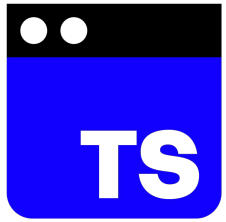
Code Academy



TypeScript
įvadas

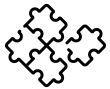
TypeScript

1 paskaita

Paskaitos eiga

Kas yra TypeScript?

TypeScript failų
transpiliavimas



Darbinės aplinkos kūrimas

Kintamųjų aprašymas tipais

Kas yra TypeScript?

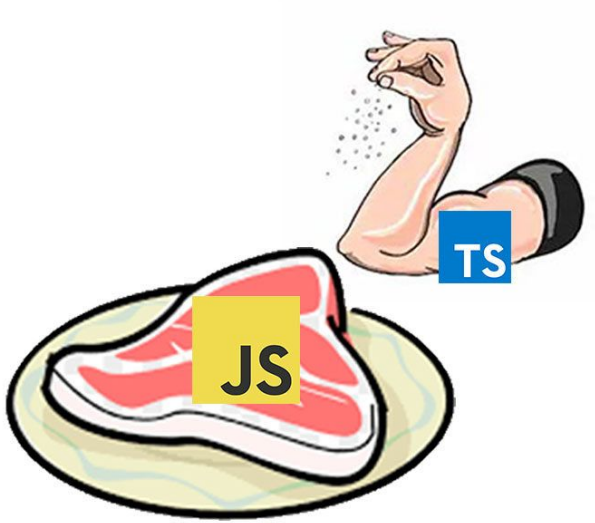
Kas yra TypeScript?

TypeScript - tai JavaScript programavimo kalbos dialektas.

TypeScript dialekto esmė, sugriežtinti kodą ir patikrinti suderinamumą rašymo metu.

Aprašant tipus ir sugriežtinant programinį kodą, mes galime matyti klaidas, transpiliuojant (compile time) kodą, o ne vykdant (runtime).

Tai sumažina tikimybę, kad bus nepastebėtos klaidos.



Kada naudojamas TypeScript?

TypeScript naudoti nebūtina, tačiau esant tam tikroms aplinkybėms - verta.

TypeScript verta naudoti, kai:

- Projektas yra didelės apimties
- Programuotojų komanda yra susipažinusi su tipais aprašytomis kalbomis
- Projekto naudojamos bibliotekos parašytos naudojant TypeScript
- Reikalingas didesnis programos greitis

