PROGRAM SZKOLENIA JĘZYKI

Szkolenie TypeScript

Nauczysz się TypeScriptu od zera i opiszesz kod typami, które łapią błędy, zanim trafią na produkcję.

CZAS TRWANIA	POZIOM	FORMA
4 dni	od podstaw	warsztat dla zespołu, zdalnie lub stacjonarnie

Znasz JavaScript, ale projekt, do którego dołączasz, jest w TypeScript, a Angular i Next.js zakładają, że rozumiesz typy. Bez tego kompilator zasypuje Cię błędami, a Ty uciszasz go przez `any` i rzutowania. Wtedy TypeScript zamiast pomagać tylko spowalnia pracę, a błąd `Cannot read properties of undefined` i tak trafia na produkcję.

Na szkoleniu zaczynasz od pierwszego typu i pustego `tsconfig.json`, a kończysz na typach generycznych, walidacji danych z API w Zod i konfiguracji projektu pod TypeScript 6. Ocenisz też, czy możesz już przejść na wersję 7 z kompilatorem w Go. Nie musisz wcześniej znać TypeScriptu, wystarczy JavaScript.

PROGRAM

4 dni

- Pierwsze kroki z TypeScript
  - Czym jest TypeScript: typy sprawdzane przed uruchomieniem, znikające w runtime
  - Instalacja, kompilator `tsc` i błędy w edytorze
  - `tsconfig.json` od zera: `strict`, `target`, `module`, `outDir`
  - Czytanie komunikatów błędów kompilatora
- Typy podstawowe i wnioskowanie
  - `string`, `number`, `boolean`, `null` i `undefined`
  - Typy literałowe
  - Wnioskowanie typów: kiedy pisać typ, a kiedy nie
  - `any`, `unknown` i `never`: czym się różnią
- Funkcje
  - Typy parametrów i wartości zwracanej, `void`
  - Parametry opcjonalne, domyślne i `...rest`
  - Typ funkcji i callbacki
  - Przeciążenia funkcji

- Obiekty i interfejsy
  - Typ obiektu, właściwości opcjonalne i `readonly`
  - `type` a `interface`: różnice i kiedy które
  - Rozszerzanie interfejsów i łączenie typów przez `&`
  - Sygnatury indeksu i `Record`
- Tablice, krotki i unie
  - Tablice, tablice `readonly` i krotki
  - Unie typów i zawężanie przez `typeof`, `in` i `instanceof`
  - Własne type guards
  - Unie dyskryminowane i sprawdzanie kompletności z `never`
  - Enum a unia literałów z `as const`
- Klasy i typy generyczne
  - Klasy: modyfikatory dostępu, klasy abstrakcyjne, implementacja interfejsu
  - Funkcje i klasy generyczne, ograniczenia `extends`
  - Dekoratory: standardowe i `experimentalDecorators`
- Zaawansowane typy
  - `keyof`, `typeof` i typy indeksowane
  - Typy mapowane, warunkowe i `infer`
  - Template literal types
  - Typy użytkowe: `Partial`, `Pick`, `Omit`, `ReturnType`
  - Operator `satisfies`
- Dane z zewnątrz
  - Odpowiedź z API, formularz i `JSON.parse` jako `unknown`
  - Walidacja w runtime z Zod i typ wyprowadzony ze schematu
- Konfiguracja kompilatora dla zaawansowanych
  - Flagi dodatkowe, np. `noUncheckedIndexedAccess`
  - `module` i `moduleResolution`: `nodenext` czy `bundler`
  - Nowe domyślne ustawienia TypeScript 6 i opcje, które znikają w 7
  - Wydajność kompilacji: project references, `skipLibCheck`, `--extendedDiagnostics`
- TypeScript 7: natywny kompilator w Go
  - Build nawet 10 razy szybszy i przejście z wersji 6 krok po kroku
  - Czego brakuje w 7.0: API kompilatora, pluginy Angulara i Vue
  - Linter w projekcie na TS 7: typescript-eslint nadal na TS 6
- Narzędzia
  - `tsc`, tryb `--watch` i `tsserver` w edytorze
  - Uruchamianie TypeScript w Node.js bez builda
  - `erasableSyntaxOnly`: enum, `namespace` i parameter properties w kodzie, który ma działać bez kompilacji
  - Moduły ES zamiast `namespace`
- Pliki deklaracji
  - Własne pliki `.d.ts` i paczki `@types`
  - `declare module` i rozszerzanie cudzych typów
  - Typy w paczce npm, którą publikujesz

