

What makes a TypeScript wizard?

a cheat sheet from Total TypeScript www.totaltypescript.com

Knowledge

Narrowing

Mutability

Open vs Closed Objects

Assignability

Strictness

Types vs Interfaces

Deriving vs Declaring

Enums

Distributivity

The Global Type Scope

Skills

IDE Tricks

Errors

Declaration Files

Developer Pipelines

any

Discriminated Unions

Utility Types

Annotations and Assertions

Type Transformations

Generics

Results

NPM Libraries

Unblocking Teammates

Rapid Development

What Bugs?

Type-Driven Development

Beautiful Codebases

A TypeScript Wizard Knows...

Narrowing

Wizards know every narrowing trick in the book. They understand the things TypeScript can't narrow, like **Boolean** and **Map.has/Map.get**.

Mutability

Wizards understand that **let** infers differently from **const**, what **as const** does, and how **readonly** properties don't really work (but **readonly** arrays do!).

Open vs Closed Objects

Wizards understand that TypeScript has open objects, not closed ones - and how this affects **Object.keys** and excess property checks.

Assignability

Wizards instinctively know when one type will be assignable to another. They know that **unknown** is the top type, and **never** is the bottom type.

Strictness

Wizards know how to dial the strictness up and down. They know the pro's and cons of function return types.

Types vs Interfaces

Wizards know that the real decision isn't between types and interfaces, but between interfaces and intersections.

Deriving vs Declaring

Wizards know the trade-offs between derived types (creating the type based on the shape of another), and redeclaring them. They can make beautiful, self-maintaining types - or they can keep it simple.

Enums

Wizards know their opinions on enums - one of TypeScript's most controversial features. They know which enums are good, and which are bad.

Distributivity

Wizards understand how some generic types automatically distribute over unions, and others don't (like **Omit**).

The Global Type Scope

Wizards understand the difference between modules and scripts, and how you can use them to type the global scope.

A TypeScript Wizard Can Do...

IDE Tricks

Wizards know every IDE trick in the book. Autocomplete, refactoring, introspection, JSDoc comments, renaming symbols.

Errors

Wizards read TypeScript errors like it's the Matrix. They know how to fix them, and how to avoid them in the first place.

Declaration Files

Wizards aren't scared of **node_modules**. They understand declaration files, and how to use go-to-definition to find their way around.

Developer Pipelines

Wizards know the best way to put TypeScript in the dev pipeline. They understand how to run TS on CI, and how to keep TS out of their way on the dev server.

any

Wizards aren't scared of **any**. They know to avoid it, but also when to judiciously apply it to keep TypeScript quiet.

Generics

Wizards know that there's no such thing as a generic. And they know the difference between type arguments, type parameters.

Utility Types

Wizards know the right utility type for the right situation. They know the difference between **Omit** vs **Exclude**, and **Extract** vs **Pick**.

Annotations and Assertions

Wizards know the trade-offs between **satisfies**, **as**, and variable annotations. They know which ones are safe, and which are risky.

Type Transformations

Wizards know how to transform any type to any other type. They can use mapped types, template literal types, conditional types and more to make powerful type utilities.

Discriminated Unions

Wizards know that discriminated unions let you model complex data structures with ease, from events to `Result` types and much more.

A TypeScript Wizard Achieves...

NPM Libraries

Wizards know how to publish npm libraries, make beautiful type-safe API's, and abstract away hard problems.

Unblocking Teammates

Wizards are the first people pinged on Slack when someone's stuck on a TypeScript problem. They might not even be seniors, but they can unblock anyone.

Rapid Development

Wizards can build features faster at record speed. TypeScript doesn't slow them down, it speeds them up.

What Bugs?

Wizards know that TypeScript prevents a whole class of errors. Wizards write simpler, more robust code.

Type-Driven Development

Wizards let their types drive their development. They know that types aren't a bolt-on after you've finished the runtime code.

Beautiful Codebases

Wizards write simple, beautiful code. They aren't scared of a bit of complexity if it makes other things more maintainable.

a cheat sheet from **Total TypeScript** www.totaltypescript.com