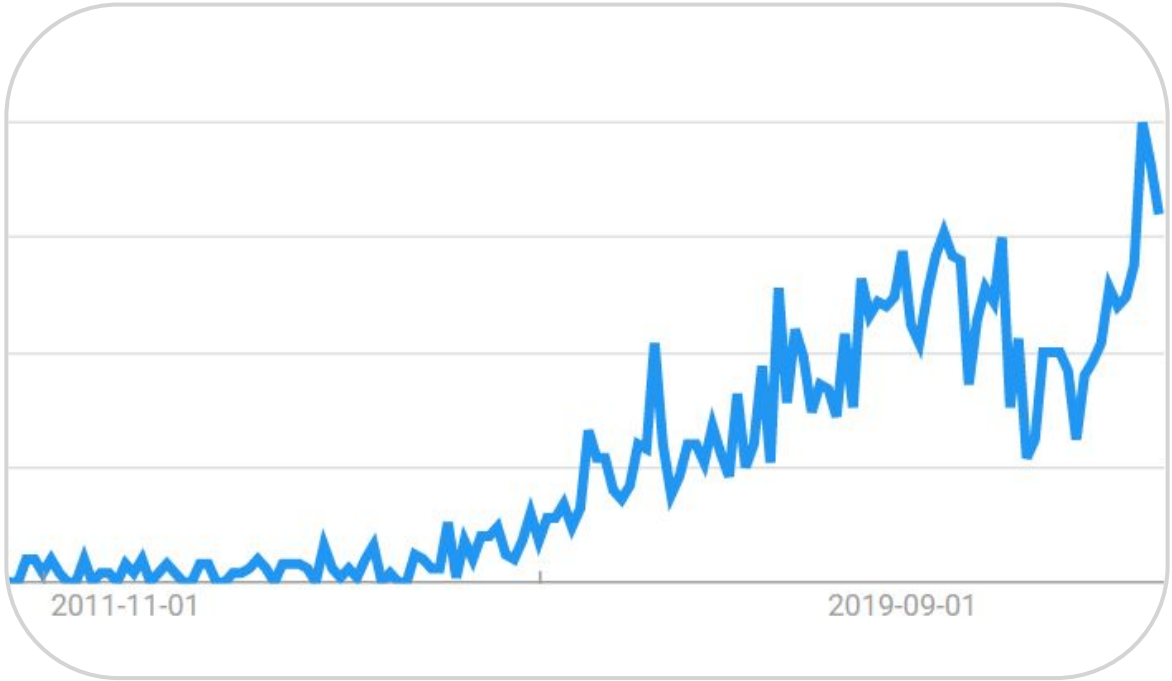
Kuriant didelės apimties projektus

Kuomet projektas yra didelės apimties ir/arba prie jo dirba daug žmonių. Didėja tikimybė, jog bus neišvengiama klaidų.

TypeScript sprendžia šią problemą tikrinant kodo kokybę ir suderinamumą kompiliacijos/rašymo metu.

Naudodami TypeScript galite aprašyti kiekvieną funkciją, klasę ar kintamuosius tipais. Taip taupomas laikas perprasti atskiroms programos dalims.

TypeScript populiarumas



TypeScript failų *transpiliavimas*

TypeScript kodo paruošimas naršyklei

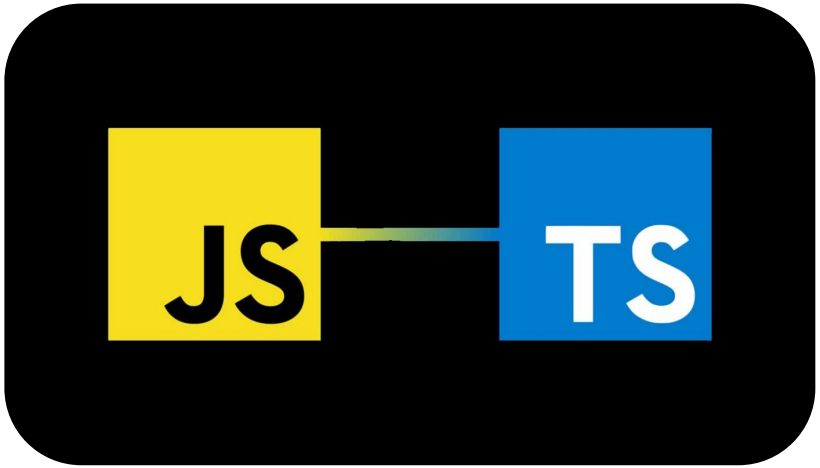
Naršyklė **negali** vykdyti TypeScript kodo.

Todėl turime parsisiųsti TypeScript transpiliatorių, kuris TypeScript kodą transpiliuoja (paverčia) į JavaScript kodą.

TypeScript transpiliatorius parsiunčiamas naudojant bibliotekų tvarkymo įrankius:

- npm
- yarn
- ir t.t.

Šiam kursui naudosime npm, kuris parsiunčiamas kartu su Node.js.



