tsx Run TypeScript directly

CONCEPTS

> tsx is node with a TypeScript loader: it strips types with esbuild and never type-checks, so run tsc --noEmit for errors.

file.ts your source
↓ esbuild strips types
tsx loader on top of node
↓ JS, no type check
node runs the code
↓ tsc --noEmit
Type errors found by tsc only

> It handles ESM and CommonJS, .ts/.tsx/.mts/.cts files and tsconfig paths in one go, with no build step.

INSTALL

npx tsx *<file.ts>*
Run once without installing

npm install -D tsx
Add to the project

npm install -g tsx
Install globally

tsx --version
Show tsx and node versions

RUN

tsx *<file.ts>*
Run a TypeScript file

tsx *<file.ts> <args>*
Pass arguments to the script

tsx -e "*<code>***"**
Evaluate a TS snippet

tsx -p "*<expr>***"**
Evaluate and print the result

RUN (cont.)

tsx --env-file=*<file> <file.ts>*
Load env vars from a file

tsx --no-cache *<file.ts>*
Skip the compile cache

> Any node flag goes before the script name and is passed to node.

WATCH

tsx watch *<file.ts>*
Rerun when imported files change

tsx watch --include "*<glob>***"** *<file.ts>*
Also watch extra paths

tsx watch --exclude "*<glob>***"** *<file.ts>*
Do not watch these paths

tsx watch --clear-screen=false *<file.ts>*
Keep earlier output

Enter (in watch mode)
Rerun by hand

> Watch ignores node_modules, dist, vendor and hidden folders by default; add them with --include.

CONFIG

tsx --tsconfig *<path> <file.ts>*
Use a specific tsconfig

USE WITH NODE

node --import tsx *<file.ts>*
Run TS with plain node

node --import tsx --test
Run the node test runner on TS

tsx --test
Same, shorter