

Low maintenance Types

TypeScript Netherlands

@ddprrt

fettblog.eu - oida.dev - typescript-book.com

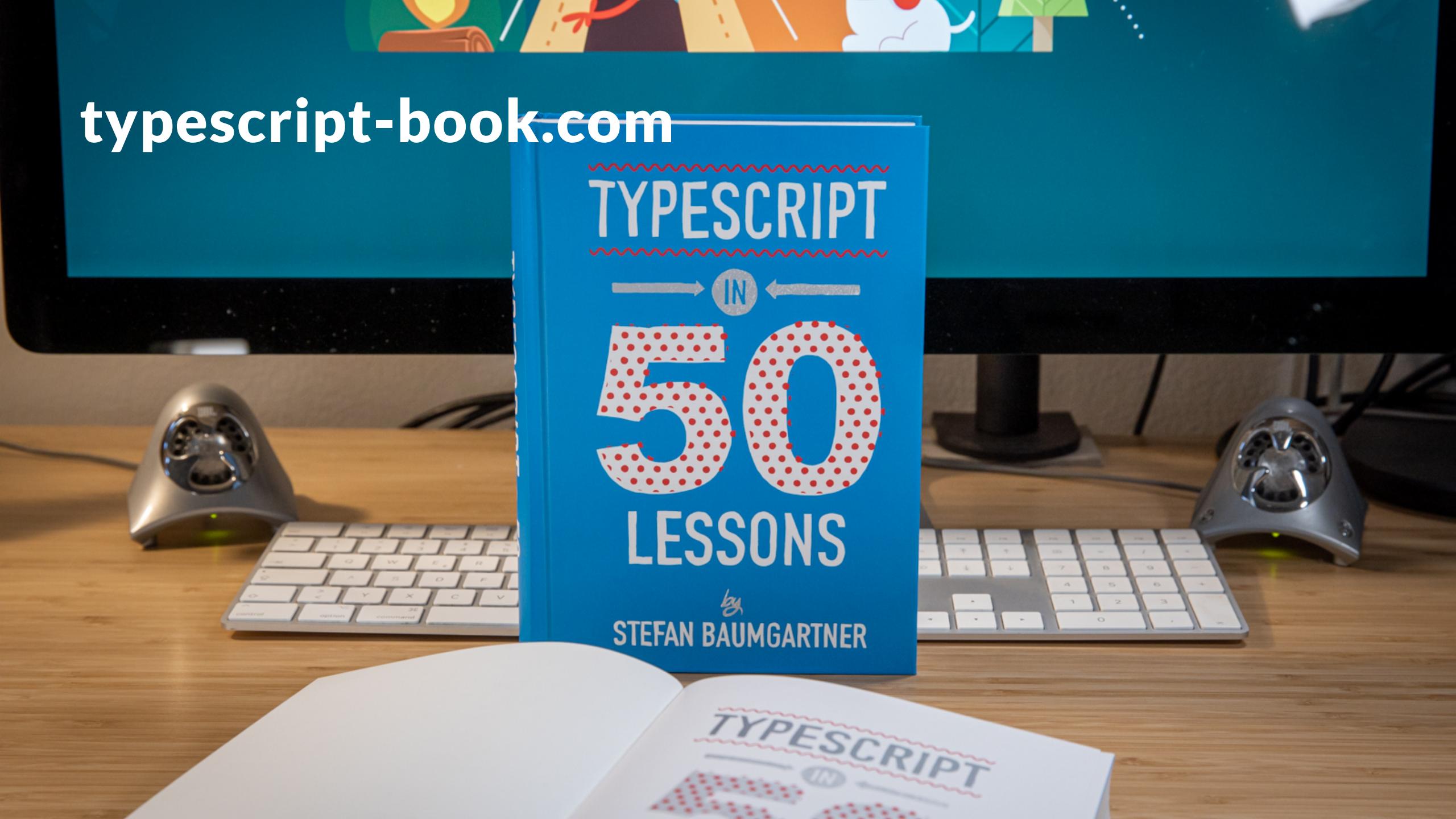
@ddprrt



typescript-book.com

TYPESCRIPT
IN
50
LESSONS

by
STEFAN BAUMGARTNER