Užduotis 1

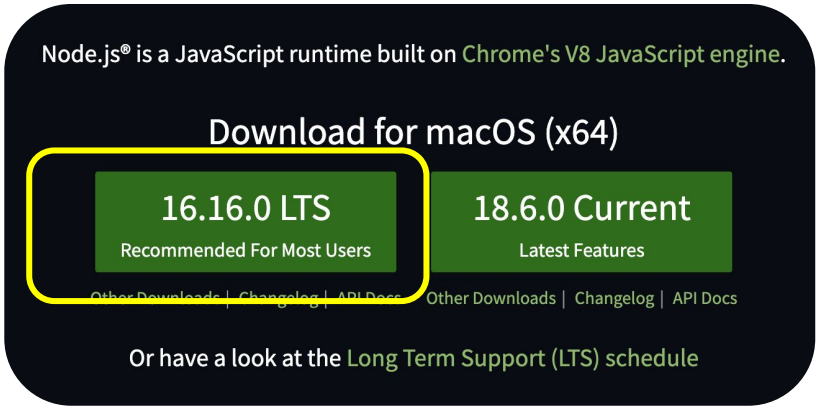
Node.js parsiuntimas

Node.js - tai C++ programa kuri naudoja Chrome V8 JavaScript variklį, kad vykdytų JavaScript kodą.

Node.js - https://nodejs.org/en/

Siųsdami Node.js programą visuomet rinkitės ne pačią naujausią versiją.

Naujos programos versijos nėra laiko patikrintos, todėl gali būti neaptiktų bėdų.



TypeScript bibliotekos įrašymas

Globalus TypeScript įrašymas

Įrašome TypeScript biblioteką globaliai →

```
npm install -g typescript
```

Įrašius biblioteką pasitikrinkite ar pavyko įrašyti biblioteką →

```
tsc -v
```

TypeScript įrašymas projektui

Pirmiausia reikia sukurti pradinį npm projektą→

```
npm init --y
```

Tuomet įrašome typescript biblioteką vystymui→

```
npm install typescript --save-dev
```

TypeScript. įrašyti globaliai ar lokaliai?

Kuriant aplikaciją profesionaliai, TypeScript reikėtų įrašyti kiekvienam projektui atskirai. Tai turi tokių pranašumų:

- Visi prie projekto dirbantys programuotojai naudos tą pačią TypeScript versiją
- Naujam programuotojui nereikia papildomai įrašyti bibliotekos globaliai, viskas parsiunčiama su standartine komanda `npm i`

Mokantis mums nereikia derintis prie kolegų ir pamokų failai nebus nuolatos vystomi po pamokos.

Todėl mokantis naudosime globaliai įrašytą TypeScript biblioteką.

TypeScript kompiliatoriaus konfigūravimas

Sukuria pradinį TypeScript konfigūravimo failą→

```
tsc --init
```

Darbiniame aplanke susikūrė failą “tsconfig.json”. Jame aprašytos taisyklės pagal kurias bus tikrinamas kodas jį transpiliuojant.
Daugiau informacijos: https://www.typescriptlang.org/tsconfig

TypeScript failų kompiliavimas→

```
tsc
```

TypeScript failų kompiliavimas stebint failus→

```
tsc --watch
```

Užduotis 2

Išanalizuokite typescript konfigūracijos nustatymus

1. Įsirašykite TypeScript biblioteką globaliai
2. Sukurkite pradinį TypeScript konfigūracijos failą
3. Atsidarykite failą `tsconfig.json`
    - Perskaitykite visų nustatymų paaiškinimus.
    - Atsidarykite dokumentaciją: https://www.typescriptlang.org
    - Detaliau panagrinėkite nustatymus, susiradę juos dokumentacijoje:
        1. `target`
        2. `rootDir`
        3. `outDir`
        4. `module`
        5. `noImplicitAny`
        6. `removeComments`

Išanalizuokite typescript konfigūracijos nustatymus

Baigę užduotį padiskutuokite su dėstytojų ar teisingai supratote dokumentacijoje perskaitytus nustatymus.

Klausimai?

