FROM RUBY TO TYPESCRIPT

A Ruby Programmer's Guide to Learning TypeScript

```
{
 : string
 : number
  interface
}
```

Joel Bryan Juliano

From Ruby to TypeScript

Joel Bryan Juliano

This book is available at https://leanpub.com/rb2ts

This version was published on 2026-05-28

Contents

Preface

I started writing Ruby in 2010. For over a decade, it was my go-to language – the one I reached for when I needed to think through a problem, build a prototype, or ship a feature. Ruby's slogan is "A Programmer's Best Friend," and I still believe it.

My serious TypeScript work began in 2015, when React was gaining popularity and suddenly the frontend needed types. I've used it across the full stack ever since – React on the frontend, Node.js on the backend – eleven years and counting.

In 2018, I started writing *From Ruby to Golang*, and published it in 2019, a book that taught Go by mapping every concept back to Ruby. It hit #3 on Amazon. Rubyists told me the Ruby-first approach was the fastest way they'd ever learned a new language. That book proved something: the best teacher for a Rubyist is Ruby itself.

This book applies the same method to TypeScript. It sat on my Leanpub dashboard for years as an unpublished draft – notes, code samples, Ruby-to-TypeScript mappings accumulating chapter by chapter – while I refined the approach that started with Go. The pattern is the same because the truth is the same: if you know Ruby by heart, the fastest way to learn a new language is through Ruby.

TypeScript is everywhere. It powers the frontend (React, Next.js, Angular), the backend (Node.js, Deno, Bun), and everything in between (VS Code extensions, CLI tools, serverless functions). If you're building for the web in 2026, you're building with TypeScript. And if you're a Rubyist who wants to go full-stack – write the backend in Ruby *and* the frontend in TypeScript, or migrate services from Rails to Node.js – you need to speak both languages fluently.

The first thing I did when learning TypeScript was relate everything back to Ruby. How do I define a class? (Almost the same – but with type annotations.) How do I handle errors? (`try/catch` looks a lot like `begin/rescue`.) How do I iterate over an array? (`.map` and `.filter` are built in – they *are* Ruby's Enumerable.) How do I do concurrency? (`async/await` is a whole new world

– Ruby has nothing quite like it.) Every answer was a small revelation about how TypeScript thinks differently, and where it thinks exactly the same.

This book teaches TypeScript by mapping every concept back to Ruby. If you know how instance variables work in Ruby, you'll learn how class properties work in TypeScript. If you know `require`, you'll learn `import`. If you know `begin/rescue/ensure`, you'll learn `try/catch/finally`. No abstract explanations – just the Ruby you know, translated to the TypeScript you want to learn.

TypeScript is a remarkable language. It adds a powerful static type system to JavaScript without losing JavaScript's flexibility. It compiles to clean, readable JavaScript that runs everywhere. It catches entire categories of bugs before your code ever runs – misspelled properties, wrong argument types, null references. The type system is so expressive that it can model things Ruby's duck typing never could: discriminated unions, exhaustive switch checks, and type-level programming that borders on magic.

But more importantly: TypeScript will make you a better programmer. Learning to think in types, to model your data precisely, to let the compiler catch your mistakes before your users do – these patterns will change how you write code in any language, including Ruby.

This book doesn't ask you to take notes or memorize syntax. Just read, type the examples, and let your Ruby knowledge do the heavy lifting.

Ruby is still my favorite language for quick scripts, CLIs, and backend services. I use it every day. This book is not about leaving Ruby behind – it's about adding TypeScript to your toolkit, so you can work confidently across the entire stack.

Let's get started.

Introduction

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2ts.

Who This Book Is For

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2ts.

How to Use This Book

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2ts.

What This Book Is Not

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2ts.

A Note on Runtime

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2ts.

About the Author

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2ts.

Chapter 1: Getting Started

Every TypeScript program needs three things: a runtime (Node.js), a way to execute TypeScript code, and an entry point. If you can set those up and write your first `console.log`, you're on your way. Let's get you running in under five minutes.

Installing Node.js and a TypeScript Runner

TypeScript compiles to JavaScript, so you need a JavaScript runtime. Node.js is the standard. Check if you have it:

```
$ node --version
v22.11.0
```

If you don't, install it from nodejs.org or via a version manager like `nvm` or `fnm`.

You also need a way to run TypeScript files without manually compiling them. This book uses `tsx` — it's fast, modern, and handles TypeScript out of the box:

```
$ npm install -g tsx
```

Now you can run `.ts` files directly, just like `ruby file.rb`:

```
$ tsx hello.ts
```

If you prefer `ts-node`, it works too. The examples in this book work with either.

