Chapter 13. Conclusion

We’re nearing the end of our journey together.

We’ve covered what types are and why they’re useful; how TSC works; what types TypeScript supports; how TypeScript’s type system handles inference, assignability, refinement, widening, and totality; the rules of contextual typing; how variance works; and how to use type operators. We’ve covered functions and classes and interfaces, iterators and iterables and generators, overloads, polymorphic types, mixins, decorators, and the various escape hatches you can use once in a while to sacrifice safety to get your code out before your deadline. We explored the different ways to handle exceptions safely and their trade-offs, and how to use types to make concurrent, parallel, and asynchronous programs safe. We dove into using TypeScript with popular frameworks like Angular and React, and how namespaces and modules work. We looked at using, building, and deploying TypeScript on the frontend and on the backend, and talked about how to gradually migrate code to TypeScript, how to use type declarations, how to publish your code to NPM so others can use it, how to safely use third-party code, and how to build your TypeScript projects.

I hope that I’ve infected you with the gospel of static types. I hope you now, at times, find yourself sketching out programs in types before implementing them, and I hope you’ve gained a deeply intuitive understanding of how you can use types to make your applications safer. I hope that I’ve changed your view of the world, at least a little bit, and that you now think in types when you write code.

You’re now equipped to teach others about TypeScript. Advocate for safety, and help make your company’s and your friends’ code better and more fun to write.

Finally, keep exploring. TypeScript probably isn’t your first language, and it probably won’t be your last. Keep learning about new ways to program, new ways to think about types, and new ways to think about the trade-offs between safety and ease of use. Maybe you’ll create the next big thing after TypeScript, and maybe I’ll be the one to write about it someday…