

TypeScript

2.2. TypeScript

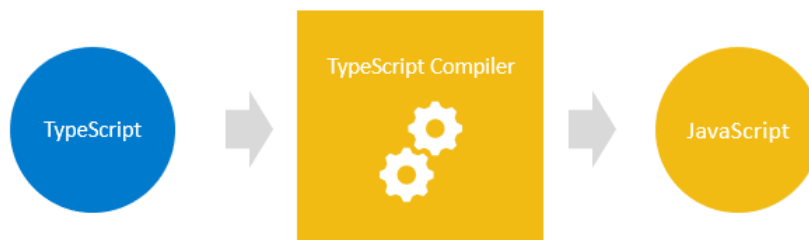
TypeScript to język programowania będący nadzbiorem języka JavaScript. To oznacza, że obowiązuje w nim składnia języka JavaScript, ale rozbudowana o dodatkowe elementy. Dzięki temu kod napisany w TypeScriptie jest czytelniejszy, a liczba popełnianych błędów — mniejsza. Wykorzystują go m.in.: Facebook, Microsoft, Google i Netflix. Mieści się on w pierwszej dziesiątce najchętniej wybieranych języków programowania, a jego popularność ciągle wzrasta (według PYPL PopularitY of Programming Language — <https://pypl.github.io/PYPL.html>).

TypeScript builds on top of JavaScript. First, you write the TypeScript code. Then, you compile the TypeScript code into plain JavaScript code using a TypeScript compiler.

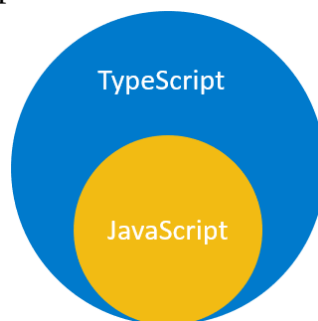
Once you have the plain JavaScript code, you can deploy it to any environments that JavaScript runs.

TypeScript files use the **.ts** extension rather than the **.js** extension of JavaScript files.

Conversion:



TypeScript jest obudową JavaScript:



Instalacja:

Strona z pomocą jak zainstalować na Windows

<https://www.typescripttutorial.net/typescript-tutorial/setup-typescript/>

1. Musisz mieć zainstalowany Node.js.
2. Tworzysz sobie folder w którym będzie Twój projekt:

```
/home/adams/PROGRAMOWANIE/NODE/pierwszytypescript
```

Jak masz Windows będzie to trochę inaczej wyglądało ale warto mieć wszystko ładnie poukładane w folderach.

3. Teraz za pomocą node tworzymy nowy projekt w tym folderze

```
adams #>: npm init -y
Wrote to /home/adams/PROGRAMOWANIE/NODE/pierwszytypescript/package.json
:
{
  "name": "pierwszytypescript",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "keywords": [],
  "author": "",
  "license": "ISC"
}
```

Powinieneś otrzymać coś takiego jak wyżej. Jeśli nie użyjesz opcji -y będziesz musiał wpisywać różne informacje.

```
package name: (pierwszytypescript)
version: (1.0.0)
description:
git repository:
author:
license: (ISC)
```

4. Następnie instalujesz sobie TypeScript:

```
npm install typescript
```

```
adams #>: npm install typescript
added 1 package, and audited 213 packages in 7s
26 packages are looking for funding
  run `npm fund` for details
found 0 vulnerabilities
```

Pojawią się następujące dwie nowe rzeczy w Twoim projekcie:

*Folder: **node_modules** - pliki odpowiedzialne za obsługę języka, dodatkowe moduły*

*plik: **package-lock.json** - informacje w formacie JSON o wersji*

5. Inicjalizujemy język TypeScript:

```
npx tsc -init
```

6. Teraz można sprawdzić wersję języka:

```
npx tsc --v
```

7. Musisz mieć Visual Studio Code (lub inny np.: Atom, SublimeTex itp.).

1. przydatny dodatek: Live Server (Visual Studio Code)

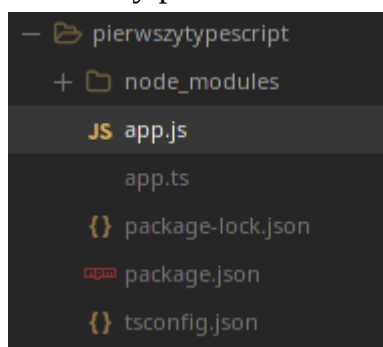
8. Tworzymy nowy plik **app.ts**.

```
let message: string = 'Hello, World!';
console.log(message);
```

9. Generujemy plik JavaScript:

```
npx tsc app.ts
```

Powinien w folderze powstać nowy plik:



10. Poprawność działania (lub otwierasz w przeglądarce internetowej index.html w którym masz skrypt app.js):

```
adams #>: node app.js
Hello, World!
```

11. Teraz czas na zautomatyzowanie wykonywanych operacji:

1. Wykonanie

`npx tsc --watch app.ts`

spowoduje że będzie prowadzony nasłuch na danym pliku – jeśli nastąpi w nim zmiana otrzymasz komunikat:

```
[08:54:14] File change detected. Starting incremental compilation...  
[08:54:15] Found 0 errors. Watching for file changes.
```

Można też użyć polecenia: (nasłuch na wszystkich plikach)

`npx tsc --watch`

12. Uruchomienie w przeglądarce:

1. Tworzymy plik index.html

```
<!DOCTYPE html>  
<html lang="en">  
  <head>  
    <meta charset="UTF-8">  
    <meta name="viewport" content="width=device-width, initial-scale=1.0">  
    <title>TypeScript: Hello, World!</title>  
  </head>  
  <body>  
    <script src="app.js"></script>  
  </body>  
</html>
```

2. Następnie zmieniamy plik app.ts na poniższy:

```
let message: string = 'Hello, World!';  
// create a new heading 1 element  
let heading = document.createElement('h1');  
heading.textContent = message;  
// add the heading the document  
document.body.appendChild(heading);
```

Teraz należy wykonać polecenie:

`npx tsc app.ts`

lub jeśli masz włączone polecenie: `npx tsc --watch` nie musisz nic robić

3. Uruchom plik **index.html** z LiveServer – w przeglądarce otrzymasz wynik!

3 zadania

Wyświetl tabelkę która będzie wyglądała jak poniżej: (napisz to w TypeScript ale w bardzo prosty sposób, dołącz css i/lub bootstrapa). W polu co się wykona ma znaleźć się Twoje wyjaśnienie

Polecenie	Co się wykona
node -v	
npm install typescript	
npx tsc -init	
npx tsc --watch	
npm init -y	
Co to jest TypeScript i kto go używa	Np: Google

Aby tabelka pojawiła się na stronie internetowej należy zmienić nieco plik app.ts.

UWAGA na znaki!!! szczególnie zobacz jaki znak otwiera string!!!

```
let message: string = `

|                     |               |
|---------------------|---------------|
| To działa           | Druga kolumna |
| Tabela jak malowana | To jest fajne |

`;
// create a new heading 1 element
let heading = document.createElement('div');
heading.innerHTML = message;
// add the heading the document
document.body.appendChild(heading);
```