Verify everything is wired up:

```
$ node --version
v22.11.0
$ tsx --version
tsx v4.19.0
```

If both commands print version numbers, you're ready. Let's write some code.

Your First TypeScript Program

Create a file called `hello.ts`:

```
console.log("Hello World");
```

Run it:

```
$ tsx hello.ts
Hello World
```

That's it. `console.log` is TypeScript's `puts` — it prints a line to the console. Unlike Ruby, where `puts` is available everywhere, `console.log` is a method on the global `console` object. But you'll never need to import it — it's available globally in Node.js and browsers.

Template Literals: String Interpolation

In Ruby, you interpolate with `#{}`:

```
name = "World"
puts "Hello #{name}!"
```

In TypeScript, you use backticks and `${}`:

```
const name = "World";
console.log(`Hello ${name}!`);
```

The backtick syntax is called a template literal. It does everything Ruby's double-quoted strings do: interpolation, multiline strings, and expressions:

```
const a = 10;
const b = 20;
console.log(`${a} + ${b} = ${a + b}`);
// 10 + 20 = 30
```

No more `"The number is " + value.to_s` — template literals handle any expression inside `${}`.

Variables: const and let

Ruby has variables without type declarations:

```
name = "Joel"
age = 40
```

TypeScript gives you two ways to declare variables:

```
const name = "Joel";   // cannot be reassigned
let age = 40;          // can be reassigned
```

Key Point: Default to `const`. Use `let` only when you need to reassign. This one habit eliminates an entire category of bugs — accidental mutation.

`const` is not a deep freeze (the object can still mutate), but the variable binding is permanent. Think of it like a Ruby frozen variable reference.

You'll also see `var` in older code. Don't use it. `var` has function scope and hoisting behavior that causes subtle bugs. `const` and `let` have block scope — the way you'd expect from Ruby.

Type Annotations (Your First Taste)

Here's the thing that makes TypeScript different from Ruby: you can tell the compiler what type a variable should be:

```
const name: string = "Joel";
let age: number = 40;
const isDeveloper: boolean = true;
```

The `: string` part is a type annotation. It says "this variable must always be a string." If you try to assign a number:

```
let name: string = "Joel";
name = 42;
// Type 'number' is not assignable to type 'string'.
```

The compiler catches it before your code runs. Ruby would let this slide and fail at runtime when something tried to call `.upcase` on a number.

Key Point: Type annotations are optional. TypeScript infers types from the value you assign. `const name = "Joel"` infers `string` automatically. Add annotations when the inference isn't enough — or when you want to be explicit at API boundaries.

Functions

TypeScript functions look familiar. Ruby:

```
def greet(name)
  "Hello, #{name}!"
end

puts greet("World")
```

TypeScript:

```typescript
function greet(name: string): string {
  return `Hello, ${name}!`;
}

console.log(greet("World"));
```

TypeScript functions are the building blocks of your program — you'll write a lot of them.

Differences from Ruby:

- Parameters need type annotations: `name: string`
- The return type goes after the parameter list: `: string`
- The function body is in curly braces
- `return` is explicit — no implicit returns

You can skip the return type annotation and TypeScript infers it:

```typescript
function greet(name: string) {
  return `Hello, ${name}!`;
}
```

TypeScript sees you're returning a string and infers the return type. But on exported functions and public APIs, explicit return types are good practice — they act as documentation and catch implementation errors.

Arrow Functions

TypeScript has a second function syntax: arrow functions. They're the closest thing to Ruby blocks and lambdas:

```ruby
# Ruby lambda
greet = ->(name) { "Hello, #{name}!" }

# Ruby block
["Alice", "Bob"].each { |name| puts greet.call(name) }
```

```typescript
// TypeScript arrow function
const greet = (name: string): string => `Hello, ${name}!`;

// Used like a Ruby block
["Alice", "Bob"].forEach(name => console.log(greet(name)));
```

Arrow functions are concise, inherit `this` from their enclosing scope, and are the standard for callbacks. We'll cover them in depth in Chapter 5. For now, know that `function greet() {}` and `const greet = () => {}` are both valid — and you'll see both in real TypeScript codebases.

The Entry Point

In Ruby, execution starts at the top of the file:

```ruby
# program.rb
def main
  puts "Starting..."
end

main if __FILE__ == $0
```

In TypeScript, execution also starts at the top of the file. There's no required `main` function — Node.js executes your file from top to bottom:

```typescript
// program.ts
function main(): void {
  console.log("Starting...");
}

main();
```

The `void` return type means "this function returns nothing." It's equivalent to a Ruby method whose return value you ignore.

