

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	5
Odczyt / Zapis LocalStorage.....	6
Wyszukiwanie	6
Commit projektu do GIT	8
Podsumowanie	8

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 przycisk 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. It features a search bar at the top. Below it is a list of tasks, each with a checkbox, a task name, a date, and a delete button (X). 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 a 'Zapisz' button.

Mockup of the application interface showing a task being edited. The search bar is empty. The list of tasks is the same as in the previous mockup. The 'chupa chups' task is selected, and its details are shown in a form below the list. The form has a text input field with 'chupa chups', a date input field with '2000-01-01', and a 'Zapisz' button. The 'Zapisz' button is highlighted in blue.

Mockup of the application interface showing search results. The search bar contains the text 'on'. The list of tasks is filtered to show only those containing 'on': macaroon and bon bons. The 'macaroon' task is highlighted in yellow. The 'Zapisz' button is highlighted in blue.

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.

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

- zadanie 1
- zadanie 2

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

KLASA TODO

Pierwszym instynktem może być chęć dodania zachowań bezpośrednio do elementów listy. 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`.

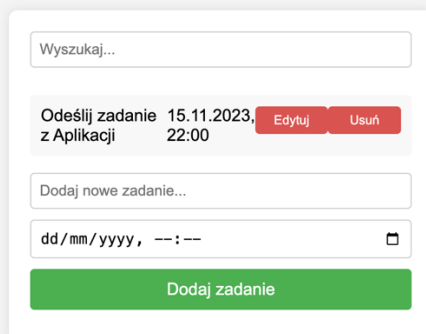
W celu sprawdzenia poprawności działania, najlepiej dostać się do tablicy `tasks` i edytować jej zawartość, po czym ręcznie wywołać metodę `draw()`. Jeśli zawartość listy wyrenderuje się na nowo poprawnie – możemy iść dalej!

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:

Wstaw zrzut ekranu listy po dodaniu nowego zadania:



The screenshot shows a task management interface. At the top is a search bar labeled 'Wyszukaj...'. Below it is a task entry: 'Odeślij zadanie z Aplikacji' with a due date of '15.11.2023, 22:00' and two red buttons labeled 'Edytuj' and 'Usuń'. Below the task entry is a text input field labeled 'Dodaj nowe zadanie...'. At the bottom is a date and time picker showing 'dd/mm/yyyy, --:--' with a calendar icon. A green button labeled 'Dodaj zadanie' is at the very bottom.

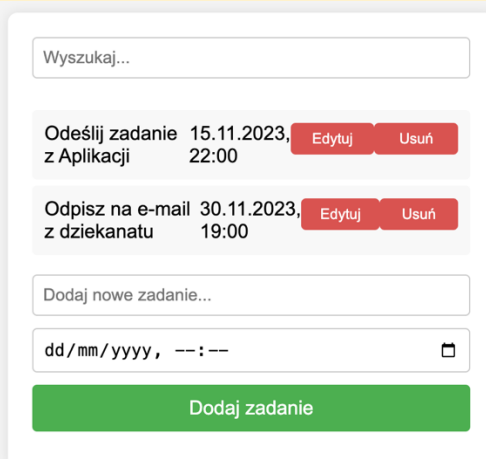
Punkty:

0

1

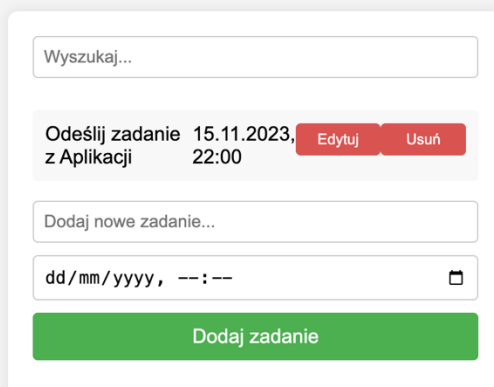
USUWANIE POZYCJI LISTY

Wstaw zrzut ekranu listy przed usunięciem wybranego zadania:



The screenshot shows the same task management interface as before, but with two tasks listed. The first task is 'Odeślij zadanie z Aplikacji' with a due date of '15.11.2023, 22:00' and 'Edytuj'/'Usuń' buttons. The second task is 'Odpisz na e-mail z dziekanatu' with a due date of '30.11.2023, 19:00' and 'Edytuj'/'Usuń' buttons. The rest of the interface (search bar, input field, date picker, and 'Dodaj zadanie' button) remains the same.

