



TYPESCRIPT & FLOW

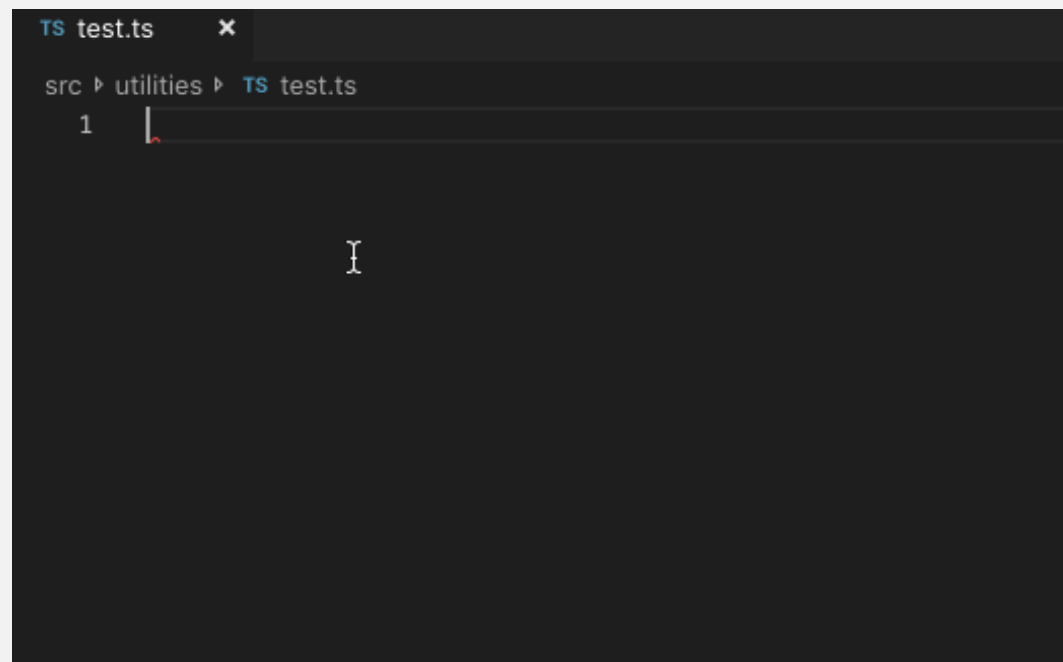


Comparing static type systems in JS

By [@chrisdopuch](#)
Software engineer - [Zapier Inc.](#)

INTRODUCTION

- Intended audience
 - New to static type systems in JS, wondering which one to try
 - Familiar with type systems looking to just compare or migrate your codebase
- These are just my subjective opinions and experiences
- So why type things statically anyways?
 - Catch bugs and refactor code easily
 - Better documentation and editor support
 - Write fewer tests and type assertions



The image shows a screenshot of a code editor window. The title bar at the top reads "TS test.ts" with a close button on the right. Below the title bar, the breadcrumb navigation shows "src > utilities > TS test.ts". The main editing area is dark-themed and contains a single line of code, "1", with a vertical cursor positioned at the end of the line. A small red squiggly line is visible under the "1", indicating a linting error. A white cursor icon is also visible in the lower right portion of the editor area.

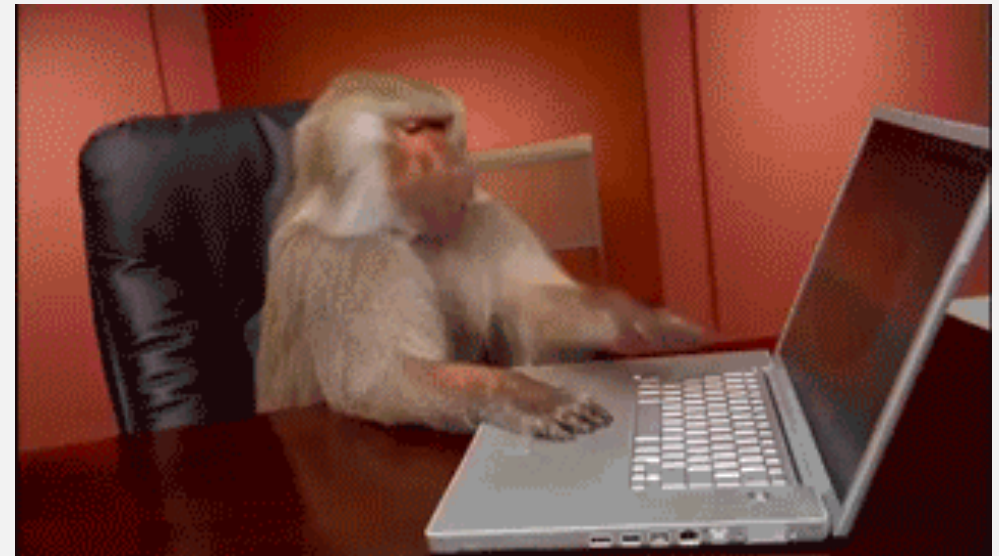
LANGUAGE FEATURES



- Both Flow and TypeScript provide excellent type checking features
 - [Deeper dive](#) on language differences, but the differences are pretty small
- Flow-typed is a static type checker for JavaScript, by Facebook
 - Added to a JS file with an `//@flow` pragma
 - Babel plugin + CLI tool for type checking
- TypeScript is a statically typed super-set of JavaScript, by Microsoft
 - `.ts` files, compiles down to `.js` files (type checking occurs at build time)
 - Can work with Babel 7 build chain, or the standalone compiler TSC

TOOLING AND SUPPORT

- TypeScript language server is very fast and reliable
- Flow language server by comparison is slower and less reliable
- Flow's release cycle includes breaking changes and sometimes difficult upgrades
 - Tightly coupled to the version of React
 - Still on 0.x.x series, by comparison TypeScript is on the 3.x.x series
- TypeScript ecosystem more isolated, but Babel 7 support may change this
- IDE integrations for TypeScript tend to be speedier and more usable
 - Especially VSCode, but many editors are well-supported



COMMUNITY



- TypeScript community in 2019 is an order of magnitude larger than Flow
 - [facebook/flow](#) is used by **48,588** repositories, while [microsoft/TypeScript](#) is used by **1,204,178**
 - TS [DefinitelyTyped](#) repo contains about **5,800** library definitions, while the [Flow-Typed](#) repo only has around **800**.
- Landmark adoption cases
 - Angular has pushed TypeScript since Angular 4.0
 - Facebook's Jest project recently migrated off Flow onto TypeScript
 - Apollo (the GraphQL client) has stopped supporting Flow, only TS

CONCLUSIONS

- In 2019, TypeScript is poised to win the battle over typed JavaScript
- Flow may still have a place if issues are addressed and the community grows
- TypeScript is a great choice for larger and more complex applications
- Flow's ease of getting started still makes it a solid choice to learn the ropes
- Let your type system work for you, not the other way around



QUESTIONS AND COMMENTS?

Thank you for listening!