

Wprowadzenie do aplikacji Internetowych

laboratorium 3

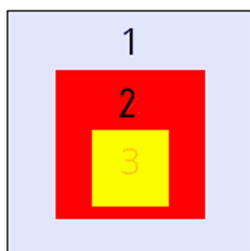
Cel zajęć:

Celem laboratorium jest przećwiczenie zagadnień związanych z programowaniem z wykorzystaniem JS. W drugiej części lab zapoznamy się z podstawami tworzenia aplikacji w Angular.

// Sekcja JS

Zadanie 0.

Stwórz stronę zawierającą 3 elementy np. divy/zdjęcia spozycjonowane tak jak na rysunku poniżej.



Niech naciśnięcie któregokolwiek z nich wyświetla odpowiedni komunikat:

Żółtego - „naciśnąłeś żółty o wartości 5”

Czerwonego – „naciśnąłeś czerwony o wartości 2”

Niebieski – „naciśnąłeś niebieski o wartości 1”

Jeśli suma naciśniętych wartości przekroczy 30 należy wyłączyć możliwość dodawania przez obiekt 2. Jeśli 50 usuwamy dodatkowo obsługę zdarzenia przez obiekt 3.

Niech na stronie będą 3 przyciski: StopStratPropagation, RemoveListener, AddListener oraz lista rozwijalna z wartościami pół.

Za pomocą tych przycisków zasymuluj działanie aplikacyjki polegające na czasowym (do kolejnego naciśnięcia przycisku) wyłączeniu obsługi zdarzenia dla obiektu 1.

RemoveListener – usuwa funkcje dla obiektu z aktualnie wybranego numeru obiektu.

AddListener – dodaje nowa funkcje kuku() { zwiększającą dodawana wartość o 3 jednostki).

Poniższe zadanie realizujemy w oparciu o przygotowaną przez Państwa stronę „Przepisy kulinarne”

Zadanie 1.

Zmodyfikuj stronę w taki sposób aby szczegóły przepisów (składniki oraz są przepis wyświetlane były tylko na życzenie). Scenariusz jest taki:

Po naciśnięciu na zdjęcie (przycisk typu button) lub samo zdjęcie – zdjęcie jest podmieniane na inne reprezentujące stan „owtarcia przepisu”. Tutaj proszę o zastosowanie jakiejś transformaty CSS (np. rozmycie, obrót wzdłuż osi Y itp.). Dodatkowo szczegóły przepisu wyświetlają się poniżej zdjęcia (również proponuje zastosowanie transformaty CSS). Naciśnięcie na zdjęcie (przycisk) drugi raz powoduje powrót do stanu pierwotnego. Podobnie powrót odbywa się w przypadku naciśnięcia innego zdjęcia celem wyświetlenia jego szczegółów. Zasada jest prosta szczegóły tylko jednego mogą być wyświetlane w danym momencie.

Zadanie 2

Niech część przepisów (zdjęcie, składniki oraz algorytm) będą ładowane dynamicznie za pomocą skryptu JS. Dane mogą być podawane za pomocą dynamicznie utworzonego formularza. Dodawanie aktywowane jest za pomocą naciśnięcia przycisku (najlepiej wysuwanego na żądanie z boku ekranu). Każdy nowo dodany element musi podlegać walidacji (już na etapie wypełniania formularza) – wymagane wypełnienie pola i np. format danych (link do zdjęcia). Każdy nowo dodany przepis musi mieć możliwość usunięcia (dodany dedykowany przycisk na dole zdjęcia).

// ES6

Zadanie 3.

Zweryfikuj czy nowe możliwości zdefiniowane w ES6 są wspierane przez Twoją przeglądarkę. Jeśli tak zrób poniższe zadanie, jeśli nie możesz samodzielnie napisać odpowiedni kod w ES6 i dokonać tranpilacji na tzw. Vanilla JavaScript za pomocą kompilatora Babel.

Kompilatora Babel możemy użyć w pracy nad projektem na kilka sposobów. Zawiera on na przykład wiersz poleceń, w którym można wykonać polecenia typu:

```
babel script.js --out-file script-compiled.js
```

Wersja przeglądarkowa jest również dostępna.

<https://babeljs.io/repl>

Ponadto możemy wykorzystać Babel jako zwykłą bibliotekę JS – wówczas kod ES6 umieszczamy w znacznikach script z typem „text/babel”.

```
1 <script src="node_modules/babel-core/browser.js"></script>
2 <script type="text/babel">
```

```
3 // twój kod ES6
4 </script>
```

Stwórz 2 tablice:

```
const tab1 = [1, 7, 12, 46, 85];
const tab2 = [10, 26, 75, 34];
```

Za pomocą operatora rozwijania (**spread syntax**) połącz te dwie tablice tak, by w wyniku uzyskać:

```
const tab3 = [1, 7, 12, 46, 85, 10, 26, 75, 34];
```

Posortuj te tablice odpowiednią metodą.

Następnie pobierz z tej tablicy największą i najmniejszą liczbę wyświetlając je na ekranie/console.

Za pomocą kompilatora babel dokonaj transpilacji stworzonego kodu. Porównaj obie wersje kodu.

// Wstęp do Angulara

Zadanie 4. Weryfikacja środowiska pracy

Sprawdź czy masz na swoim komputerze zainstalowane następujące oprogramowanie:

```
npm - npm -v
```

Zapoznaj się z krótkim tutorialiem dotyczącym instalacji i używania Angulara

<https://angular.io/guide/quickstart>

Na jego podstawie sprawdź czy masz zainstalowanego Angulara.

```
ng version lub ng -v
```

W przypadku gdyby nie było zainstalowanego angulara proszę o jego instalację.

