

TypeScript - podstawy typowania



Hello

Adrian Kukowski

Senior Frontend Developer at LIBRUS

Agenda

1. Czym jest TypeScript?
2. Co nam daje TypeScript?
3. Jakie narzuca utrudnienia
4. Konfiguracja środowiska
5. Podstawowe typy
6. Interface
7. Enum
8. Any / unknown

Agenda

- 9. never
- 10. Łączenie typów
- 11. Typy generyczne

TYPESCRIPT

TypeScript rozszerza JavaScript o możliwość typowania. Dzięki czemu jesteśmy w stanie wyłapać więcej błędów zanim oprogramowanie trafi na produkcję.

Co nam daje TypeScript?

1. Ułatwia kontrolę nad aplikacją
2. Większa czytelność kodu
3. Wymusza mniejsze funkcje
4. Podpowiedzi w edytorze kodu
5. Sprawdza poprawność typów podczas kompilacji
6. Każdy kod JS jest prawidłowym kodem TS
7. TS finalnie jest kompilowany do JS

Jakie narzuca utrudnienia

1. Zwaln timer wydawania oprogramowania
2. Dodatkowa konfiguracja przy starcie projektu

Konfiguracja środowiska



instalacja

Dodanie polecenia
kompilacji

Sprawdzenie
efektu

Podstawowe typy

string

number

boolean

void

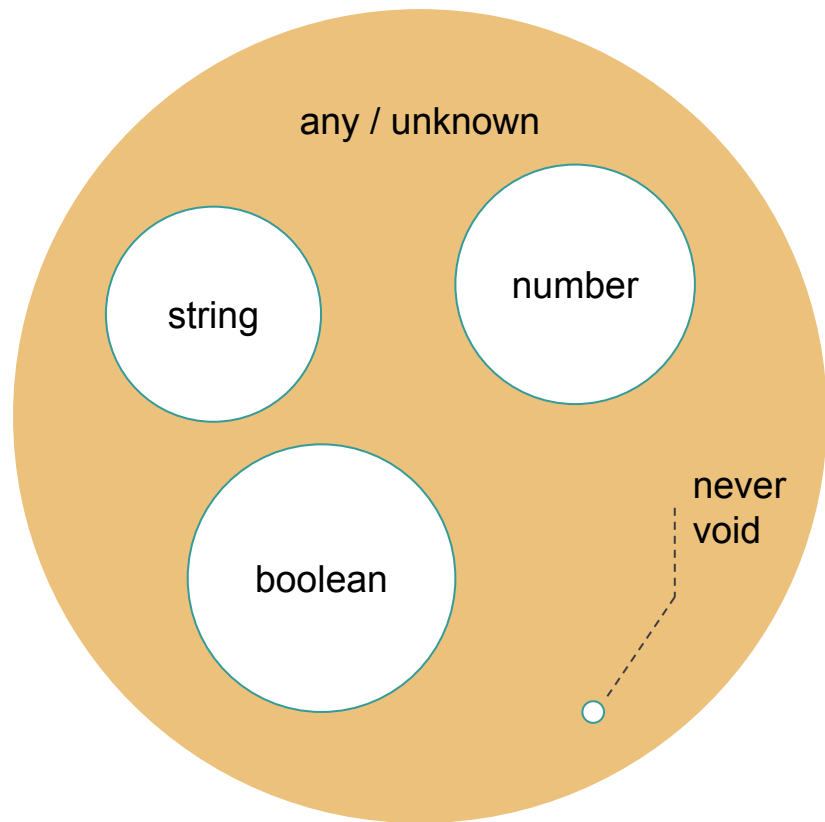
null

undefined

never

any

unknown



Interface

Tworzymy własne typy

Enum

Any / unknown

never

Union & Intersection

Union

type C = type A | type B

Intersection

type C = type A & type B

```
interface User {  
  name: string,  
}  
  
// Union  
interface Student {  
  age: number | string,  
}  
  
// Intersection  
const sayHello = (student: Student  
& User) => {  
}
```

Typy generyczne

[snippets/generic-types.ts](#)
[snippets/utility-types.ts](#)
[snippets/dead-branch.ts](#)



Dzięki



adrian-kukowski



adrian.kukowski@gmail.com