

# Multimedialne i responsywne strony WWW

## laboratorium 4

### Java Script

#### Cel zajęć:

##### Zadanie 1.

A . Zadeklaruj tablicę `tab` z 5 losowymi wyrazami. Wypisz w konsoli tą tablicę i jej długość. Następnie wypisz pierwszy i ostatni element oraz ich długość.

B. Robiąc pętlę po powyższej tablicy wypisz każdy element dużymi literami oraz jego długość.

Dodaj na koniec tekst „PAW”. Skorzystaj z pętli `for`, `for of` i `forEach` oraz funkcji map.

C. Do tabeli z 1 zadania dodaj na końcu i początku po 1 nowym elemencie.

Wykorzystaj odpowiednie metody. Po dodaniu elementów wypisz długość tablicy oraz całą tablicę za pomocą `console.table()`

D Usuń z tablicy 3 element ale tylko wtedy gdy ma ona tyle elementów.

E. Stwórz tablice 6 obiektów przechowujących imię i wiek Przykładowa zapis:

```
const Users = [  
  {name : "Grzegorz", age: 102},  
  {name : "Ula", age: 18},
```

Za pomocą dowolnej metody (petli lub funkcji) wypisz w konsoli imiona użytkowników, którzy są pełnoletni. Spróbuj zrobić to zadanie na tak wiele sposobów jak potrafisz.

##### Zadanie 2.

W sekcji zadanie 2 znajdziesz plik cities.js`.

W pliku cities.js znajduje się tablica cities z miastami - każde miast to obiekt.

Przy rozwiązywaniu punktów staraj się unikać klasycznej pętli for/while.

Wypisz ile miast jest w Polsce

Wypisz ile jest ludzi w tych miastach

Wypisz pierwsze miasto, w którym jest ponad 10000 ludzi

Wypisz miasta, w których ludzi jest ponad średnią

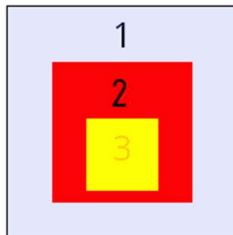
Wypisz nazwy wszystkich miast, w których jest ponad 10000 ludzi

Wypisz czy więcej jest miast z > 10000 ludzi czy mniejszych

Zadanie 3 Przetaczanie pomiędzy dwoma zdjęciami. Niech na stronie wyświetlone jest zdjęcie z gór w czerwonym obramowaniu. Po naciśnięci przycisku zdjęcie zamieni się na zdjęcie morze w niebieskim obramowaniu. Kolejne naciśnięcie to ponowny powrót do wersji z górami itp.

Zadanie 4. Napisz skrypt, który dodaje lub usuwa obsługę przycisku testowego. Naciśnięcie przycisku inkrementuje licznik przyciśnięć. Odpięcie zdarzenia resetuje licznik.

Zadanie 5. 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 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 kolekcje 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 jakiegos managera 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 będącą 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 wyświetl tylko obiektu o imionach zawierających literkę r w imieniu oraz drugi najstarszy i trzeci najmłodszy obiekt w tablicy.

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.

**Zad 8.** Napisz skrypt, w którym za pomocą przycisk Dodaj będzie dodawał nową pozycję do książki telefonicznej (imię, nr telefonu). Dane podajemy za pomocą formularza. Dodawana pozycja zawiera również przycisk (w formie ikony – kosz) pozwalający na usuwanie elementów z książki adresowej. Przykład poniżej

The screenshot displays a web interface for managing a contact list. At the top, there are two existing contact entries, each in a white box with a light blue border. The first entry shows 'Grzegorz Rogs' with the phone number '500 600 700' and a green trash icon on the right. The second entry shows 'Jacek Dajda' with the same phone number and a green trash icon. Below these entries is a form to add a new contact. The form has two input fields: 'IMIĘ I NAZWISKO' (Name and Surname) and 'TELEFON' (Phone). To the right of these fields is a green button labeled 'Dodaj' (Add).

Niech wszystkie pola podlegają walidacji. Nie jest możliwe dodanie pustej pozycji do listy.

### zad. 10

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.

### zad. 11.

Napisz aplikację w której będzie przesuwala kulke w miejsce gdzie została naciśnięty prawy klawisz myszki.

### zad. 12. Przetwarzanie plików zewnętrznych

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.

Treść właściwa zadania.

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.

Przykład: Plik produktyA.json zawiera:

```
„Samochody” : [  
  „nazwa”: „Fiat”,  
  „marka”: „BMW”,  
  .....  
]
```

Plik produktyB.json zawiera:

```
Plik produktyB.json zawiera:  
„Samochody” : [  
  „nazwa”: „Fiat”,  
  „marka”: „Mercedes”,  
  .....  
]
```

„Słodczyce” : [ „nazwa”: „Czekolada”, „nazwa”: „Cukierki”, ..... ]	] „Rowery” : [ „nazwa”: „Kross”, „nazwa”: „Merida”, ..... ]
--	--

Napisz aplikację, która odczyta dane z plików i na ich podstawie wyświetli dane w postaci rozwijalnej 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)

---

▼

1

☒ 147814298
   
☐ 299155077
   
☐ 868566152
   
☐ 1187187215
   
☒ 743084005

▼

2

☒ 1
   
☒ 1765323542

▼

3

☐ 248207181

>

45

☐ 602344033

▼

699114243

☐ -35562339