

Frontender[1.0] HTML, CSS, JavaScript - Todo app advanced. Продвинутое Todo приложение

	https://youtu.be/gH4dUH9yk1w
	https://t.me/Dmitry_Kolotilshikov
	https://github.com/DmitryKolotilshikov/
⊗ Boosty	https://boosty.to/dmitry_ko
# Номер урока	36



TODO APP ADVANCED - HTML, CSS, JAVASCRIPT

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <script type="module" src="utils.js" defer></script>
    <script type="module" src="script.js" defer></script>
    <link rel="stylesheet" href="styles.css">
   <title>todo app</title>
</head>
<body>
    <header class="header">
        <h1>TODO APP</h1>
    </header>
    <main>
        <section class="add-input-container">
            <input type="text" placeholder="add todo" class="add-todo-input" data-add-todo-in</pre>
put>
            <button class="add-todo-btn" data-add-todo-btn>add todo
        </section>
        <section class="search-input-container">
            <input type="text" placeholder="search todo" class="search-input" data-search-tod</pre>
o-input>
        </section>
        <section class="todo-list-container" data-todo-container></section>
   </main>
```

```
<template data-todo-template>
           <div class="todo">
                <input type="checkbox" class="todo-checkbox" data-todo-checkbox>
                <div class="todo-info">
                   go to jym
                   <span class="todo-creation-date" data-todo-date>03-10-24</span>
                </div>
                <button class="remove-todo-btn" remove-todo-btn>X</button>
           </div>
   </template>
</body>
</html>
// Utils.js
export const getDateRepresentation = (date) => {
    return Intl.DateTimeFormat("ru-RU", { day: "numeric", month: "numeric", year: "numeric"
}).format(date);
export const saveTodosIntoLocalStorage = (todos) => {
    localStorage.setItem("todos", JSON.stringify(todos));
}
export const getTodosFromLocalStorage = () => {
    return JSON.parse(localStorage.getItem("todos")) || [];
// script.js
import { getDateRepresentation, saveTodosIntoLocalStorage, getTodosFromLocalStorage } from
"./utils.js";
const addTodoInput = document.querySelector("[data-add-todo-input]");
const addTodoBtn = document.querySelector("[data-add-todo-btn]");
const searchTodoInput = document.querySelector("[data-search-todo-input]");
const todosContainer = document.querySelector("[data-todo-container]");
const todoTemplate = document.querySelector("[data-todo-template]");
let todoList = getTodosFromLocalStorage();
let filteredTodoList = [];
addTodoBtn.addEventListener("click", () => {
    if (addTodoInput.value.trim()) {
        const newTodo = {
           id: Date.now(),
           text: addTodoInput.value.trim(),
           completed: false,
           createdAt: getDateRepresentation(new Date()),
       todoList.push(newTodo);
        addTodoInput.value = "";
        saveTodosIntoLocalStorage(todoList);
        renderTodos();
})
```

```
addTodoInput.addEventListener("input", () => {
    if (searchTodoInput.value.trim()) {
        searchTodoInput.value = "";
        renderTodos();
})
searchTodoInput.addEventListener("input", (e) => {
    const searchValue = e.target.value.trim();
    filterAndRenderFilteredTodos(searchValue);
})
const filterAndRenderFilteredTodos = (searchValue) => {
    filteredTodoList = todoList.filter((t) => t.text.includes(searchValue));
    renderFilteredTodos();
}
const createTodoLayout = (todo) => {
    const todoElement = document.importNode(todoTemplate.content, true);
    const checkbox = todoElement.querySelector("[data-todo-checkbox]");
    checkbox.checked = todo.completed;
    const todoText = todoElement.querySelector("[data-todo-text]");
    todoText.textContent = todo.text;
    const todoCreatedDate = todoElement.querySelector("[data-todo-date]");
    todoCreatedDate.textContent = todo.createdAt;
    const removeTodoBtn = todoElement.querySelector("[remove-todo-btn]");
    removeTodoBtn.disabled = !todo.completed;
    checkbox.addEventListener("change", (e) => {
        todoList = todoList.map((t) => {
            if (t.id === todo.id) {
                t.completed = e.target.checked;
            }
            return t;
        })
        saveTodosIntoLocalStorage(todoList);
        if (searchTodoInput.value.trim()) {
            filterAndRenderFilteredTodos(searchTodoInput.value.trim());
        } else {
            renderTodos();
    })
    removeTodoBtn.addEventListener("click", () => {
        todoList = todoList.filter((t) => {
            if (t.id !== todo.id) {
                return t;
            }
        })
        saveTodosIntoLocalStorage(todoList);
```