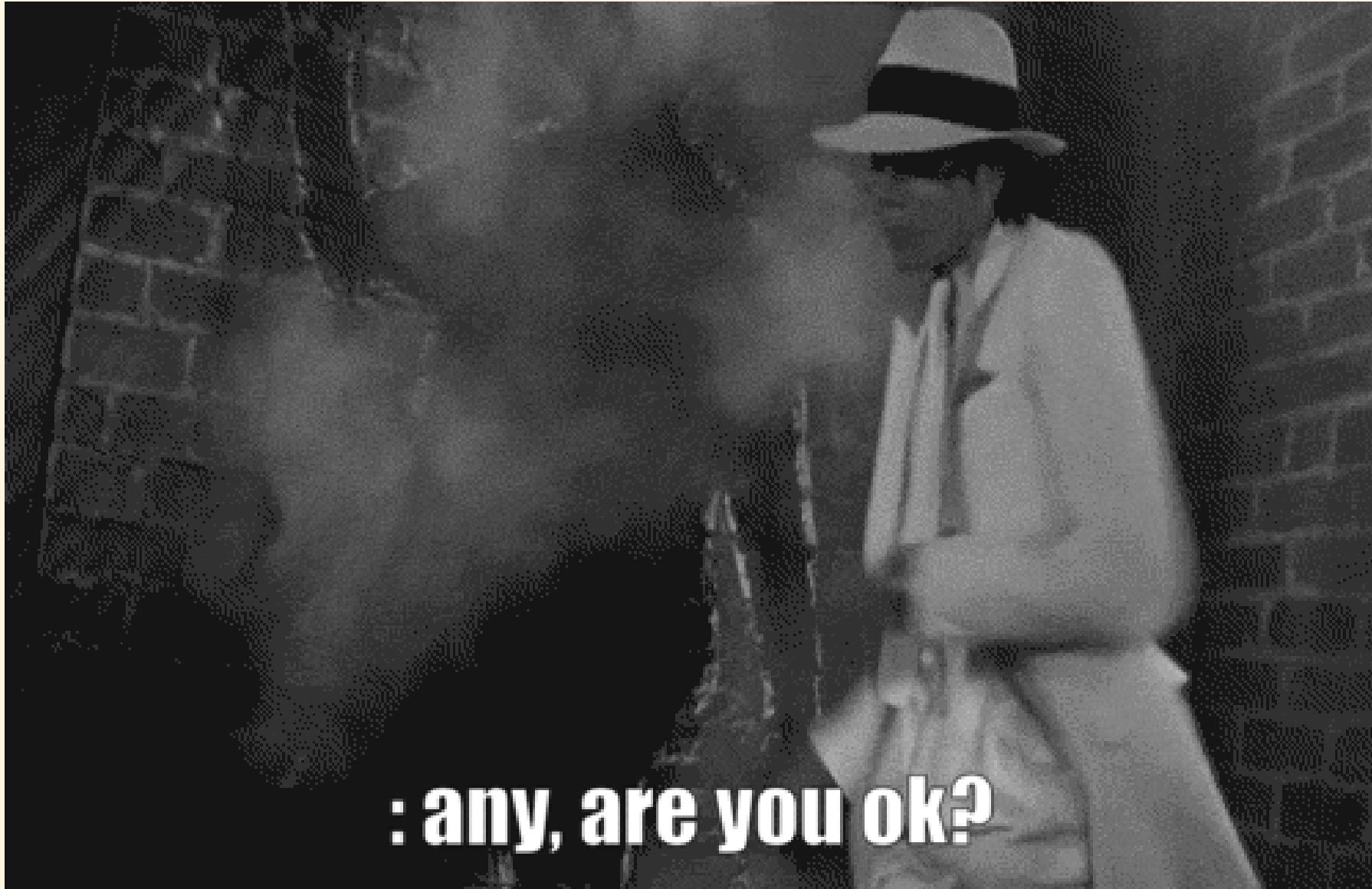


So... TypeScript!

When I started with TypeScript, I actually hated it!

TypeScript came in 3 waves...

1. Java developers told me this was the way to go!
2. Angular told me this is the way to go!



: any, are you ok?

