

# Wprowadzenie do aplikacji Internetowych

## laboratorium 3

### Cel zajęć:

Celem laboratorium jest przećwiczenie zagadnień związanych z programowaniem z wykorzystaniem JS. Skoncentrujemy się na aspektach zarządzania zdarzeniami oraz modyfikacji modelu DOM.

### Zadania na rozgrzewkę ( bez oceniania).

Zadanie 1. Po naciśnięciu na przycisk wyświetl okno dialogowe typu prompt pozwalające na wprowadzenie swojego imienia. Podane imię ma być wstawione do przygotowanej sekcji.

Zadanie 2. Przełączanie pomiędzy dwoma zdjęciami. Niech na stronie wyświetlone jest zdjęcie z gór w czerwonym obramowaniu. Po naciśnięciu przycisku zdjęcie zamieni się na zdjęcie morze w niebieskim obramowaniu. Kolejne naciśnięcie to ponowny powrót do wersji z górami itp.

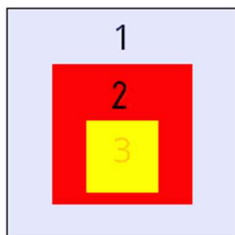
### Zadania punktowane.

**Sekcja podstawowa** (max 10 pkt) + 4pkt (sekcja podstawowa zadanie 11)

Zadanie 3. Napisz skrypt, który dodaje lub usuwa obsługę przycisku testowego. Naciśnięcie przycisku inkrementuje licznik przyciśnięć. Odpięcie zdarzenia resetuje licznik. **( 1pkt )**

Zadanie 4. Napisz skrypt, w którym za pomocą 2 przycisków modyfikujesz zawartość listy elementów. Jeden przycisk dodaje a drugi usuwa elementy z listy. Niech usuwane będzie zawsze pierwszy element listy. **( 1pkt )**

Zadanie 5. Stwórz stronę zawierającą 3 elementy np. divy/zdjęcia spozycjonowane tak jak na rysunku poniżej. **( 2pkt )**



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 jeszcze 2 przyciski: StopStartPropagation, Reset .

StopStartPropagation – włącza i wyłącza propagację.

Reset – wraca do stanu początkowego – reset licznika punktów oraz podpięcie funkcji obsługi zdarzeń typu click

zad. 6. Masz plik json zawierający kolekcję obiektów składających się z imię oraz liczbę lat. Niech kolekcje liczy co najmniej 10 osób. Odczytaj zawartość pliku.

---

Celem ćwiczenia jest weryfikacja umiejętności przetwarzania plików zewnętrznych z wykorzystaniem Fetch API. Aby można było przetwarzać pliki zewnętrzne (pliki json-owe) potrzebujemy serwera który będzie je hostował. Proponuje użycie json-server. W celu jego zainstalowania potrzeba posiadania jakiegoś menedżera pakietów: npm lub yarn.

Aby zainstalować JSON Server, należy otworzyć konsolę i wpisać poniższe polecenie:

```
npm install -g json-server
```

Flaga -g spowoduje, że serwer zostanie zainstalowany globalnie w systemie, co pozwoli uruchomić go z każdego miejsca.

Aby uruchomić JSON Server, proszę otworzyć wiersz poleceń i wpisać:

```
json-server db.json
```

W tym momencie serwer udostępnia pliki db.json i będzie można przy użyciu Fetch API przeczytać jego zawartość.

---

Napisz funkcję która na końcu każdego imienia doda wartość będąca ilością znaków w imieniu. Wyliczy średnio lat dla wszystkich użytkowników. Wyniki (zmodyfikowana lista imion wraz z liczbą lat oraz średnia z lat) wyświetl na ekranie przeglądarki. Dodatkowo w

sekcji poniżej wyświetli tylko obiektu o imionach zawierających literkę r w imieniu oraz drugi najstarszy i trzeci najmłodszy obiekt w tablicy. **( 2pkt )**

zad. 7. Walidacja formularza. Stwórz formularz do składania wniosku o emeryturę. Formularz zawiera następujące informacje: Imię, nazwisko, nr telefonu kontaktowego, płeć, rok urodzenia, planowany rok przejścia na emeryturę. Niech formularz weryfikuje czy osoba składająca wniosek ma uprawnienia do przejścia na emeryturę. Dodatkowo każde pole ma być walidowane pod kątem poprawności wprowadzanych danych. **(1pkt)**

zad 8. Napisz funkcję, która jako argumenty przyjmuje ciąg liczb. Liczba argumentów może być dowolna, ale nie mniejsza niż 2 (czyli musimy do funkcji podać minimum dwie liczby). Funkcja ma na celu zwrócić liczbę, która jest największa z podanego jako argumenty zbioru. Jeśli do funkcji zostaną podane argumenty, które nie są liczbami, powinien się wyświetlić odpowiedni komunikat informujący o tym, co należy podać do funkcji (w formie alertu lub console.log). **(1 pkt)**

**Zad 9.** Napisz skrypt, w którym za pomocą przycisk Dodaj będzie dodawał nowa pozycje do książki telefonicznej (imię, nr telefonu). Dane podajemy za pomocą formularza. Dodawana pozycja zawiera również przycisk ( w formie ikony – kosz) pozwalający na usuwanie elementy z książki adresowej. Przykład poniżej. **(2pkt)**

The screenshot displays a web application for managing a contact list. At the top, there are two entries for existing contacts. Each entry consists of a name and a phone number, followed by a green square button with a white trash icon for deleting the contact. The first contact is 'Grzegorz Rogs' with the number '500 600 700'. The second contact is 'Jacek Dajda' with the number '500 600 700'. Below these entries is a form for adding a new contact. It features two input fields: one for 'IMIĘ I NAZWISKO' (Name and Surname) and another for 'TELEFON' (Phone). To the right of these fields is a green button labeled 'Dodaj' (Add).

