

41 | TypeScript扩展了哪些应用特性

周爱民 (Aimingoo)

目录

1 回顾

2 兼容性：规范或环境兼容，js 兼容，以及对 jsx 等的兼容

3 声明文件

4 其它特性

5 总结

之前课程中相关内容的回顾

1. 对 js 进行核心类型概念的扩展

- ▶ 与一般类型相关的概念：类型标注、类型断言(as)
- ▶ 与函数相关的类型签名：函数签名、构造签名、重载签名；断言签名、谓词签名；装饰器 (第43讲)
- ▶ 类声明中的扩展语法：构造参数、可见性修饰符、只读 (readonly) 与覆盖 (override) 等
- ▶ 与泛型相关的扩展语法：泛型函数、泛型类

2. 在 js 在应用的新类型

- ▶ 元组 (and readonly array)
- ▶ 枚举 (object map)
- ▶ 名字空间 (函数的静态成员)

其它应用特性

- 配置、编译与转换 (palyground)
 - TypeScript 4.0+ 的特殊支持
 - --target, --lib
- 对 .js 文件的支持
 - --allowJs
 - --checkJs
- 模块
 - --module
 - configurations of module in tsconfig.json
- 类型声明文件 (.d.ts)
 - declaration
 - declarationMap
 - declarationDir
 - emitDeclarationOnly
- 开发支持
 - 外部开发环境集成 (Node.js, Browser / Editor Support)
 - 运行库定制 (lib / noLib / *LibCheck / jsx* / reactNamespace / ...)
 - watch

H: TypeScript应用体验

41 TypeScript扩展了哪些应用特性

42 管理与组织TypeScript文件与工具套件 ⑨

43 装饰器的用法

44 为TypeScript代码写测试用例

45 配置基础概念之模块：解析，目标和映射

46 配置选项详解 - Node.js集成

47 配置选项详解 - 浏览器输出

48 配置选项详解 - 第三方工具集成

49 配置选项详解 - 其它编译选项与环境

总结

1. TypeScript 扩展了JavaScript 的语言能力、兼容性

- ▶ TypeScript “总是会” 提前实现一些 ECMAScript 未发布最终规范的特性，但（通常）不会实现那些早于 Stage 3 的提案
- ▶ TypeScript 扩展 JavaScript 的关键配置是 lib 和 target，它们决定了最终运行环境
- ▶ TypeScript 总是编译输出目标平台（最终运行环境）所支持的 .js 代码

2. TypeScript 扩展的应用特性

- ▶ 编辑器集成
- ▶ 运行环境的支持
- ▶ 插件

3.3. 参考

- ▶ TS Playground 演练场 @see <https://www.typescriptlang.org/play>
- ▶ tsconfig.json 配置手册 @see <https://www.typescriptlang.org/tsconfig>

THANKS