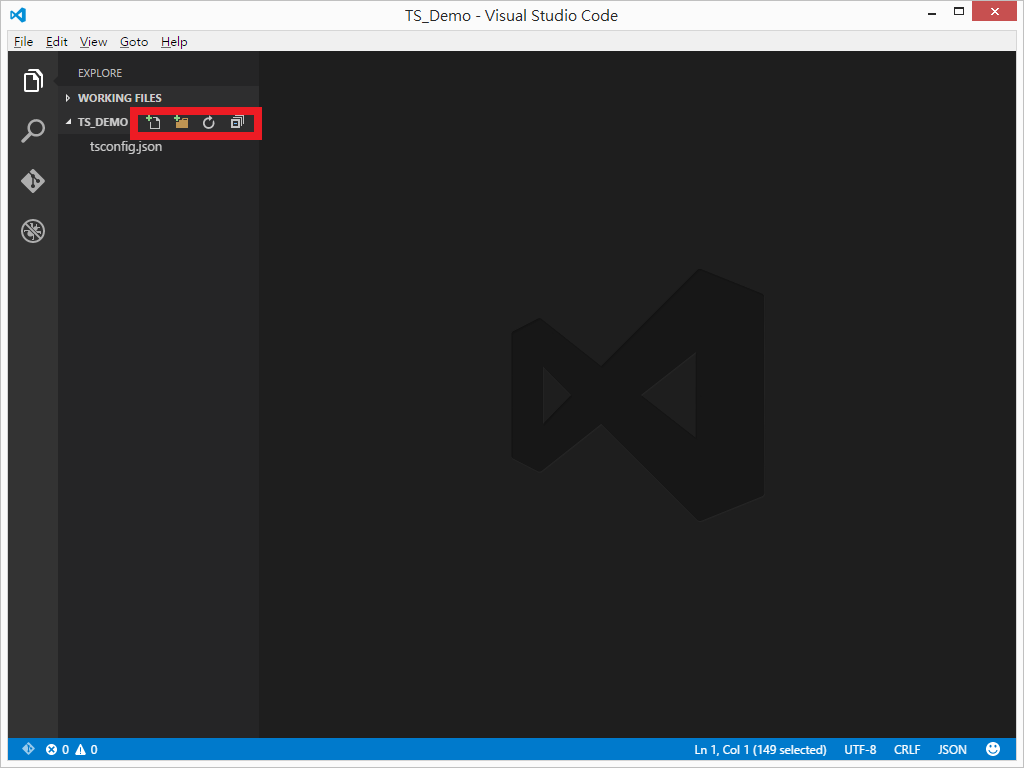
"isShellCommand": true,

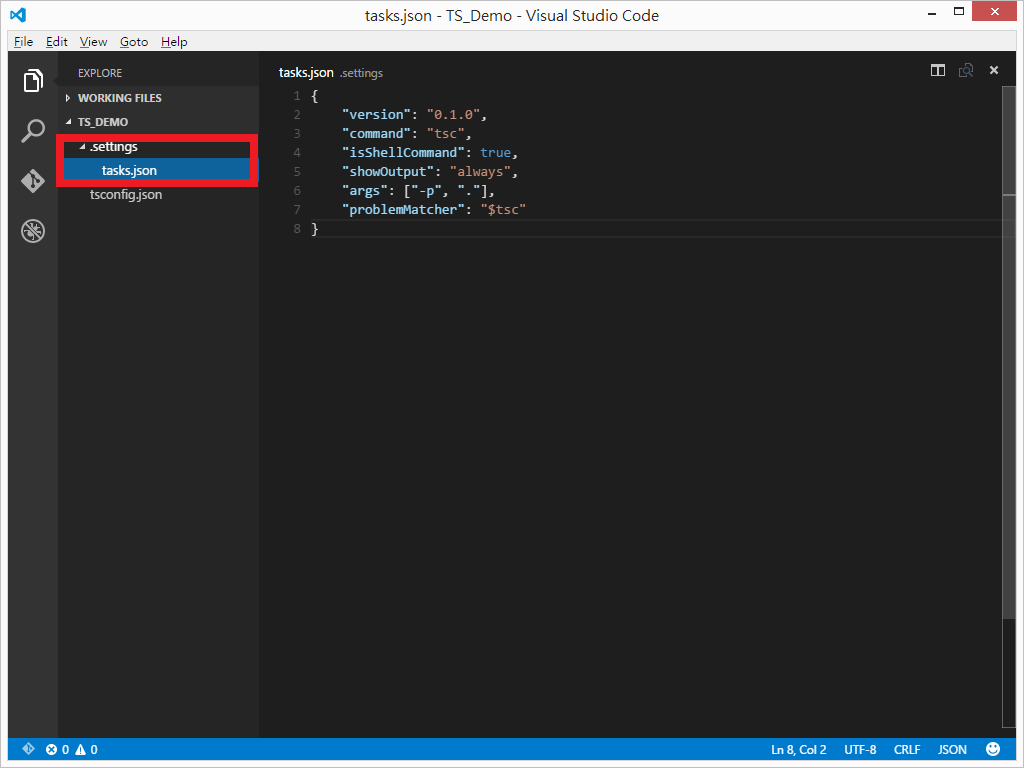
"showOutput": "always",

"args": ["-p", "."],

"problemMatcher": "$tsc"

}





**开发main.ts**

完成上述步骤后，在Workspace加入一个新档案「main.ts」，并且输入下列TypeScript程序代码。

class Greeter {

data: string;

constructor(data: string) {

this.data = data;

}

run() {

alert(this.data);

}

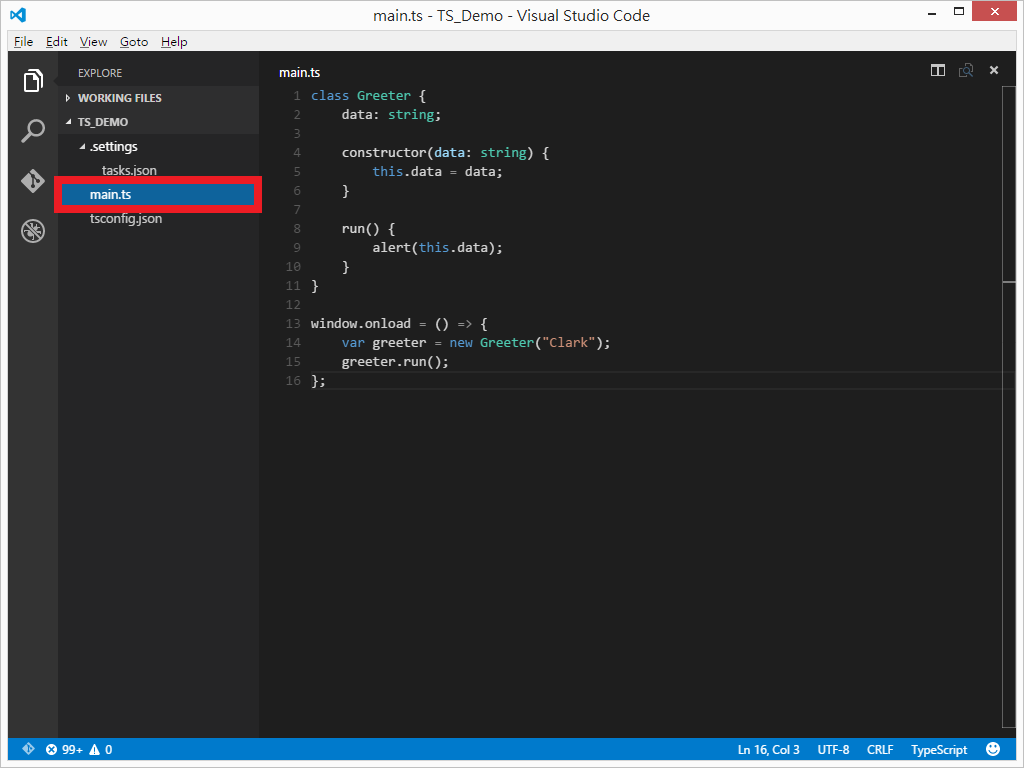
}

window.onload = () => {

var greeter = new Greeter("Clark");

greeter.run();

};



最后按下快捷键「Ctrl+Shift+B」，就可以看到Visual Studio Code编译TypeScript,并且输出对应的JavaScript档案：main.js。

var Greeter = (function () {

function Greeter(data) {

this.data = data;

}

Greeter.prototype.run = function () {

alert(this.data);

};

return Greeter;

})();

window.onload = function () {

var greeter = new Greeter("Clark");

greeter.run();

};

//# sourceMappingURL=main.js.map