

Qualities of TS to write
sustainable codes

1. **Introduction**
2. **The “Problem”**
3. **The “healthy” relationship**
4. **Wrap up**

Who is Kei?

A TypeScript enthusiast located in Vancouver, BC.

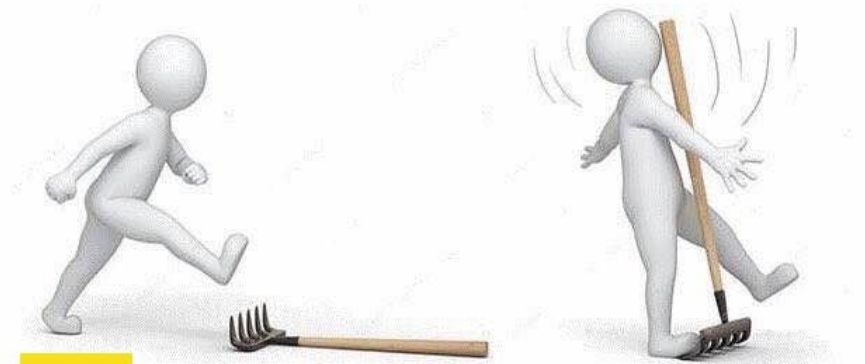
Having fun writing TypeScript for 4 years and actively helping beginners to learn the language by holding a bookclub (The last book was Learning TypeScript by Josh Goldberg).

Github: www.github.com/kei95

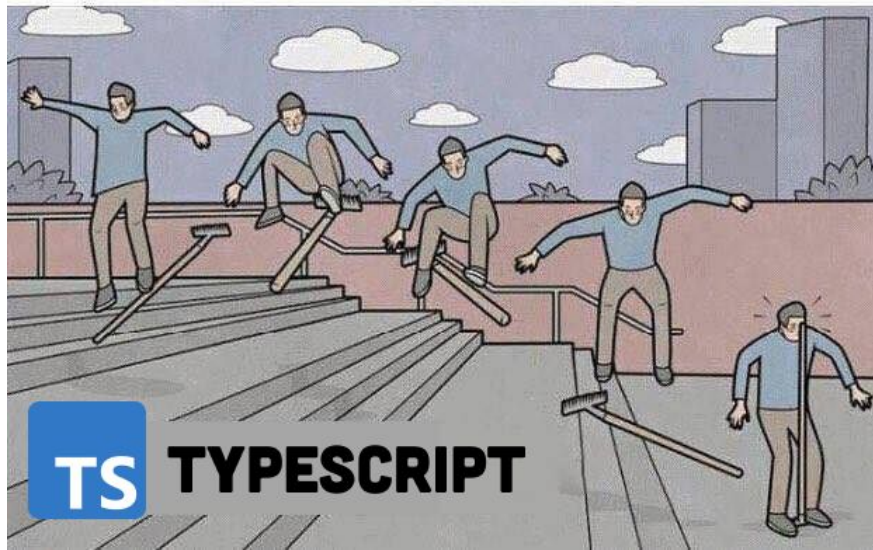
Twitter: www.twitter.com/kei957

LinkedIn: www.linkedin.com/in/keisuke-yamashita-075164178





JS JAVASCRIPT



TS TYPESCRIPT

1. **Introduction**
2. **The “Problem”**
3. **The “healthy” relationship**
4. **Wrap up**

▲ [throw_m239339](#) on Nov 20, 2020 | [parent](#) | [context](#) | [favorite](#) | on: [TypeScript 4.1](#)

I love typescript but its types are becoming so complicated it's turning into a dynamic language...

I mean one can't represent the all possibilities of JavaScript "types" without re-implementing JavaScript itself, only a subset. I wonder where Typescript will stop.

I'd like to see Typescript compiling to "native" as well.

**“I love TypeScript but its types
are becoming so complicated it’s
turning into a dynamic
language...”**

Why that is a case?

Trade-offs of complex util types

- 1. Readability**
- 2. Maintainability**
- 3. Soundness**

Trade-offs of complex util types

1. Readability
2. Maintainability
3. Soundness

Trade-offs of complex util types

1. **Readability**
2. **Maintainability**
3. **Soundness**

What is soundness?

“*Soundness* is about ensuring your program can’t get into certain invalid states. A sound *type system* means you can never get into a state where an expression evaluates to a value that doesn’t match the expression’s static type.” - Dart docs

What is soundness?

“*Soundness* is about ensuring your program can’t get into certain invalid states. A sound *type system* means you can never get into a state where an expression evaluates to a value that doesn’t match the expression’s static type.” - Dart docs

in other words...

What is soundness?

