

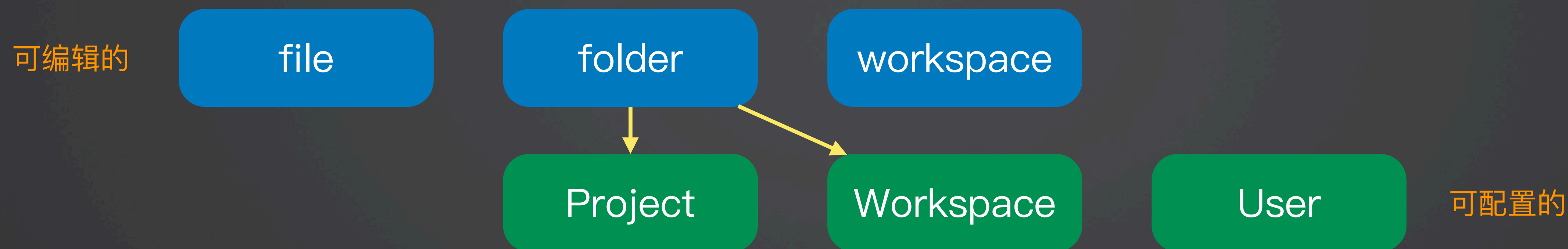
36 | 在 VSCode 中管理 TypeScript

周爱民 (Aimingoo)

目录

- 1 有哪些 TypeScript
- 2 有哪些 Node.js
- 3 以文件夹或工作空间来组织项目
- 4 常见的配置项与概念入门
- 5 总结

VSCode 基本规则



常用命令

1. 使用 tsc

```
> tsc --init ... // 初始化项目目录 (create tsconfig.json)
> tsc ... --showConfig // 不编译, 仅显示该命令行最终使用的配置
> tsc ...           // 缺省等义于 --project tsconfig.json
> tsc --project ... // 编译 (注意: 命令行上不能出现文件名)
> tsc --build ...   // 构建 (will resolve dependencies)
> tsc ... t.ts t1.ts ... // 编译入口文件 (support multi entries)
> tsc @files.txt    // 文件 @files.txt 是一个文件列表
```

2. 使用 Node.js

```
> npm ls -depth 0 // 查看npm包安装路径和已安装的包
> npm init -y // 初始化项目目录 (create package.json)
> mkdir node_modules // 手工创建模块目录 (可以在没有package.json的情况下使用Node.js)
> npm install -D typescript // 在当前项目中安装TS, 写入开发依赖 (或使用--no-save不写入)
> npm install @types/node // 为指定的库或环境安装TypeScript类型包 (示例用于支持Node.js的内建库)
```

总结

1. 注意不同环境（尤其是控制台/终端）中的node/tsc/npm是不同的
2. 注意 VSCode 的全局、工作空间和项目（当前文件夹）都可以配置各自独立的 tsc
 - 右下角指示的是“当前项目”所使用的 tsc
 - 在右下角切换 tsc 使用版本时可能导致当前项目自动生成 .vscode 目录
3. tsc 的三种基本使用模式（文件入口、编译项目 和 构建项目组）

THANKS