

JS I DOM NA PRZYKŁADZIE LISTY TODO

SPIS TREŚCI

Spis treści.....	1
Cel zajęć.....	1
Rozpoczęcie.....	1
Uwaga.....	2
Wymagania.....	2
Strona HTML.....	2
Klasa Todo.....	3
Dodawanie pozycji listy.....	3
Usuwanie pozycji listy.....	4
Edycja pozycji listy.....	4
Odczyt / Zapis LocalStorage.....	5
Wyszukiwanie.....	6
Commit projektu do GIT.....	6
Podsumowanie.....	7

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 Usun / Śmietnik
- wpisy na liście zapisują się do Local Storage
- po odświeżeniu strony lista wypełnia się wpisami z Local Storage

Mockupy:

Mockup of the application interface showing a list of tasks. The interface includes a search bar at the top. Below it, a list of tasks is displayed, each with a checkbox, a name, a date, and a delete button (trash icon). The tasks are: chocolate (2000-01-01), macaroon, chupa chups, candy canes (2000-01-05), and bon bons. At the bottom, there is a text input field labeled 'do zrobienia...', a date input field with a calendar icon and the date '2000-01-01', and a 'Zapisz' button.

Mockup of the application interface showing a task being edited. The interface is similar to the previous one, but the 'chupa chups' task is selected, and its details are shown in a form below the list. The form includes a text input field with 'chupa chups', a date input field with a calendar icon, and a 'Zapisz' button. The date input field is currently empty.


Mockup of the application interface showing search results. The search bar contains the text 'on'. Below it, the list of tasks is filtered to show only those containing the search term. The tasks shown are 'macaroon' and 'bon bons', both highlighted in yellow. The interface also includes the same bottom section with the 'do zrobienia...' input, date input, and 'Zapisz' button.




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 łatwe, ale nie musi. Nie trać za dużo czasu na CSS.

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

Lista zadań

 Wyszukaj zadania...

Przykładowe zadanie 1	Termin: 2024-10-31	
Przykładowe zadanie 2	Termin: 2024-11-05	
Przykładowe zadanie 3	Termin: brak	

 dd.mm.yyyy --:--

Dodaj zadanie

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:

Lista zadań

 Wyszukaj zadania...

Nowe zadanie

 dd.mm.yyyy --:-- 

Dodaj zadanie

Wstaw zrzut ekranu listy po dodaniu nowego zadania:

Lista zadań

Q

Wyszukaj zadania...

Zrobić AI ASAP

Termin: 2024-10-25T08:00

Nowe zadanie

dd.mm.yyyy --:--


Dodaj zadanie



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

USUWANIE POZYCJI LISTY


Wstaw zrzut ekranu listy przed usunięciem wybranego zadania:


Lista zadań

 Wyszukaj zadania...

Zrobić AI ASAP	Termin: 2024-10-25T08:00	
Rzucić wszystko i wyjechać w Bieszczady	Termin: brak	

Nowe zadanie

 dd.mm.yyyy --:--



Dodaj zadanie

Wstaw zrzut ekranu listy po usunięciu zadania:

Lista zadań

 Wyszukaj zadania...

Zrobić AI ASAP	Termin: 2024-10-25T08:00	
----------------	--------------------------	---

Nowe zadanie

 dd.mm.yyyy --:--



Dodaj zadanie

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

EDYCJA POZYCJI LISTY

Wstaw zrzut ekranu listy przed edycją wybranego zadania:

Lista zadań

Q

Wyszukaj zadania...

Zrobić AI ASAP

Termin: 2024-10-25T08:00

Nowe zadanie

dd.mm.yyyy --:--

Dodaj zadanie

Wstaw zrzut ekranu listy w trakcie edytowania zadania i daty:

Lista zadań

 Wyszukaj zadania...

Zrobić AI ASAP

25.10.2024 02:00 



Nowe zadanie

 dd.mm.yyyy --:-- 

Dodaj zadanie

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

Lista zadań

 Wyszukaj zadania...

Zrobić AI (kiedyś)

Termin: brak



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:

Lista zadań

Wyszukaj zadania...

Zrobić AI (kiedyś)

Termin: brak

Nowe zadanie

dd.mm.rrrr --:--

Dodaj zadanie

DevTools is now available in Polish! Always match Chrome's language Switch DevTools to Polish Don't show again

Elements

Console

Sources

Network

Performance

Memory

Application

Security

Lighthouse

Recorder

>>

Application

Manifest

Service workers

Storage

Storage

Local storage

file://

Session storage

IndexedDB

Cookies

Private state tokens

Interest groups

Shared storage

Cache storage

Storage buckets

Background services

Back/forward cache

Background fetch

Background sync

Bounce tracking mitigati...