“*Soundness* is about ensuring your program can’t get into certain invalid states. A sound *type system* means you can never get into a state where an expression evaluates to a value that doesn’t match the expression’s static type.” - Dart docs

in other words...

**The type written in the file is
the same on runtime**

Trade-offs of complex util types

- 1. Readability**
- 2. Maintainability**
- 3. Soundness**

1. **Introduction**
2. **The “Problem”**
3. **The “healthy” relationship**
4. **Wrap up**

The “healthy” relationship

The “healthy” relationship

- Help TypeScript figures out types rather than you tell it what to do**

The “healthy” relationship

- Help TypeScript figures out types rather than you tell it what to do**
- Don't think you are smarter than TypeScript**

Matt Pocock's TS feature tier list



Matt Pocock
@mattpocockuk



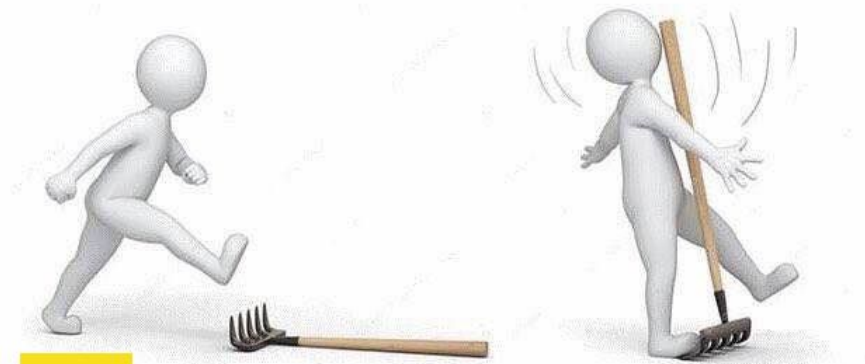
An extremely incomplete TypeScript features type-safety tier list.



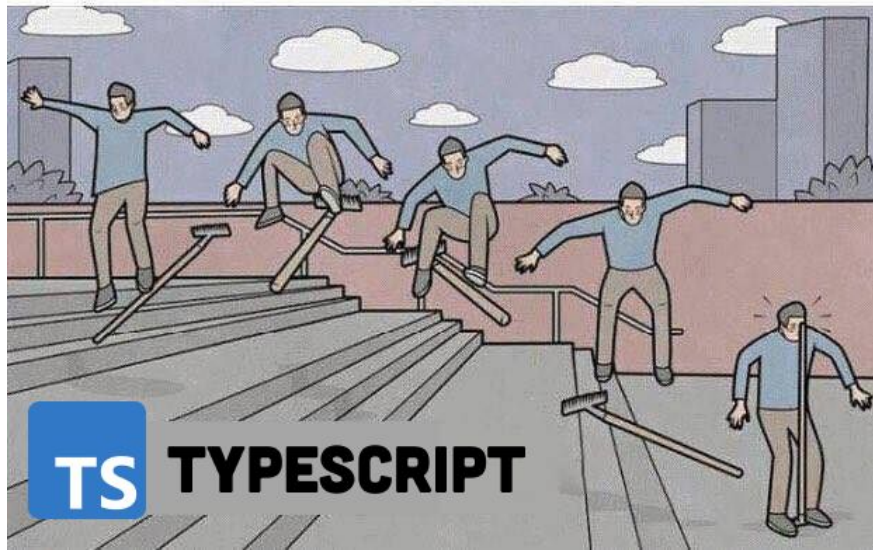
10:09 AM · 17 Feb, 2023

8 replies 49 shares 474 likes

<div>A</div> <div>Help TS to figure out types</div>	<div>satisfies</div> <div>Variable annotations (let a: string)</div> <div>Discriminated Unions</div> <div>Union Types</div> <div>Type helpers</div> <div>Narrowing with keywords (in, typeof, instanceof)</div> <div>Exhaustiveness checking</div>
<div>B</div> <div>Great from consumer</div>	<div>Generics</div> <div>Function overloads</div>
<div>C</div> <div>I know better than TypeScript</div>	<div>'as' assertions (as string, as number etc)</div> <div>Type predicates</div> <div>Assertion functions</div>
<div>D</div> <div>Inevitable, but brings “any”s</div>	<div>JSON.parse</div> <div>fetch</div> <div>localStorage.getItem</div> <div>anything in lib.dom.d.ts that returns any</div>
<div>F</div> <div>Usually there’s a better option</div>	<div>as any</div> <div>as unknown as any</div> <div>@ts-ignore</div> <div>@ts-except-error</div>



JS JAVASCRIPT



TS TYPESCRIPT

**How much
complex code
you write**

!=

**Your value as
a developer**

Thank you