- Migracja z JavaScript
  - `allowJs`, `checkJs` i typy w komentarzach JSDoc
  - Stopniowe zaostrzanie kompilatora, plik po pliku
  - `// @ts-expect-error` zamiast `// @ts-ignore`
- Testowanie typów
  - `expectTypeOf` w Vitest
  - Testy, które pilnują, że błędny kod się nie kompiluje

DLA KOGO?

- Dla programistów JavaScript, którzy zaczynają pracę z TypeScript.
- Dla osób, które przed szkoleniem Angular albo Next.js chcą najpierw poznać typy.
- Dla programistów, którzy piszą już w TypeScript, ale walczą z kompilatorem albo uciszają go przez `any`.
- Dla zespołów, które migrują kod z JavaScript na TypeScript albo planują przejście na TypeScript 7.

CO MUSISZ UMIEĆ?

- Programować w JavaScript: funkcje, obiekty, metody tablic, klasy, moduły ES i `async` / `await`.
- Pracować z npm i terminalem.

CZEGO SIĘ NAUCZYSZ?

- Skonfigurujesz projekt TypeScript od zera i odczytasz błędy kompilatora.
- Opiszesz typami funkcje, obiekty, tablice i klasy.
- Opiszesz typami odpowiedzi API i formularze, więc literówka w nazwie pola nie przejdzie builda.
- Użyjesz unii dyskryminowanych, typów generycznych, mapowanych i warunkowych tam, gdzie upraszczają kod.
- Zwalidujesz dane z zewnątrz w Zod zamiast ufać rzutowaniu.
- Skonfigurujesz `tsconfig.json` pod TypeScript 6 i 7 oraz linter, który działa przy obu wersjach.
- Ocenisz, czy Twój projekt może już przejść na natywny kompilator.
- Dopiszesz typy do biblioteki, która ich nie ma.
- Przeprowadzisz migrację modułu z JavaScript na TypeScript.
- Napiszesz testy, które sprawdzają same typy.

OPINIE UCZESTNIKÓW

> *„Oprócz samego Angulara Piotr zdążył nam przekazać mnóstwo wiedzy na temat Git, Visual Studio Code, TypeScript i dobrych praktyk programistycznych - czego oczywiście nie było w programie.”*
>
> Sebastian, Full-Stack Developer (szkolenie z Angulara)

> *„The whole team was able to learn from the mini-workshops and presentations he regularly shared with us, covering advanced Git tricks, React, JavaScript and more.”*
>
> Paul Ngei, współpracownik z zespołu w Box (rekomendacja na LinkedIn)

TRENER

Piotr Kowalski

W branży IT od 2008 roku. Od 2014 jest mentorem dla wielu developerów. Jako trener spędził na sali szkoleniowej ponad 1000 godzin, a na LinkedIn ma 52 rekomendacje. Każde jego szkolenie to dawka solidnej wiedzy oraz humoru.

Więcej: piecioshka.pl/piotr-kowalski

ZORGANIZUJMY TO SZKOLENIE W TWOIM ZESPOLE

Program dopasuję do Waszego projektu i poziomu grupy. Napisz, z czym pracujecie, a odpowiem z propozycją terminu i zakresu.

KONTAKT	**piecioshka.pl/kontakt**
E-MAIL	**piecioshka+blog [małpka] gmail.com**
SZKOLENIE	**piecioshka.pl/szkolenia/typescript/**