TypeScript compiler (tsc)

Typed JS compiler

CONCEPTS

> tsc finds tsconfig.json by walking up from the current directory. Naming files on the command line makes it refuse to run unless you add --ignoreConfig.

tsconfig.json options + files
↓ tsc
Type check errors, exit code 1
↓ emit
JavaScript outDir

> Type errors do not stop emit unless noEmitOnError is set. A non-zero exit code only tells you there were errors.

PROJECT SETUP

`tsc --init`
Create a tsconfig.json

`tsc`
Compile using the nearest tsconfig.json

`tsc -p <path>`
Compile a config file or directory

`tsc --noEmit`
Type-check without writing files

`tsc --watch`
Recompile when files change

`tsc --ignoreConfig <file>`
Compile given files, ignore tsconfig

`tsc --version`
Show the compiler version

`tsc --help --all`
List every compiler option

TSCONFIG KEYS

`"extends": "<file>"`
Inherit another config

`"include": ["src"]`
Files to compile

`"exclude": ["dist"]`
Files to skip

`"compilerOptions": {}`
Where all options below go

LANGUAGE & MODULES

`--target esnext`
Syntax level of emitted JS

`--module nodenext`
Module format for Node projects

`--module preserve`
Keep imports as written (bundlers)

`--moduleResolution bundler`
Resolution for bundler setups

`--lib <list>`
Built-in type libraries to include

`--types <list>`
Which @types packages to load

`--jsx react-jsx`
How JSX is compiled

`"paths": {"@/*": ["./src/*"]}`
Alias import prefixes

TYPE CHECKING

`--strict`
Enable all strict checks

`--noUncheckedIndexedAccess`
Indexed access may be undefined

`--skipLibCheck`
Skip checking .d.ts files

EMIT

`--outDir <dir>`
Write output files here

`--rootDir <dir>`
Root of the input tree

`--declaration`
Emit .d.ts files

`--sourceMap`
Emit .js.map files

`--verbatimModuleSyntax`
Keep imports/exports as written

PROJECTS & BUILDS

`tsc --build`
Build projects whose inputs changed

DEBUGGING

`tsc --showConfig`
Print the final merged config