

JS I DOM NA PRZYKŁADZIE LISTY TODO

SPIS TREŚCI

Spis treści	1
Cel zajęć.....	1
Rozpoczęcie.....	1
Uwaga	2
Wymagania.....	2
Strona HTML	3
Klasa Todo	3
Dodawanie pozycji listy	3
Usuwanie pozycji listy	4
Edycja pozycji listy	5
Odczyt / Zapis LocalStorage	7
Wyszukiwanie.....	8
Commit projektu do GIT.....	9
Podsumowanie.....	10

CEL ZAJĘĆ

Celem głównym zajęć jest zdobycie następujących umiejętności:

- przemieszczania się po drzewie DOM;
- dodawania, usuwania, edytowania elementów drzewa DOM.

W praktycznym wymiarze utworzona zostanie dynamiczna lista czynności do zrobienia (lista To Do).

ROZPOCZĘCIE

Rozpoczęcie zajęć. Powtórzenie metod przemieszczania się po drzewie DOM.

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 B przygotowane powinny zostać:

- pojedyncza strona HTML ze skryptem ładowanym z zewnętrznego pliku JS
- lista zadań
- na dole listy pole tekstowe do dodawania nowych zadań, pole typu data/czas do określenia terminu wykonania zadania, przycisk dodawania zadania
- walidacja nowych zadań: co najmniej 3 znaki, nie więcej niż 255 znaków, data musi być pusta albo w przyszłości
- na górze listy pole wyszukiwarki
- po wpisaniu w wyszukiwarkę co najmniej 2 znaków na liście wyświetlają się wyłącznie pozycje zawierające wpisaną w wyszukiwarkę frazę
- wyszukiwana fraza zostaje wyróżniona w każdym wyniku wyszukiwania
- kliknięcie na dowolną pozycję listy zmienia ją w pole edycji; kliknięcie poza pozycję listy zapisuje zmiany
- obok każdej pozycji listy znajduje się przycisku Usuń / Śmiertnik
- wpisy na liście zapisują się do Local Storage
- po odświeżeniu strony lista wypełnia się wpisami z Local Storage

Mockupy:

The left mockup shows a search bar at the top. Below it is a list of tasks:

<input type="checkbox"/> chocolate	2000-01-01	
<input type="checkbox"/> macaroon		
<input type="checkbox"/> chupa chups		
<input type="checkbox"/> candy canes	2000-01-05	
<input type="checkbox"/> bon bons		

Below the list are three input fields: "do zrobienia...", a date picker, and a "Zapisz" button.

The right mockup is similar but highlights the "chupa chups" task. The task details are shown in a larger box below the list:

<input type="checkbox"/> chupa chups			Zapisz	
--------------------------------------	--	--	--------	--

Below the highlighted task are the same input fields as the left mockup.

The search bar contains the placeholder "do zrobienia...". Below it is a date picker and a "Zapisz" button.

The list of tasks includes:

<input type="checkbox"/> macaroon
<input type="checkbox"/> bon bons

STRONA HTML

Prace rozpocznij od implementacji HTML z danymi wpisanymi „na sztywno”. Upewnij się, że wstawione zostały wszystkie wymagane elementy – pole wyszukiwarki, lista, pole dodawania, przycisk usuwania. To laboratorium koncentruje się na JS, więc może być ładne, ale nie musi. Nie trać za dużo czasu na CSS.

Wstaw zrzut ekranu przedstawiający stronę HTML z polem wyszukiwarki, listą, polem dodawania, przyciskami usuwania:



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

KLASA TODO

Pierwszym instynktem może być chęć dodania zachowań bezpośrednio do elementów listy w drzewie DOM. Chociaż na krótką metę wydaje się być to najprostsze rozwiązanie, za chwilę okaże się krótkowzroczne i trudne do implementacji przy kolejnych punktach 😊

Najlepszym sposobem rozwiązania tego laboratorium jest utworzenie klasy Todo (albo po prostu obiektu z kilkoma metodami). Bez względu na przyjętą strategię, należy w tym nowoutworzonym bycie utworzyć tablicę `tasks` oraz metodę `draw()`, która wyczyści `div` z obecną wizualizacją zadań do zrobienia i wygeneruje ją na nowo na podstawie tablicy `tasks`.

W celu sprawdzenia poprawności działania, najlepiej dostać się do tablicy `tasks` i edytować jej zawartość, po czym ręcznie wywołać metodę `draw()`. Jeśli zawartość listy wyrenderuje się na nowo poprawnie – możemy iść dalej!

Zaimplementuj dodawanie, usuwanie, edycję pozycji listy – wszystko modyfikujące tablicę `tasks` i wywołujące na koniec metodę `draw()`.