For larger applications, you'll often see an async entry point:

```
async function main(): Promise<void> {
  console.log("Starting...");
  // database calls, API requests, etc.
}

main().catch(console.error);
```

We'll explain `async`, `Promise`, and `catch` in Chapter 10. For now, the `main()` pattern is enough.

tsconfig.json: Project Configuration

Every TypeScript project needs a `tsconfig.json` – the equivalent of a `.rubocop.yml` mixed with Gemfile-level project settings. It tells the TypeScript compiler how to process your code.

Create one in your project root:

```
{
  "compilerOptions": {
    "target": "ES2022",
    "module": "ESNext",
    "moduleResolution": "bundler",
    "strict": true,
    "esModuleInterop": true,
    "skipLibCheck": true,
    "outDir": "dist",
    "rootDir": "src"
  },
  "include": ["src"]
}
```

The key settings:

- **strict**: Enables all type-checking flags. Always use this for new projects. It's the equivalent of `# frozen_string_literal: true` – a gate that prevents whole classes of bugs.
- **target**: Which JavaScript version to compile to. `ES2022` is safe for Node.js 18+.
- **module**: How imports/exports are compiled. `ESNext` keeps native ES modules.

Run `npx tsc --init` to generate a default `tsconfig.json` with all available options commented out. For now, the config above is all you need.

Chapter Exercises

1. Install Node.js and `tsx`. Write a program that prints your name, your favorite programming language, and the current time (`new Date().toISOString()`). Use `console.log`, template literals, and at least one type annotation.

2. Write a `greet` function that takes a `name: string` and returns a greeting. Call it from a `main()` function. Then change the parameter type to `number` — what does the compiler tell you when you pass a string? This is your first compiler error, and it just saved you from a runtime bug.

3. Create a `tsconfig.json` with `strict: true`. Write a program that declares a variable without a type annotation and assigns a string. Hover over the variable in VS Code (or run `npx tsc --noEmit`) — TypeScript infers the type. This inference is why you don't need to annotate everything.

Summary

You can now write and run TypeScript programs. `console.log` replaces `puts`. Template literals (`` `Hello ${name}` ``) replace string interpolation. `const` and `let` replace Ruby's untyped variables — and TypeScript infers types from your values so you don't have to annotate everything.

You learned two function syntaxes: traditional `function` declarations and arrow functions (`=>`). You know that `tsconfig.json` with `strict: true` is the starting point for every project. And you ran your first TypeScript program.

These building blocks never change. Come back here when you need a refresher.

In the next chapter, you'll learn TypeScript's type system in depth — the single biggest shift from Ruby, and the feature that gives TypeScript its name.

Chapter 2: Types & Type Annotations

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2ts.

The Basic Types

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2ts.

Type Annotations

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2ts.

Type Inference

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2ts.

null and undefined

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2ts.

Union Types

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2ts.

Literal Types and Enums

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2ts.

Type Aliases

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2ts.

any and unknown

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2ts.

Type Assertions

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2ts.

Chapter Exercises

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2ts.

Summary

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2ts.

Chapter 3: Arrays, Tuples & Iteration

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2ts.

Typed Arrays

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2ts.

Mixed-Type Arrays

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2ts.

Tuples

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2ts.

readonly Arrays and Tuples

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2ts.

Iteration: for...of (The Each Replacement)

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2ts.

forEach: Each with an Index

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2ts.

Traditional for Loop (C-Style)

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2ts.

map: Transform Every Element

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2ts.

filter: Select Elements

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2ts.

reduce: Aggregate Values

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2ts.

some and every: Predicate Checks

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2ts.

Spread and Rest: The Splat Operator

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2ts.

Chapter Exercises

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2ts.

Summary

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2ts.

Chapter 4: Objects, Records & Maps

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2ts.

Object Types: The Hash Replacement

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2ts.

Optional Properties

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2ts.

Index Signatures: Hash-Like Objects

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2ts.

Record<K, V>: Typed Hash

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2ts.

Map<K, V>: When Keys Are Dynamic

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2ts.

Set: Unique Collections

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2ts.

Object Destructuring: Hash Splat

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2ts.

Spread: Merging Objects

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2ts.

readonly Properties

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2ts.

Chapter Exercises

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2ts.

Summary

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2ts.

Chapter 5: Functions Deep Dive

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2ts.

Function Declarations vs Arrow Functions

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2ts.

Typed Parameters and Return Types

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2ts.

Optional and Default Parameters

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2ts.

Rest Parameters: Ruby's Splat

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2ts.

Callbacks and Function Types

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2ts.

