# TypeScript usage explained

## How TypeScript is related to JavaScript/ECMAScript and what steps are needed

2.10.2024


Juhani Välimäki

# Installing TypeScript to your computer

- Assuming you have Node.js already installed to your computer, continue by installing TypeScript, e.g. this might be the correct command:
- npm i --save -g typescript @types/node

# Starting to turn JavaScript project to TypeScript

- …

# Initialize the project as TS project

- tsc init
- => tsconfig.json

# tsconfig.json

- E.g.
- source folder (for TS)
- Output/dist folder (for compiled JS files)
- How strict TypeScript required/followed?
- What version of ES should the output be?

# Package.json npm run/build etc scripts changed

- …from using JS tools, to use TS tools like compiler
- E.g. ts-watch could look for changing .ts files and compile them to .js

# Renaming source files from .js to .ts

- Beeee

# Run some checker that forces to use TS features

- E.g. biome

# Run some checker that forces to use TS features

- E.g. biome

# Install the TS versions of libraries, with their type definition modules

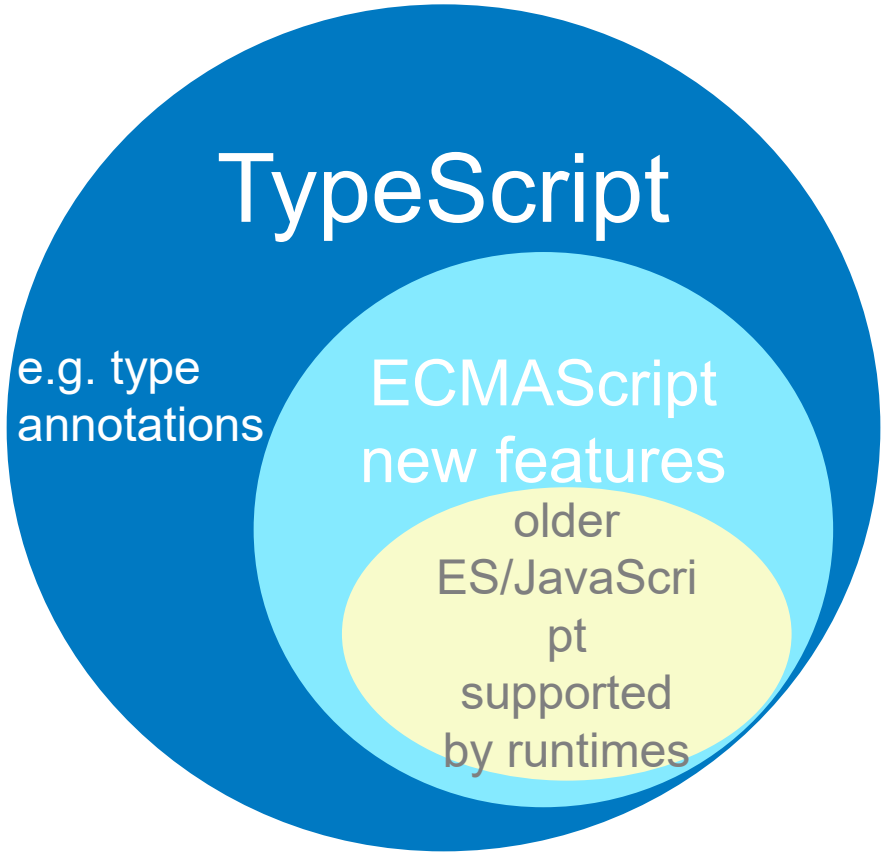
- E.g.
- npm i --save express @types/node @types/react @types/react-dom @types/jest

# Understand compilation-time vs run-time

- - 1. Compilation time:  tsc (TypeScript build) .ts->.j
- - 2. Runtime:  run the .js

- See also how all TS tools are in DevDependencies in package.json, where as JS tools and modules are for running time

# Principles as a picture



# tsc compilation as the image