Užduotis 3

Sukurkite naują TypeScript projektą

1. Sukurkite naują `npm` projektą
2. Įrašykite typescript biblioteką privačiai
3. Sukurkite pradinį typescript konfigūracijos failą
    - Nurodykite, kad pradinis typescript programinis aplankas bus `src/`
    - Nurodykite, kad *transpiliuoti* javascript failai bus saugomi į aplanką `public/js/`
4. Sukurkite pradinį typescript programinį failą
    - Faile parašykite testinį kodą savo nuožiūra

Sukurkite naują TypeScript projektą

5. Paleiskite typescript transpiliatorių su stebėjimo režimu
6. public aplanke sukurkite failą `index.html`
    - Sukurkite pradinį HTML failo skeletą
    - Įtraukite *transpiliuotą* javascript failą
7. Peržiūrėkite `index.html` failą naršyklėje ir patikrinkite savo sukurto kodo vykdymą

Klausimai?

Darbinės aplinkos kūrimas

ESlint

ESLint - tai kodo kokybės užtikrinimo įrankis. Jis skirtas patikrinti ar jūsų kodas atitinka iš anksto nustatytus ir sukonfiguruotus kodo kokybės standartus.

Jeigu įsirašytumėte savo ekraną kuriant bet kokį programinį kodą, pamatytumėte, kad jūs labiau skaitot kodą nei jį rašot.

ESlint

Dažniausiai skaitymas užtrunka 5 kartus daugiau laiko nei rašymas. Jeigu kodas rašomas vienoda stilistika ir tvarka jį daug lengviau skaityti.

Ypač tai aktualu kuomet dirbate prie projekto komandoje:

P.S.: beveik visada dirbate komandoje.

Tam ir skirtas ESLint įrankis, kad jūsų kodas būtų vienodas, nuspėjamas ir lengvai skaitomas. Taupytų jūsų laiką!

ESLint instaliavimas

Globalus ESLint įrašymas→

```
npm install -g eslint
```

ESLint konfigūracijos kūrimas→

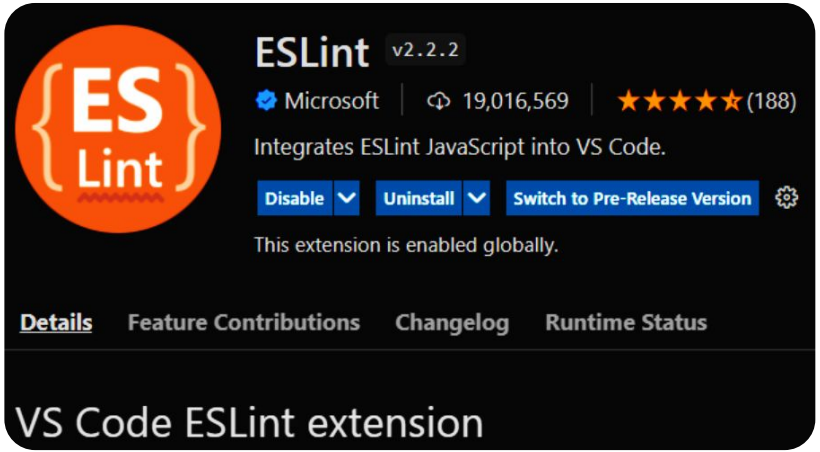
```
eslint --init
```

ESLint instaliavimas

```
$ eslint --init
You can also run this command directly using 'npm init @eslint/config'.
√ How would you like to use ESLint? · style
√ What type of modules does your project use? · esm
√ Which framework does your project use? · none
√ Does your project use TypeScript? · No / Yes
√ Where does your code run? · browser
√ How would you like to define a style for your project? · guide
√ Which style guide do you want to follow? · airbnb
√ What format do you want your config file to be in? · JavaScript
Checking peerDependencies of eslint-config-airbnb-base@latest
Local ESLint installation not found.
The config that you've selected requires the following dependencies:

@typescript-eslint/eslint-plugin@latest eslint-config-airbnb-base@latest eslint@^7.32.0 || ^8.2.0 eslint
@latest
√ Would you like to install them now with npm? · No / Yes
Installing @typescript-eslint/eslint-plugin@latest, eslint-config-airbnb-base@latest, eslint@^7.32.0 ||
t-eslint/parser@latest

added 183 packages, and audited 184 packages in 7s

57 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
```

ESLint extension

Šis įskiepis tikrina jūsų parašytą kodą pagal sukurtą ESLint konfigūracijos failą.