Void and Never Return Types

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2ts.

Function Overloads

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2ts.

Chapter Exercises

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2ts.

Summary

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2ts.

Chapter 6: Classes

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2ts.

A Ruby Class, Translated

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2ts.

Access Modifiers: public, private, protected

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2ts.

JavaScript Private Fields (#)

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2ts.

Parameter Properties (Shorthand)

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2ts.

Inheritance: extends

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2ts.

Abstract Classes

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2ts.

Getters and Setters

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2ts.

this and Arrow Functions

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2ts.

Chapter Exercises

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2ts.

Summary

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2ts.

Chapter 7: Interfaces & Type Aliases

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2ts.

Interfaces: The Basics

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2ts.

interface vs type

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2ts.

Extending Interfaces

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2ts.

Optional and readonly Properties in Interfaces

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2ts.

Index Signatures in Interfaces

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2ts.

Structural Typing in Practice

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2ts.

When Interfaces Feel Like Ruby Modules

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2ts.

Chapter Exercises

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2ts.

Summary

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2ts.

Chapter 8: Modules & Package Management

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2ts.

import and export: The New require

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2ts.

Named Exports vs Default Exports

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2ts.

Re-exports and Barrel Files

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2ts.

package.json: The Gemfile

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2ts.

npm, Yarn, and pnpm

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2ts.

node_modules and the Magic of Resolution

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2ts.

Your First TypeScript Project (Walkthrough)

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2ts.

Step 1: Initialize the Project

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2ts.

Step 2: Add a Runtime Dependency

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2ts.

Step 3: Write the Code

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2ts.

Step 4: Add a Run Script

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2ts.

What Just Happened?

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2ts.

tsconfig.json Paths: Module Aliases

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2ts.

Installing Type Definitions

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2ts.

Chapter Exercises

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2ts.

Summary

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2ts.

Chapter 9: Error Handling

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2ts.

try/catch/finally: The Direct Mapping

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2ts.

Typed Catch with instanceof

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2ts.

Custom Error Classes

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2ts.

The TypeScript Way: Errors as Values

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2ts.

Strict Null Checks: Eliminating NoMethodError on nil

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2ts.

Optional Chaining: Safe Navigation

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2ts.

Nullish Coalescing: The || Alternative

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2ts.

Chapter Exercises

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2ts.

Summary

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2ts.

Chapter 10: Async Programming

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2ts.

The Event Loop

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2ts.

Callbacks and Callback Hell

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2ts.

Promises: The Foundation

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2ts.

async/await: Write Async Code That Looks Synchronous

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2ts.

The async Main Function

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2ts.

Async Iteration

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2ts.

Ruby Threads vs TypeScript Promises

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2ts.

Chapter Exercises

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2ts.

Summary

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2ts.

Chapter 11: JSON & Type Narrowing

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2ts.

JSON.stringify: Value to JSON

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2ts.

JSON.parse: JSON to Value (and the any Problem)

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2ts.

Type Assertions: Trust Me

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2ts.

Type Guards: Prove It at Runtime

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2ts.

Zod: Schema Validation

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2ts.

Discriminated Unions: The JSON Pattern

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2ts.

Working with Unknown JSON Shapes

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2ts.

Chapter Exercises

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2ts.

Summary

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2ts.

Chapter 12: Generics & Utility Types

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2ts.

The Problem Generics Solve

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2ts.

Generic Functions

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2ts.

Generic Interfaces and Types

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2ts.

Generic Constraints

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2ts.

Built-in Utility Types

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2ts.

Partial — Make all properties optional

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2ts.

Required — Make all properties required

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2ts.

Pick<T, K> — Select specific properties

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2ts.

Omit<T, K> — Remove specific properties

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2ts.

Record<K, V> — Map keys to values

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2ts.

Pick<T, K> & Omit<T, K> — Composing Types

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2ts.

ReturnType — Extract the return type of a function

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2ts.

Awaited — Unwrap a Promise type

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2ts.

Generics vs Duck Typing

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2ts.

Chapter Exercises

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2ts.

Summary

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2ts.

Appendix A: Ruby ↔️ TypeScript Quick Reference

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2ts.

Syntax & Structure

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2ts.

Variables & Types

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2ts.

Data Structures

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2ts.

Functions & Methods

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2ts.

Error Handling

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2ts.

Concurrency & Async

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2ts.

Modules & Packages

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2ts.

Common Patterns

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2ts.

TypeScript CLI Cheat Sheet

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2ts.

File Layout

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2ts.

Glossary

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2ts.

Acknowledgements

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2ts.

Credits

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/rb2ts.