Цели задачи

* Самостоятельно получить опыт работы с событиями и изменением существующего кода.
* Научиться создавать веб-приложения на примере приложения TODO.

Этап 1

1. Напишите код приложения TODO, повторяя за спикером из видео. Проверьте его работоспособность.
2. Изучите написанный код. Рекомендуем каждую строку кода дополнить комментарием с описанием назначения команды.
3. Измените функцию создания дела, чтобы она принимала и корректно обрабатывала объект { name, done }, а не просто название. Это пригодится для преобразования объекта дела в DOM-элемент.

Этап 2

Сделайте так, чтобы у кнопки в форме устанавливался атрибут **disabled**, когда поле ввода пустое. Не забудьте, что **disabled** должен устанавливаться и при загрузке приложения, так как изначально поле тоже пустое.

Этап 3

Каждое созданное дело должно храниться в массиве дел в виде объекта. При создании нового дела в массив должен добавляться новый объект с тремя параметрами: **name** — название дела, **done** — статус дела, который может принимать значения **true** или **false**, и id — уникальный числовой параметр.

Чтобы сгенерировать id, вы можете выбрать один из двух способов:

1. Для каждого нового объекта (дела) можно генерировать **id** с произвольным, рандомным числом.
2. Написать функцию, которая будет искать максимальный id в массиве дел и прибавлять к максимальному id число 1.

Пример массива с объектами дел

[

    {id: 1, name: ‘Купить хлеб', done: false},

    {id: 2, name: ‘Покормить кота’, done: true},

    {id: 3, name: ‘Сделать задание по JS’, done: true}

]

Этап 4

Сделайте так, чтобы при нажатии на кнопку «Удалить» удалялся не только DOM-элемент **li** из DOM-дерева, но и объект в массиве дел. Для решения этой задачи при клике на кнопку «Удалить» необходимо найти в массиве дел объект с таким же **id** и удалить его. Искать элементы в массиве вы уже умеете.

Так же поступите и с кнопкой изменения статуса. При клике на неё найдите в массиве объектов нужный объект и измените параметр **done** на противоположный.

Этап 5

Создайте функцию сохранения массива дел в localStorage. Вызывайте функцию сохранения каждый раз, когда вы изменяете список дел, а именно: добавляете новое дело, удаляете дело или изменяете статус.

Перед выполнением этого этапа рекомендуем почитать материалы:

* [JSON.stringify](https://developer.mozilla.org/ru/docs/Web/JavaScript/Reference/Global_Objects/JSON/stringify)
* [JSON.parse](https://developer.mozilla.org/ru/docs/Web/JavaScript/Reference/Global_Objects/JSON/parse)
* [Window.localStorage](https://developer.mozilla.org/ru/docs/Web/API/Window/localStorage)

Учтите, что список дел разный для каждой страницы, поэтому нужно использовать разные ключи в localStorage. Ключ можно передавать отдельным параметром в createTodoApp.

Добавьте в функцию **createTodoApp()**третий параметр **listName**, который поможет вам создать собственный список для каждого пользователя.

Пример запуска приложения на странице пользователя:

document.addEventListener('DOMContentLoaded', function(){

      createTodoApp(document.getElementById('todo-app'), 'Мои дела', '**my**');

});

Этап 6

Сейчас ваше приложение может сохранять список дел. На данном этапе нужно сделать так, чтобы при запуске приложения в функции **createTodoApp()** была выполнена проверка на наличие данных в localStorage. Если в localStorage есть данные, то их нужно расшифровать и полученный массив отрисовать на экране в виде DOM-элементов списка.