# Programowanie Aplikacji Internetowych

Javascript / Typescript stack

# Plan na dziś

- Javascript
- React
- Express
- Typescript

# Tech Stack 2023: Frontend

1. React
2. Next.js flavoured React (the best choice)
3. Typescript

Nice summary: blog post

# Tech Stack 2023: Backend

No one clean winner:

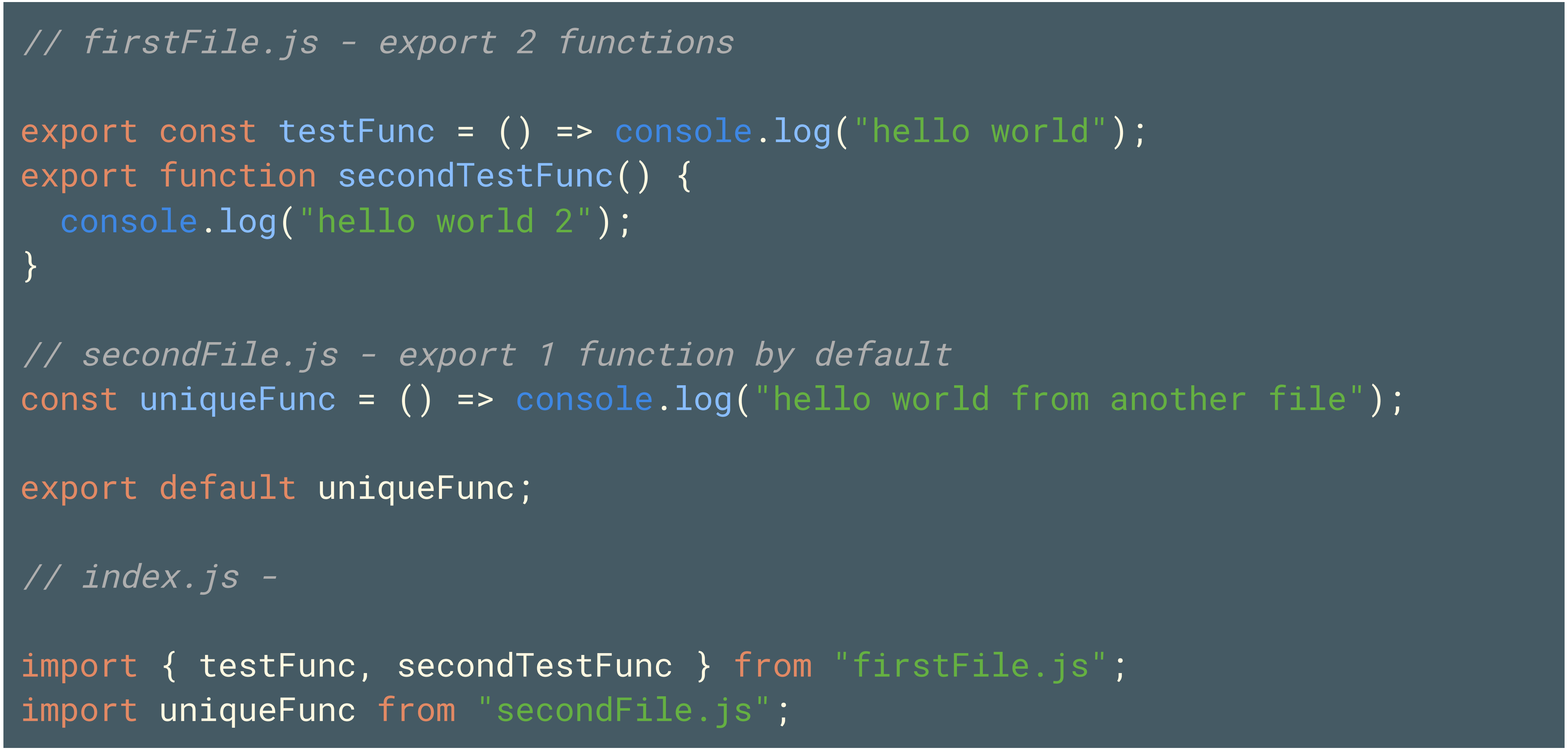
- Python (Flask, FastAPI, Django)
- JS/TS frameworks
- Java/C#
- Golang

# Javascript

- Lightweight and interpreted,
- just-in-time compiled,
- first-class functions,
- imperative, and declarative (e.g. functional programming) styles;
- ECMAScript.

[więcej na developer.mozilla.org]

# Javascript



```javascript
// firstFile.js - export 2 functions

export const testFunc = () => console.log("hello world");
export function secondTestFunc() {
  console.log("hello world 2");
}

// secondFile.js - export 1 function by default
const uniqueFunc = () => console.log("hello world from another file");

export default uniqueFunc;

// index.js -

import { testFunc, secondTestFunc } from "firstFile.js";
import uniqueFunc from "secondFile.js";
```

# Javascript

Javascript for Python programmers

# Javascript

JavaScript style:

- camelCase instead of snake_case for variables, still TitleCase for classes and ALL_CAPS for global constants;
- Standard JavaScript style prescribes semicolons;
- Two spaces per indentation level instead of four.

# Nodejs

- Node.js to otwarto źródłowe i między platformowe środowisko uruchomieniowe dla języka JavaScript;
- Oparte jest o silnik wykonywania kodu JavaScript o nazwie V8;

# npm

- manager pakietów;
- najlepiej zarządzać z pomocą nvm;
- `npx` - package executor.

express

# Express

- Fast server-side development;
- Middleware;
- Routing;
- Templating.

# Express - helloworld

```
const express = require("express");
const app = express();
const port = 3000;

app.get("/", (req, res) => {
    res.send("Hello World!");
});

app.listen(port, () => {
   console.log(`Example app listening on port ${port}!`);
});
```

# Express

- Struktura wygenerowanego projektu:

```
npx express-generator  --view=pug myapp
```

- `app.js`, `routes/`, `public/` i `views/`.

# Express

- routes;
- template engines;
- static.

# Typescript

- vs Javascript:

```
let var1 = "Hello";
var1 = 10;
console.log(var1);
```

- IDE support

# Typescript

- TypeScript is JavaScript with syntax for types;
- `npm install @types/http`.

# Dziękuję za uwagę

# Backup slides