## Sekcja rozszerzona (max 9 pkt)

### Zad 10. Przetwarzanie plików zewnętrznych (5 pkt)

Mamy trzy pliki typu json. Pierwszy zawiera listę kategorii produktów przemysłowych i spożywczych. Dwa pozostałe pliki zawierają list konkretnych produktów przypisanych do kategorii. W obu plikach mogą pojawić się te same kategorie jak również produkty czyli dane mogą się powtarzać. Nie jest wymagań aby wszystkie pozycje z kategorii pojawiły się w każdym pliku. Niech lista kategorii liczy przynajmniej 5 pozycji.

Plik produktyB.json zawiera:

<pre> „Samochody” : [     „nazwa”: „Fiat”,     „nazwa”: „BMW”,     .... ]  „Stodczyce” : [     „nazwa”: „Czekolada”,     „nazwa”: „Cukierki”,     .... ]  „Soki” : [     „nazwa”: „Jupi”,     „nazwa”: „Kubuś”,     „nazwa”: „Hoop”,     .... ]  .... </pre>	<p>Plik produktyB.json zawiera:</p> <pre> „Samochody” : [     „nazwa”: „Fiat”,     „nazwa”: „Mercedes”,     .... ]  „Rowery” : [     „nazwa”: „Kross”,     „nazwa”: „Merida”,     .... ]  „Soki” : [     „nazwa”: „Jupi”,     „nazwa”: „Kubuś”,     .... ]  .... </pre>
--	---

Napisz aplikacyjne która odczyta dane z plików i na ich podstawie wyświetli dane w postaci rozwijalne listy. Elementy na poszczególnych gałęziach nie mogą się powtarzać.

Przykładowo wyglądałby to tak jak na zdjęciu poniżej (podane wartości są tylko przykładowe i w żaden sposób nie dotyczą treści zadania)

Sekcja menu	Sekcja główna
<div> <div> <div>▼</div> <div> <div>☑</div> <div>1</div> </div> </div> <div> <div>☑</div> <div>147814298</div> </div> <div> <div>☐</div> <div>299155077</div> </div> <div> <div>☐</div> <div>868566152</div> </div> <div> <div>☐</div> <div>1187187215</div> </div> <div> <div>☑</div> <div>743084005</div> </div> <div> <div>▼</div> <div> <div>☑</div> <div>2</div> </div> <div> <div>☑</div> <div>1</div> </div> <div> <div>☑</div> <div>1765323542</div> </div> </div> <div> <div>▼</div> <div> <div>☐</div> <div>3</div> </div> <div> <div>☐</div> <div>248207181</div> </div> </div> <div> <div>&gt;</div> <div> <div>☐</div> <div>45</div> </div> <div> <div>&gt;</div> <div> <div>☐</div> <div>602344033</div> </div> </div> <div> <div>▼</div> <div> <div>☐</div> <div>699114243</div> </div> <div> <div>☐</div> <div>-35562339</div> </div> </div> </div></div>	<div>147814298, 743084005, 1, 1765323542</div>

Wybrane elementy listy niech wyświetlają się w sekcji głównej na stronie. W przypadku odznaczenia pozycji z listy rozwijalnej dana pozycja znika oczywiście natychmiast z sekcji głównej. Analogicznie jest przy zaznaczaniu – dołącza do pozycji w sekcji głównej

**Zadanie 11.** Zadanie: Gra "Balloons". **(8 punktów)**

Twoim zadaniem jest implementacja gry widocznej na filmie i screenshocie.

Na wejściu dostaniesz szablon projektu z gotowym plikiem HTML, CSS oraz dźwiękami gry. Twoim zadaniem jest tylko i wyłącznie implementacja pliku script.js

Poniżej znajdziesz założenia gry i punktację

**# Wersja podstawowa, zawiera (4 punktów) - czesc podstawowa**

- podstawowy mechanizm strzelania do balonów
- balony pojawiają się jeden po drugim w obszarze gry, na 2 sekundy
- gra w 30 rundach (30 balonów)
- punktacja statyczna - za każdy balon naliczane punkty, za niecelny strzał ujemne punkty

**# Dodatki (2 punkty)**

- ilość punktów zależy od reakcji (im dłużej tym mniej punktów)
- odgrywanie dźwięków dla niecelnego strzału i celnego
- miganie na czerwono tła by zaznaczyć trafienie w balona
- czas pojawienia się balonu zależy od aktualnej rundy, im dalsza runda tym mniejszy czas
- obliczanie dodatkowych statystyk i wyświetlanie ich nad planszą do gry

**# Ranking (highscores) (2 punktów)**

- na wstępie gra prosi o podanie nicka (do skutku) - potem ten nick jest wyświetlany nad planszą do gry
- po zakończeniu gry pobieramy zapisane na serwerze wyniki
- dodajemy swój wynik, sortujemy, odcinamy pierwsze 7 wyników i zapisujemy znowu na serwerze
- ranking zawiera pozycję, nicka, ilość punktów i datę wpisu
- przy prezentacji highscores nie ma możliwości strzelania a kursor zmienia się na zwykły
- pojawia się guzik umożliwiający powrót do nowej rozgrywki