

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	5
Dodawanie pozycji listy	5
Usuwanie pozycji listy	7
Edycja pozycji listy.....	9
Odczyt / Zapis LocalStorage	10
Wyszukiwanie.....	12
Commit projektu do GIT.....	14
Podsumowanie.....	14

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 first mockup shows a simple list of tasks:

checkbox	task	date	trash icon
<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		

The second mockup shows the 'chupa chups' task selected:

checkbox	task	date	trash icon
<input type="checkbox"/>	chupa chups		
<input type="checkbox"/>	macaroon		
<input type="checkbox"/>	candy canes	2000-01-05	
<input type="checkbox"/>	bon bons		

The search bar contains 'do zrobienia...' and 'on'. The list of tasks includes:

checkbox	task	date	trash icon
<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** – to jest laboratorium z JS.

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

TODO List

Zarządzaj swoimi zadaniami efektywnie

9 lis 2025, 18:05

Sprawdzenie
Sprawdzeni zadań na obecny weekend

Programowanie
napisać program dotyczący mierzenia czasu pracy CPU i GPU
1 lis 2025, 16:04

Testowe
Testowy opis do zadania - !

Aplikacje Internetowe
Lab 2 do wykonania
25 paź 2025, 12:00

Łącznie zadań: 5

+ Dodaj nowe zadanie

Tytuł zadania:
Wprowadź tytuł zadania (3-100 znaków)...

Opis zadania (opcjonalnie):
Szczegółowy opis zadania (max 500 znaków)...

Termin wykonania (opcjonalnie):
dd.mm.rrrr --:--

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:



Wyszukaj zadania (min. 2 znaki)...



Brak zadań

Dodaj swoje pierwsze zadanie poniżej!

Łącznie zadań: 0

[Dodaj nowe zadanie](#)

Wstaw zrzut ekranu listy po dodaniu nowego zadania:

 Wyszukaj zadania (min. 2 znaki)...

Zadanie Domowe

Aplikacja internetowa

 2 lis 2025, 21:41

 Usuń

Łącznie zadań: 1

Punkty:

0

1

USUWANIE POZYCJI LISTY

Wstaw zrzut ekranu listy przed usunięciem wybranego zadania:

 9 lis 2025, 18:05

Sprawdzeneie

Sprawdzeeni zadan na obecny weekend

 Usuń

Programowanie

nаписаць програму дзялчыці мірэння часу працы CPU і GPU

 1 lis 2025, 16:04

 Usuń

Testowe

Testowy opis да zadania - !

 Usuń

Aplikacje Internetowe

Lab 2 да выполнения

 25 па́з 2025, 12:00

 Usuń

Łącznie заданий: 5

Wstaw зrzut ekranu listy по usunięciu zadania:



Wyszukaj zadania (min. 2 znaki)...



Brak zadań

Dodaj swoje pierwsze zadanie poniżej!

Łącznie zadań: 0

[Dodaj nowe zadanie](#)

Punkty:

0

1

EDYCJA POZYCJI LISTY

Wstaw zrzut ekranu listy przed edycją wybranego zadania:

