Temat

Aplikacja todo

Imię i nazwisko

Adam Orłowski

Grupa labolatoryjna

Lab 3

Data wykonania

2021.01.24

Data oddania

2021.01.24

Nazwa pliku

Index.html

Typ dokumentu

HTML5

Kodowanie znaków

UTF-8

Tytuł strony

TODO

Budowa strony

@font-face {  
 font-family: Exo2;  
 src: url("font/Exo2.ttf");  
}  
  
\* {  
 font-family: Exo2, serif;  
}  
  
body {  
 background-color: #A3235C;  
 color: white;  
}  
  
h1 {  
 font-weight: bold;  
}  
  
hr {  
 height: 3px;  
 background-color: #F0E271;  
}  
  
dd {  
 word-wrap: break-word;  
}  
  
**.**container {  
 padding-left: 5px;  
 padding-right: 5px;  
 padding-bottom: 1px;  
 margin-right: 30%;  
 margin-left: 30%;  
 background-color: rgba(0, 0, 0, 0.3);  
 border-radius: 5px;  
}  
  
**.**intro {  
 margin-top: 50px;  
 margin-bottom: 50px;  
}  
  
**.**main {  
 text-align: center;  
}  
  
**.**task-wrapper {  
 border-radius: 3px;  
}  
  
**.**task-margin {  
 margin-left: 5%;  
 margin-right: 5%;  
}  
  
**.**task-title {  
 font-size: 25px;  
}  
  
label {  
 display: block;  
 margin: 5px;  
}  
  
input[type = "text"], textarea {  
 color: white;  
 background: none;  
 border: 2px solid #3493A3;  
 padding: 10px 10px;  
 outline: none;  
 border-radius: 24px;  
 transition: 0.25s;  
}  
  
#task-title-input {  
 width: 40%;  
}  
  
#task-title-input:focus {  
 width: 50%;  
 border-color: #F0E271;  
}  
  
#task-description-input {  
 width: 80%;  
 max-width: 90%;  
}  
  
#task-description-input:focus {  
 width: 90%;  
 border-color: #F0E271;  
}  
  
input[type = "submit"] {  
 color: white;  
 background: none;  
 border: 2px solid #3493A3;  
 padding: 10px 10px;  
 margin: 10px;  
 width: 20%;  
 outline: none;  
 border-radius: 24px;  
 transition: 0.25s;  
}  
  
input[type = "submit"]:hover {  
 border: 2px solid #F0E271;  
 background: #F0E271;  
 cursor: pointer;  
 color: black;  
}  
  
#validation-error {  
 display: none;  
 color: rgb(224, 64, 6);  
 font-size: 20px;  
}  
  
**.**err-img {  
 width: 20px;  
 height: auto;  
}  
  
span {  
 cursor: pointer;  
 font-size: 12px;  
 margin-left: 5px;  
 margin-right: 5px;  
 transition: 0.25s;  
}  
  
span:hover {  
 color: black;  
 background-color: #F0E271;  
}

**.**square {  
 z-index: -1;  
 position: absolute;  
 width: 120px;  
 height: 120px;  
}  
  
**.**square1 {  
 top: 50%;  
 left: 18%;  
 transform: rotate(20deg);  
 background-color: #F0E271;  
}  
  
**.**square2 {  
 top: 25%;  
 left: 70%;  
 transform: rotate(65deg);  
 background-color: #3493A3;  
}  
  
**.**square3 {  
 top: 70%;  
 left: 50%;  
 transform: rotate(50deg);  
 background-color: #F0599D;  
}  
  
**.**circle {  
 z-index: -1;  
 position: absolute;  
 width: 80px;  
 height: 80px;  
 border-radius: 50%;  
}  
  
**.**circle1 {  
 top: 5%;  
 left: 10%;  
 background-color: #F0599D;  
}  
  
**.**circle2 {  
 top: 80%;  
 left: 25%;  
 background-color: #3493A3;  
}  
  
**.**circle3 {  
 top: 70%;  
 left: 80%;  
 background-color: #F0E271;  
}

Zawartość strony

div.square .square1

div.square .square2

div.square .square3

div.circle .circle1

div.circle .circle2

div.circle .circle3

div.container

div.intro

h1: todo list

p: Projekt: Adam Orłowski

hr

div.main

form#form

label: Task title

input#task-title-input type:text; placeholder: Enter task title; name: title

br

label: Task description

textarea#task-description-input placeholder: Enter task description; name: description

input value: add

p#validation-error

img: error-icon.png

Title and description cannot be empty!

img: error-icon.png

hr

my todos:

div.task-wrapper

div.task-margin

dl#tasks

Założenia projektu

Prosta aplikacja służy do zapisywania oraz usuwania zadań do realizacji.

Opis zmiennych

state – obiekt, w którym ustawiany jest tytuł oraz opis danego taska

taskElement – element div, w którym umieszczane są wszystkie taski

todos – obiekt zawierający wszystkie taski zapisane w pamięci podręcznej

Opis funkcji

W skrypcie behaviour.js zdefiniowane są 3 funkcje:

* setState(event)
  + ustawia zmienne w obiekcie state
  + usuwa alert dotyczący walidacji
* handleFormSubmit(event)
  + sprawdza, czy tytuł i opis taska nie jest pusty
  + tworzy obiekt todo
  + dodaje to tablicy tasków kolejnego taska, jeśli ich nie ma tworzy nową tablicę
  + zapisuje wszystkie taski w pamięci podręcznej
* showErrorMessage(show)
  + pokazuje/usuwa alert dotyczący walidacji zależny od argumentu show

W skrypcie util.js zdefiniowane są 3 funkcje:

* addTask(todo)
  + przyjmuje obiekt todo zawierający zmienne id, title oraz description
  + pobiera element div o id tasks
  + tworzy element div i nadaje mu id ze zmiennej id obiektu todo
  + tworzy element dt zawierający tytuł taska i nadaje mu klasę ‘task-title’
  + tworzy element dd zawierający opis taska
  + tworzy element span zawierający napis ‘done’ i ustawia *onClickListener*
* removeTaskFromStorage(taskId)
  + pobiera z pamięci lokalnej listę tasków
  + usuwa poszczególny task w zależności od podanego argumentu taskId
  + zapisuje taski do pamięci lokalnej
* deleteTask(taskId)
  + usuwa element div taska

Inicjalizacja skryptu

Przy otworzeniu projektu w przeglądarce, uruchamia się skrypt loadTodos.js który pobiera z pamięci lokalnej tablicę tasków. Następnie iteruje te tablice wykonując funkcję addTask() przekazując w argumencie funkcji poszczególny task.

Dodawanie i usuwanie tasków

Dodawanie realizowane jest poprzez kliknięcie przycisku Add, który wywołuje funkcję handleFormSubmit(). Tytuł ani opis taska nie może być pusty, w przeciwnym razie wywołuje się funkcja showErrorMessage().

Usuwanie taska realizowanie jest poprzez kliknięcie na poszczególny napis „done”, który wywołuje funkcję deleteTask() usuwający dany task.