W przypadku korzystania z komputerów w lab może pojawić się problem rejestracji ścieżki do zainstalowanego środowiska ng (Angulara). Proszę samodzielnie ustawić lokalną zmienną systemową PATH dla zainstalowanego ng.

Potrzebujemy także edytora tekstowego obsługującego TypeScript. Proponuje korzystać z Visual Studio Code.

Zapoznaj się z możliwościami środowiska CLI - <https://cli.angular.io/>

Zadanie 5. Podstawowym językiem tworzenia oprogramowania w Angularze jest TypeScript.

Przeglądaj tutorial dotyczący TypeScript

<https://www.typescriptlang.org/docs/tutorial.html>

Zapoznając się z najważniejszymi zasadami i konstrukcjami języka.

Zadanie 6. Stwórz nowy projekt w Angularze typu HelloWorld

Zapoznaj się z wygenerowaną strukturą projektu. Przeglądnij zawartość najważniejszych plików projektu.

Generalnie interesują nas tylko te cztery pliki:

src/app/app.module.ts,
src/app/app.component.ts,
src/index.html,
src/main.ts

W utworzony projekcie proszę dokonać następujących zmian, najlepiej, krok po kroku, po każdej całościowej i spójnej zmianie uruchamiając aplikację:

- zmieniamy komponent app na hello czyli:
 - zmieniamy nazwę pliku app.component.ts na hello.component.ts,
 - zmieniamy nazwę komponentu z AppComponent na HelloComponent
 - zmieniamy selektor (znacznik html) w szablonie i pliku index.html na hello-app,
 - zmieniamy treść szablonu na „Witaj Jan Kowalski”, gdzie Jan Kowalski powinno być zmiennymi imię i nazwisko przekazany do eksportowanej klasy wykorzystanej w szablonie: `<h1>Witaj {{imie}} {{nazwisko}}</h1>`,
- zmieniamy moduł app na main czyli:
 - zmieniamy nazwę pliku app.module.ts na main.module.ts,
 - zmieniamy nazwę modułu z AppModule na MainModule,
- **Dla ambitnych:** zmieniamy nazwę głównego katalogu aplikacji z src/app na src/hello, będzie wymagało również zmiany w pliku systemjs.config.js albo w pliku .angular-cli.json,
- zmieniamy port, na którym uruchamiana jest aplikacja na 5000:
<http://stackoverflow.com/questions/37154813/angular-cli-server-how-to-specify-default-port>

Takie zabawy z Hello World, żeby lepiej zrozumieć, jak wczytywane są komponenty, importowane moduły, jak jest uruchamiana aplikacja. To ważne aby swobodnie poruszać się w tym środowisku.

Zadanie 5. Niech na stronie głównej wyświetla się obok tekstu powitalnego pole tekstowe pozwalające na wpisywanie nazwy imienia. Po wpisaniu tam wartości tekst ten powinien pojawić się od razu na ekranie w tekście powitania.

Wykorzystując interpolację zweryfikuj wynik działania następujących operacji:

`{{2+2}}`

```
{{ imie.length }}
```

```
{{ imie.toUpperCase() }}
```

```
{{ a = 2+3 }}
```

```
{{ windows.location.href }}
```

Wyciągnij wnioski z otrzymanych wyników i ewentualnie tak zmodyfikuj kod aby zadziałał.

Zadanie 6. Stwórz nowy komponent o nazwie Produkty. Zmodyfikuj tak kod aby nowy komponent był komponentem wyświetlanym na stracie naszej aplikacji.

Komponent Produkty powinien wyświetlać listę produktów. Sama lista może być zaszyta bezpośrednio w kodzie w postaci tablicy obiektów. Pojedynczy obiekt Produkt powinien zawierać następujące pola:

Nazwa, ilość produktów, cena jednostkowa, opis oraz link do poglądowego zdjęcia.

Stwórz tablice w głównym komponencie o co najmniej 8 elementach.

Wyświetl zawartość tablicy obiektów w szablonie komponentu głównego - dyrektywa *ngFor

Wyświetlane zdjęcia produktów proszę wyświetlać jako okrągłe.

Przy każdym produkcie powinny znajdować się 2 przyciski + i - pozwalające na dodawanie produktu do listy zakupów lub jego oddawanie.

Jeśli wartość produktu znajdującego się w tablicy będzie wynosiła 0 to należy wyświetlić inny komunikat niż gdy ilość dostępnych produktów jest większa od 0.

W przypadku gdy ilość produktu spadnie do zera przycisk - powinien zostać ukryty. Nie chcemy przecież rezerwować produktu którego już nie posiadamy. - dyrektywy ngStyle lub ngClass

Gdy ilość produktu będzie zbliżała się do 0 (np. od 3 w dół) należy zaznaczyć to w sposób graficzny np. inne tło, kolor czcionki, wielkość fontów lub inny wizualny sposób.

Podobnie należy rozróżnić produkt o najniższej cenie jednostkowej oraz najwyższej - za pomocą dodatkowego obramowania obejmującego dany produkt - zielone - najdroższy, czerwone najtańszy.

Wypisz całą nazwę produktu dużymi literami -> skorzystaj z odpowiedniego typu pipe.

Wyświetl cenę produktu wraz z nazwą (lub znakiem płatniczym) skojarzonym z walutą np. USD - \$, euro lub złotówki.

Wyświetl również sumaryczną ilość aktualnie dostępnych produktów - jeśli wynosi on więcej niż 10 ma być wyświetlana na zielonym tle, jeśli poniżej 10 na czerwonym tle.