Notifications

Payment handler

Periodic background sync

Speculative loads

Push messaging

Reporting API

Frames

top

file://

Origin file://

Key	Value
tasks	[{"text": "Zrobić AI (kiedyś)", "date": "brak"}]

▼ [{"text": "Zrobić AI (kiedyś)", "date": "brak"}]

0: {"text": "Zrobić AI (kiedyś)", "date": "brak"}

Strona 10 z 16

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:

Lista zadań



Zrobić AI (kiedyś)

Termin: brak

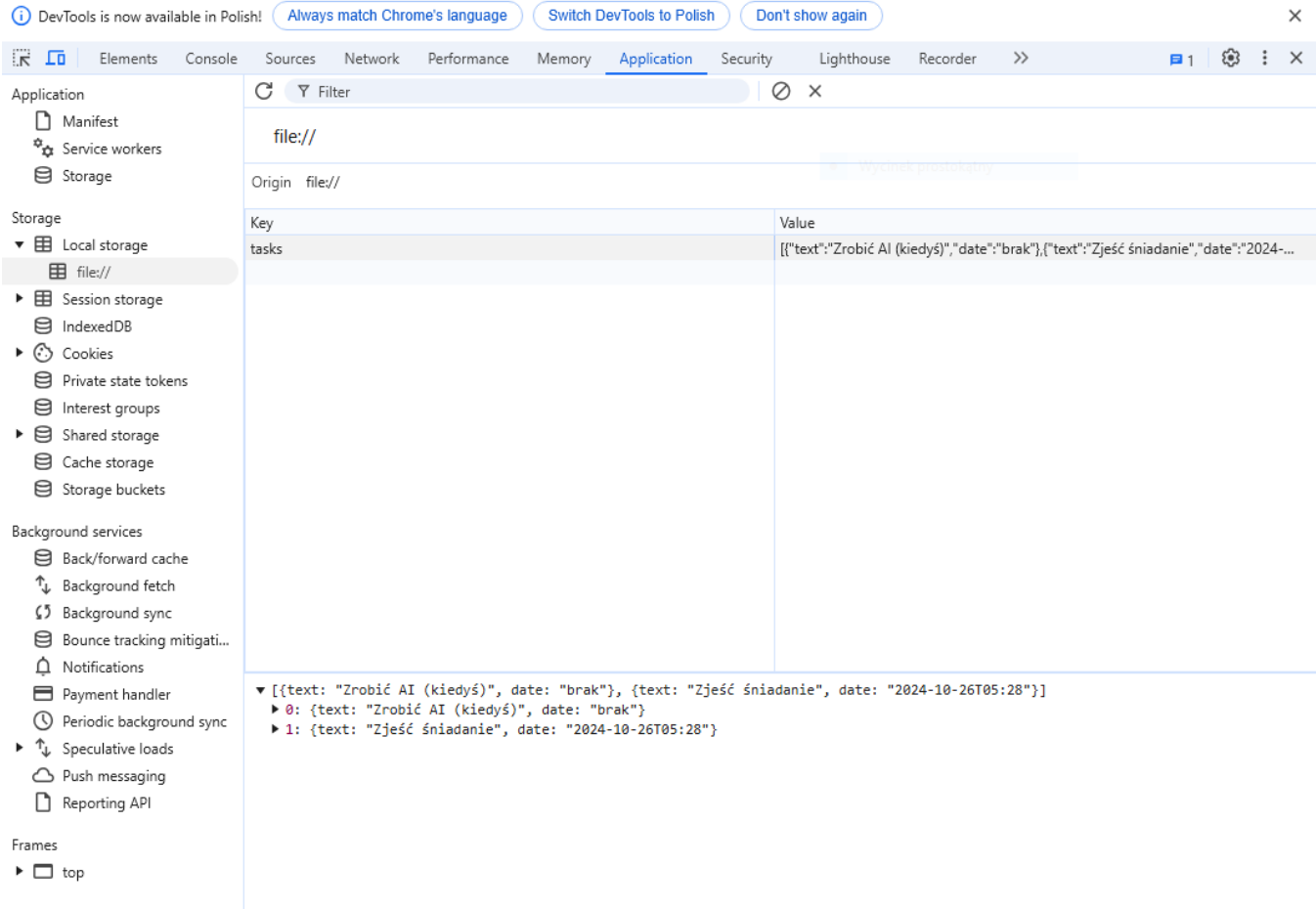


Zjeść śniadanie

Termin: 2024-10-26T05:28



Dodaj zadanie



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 zwracać będzie te elementy tablicy `tasks`, które odpowiadają zapytaniu. Można użyć funkcji wyższego rzędu `filter()`.

Wstaw zrzut ekranu listy, gdy pole wyszukiwania jest puste:

Lista zadań

Wyszukaj zadania...

Zrobić AI (kiedyś)	Termin: brak	
Zjeść śniadanie	Termin: 2024-10-26T05:28	
Zrobić POI (może)	Termin: 2024-10-26T22:30	
Zrobić Projekt	Termin: brak	
Policzyć klocki LEGO	Termin: 2024-10-30T14:28	

Nowe zadanie

dd.mm.yyyy

--:--




Dodaj zadanie

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



Lista zadań

Q

zr

Zrobić AI (kiedyś)	Termin: brak	
Zrobić POI (może)	Termin: 2024-10-26T22:30	
Zrobić Projekt	Termin: brak	

Nowe zadanie

 dd . mm . rrrr -- : -- 

Dodaj zadanie

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**:

Lista zadań



ko

Ala ma kota

Termin: brak



Nowe zadanie



dd.mm.rrrr --:--



Dodaj zadanie

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:

<https://github.com/Saggy28/AI1/tree/lab-b>

PODSUMOWANIE

W kilku zdaniach podsumuj zdobyte podczas tego laboratorium umiejętności.

Nauczyłem się podstaw programowania w JavaScriptcie. Dowiedziałem się jak go połączyć z HTML'em. Nauczyłem się pisać listę zadań todo. Dodawać, usuwać i wyszukiwać w niej wartości. Dowiedziałem się również jak znaleźć logi i LocalStorage w DevTools.

Zweryfikuj kompletność sprawozdania. Utwórz PDF i wyślij w terminie.