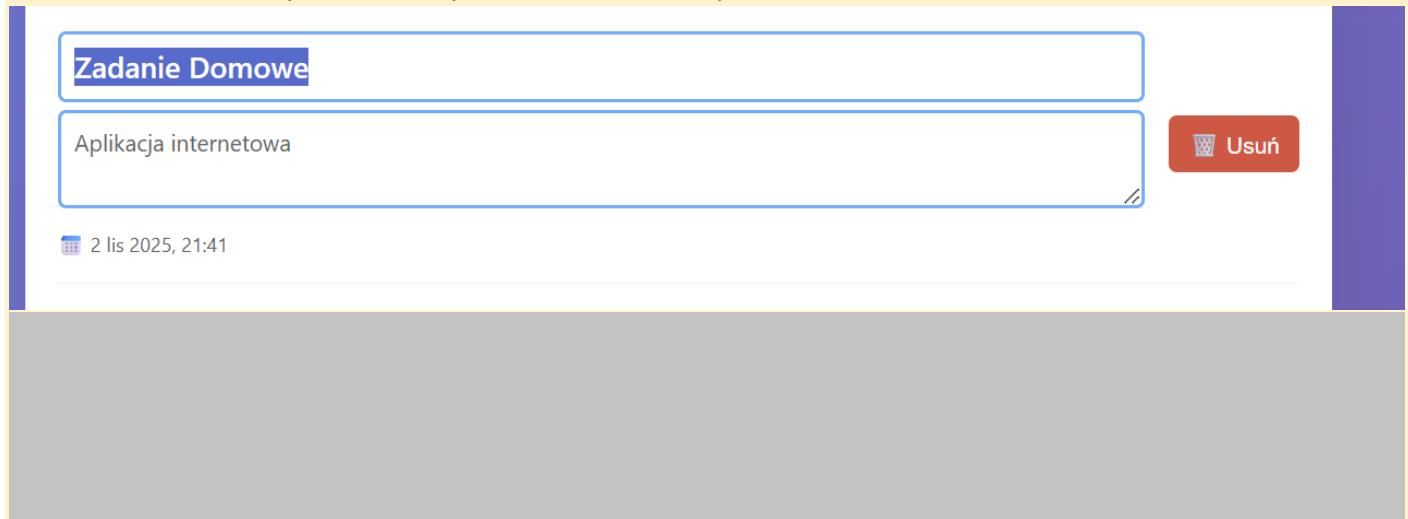
Zadanie Domowe

Aplikacja internetowa

2 lis 2025, 21:41

Usuń

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:

A screenshot of a task list interface. The list item has been updated to 'Zadanie domowe'. The date and time are now '31.10.2025 22:00'. The red 'Usuń' button is still present to the right of the list item. Below this section, there is a table with three columns: 'Punkty:' (Points), '0', and '1'.

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:

Dimensions: Responsive ▾ 836 x 355 100% ▾ No throttling ▾ 🔍 Laptop – 1024px

Wyszukaj zadania (min. 2 znaki)...

231231
3212312321
📅 Brak terminu

aasdad
aasdadsad
📅 Brak terminu

Zadanie Domowe
Zadanie domowe
📅 31 paź 2025, 22:00

Console Sources Network Performance Memory Application Privacy and security Lighthouse Recorder

C ⌂ Filter

file://

Origin	file://
Key	Value
todoApp_tasks	[{"id": 1761422563893.902, "title": "231231", "description": "3212312321", "dueDate": null, ...}, {"id": 1761422563893.902, "title": "231231", "description": "3212312321", "dueDate": null, ...}, {"id": 1761422558311.612, "title": "aasdad", "description": "aasdadsad", "dueDate": null, ...}, {"id": 1761422351440.0256, "title": "Zadanie Domowe", "description": "Zadanie domowe", ...}]

Wstaw zrzut 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 web application interface with a header "Zarządzaj swoimi zadaniami efektywnie". Below it is a search bar with placeholder "Wyszukaj zadania (min. 2 znaki)...". A section titled "Nowe zadanie" displays a new task: "Nowo dodane zadanie do listy" (8 lis 2025, 22:04), with a red "Usuń" button. Another task, "231231" (3212312321, Brak terminu), also has a "Usuń" button. The bottom part of the screen shows the browser's Application tab in developer tools, displaying the key-value pair "todoApp_tasks" with the value being an array of task objects.

Key	Value
todoApp_tasks	[{"id": 1761422645312.0942, "title": "Nowe zadanie", "description": "Nowo dodane zadanie do listy", ...}, {"id": 1761422645312.0942, "title": "Nowe zadanie", "description": "Nowo dodane zadanie do listy", ...}, {"id": 1761422563893.902, "title": "231231", "description": "3212312321", "dueDate": null, ...}, {"id": 1761422558311.612, "title": "aasdad", "description": "aasdadsad", "dueDate": null, ...}, {"id": 1761422351440.0256, "title": "Zadanie Domowe", "description": "Zadanie domowe", ...}]

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:

 Wyszukaj zadania (min. 2 znaki)...

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

za

Za danie Domowe

Za danie domowe

31 paź 2025, 22:00

 Usuń

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**:

za

Za danie Domowe

Za danie domowe

31 paź 2025, 22:00

Usuń

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/inazwisko/ai1-lab/tree/lab-b...>

Link do github: <https://github.com/Griniuu/Aplikacje-Internetowe.git>

PODSUMOWANIE

W kilku słowach/zdaniach napisz swoje przemyślenia odnośnie tego laboratorium. Nie używaj LLM.

...podsumowanie...

Pomyśl zadania ciekawy i przydatny do codziennych zadań na studia czy nawet i w pracy do robienia tasków itp.
Przydatna aplikacja i można ją dobrze rozwinąć

Zweryfikuj kompletność sprawozdania. Utwórz PDF i wyślij w terminie.