Užduotis 4

Papildykite savo projektą ESLint konfigūracija

1. Parsisiųskite ESLint
2. Sukurkite pradinį ESLint konfigūracijos failą, pagal praeitos skaidrės pavyzdį
3. Įsirašykite ESLint įskiepį į VisualStudio Code
4. Sukurkite failą `.vscode/settings.json`

```
{
   "editor.codeActionOnSave":{
      "source.fixAll.eslint":true
   }
}
```

Papildykite savo projektą ESLint konfigūracija

5. Perkraukite VisualStudio Code (tik tuomet pradės veikti nustatymai!)
6. Pabandykite parašyti *netvarkingą* kodą savo programiniame faile
    - daug tuščių eilučių
    - kabliataškių trūkumas ir t.t.
7. Išsaugokite failą. Ar kodas susitvarkė? **Jei ne, kreipkitės pagalbos į dėstytoją**

Klausimai?

Kintamųjų aprašymas tipais

Kintamųjų aprašymas

Kintamųjų tipai aprašomi po kintamojo pavadinimo parašant dvitaškį ir TypeScript tipą.
Jeigu tipo neaprašėte, “TypeScript server” (fone veikianti programa) bandys atspėti kintamojo tipą pagal tai ką parašėte dešinėje priskyrimo(lygybės ženklo) pusėje.

```ts
const firstName: string = 'Opelijus';
const age: number = 21;
const numbers: number[] = [1, 1, 2, 3, 5, 8,
13, 21];

type Person = {
  name: string,
  surname: string,
  age: number,
};
```

```ts
const student1: Person = {
  name: 'Faubas',
  surname: 'Madvalas',
  age: 11
};

const students: Person[] = [
  { name: 'Faubas', surname: 'Nikalas', age: 11 },
  { name: 'Micius', surname: 'Mežonas', age: 11 },
  { name: 'Veira', surname: 'Marcytė', age: 11 },
];
```

Užduotis 5

Kintamųjų aprašymas

1. Sukurkite kintamuosius ir aprašykite juos TypeScript tipais:
    - vardas
    - amžius
2. Monetos esančios piniginėje (masyvas iš skaičių)
3. Prekė (reikia sukurti atskirą tipą - įvardinti objektu)
    - pavadinimas
    - kaina

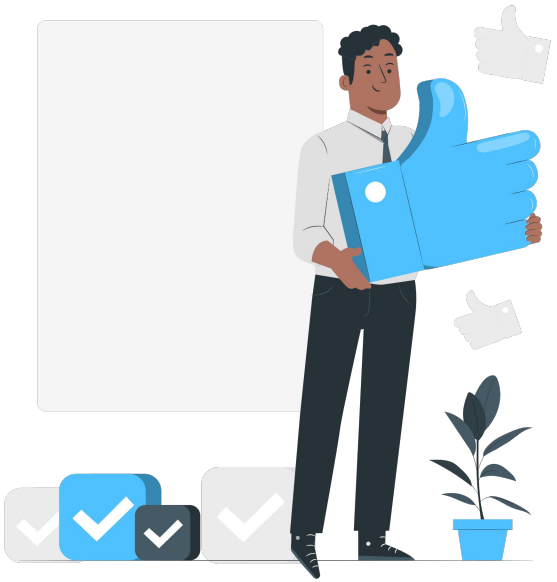
Kintamųjų aprašymas

4. Prekių sąrašas (masyvas iš prekių)
5. Žmogus
    - vardas
    - amžius
    - statusas (vedęs/ištekėjusi, išsiskyręs/-usi, našlys/-ė ir t.t)
6. Draugų sąrašas (masyvas iš žmonių)

Klausimai?

Projekto šablonas

Peržiūrėkite aplanką “TS projekto skeletas” ir naudokite jį atlikti praktikos užduotims.



Iki kito karto!