

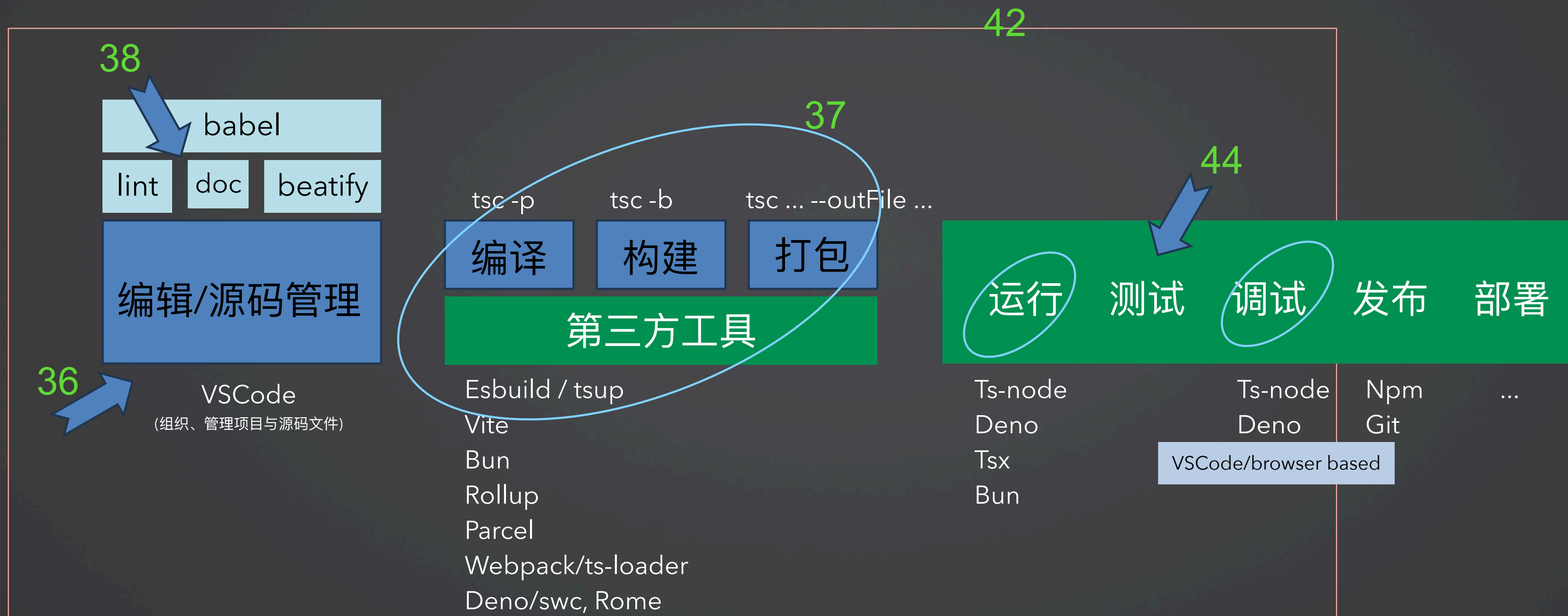
37 | 在 VSCode 环境下的 TypeScript 开发过程全解

周爱民 (Aimingoo)

目录

- 1 开发过程概要
- 2 运行环境
- 3 调试与环境配置
- 4 第三方工具的打包
- 5 总结

开发过程概要



支持TypeScript的执行器 (runtime/execute)

TS-Node - Node/V8 based

Vite-node (Vite as Node runtime, vitest sub-project)

privatenumber/tsx (Esno 1.5+)

unjs/jit

Bun – JavaScriptCore, ZIG based

Deno – V8, Rust base

edge-runtime (A server based on Deno runtime)

Elsa – QuickJs, Go based

Others

- kaleidawave/ezno 是js编译器和基于ts的类型检查器。Rust based
- ASDAlexander77/TypeScriptCompiler 是ts的编译器。C++ based, by LLVM

Project builder/bundler
ex: esbuild

Package Manage
ex: npm

Runtime + REPL.
ex: nodejs

JS Engine
ex: v8

安装

1. 安装deno, bun等

```
> curl -fsSL https://bun.sh/install | bash
> curl -fsSL https://deno.land/install.sh | sh
# (OR,)
> brew install deno oven-sh/bun/bun // 使用 brew 或其它安装工具
```

2. 安装ts-node, rollup, 以及相关插件

```
> npm install -g ts-node rollup // 安装在全局
> npm install -D typescript rollup-plugin-typescript2 // 安装在当前项目
```

3. 参考

- ▶ https://docs.deno.com/runtime/manual/getting_started/installation
- ▶ <https://bun.sh/docs/installation>
- ▶ <https://esbuild.github.io/getting-started>
- ▶ <https://typestrong.org/ts-node/docs/installation>
- ▶ <https://tsx.is/getting-started>

总结

1. TypeScript 通过 tsc 提供简单的编译、构建与打包功能，编译是基础，最为关键

```
> tsc // 缺省等义于 --project tsconfig.json
> tsc a.ts b.ts ... // 编译指定文件入口的列表
> tsc --project ... // 编译指定项目（注意：命令行上不能出现文件名）
> tsc --build xxx.json // 构建：使用 xxx.json 中的引用列表
> tsc ... --outFile ... // 打包：将文件列表（或项目）中的文件编译并打包到目标文件
```

2. VSCode 本身不具备运行和调试 .ts 的能力，需要配置

- ▶ **要点：**最终能运行的，“总是” .js 代码
- ▶ 需要安装执行器或运行环境（例如TS-Node， Bun等），并配置到 VSCode 中
- ▶ launch.json 用于配置和管理运行与测试环境

3. 在工程中通常使用第三方工具来打包

- ▶ TypeScript缺省时使用 --outFile 参数来打包，但只支持none/amd/system三种包格式
- ▶ 第三方工具可以用外部工具来支持足够复杂的打包行为和包格式
- ▶ 第三方工具总是基于 tsc 编译的

作业

1. 课外阅读：关于 TypeScript 的打包行为，了解第三方的打包工具

@see: 常见的6种打包器: <https://blog.csdn.net/olderandyanger/article/details/134883499>

2. 课外阅读：进一步了解如何在浏览器中调试 TypeScript

@see: <https://vscode.github.net.cn/docs/nodejs/browser-debugging>

3. 推荐了解如下工具的使用：TS-Node, ESBuild, Bun/Tsu

THANKS