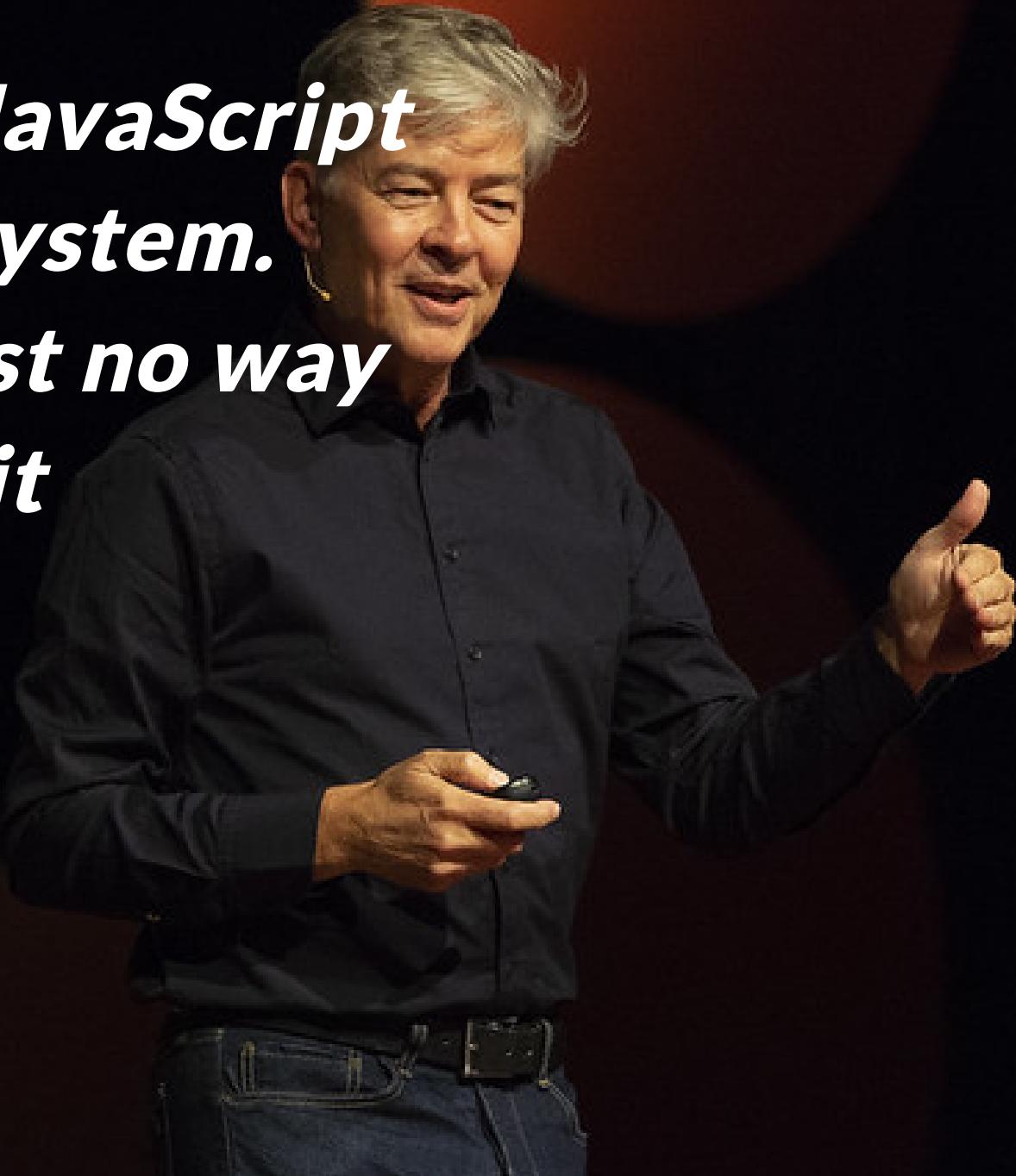
Anders Hejlsberg



The third wave ...

1. TypeScript is closer to JavaScript than I thought!
2. It's a thin, thin layer on top of JavaScript

*It's not that JavaScript
has no type system.
There was just no way
to formalize it*



- ❗ **TypeScript IS JavaScript**
- 🧪 **Language innovation through ECMAScript**
- 🔬 **Innovate in Type System through use cases**
- 🔧 **Tooling as primary citizen**

Formalize JavaScript

We need

- Structural Typing
- Bendable Type System
- Dynamic Static Typing
- Low Maintenance Types

Let's code!

Example 1: Dynamic Static Typing

```
app.get("/api/user/:userID", (req, res) => {
  if(req.method === "POST") {
    res.status(200).send();
  }
});
```

More code!

Example 2: Low Maintenance Types

```
type Toy = BoardGame | Puzzle | Doll;  
  
type ToyKind = "boardgame" | "puzzle" | "doll";  
  
type GroupedToys = {  
    boardgame: Toy[];  
    puzzle: Toy[];  
    doll: Toy[];  
};
```

A little bit more

Example 3: No Maintenance Types

```
const defaultOptions = {  
  from: "./src",  
  to: "./dest",  
};  
  
function copy(options) { /* ... */ }
```

Resources

- Book: [TypeScript in 50 Lessons](#)
- Article: [Low maintenance types](#)
- Article: [Dynamic static typing](#)
- Series: [Tidy TypeScript](#)
- Guide: [Advanced TypeScript](#)

fin

@ddprrt