DODAWANIE POZYCJI LISTY

Wstaw zrzut ekranu listy przed dodaniem nowego zadania:

MOJA LISTA ZADAŃ

Szukaj zadań...



Nowe zadanie

dd.mm.yyyy, -- : --



Dodaj zadanie

Wstaw zrzut ekranu listy po dodaniu nowego zadania:

MOJA LISTA ZADAŃ

Szukaj zadań...



nowe zadanie | 26.10.2024, 11:30



Nowe zadanie

dd.mm.yyyy, -- : --



Dodaj zadanie

Punkty:

0

1

USUWANIE POZYCJI LISTY

Wstaw zrzut ekranu listy przed usunięciem wybranego zadania:

MOJA LISTA ZADAŃ

Szukaj zadań... 

nowe zadanie | 26.10.2024, 11:30 

drugie zadanie 

trzecie zadanie 

Nowe zadanie

dd.mm.aaaa, --:-- 

Dodaj zadanie

Wstaw zrzut ekranu listy po usunięciu zadania:

MOJA LISTA ZADAŃ

Szukaj zadań... 

trzecie zadanie 

Nowe zadanie

dd.mm.aaaa, --:-- 

Dodaj zadanie

Punkty:

0

1

EDYCJA POZYCJI LISTY

Wstaw zrzut ekranu listy przed edycją wybranego zadania:

MOJA LISTA ZADAŃ

Szukaj zadań...



zadanko | 22.10.2024, 22:30



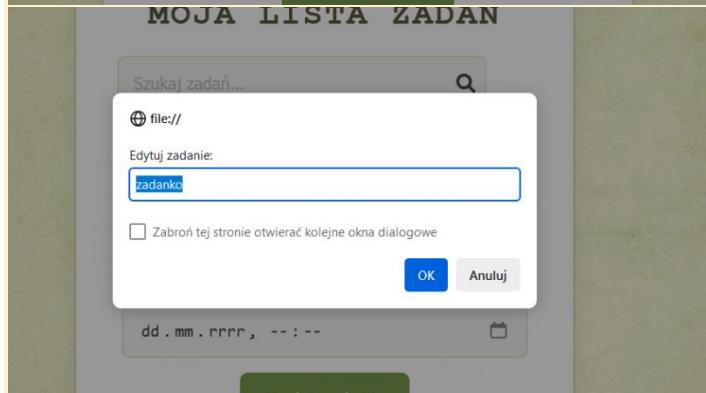
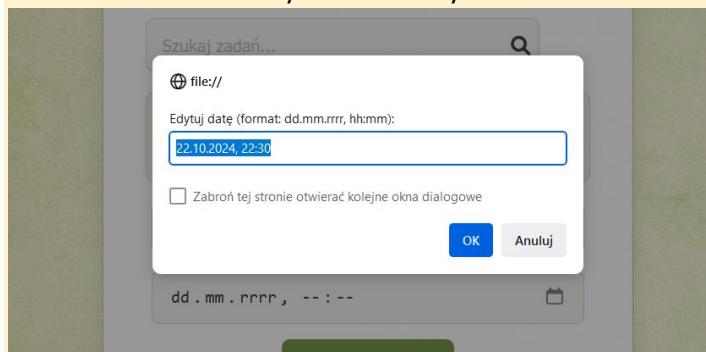
Nowe zadanie

dd.mm.yyyy, --:--



Dodaj zadanie

Wstaw zrzut ekranu listy w trakcie edytowania zadania i daty:



Wstaw zrzut ekranu listy po edycji zadania i daty. Upewnij się, że dane się zapisały i zadanie jest zmienione:

MOJA LISTA ZADAŃ

Szukaj zadań... 

edytowane | 24.10.2024, 12:45 

Nowe zadanie

dd.mm.yyyy, --:-- 

Dodaj zadanie

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

ODCZYT / ZAPIS LOCALSTORAGE

Zastosowanie klasy Todo w realizacji tego laboratorium pozwala w bardzo łatwy sposób odczytywać i zapisywać stan listy do pamięci przeglądarki. Wystarczy serializacja / deserializacja za pomocą JSON.parse() i JSON.stringify().