Wstaw zrzut ekranu listy po usunięciu zadania:



The screenshot shows the task management interface after one task has been deleted. Only the first task, 'Odeślij zadanie z Aplikacji' with a due date of '15.11.2023, 22:00', remains. The 'Odpisz na e-mail z dziekanatu' task is no longer visible. The rest of the interface is identical to the previous screenshot.

Punkty:

0

1

EDYCJA POZYCJI LISTY

Wstaw zrzut ekranu listy przed edycją wybranego zadania:

Wyszukaj...

Odeślij zadanie z Aplikacji - 15.11.2023, 00:00 Edytuj Usuń

Odpisz na e-mail z dziekanatu - 21.02.2023, 13:00 Edytuj Usuń

Dodaj nowe zadanie...

dd/mm/yyyy, --:--

Dodaj zadanie

Wstaw zrzut ekranu listy w trakcie edytowania zadania i daty:

Wyszukaj...

Odeślij zadanie z Aplikacji - 15.11.2023, 00:00 Edytuj Usuń

Odpisz na e-mail z dziekanatu - 21.02.2023, 13:00 Edytuj Usuń

Pojedź do dziekanatu WIZUT

16/11/2023, 08:30

Zapisz zmiany

Anuluj

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

Wyszukaj...

Odeślij zadanie z Aplikacji - 15.11.2023, 00:00 Edytuj Usuń

Pojedź do dziekanatu WIZUT - 16.11.2023, 08:30 Edytuj Usuń

Dodaj nowe zadanie...

dd/mm/yyyy, --:--

Dodaj zadanie

Punkty:

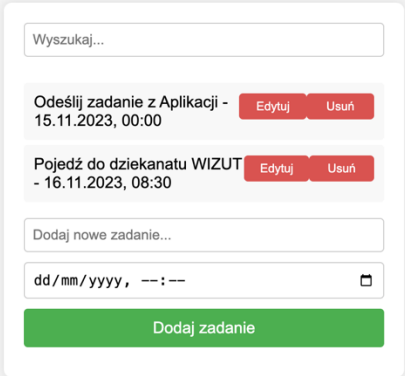
0

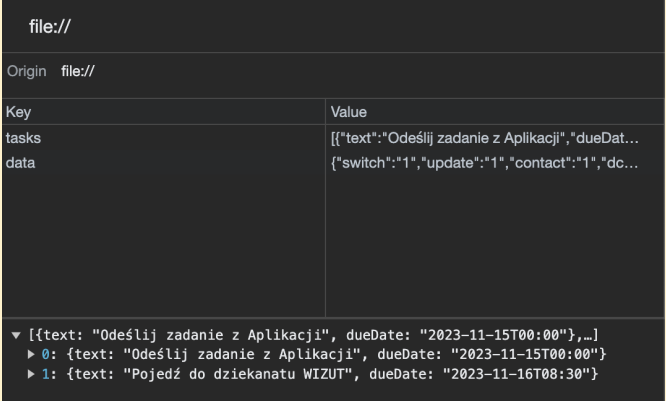
1

ODCZYT / ZAPIS LOCALSTORAGE

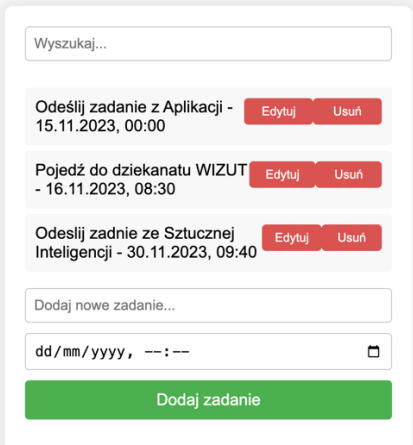
Zastosowanie klasy Todo w realizacji tego laboratorium pozwala w bardzo łatwy sposób odczytywać i zapisywać stan listy do pamięci przeglądarki. Wystarczy serializacja / deserializacja za pomocą `JSON.parse()` i `JSON.stringify()`.

