

# Les 4: TypeScript Escaperoom

*Fix de types. Ontsnap uit de kamers. Word een TypeScript master.*

**Tijdsduur:** 50-60 minuten (in de les) | **Type:** Hands-on Escaperoom

**Benodigheden:** Cursor (AI uit!), Terminal, Node.js

**Werkvorm:** Teams van 2

## WAT GA JE DOEN?

Je bent opgesloten in **10 kamers** vol TypeScript-puzzels. Elke kamer bevat code met type-fouten. Fix alle errors zodat de code compileert, en je ontvangt een **escape code** per kamer.

Los alle 10 kamers op en je krijgt de volledige escape code — het bewijs dat je bent ontsnapt!

## SETUP (5 MINUTEN)

- Download `les4-typescript-escaperoom-v2.zip` van Teams
- Unzip en open in Cursor

■ Installeer dependencies:

```
npm install
```

■ Zet Cursor AI uit!

```
Cmd+Shift+P → "Disable Cursor Tab"
```

■ Check je startpositie:

```
npm run check
```

**Belangrijk:** Je ziet errors — dat is de bedoeling! Elke error is een puzzel die jij moet oplossen.

## REGELS

### Mag WEL

- ✓ Cursor error meldingen lezen
- ✓ Hover over rode squiggles
- ✓ Terug naar slides kijken
- ✓ Je teamgenoot vragen
- ✓ De docent vragen (laatste resort!)

### Mag NIET

- ✗ AI gebruiken (Cursor Tab moet UIT)
- ✗ any gebruiken om errors te omzeilen
- ✗ @ts-ignore gebruiken
- ✗ Antwoorden kopiëren van anderen
- ✗ Opgeven!

## DE 10 KAMERS

| # | Concept               | Niveau      |
|---|-----------------------|-------------|
| 1 | Basic Types           | Easy        |
| 2 | Type Inference        | Easy        |
| 3 | Interfaces            | Easy-Medium |
| 4 | Optional Properties   | Easy-Medium |
| 5 | Union Types & Aliases | Medium      |
| 6 | Literal Types         | Medium      |

|    |                         |           |
|----|-------------------------|-----------|
| 7  | Type Narrowing          | Hard      |
| 8  | Function Types          | Hard      |
| 9  | Intersection & Readonly | Hard      |
| 10 | Boss Room (alles samen) | Very Hard |

**Doel:** Minimaal kamer 1-6 halen. Kamer 7-10 zijn bonus.

## TIMING

| Fase        | Tijd   | Wat  |
|-------------|--------|--|
| Setup       | 5 min  | Download, install, AI uitzetten                        |
| Kamers 1-4  | 15 min | Warm-up: basic types, inference, interfaces, optionals |
| Kamers 5-6  | 15 min | Union types, literal types                             |
| Kamers 7-10 | 25 min | Narrowing, functies, intersection, boss room           |
| Escape      | 5 min  | <code>npm run escape</code> + screenshot               |

## WAT LEER JE PER KAMER?

### Kamer 1: Basic Types — `src/kamer1-basics.ts`

Fix type annotations op variabelen. Leer `string`, `number`, `boolean` en `arrays` typen.

### Kamer 2: Type Inference — `src/kamer2-inference.ts`

Voeg ontbrekende type annotations toe aan functies. Leer wanneer TypeScript types zelf kan afleiden.

### Kamer 3: Interfaces — `src/kamer3-interfaces.ts`

Schrijf zelf interfaces voor objecten. Leer de "vorm" van data beschrijven.

### Kamer 4: Optional Properties — `src/kamer4-optionals.ts`

Gebruik `?` voor properties die niet altijd aanwezig zijn. Voeg `null` checks toe.

### Kamer 5: Union Types — `src/kamer5-unions.ts`

Definieer union types en fix ongeldige waarden. Leer `type Status = "pending" | "shipped"`.

### Kamer 6: Literal Types — `src/kamer6-literals.ts`

Fix ongeldige literal waarden en voeg ontbrekende cases toe aan functies.

### Kamer 7: Type Narrowing — `src/kamer7-narrowing.ts`

Gebruik `typeof` en `kind` checks om union types te narrowen.

### Kamer 8: Function Types — `src/kamer8-functions.ts`

Fix function signatures, return types, en callback typering.

## Kamer 9: Intersection & Readonly — `src/kamer9-intersection.ts`

Definieer intersection types met `&` en respecteer `readonly` properties.

## Kamer 10: Boss Room — `src/kamer10-boss.ts`

Combineer ALLES: interfaces, unions, narrowing, readonly, functies. Een compleet API systeem typen.

## HOE TRACK JE JE VOORTGANG?

**Check je status (regelmatig doen!):**

```
npm run check
```

Dit toont alle TypeScript errors per kamer. Hoe minder errors, hoe dichterbij de uitgang!

**Alle kamers opgelost? Run de escape:**

```
npm run escape
```

Dit compileert en runt je project. Als alles klopt, krijg je de **escape code**!

Laat je escape code zien aan de docent — dat is je bewijs!

## TIPS & TRICKS

### 1. Lees de error messages!

TypeScript vertelt je precies wat fout is. Hover over de rode squiggles in Cursor.

### 2. Check regelmatig

Run `npm run check` na elke wijziging. Zo weet je of je op de goede weg bent.

### 3. Ctrl+Click op types

Klik op een type om te zien hoe het gedefinieerd is. Handig voor het begrijpen van bestaande code.

### 4. Begin simpel

Kamer 1-2 zijn opwarming. Neem je tijd daar — de concepten komen terug in latere kamers.

### 5. Werk samen

Bespreek met je teamgenoot. Twee paar ogen zien meer dan een!

### 6. Fix cascade errors

Soms veroorzaakt 1 fout meerdere errors. Fix de eerste error en kijk opnieuw.

## HULP NODIG?

Type 'X' is not assignable to type 'Y' → Je gebruikt het verkeerde type

Cannot find name 'X' → Het type/interface bestaat nog niet — definieer het!

Property 'X' is missing → Je object mist verplichte velden

Parameter 'X' implicitly has an 'any' type → Voeg een type toe aan de parameter

Property does not exist on type → Je moet eerst type narrowing doen

Cannot assign to 'X' (read-only) → Verwijder de toewijzing — readonly kan niet gewijzigd worden

Function lacks ending return statement → Voeg een return toe voor alle mogelijke cases

Echt vast? → Vraag je teamgenoot, kijk terug naar de slides, of steek je hand op!

## DELIVERABLES

- `npm run check` geeft 0 errors
- `npm run escape` toont je escape code
- Screenshot van je escape code
- Laat je code zien aan de docent

**Niet alles af in de les?** Geen stress — maak thuis af. De escape code is je bewijs.