Wstaw zrzuty ekranu przedstawiające wygląd listy i zawartość local storage gdy na liście są pewne zadania:

MOJA LISTA ZADAŃ

Szukaj zadań... 

polska | 23.10.2024, 11:00 

francja 

niemcy 

anglia | 03.11.2024, 13:00 

Nowe zadanie

dd.mm.yyyy, --:-- 

Dodaj zadanie

```
▼ tasks:Array
  ▼ 0:Object
    text:"polska"
    date:"23.10.2024, 11:00"
    completed:false
    ▶ __proto__:Object
  ▼ 1:Object
    text:"francja"
    date:null
    completed:false
    ▶ __proto__:Object
  ▼ 2:Object
    text:"niemcy"
    date:null
    completed:false
    ▶ __proto__:Object
  ▼ 3:Object
    text:"anglia"
    date:"03.11.2024, 13:00"
    completed:false
    ▶ __proto__:Object
    length:4
    ▶ __proto__:Array
```

Wstaw zrzuty ekranu przedstawiające wygląd listy i zawartość local storage po dodaniu nowej pozycji listy. Upewnij się, że widoczne w local storage są dane dotyczące nowego zadania:

The screenshot shows a user interface for a task list application. On the left, there's a search bar labeled "Szukaj zadań..." with a magnifying glass icon. Below it is a list of five tasks, each with a checkbox, a text field, a date field, and a delete button. The tasks are:

- ☐ polska | 23.10.2024, 11:00
- ☐ francja
- ☐ niemcy
- ☐ anglia | 03.11.2024, 13:00
- ☐ Berlin

Below the tasks is a button labeled "Nowe zadanie". Underneath that is a date input field with placeholder "dd.mm.aaaa, --:--" and a calendar icon. At the bottom is a green button labeled "Dodaj zadanie".

On the right, a modal window displays the "tasks" array in a hierarchical tree view. The array contains five objects, indexed 0 to 4. The object at index 4, which corresponds to the task "Berlin", is highlighted with a blue background.

```
▼ tasks:Array
  ▼ 0:Object
    text:"polska"
    date:"23.10.2024, 11:00"
    completed:false
    ► __proto__:Object
  ▼ 1:Object
    text:"francja"
    date:null
    completed:false
    ► __proto__:Object
  ▼ 2:Object
    text:"niemcy"
    date:null
    completed:false
    ► __proto__:Object
  ▼ 3:Object
    text:"anglia"
    date:"03.11.2024, 13:00"
    completed:false
    ► __proto__:Object
  ▼ 4:Object
    text:"Berlin"
    date:null
    completed:false
    ► __proto__:Object
    length:5
    ► __proto__:Array
```

Punkty:

0

1

WYSZUKIWANIE

Na koniec zostało filtrowanie wyników. Proponowanym podejściem do tego tematu jest umieszczenie w klasie Todo właściwości `term` – frazy wyszukiwanej przez użytkownika. Następnie można utworzyć metodę `getFilteredTasks`, albo getter `filteredTasks`, która zwróci będzie te elementy tablicy `tasks`, które odpowiadają zapytaniu. Można użyć funkcji wyższego rzędu `filter()`.

Wstaw zrzut ekranu listy, gdy pole wyszukiwania jest puste:

A screenshot of a search bar. It contains the placeholder text "Szukaj zadań..." and a magnifying glass icon to its right.

Wstaw zrzut ekranu listy, gdy w polu wyszukiwania wpisano wystarczająco dużo znaków, by zadziałało filtrowanie. Upewnij się, że chociaż 2 wyniki będą wciąż widoczne:

A screenshot of a web-based task management application. On the left, there is a sidebar with a search bar containing the text 'po'. Below the search bar are two items: 'polska' and 'powidla', each with a checkbox and a trash can icon. A button labeled 'Nowe zadanie' (New task) is present, along with a date input field and a 'Dodaj zadanie' (Add task) button.

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

Wstaw zrzut ekranu przedstawiający podświetlenie szukanej frazy w wynikach wyszukiwania, przykładowo dla frazy ko i zadania Ala ma kota otrzymujemy: Ala ma **kota**:

A screenshot of the same task management application. The search bar now contains 'ko'. The result 'ala ma **kota**' is shown with the word 'kota' highlighted in yellow. The sidebar elements remain the same: 'Nowe zadanie', date input, and 'Dodaj zadanie' button.

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

COMMIT PROJEKTU DO GIT

Zacommituj 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-b na podstawie głównej gałęzi kodu.

Podaj link do brancha lab-b w swoim repozytorium:

...link, np. <https://github.com/kapikoks1/a1/tree/main/l2>

PODSUMOWANIE

W kilku zdaniach podsumuj zdobyte podczas tego laboratorium umiejętności.

Ciekawe zadanie, nauczyłem się wyszukiwać fraze po wpisaniu kilku liter i przede wszystkim pamiętalem o zapisie sprawozdania do pdf ...

Zweryfikuj kompletność sprawozdania. Utwórz PDF i wyślij w terminie.