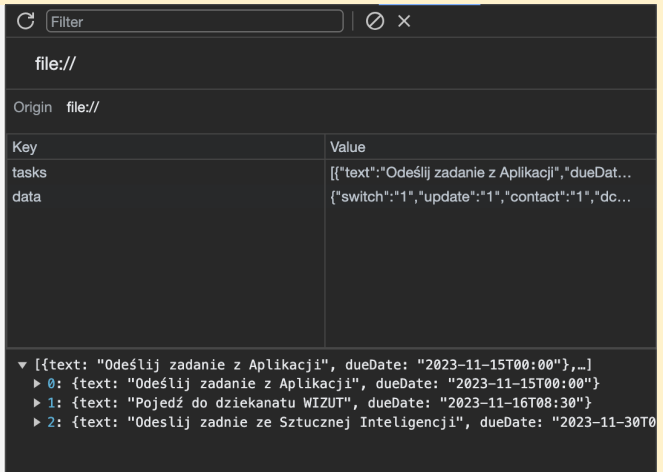
Wstaw zrzuty ekranu przedstawiające wygląd listy i zawartość local storage gdy na liście są pewne zadania:





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:





Punkty:

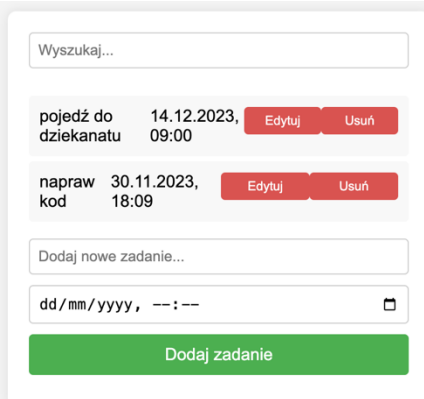
0

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...

pojedź do dziekanatu 14.12.2023, 09:00 Edytuj Usuń

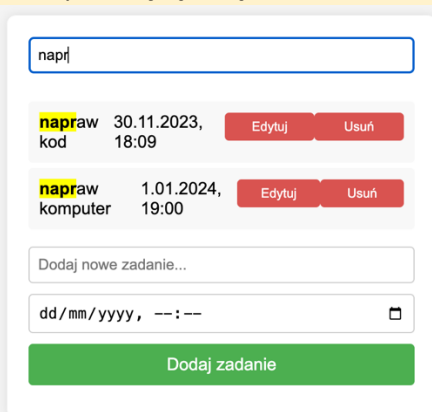
napraw kod 30.11.2023, 18:09 Edytuj Usuń

Dodaj 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:



napr

napraw kod 30.11.2023, 18:09 Edytuj Usuń

napraw komputer 1.01.2024, 19:00 Edytuj Usuń

Dodaj nowe zadanie...

dd/mm/yyyy, --:-- 📅

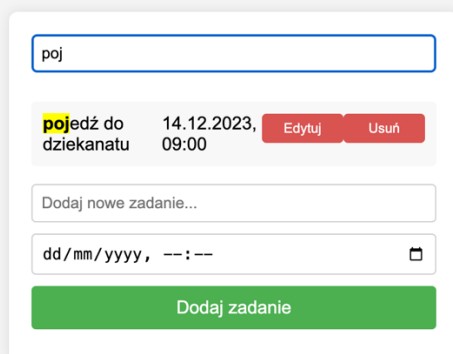
Dodaj zadanie

Punkty:

0

1

Wstaw zrzut ekranu przedstawiający podświetlenie szukanej frazy w wynikach wyszukiwania, przykładowo dla frazy **imp** i zadania **implementacja** otrzymujemy: **implementacja**:



poj

pojedź do dziekanatu 14.12.2023, 09:00 Edytuj Usuń

Dodaj nowe zadanie...

dd/mm/yyyy, --:-- 📅

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:

...link, np. <https://github.com/gracjanjasnos/main/tree/main/lab02>

PODSUMOWANIE

W kilku zdaniach podsumuj zdobyte podczas tego laboratorium umiejętności.

Podczas tego laboratorium zdobyłem umiejętność tworzenia interaktywnych stron internetowych przy użyciu HTML, CSS i JavaScript. Nauczyłem się obsługi zdarzeń, manipulowania elementami DOM, korzystania z Local Storage do przechowywania danych, implementacji funkcji dodawania, usuwania i edycji elementów listy zadań. Ponadto, wprowadziłem filtrowanie i podświetlanie wyników wyszukiwania. Praktyczne zastosowanie tych umiejętności umożliwiło mi stworzenie funkcjonalnej aplikacji do zarządzania listą zadań z nowoczesnym i estetycznym interfejsem.

Zweryfikuj kompletność sprawozdania. Utwórz PDF i wyślij w terminie.