HOW SOFTWARE WORKS · BUILDER · 13 MIN READ

TypeScript basics: types and reading the errors

What types are, the handful of TypeScript you'll see in AI-built apps, and how to read the type errors that stop a deploy.

Pass the check (4 of 5) online to earn the “Type Checker” badge:
https://digitalbuilders757.com/university/study/typescript-basics-read-the-errors

Why TypeScript

TypeScript is JavaScript with types: labels that say what kind of value something holds. A type checker reads your code before it runs and flags mistakes like a misspelled field or text where a number belongs.

Browsers run plain JavaScript, so TypeScript is checked and then turned into JavaScript during the build. Most AI app builders write TypeScript (files ending in .ts and .tsx) because the checks catch the AI's slips too. When a deploy fails with 'Type error', this is what's talking.

The types you'll see most

- Basic types: string (text), number, boolean (true or false), and arrays like string[] (a list of text).
- Object shapes: type User = { name: string; email: string; age?: number }. The ? means optional: age may be missing, so it can be undefined.
- Unions: type Status = "draft" | "sent" | "paid" allows only those three values, and string | null means 'some text, or nothing'.
- Functions: function total(prices: number[]): number says what goes in (a list of numbers) and what comes out (a number).
- any switches checking off for a value. AI tools sometimes add it to make an error go away. Treat it as a warning sign, not a fix.

Reading a type error

A type error names the file and line, then the mismatch. Read it as 'you gave X, but Y was expected'.

Type 'string' is not assignable to type 'number' means a value is text where the type says number, often a form field: convert it with Number(). Property 'titel' does not exist on type

'Post' usually means a typo, or a field the type doesn't define. Object is possibly 'undefined' means the value might not exist yet: check it first with if (user), or read it safely with user?.name and give a fallback with ??, as in user?.name ?? 'Guest'.

Fixing type errors with AI

- Run the type check before you push (your build, or npx tsc --noEmit) so the deploy doesn't fail in front of everyone.
- Paste the full error and the code around it. Ask for the cause first, then the smallest fix.
- Be suspicious of fixes that add any, as unknown as, or // @ts-ignore. They hide the problem instead of solving it.
- If a type and the database disagree, fix the source of truth (the type or the column), not just the line that errors.

Key terms

TypeScript
JavaScript with types that catch mistakes before the code runs.

Type (TypeScript)
A label for the kind of value something holds, like string, number or a User object.

Union type
A type that allows one of several options, like "draft" | "sent" | "paid", or string | null.

Optional chaining (?.)
Reads a property only if the thing before it exists; otherwise gives undefined instead of crashing.

any (TypeScript)
A type that switches checking off for a value. In AI-written code it often hides a real mistake.

JavaScript
The programming language that makes web pages interactive.

Deploy
Publish a version of your app to a host.

Debugging
Finding and fixing bugs, often by reading the exact error message.

Try it: Fix three type errors

1. Ask an AI for a tiny TypeScript file with a User type (name, email, optional age) and a function that greets a user.
2. Paste it into the TypeScript Playground (typescriptlang.org/play) and hover over names to see their types.
3. Make three mistakes on purpose: misspell a field, pass a number as text, and read age without checking it.
4. Read each error out loud as 'you gave X, but Y was expected', then fix each one without using any.

KNOWLEDGE CHECK

TypeScript basics: types and reading the errors

5 questions. Circle one answer for each; 4 or more correct passes. Answers are at the back.

1. What does TypeScript add to JavaScript?

A. A faster engine, so apps run twice as quickly in the browser
B. Types that are checked before the code runs
C. Automatic tests that click through every page for you
D. A new language that browsers run instead of JavaScript

2. In type User = { name: string; age?: number }, what does the ? mean?

A. age is a question the app asks the user
B. age must be a whole number
C. age is optional, so it may be undefined
D. age is private and hidden from the other files in the project

3. The build fails: Type 'string' is not assignable to type 'number'. What does it mean?

A. The text is too long to fit in a number field, so shorten it
B. Numbers aren't allowed in TypeScript
C. The server is down
D. A value is text where a number was expected

4. An AI 'fixes' a type error by adding as any. What's the problem?

A. It hides the mistake instead of fixing it
B. Nothing: any is the recommended way to fix type errors
C. any only works in the browser, so the server will crash
D. any makes the app slower for every visitor

5. TypeScript says Object is possibly 'undefined' at user.name. A good fix?

A. Delete the type so TypeScript stops checking it entirely
B. Check it first, or use user?.name with a fallback
C. Add // @ts-ignore above the line
D. Rename user to something shorter

Official sites

TypeScript Handbook: Everyday types:
https://www.typescriptlang.org/docs/handbook/2/everyday-types.html
TypeScript for JavaScript programmers:
https://www.typescriptlang.org/docs/handbook/typescript-in-5-minutes.html

Last reviewed 2026-11-25. AI tools change fast; check each tool's own site for current features, plans and prices.

ANSWER KEY

TypeScript basics: types and reading the errors

1. B: Types that are checked before the code runs
The types are checked during development and the build, then removed: browsers still run plain JavaScript.

2. C: age is optional, so it may be undefined
Optional fields can be missing, so TypeScript makes you check before using them.

3. D: A value is text where a number was expected
Often it's a form value, which is always text. Convert it with Number() before using it as a number.

4. A: It hides the mistake instead of fixing it
any turns checking off, so the bug the error pointed at is still there, just quieter. Ask for the cause and a real fix.

5. B: Check it first, or use user?.name with a fallback
if (user) or user?.name ?? 'Guest' handles the missing case instead of hiding it.