```
if (searchTodoInput.value.trim()) {
            filterAndRenderFilteredTodos(searchTodoInput.value.trim());
        } else {
            renderTodos();
    });
    return todoElement;
}
const renderTodos = () => {
    todosContainer.innerHTML = "";
    if (todoList.length === 0) {
        todosContainer.innerHTML = `<h3>No todos...</h3>`;
        return;
    }
    todoList.forEach((todo) => {
        const todoElement = createTodoLayout(todo);
        todosContainer.append(todoElement);
   })
}
const renderFilteredTodos = () => {
    todosContainer.innerHTML = "";
    if (filteredTodoList.length === 0) {
        todosContainer.innerHTML = `<h3>No todos found...</h3>`;
        return;
    }
    filteredTodoList.forEach((todo) => {
        const todoElement = createTodoLayout(todo);
        todosContainer.append(todoElement);
   })
}
renderTodos();
* {
    box-sizing: border-box;
p, body {
    margin: 0;
body {
    background-color: #bbbfb9;
}
.header {
    width: 100%;
    height: 130px;
    background: url("assets/bg_forrest.webp") center/cover no-repeat;
    color: #fff;
    font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;
```

```
display: flex;
    justify-content: center;
    align-items: center;
    font-size: 32px;
   h1 {
        text-shadow: 2px 2px 13px #000;
        user-select: none;
}
.add-input-container {
    width: 100%;
    max-width: 600px;
    height: 50px;
    margin: 30px auto;
    padding: 0 10px;
    display: flex;
    justify-content: flex-start;
    align-items: stretch;
    gap: 10px;
    .add-todo-btn {
        font-size: 20px;
        flex-grow: 1;
        user-select: none;
}
.search-input-container {
    width: 100%;
    max-width: 600px;
    height: 50px;
    margin: 0 auto 30px;
    padding: 0 10px;
    display: flex;
    justify-content: flex-start;
    align-items: stretch;
    gap: 10px;
}
.add-todo-input,
.search-input {
    border: none;
    font-size: 24px;
    flex-grow: 3;
}
.todo-list-container {
    width: 100%;
    max-width: 580px;
    margin: 0 auto 30px;
    display: flex;
    flex-direction: column;
    justify-content: flex-start;
    align-items: stretch;
    gap: 15px;
```

```
.todo {
        height: 60px;
        padding: 0 10px;
        display: flex;
        justify-content: flex-start;
        align-items: center;
        gap: 10px;
        background-color: #e0e4bd;
        box-shadow: 0 0 15px 3px #00000055;
    .todo-checkbox {
        accent-color: #3db86a;
        width: 25px;
        height: 25px;
   }
    .todo-info {
        flex-grow: 1;
        display: flex;
        flex-direction: column;
        justify-content: flex-start;
        align-items: flex-start;
        margin: 0;
        .todo-text {
            margin: 0 0 3px 0;
        .todo-creation-date {
            font-size: 12px;
            font-style: italic;
   }
    .remove-todo-btn {
        background-color: #b5a2a2;
        border: none;
        color: #fff;
        padding: 5px 10px;
        background-color: #b83d3d;
        cursor: pointer;
        &:disabled {
            background-color: #b79999;
            cursor: not-allowed;
    .todo:has(.todo-checkbox:checked) {
        background-color: #6acb8e;
   }
    .todo-checkbox:checked ~ .todo-info {
        text-decoration: line-through;
   }
}
```