

## Typescript

### SPIS TREŚCI

Spis treści .....	1
Cel zajęć.....	1
Rozpoczęcie .....	1
Uwaga .....	1
Wymagania.....	2
Instalacja Node.js .....	2
Konfiguracja projektu .....	2
Dynamicznie podłączany styl .....	5
Dynamicznie tworzony obszar z linkami .....	6
Commit projektu do GIT.....	8
Podsumowanie.....	8

### CEL ZAJĘĆ

Celem głównym zajęć jest zdobycie następujących umiejętności:

- konfiguracja środowiska do programowania i kompilacji aplikacji z wykorzystaniem TypeScript
- budowa skryptów z wykorzystaniem języka TypeScript

W praktycznym wymiarze uczestnicy zmodyfikują witrynę z LAB A (CSS Garden) do postaci single-page application (SPA), w której style będą wczytywane dynamicznie za pomocą skryptów napisanych w TypeScript.

### ROZPOCZĘCIE

Rozpoczęcie zajęć. Powtórzenie podstaw składni TypeScript.

Wejściówka?

### UWAGA

Ten dokument aktywnie wykorzystuje niestandardowe właściwości. Podobnie jak w LAB A wejdź do **Plik** -> **Informacje** -> **Właściwości** -> **Właściwości zaawansowane** -> **Niestandardowe** i zaktualizuj pola. Następnie uruchom ten dokument ponownie lub **Ctrl+A** -> **F9**.

## WYMAGANIA

W ramach LAB E zmodyfikowany zostanie kod z LAB A:

- w przeciwieństwie do pierwotnej postaci, witryna ma teraz składać się z jednej strony index.html oraz wielu stylów CSS;
- do strony podpięty jest skrypt JS z zewnętrznego pliku budowanego za pomocą webpacka
- w skrypcie napisanym w TS przechowywany jest stan aplikacji:
  - nazwa bieżącego stylu i jego plik;
  - słownik dostępnych stylów i ich plików;
- fragment strony zawierający linki do stron w innym stylu zostają zastąpione przez linki z wywołaniami aplikacji w TS;
- po wywołaniu aplikacji z linka, z DOM usunięte zostaje odwołanie do starego stylu CSS, a dodane zostaje odwołanie do stylu CSS powiązanego w słowniku aplikacji z wybranym stylem

Prowadzący omówi powyższe wymagania. Upewnij się, czy wszystko rozumiesz.

Tu umieść swoje notatki:

...notatki...

## INSTALACJA NODE.JS

Wejdź na stronę <https://nodejs.org/en/download/current>. Pobierz wersję Current -> Windows Binary (.zip) -> 64-bit. Rozpakuj archiwum do I:\node. Dodaj ten folder do zmiennej środowiskowej Path użytkownika.

Uruchom nowe okno wiersza poleceń. Wykonaj polecenie `npm -v`. Powinno zadziałać – udało się zainstalować NODE i NPM lokalnie dla użytkownika. Jeśli nie działa w Twoim terminalu, spróbuj w innym (CMD, PowerShell) albo wyloguj się i zaloguj ponownie (<https://ispot.link/hyttiooa>).

## KONFIGURACJA PROJEKTU

Powiel projekt z LAB A. Zaktualizuj strukturę projektu:

- pozostaw pojedynczy plik HTML
- pliki CSS umieść w osobnym podkatalogu
- utwórz plik `script.ts` o zawartości:

```
const msg: string = "Hello!";
alert(msg);
```

Do pliku HTML podłącz plik `dist/script.js`:

```
<head>
...
  <script src="dist/script.js" defer></script>
</head>
```

Zainicjalizuj pakiet NPM i zainstaluj paczkę mix oraz zależności:

```
npm install --save-dev laravel-mix ts-loader typescript  
npx tsc --init
```

Utwórz plik `webpack.mix.js`:

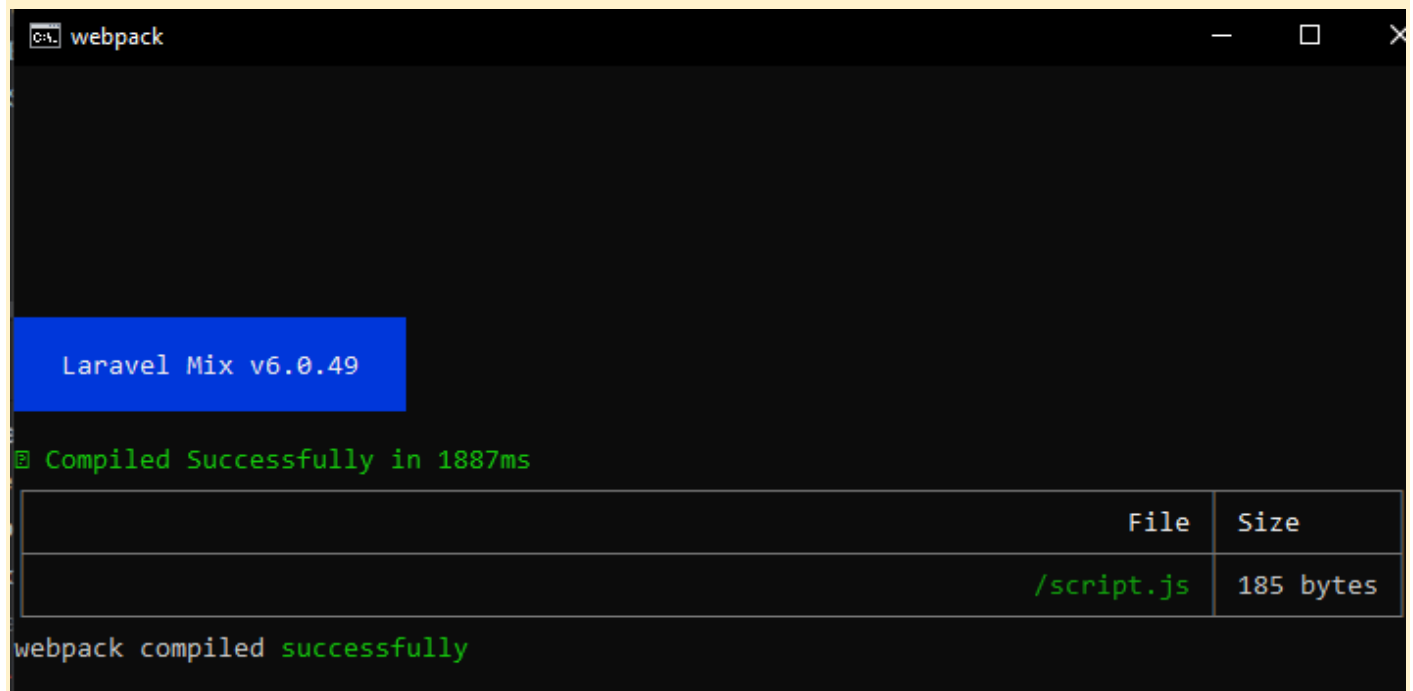
```
let mix = require('laravel-mix');  
mix.ts('script.ts', 'dist').setPublicPath('dist');
```

Włącz kompilację ciągłą:

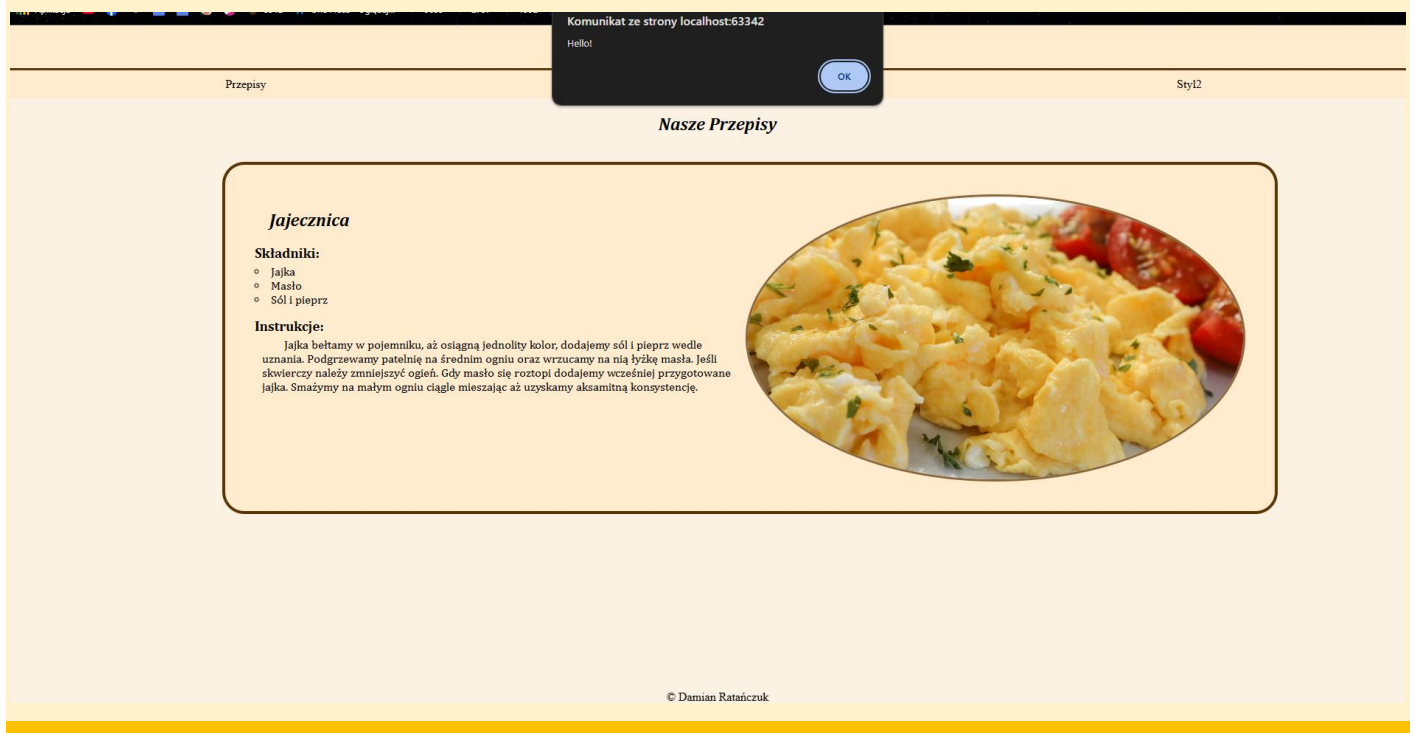
```
npx mix watch
```

Uruchom stronę w przeglądarce.

Wstaw zrzut ekranu komunikatu o sukcesie kompilacji:



Wstaw zrzut ekranu zbudowanej strony:



Punkty:	0	1
---------	---	---

## DYNAMICZNIE PODŁĄCZANY STYL

Wstaw zrzut ekranu kodu odpowiedzialnego za dynamiczne podłączanie stylu CSS:

```
function changeStyle(styleName: string): void { Show usages new *
  console.log(`Attempting to change style to: ${styleName}`);
  const stylePath :string = appState.styles[styleName];
  if (!stylePath) {
    console.error(`Style "${styleName}" not found.`);
    return;
  }

  const existingLink :Element| null = document.querySelector('link[data-app-style]');
  if (existingLink) {
    console.log('Removing existing style.');
    existingLink.remove();
  }

  const link :HTMLLinkElement = document.createElement('link');
  link.rel = 'stylesheet';
  link.href = stylePath;
  link.setAttribute('data-app-style', 'true');
  document.head.appendChild(link);

  appState.currentStyle = styleName;
  console.log(`Style successfully changed to: ${styleName}`);
}
```

Punkty:

0

1

Wstaw zrzut ekranu kodu odpowiedzialnego za dynamiczne generowanie linków do stylów:

```
function generateStyleLinks(): void { Show usages new *
  const container :HTMLElement| null = document.getElementById('style-links');
  if (!container) {
    console.error('Container for style links not found. ');
    return;
  }

  container.innerHTML = '';
  for (const styleName of Object.keys(appState.styles)) {
    const link :HTMLAnchorElement = document.createElement('a');
    link.href = '#';
    link.textContent = styleName;
    link.onclick = (event :MouseEvent ) :void => {
      event.preventDefault();
      changeStyle(styleName);
    };
    container.appendChild(link);
    container.appendChild(document.createElement('br'));
  }
}
```

Wstaw zrzut ekranu obszaru linków na stronie:

Przepisy Kulinarne		
Styl 1		Styl 2

Wstaw zrzut ekranu obszaru linków po dodaniu kolejnego stylu do tablicy stylów:

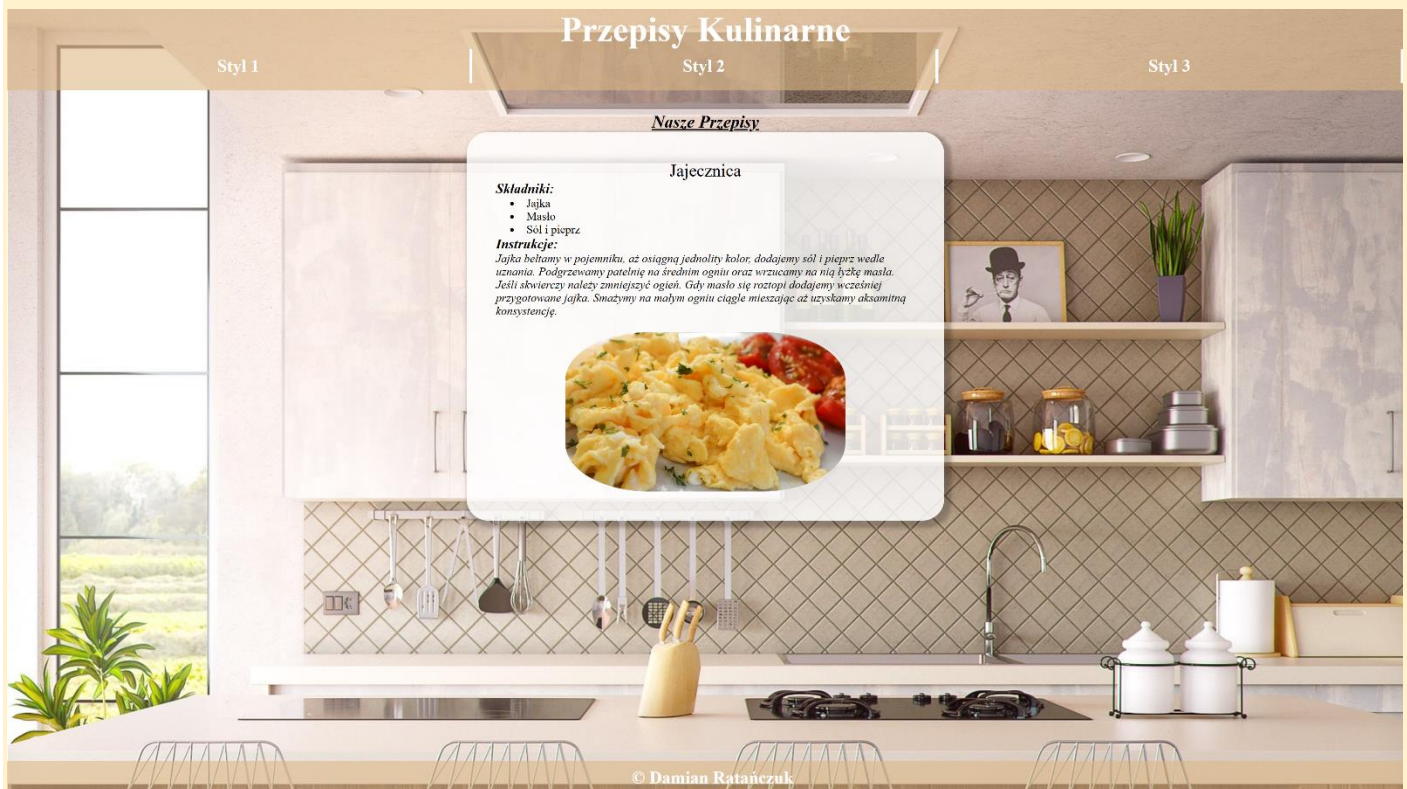
Przepisy Kulinarne		
Styl 1	Styl 2	Styl 3

Punkty:	0	1
---------	---	---

Wstaw zrzut ekranu kodu odpowiedzialnego za zmianę stylu po kliknięciu na link:

```
link.onclick = (event :MouseEvent) :void => {
  event.preventDefault();
  changeStyle(styleName);
};
container.appendChild(link);
```

Wstaw zrzut ekranu strony po kliknięciu na link zmiany stylu:



Wstaw zrzuty ekranu fragmentu kodu strony (narzędzia developerskie) z podłączonymi linkami przed i po zmianie stylu:

```
<head> == $0
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Przepisy Kulinarne</title>
<script src="dist/script.js" defer</script>
<link rel="stylesheet" href="./styles/page1.css" data-app-style="true">
</head>

<head> == $0
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Przepisy Kulinarne</title>
<script src="dist/script.js" defer</script>
<link rel="stylesheet" href="./styles/page2.css" data-app-style="true">
</head>
```

Punkty:	0	1
---------	---	---

## COMMIT PROJEKTU DO GIT

Zacommittuj i pushnij swoje rozwiązanie do repozytorium GIT.

Upewnij się, czy wszystko dobrze się wysłało. Jeśli tak, to z poziomu przeglądarki utwórz branch o nazwie `lab-e` na podstawie głównej gałęzi kodu.

Podaj link do brancha `lab-e` w swoim repozytorium:

<https://github.com/Xomil/Aplikacje-Internetowe-1/tree/main/lab-e>

## PODSUMOWANIE

W kilku zdaniach podsumuj zdobyte podczas tego laboratorium umiejętności.

...podsumowanie...

Zweryfikuj kompletność sprawozdania. Utwórz PDF i wyślij w terminie.