



Frontender[1.0] пишем TODO ПРИЛОЖЕНИЕ. практика - JavaScript, DOM, Events, Storage

| | |
|--|---|
| YouTube | https://youtu.be/KidK78UXKiw |
| Telegram | https://t.me/Dmitry_Kolotilshikov |
| Github | https://github.com/DmitryKolotilshikov/ |
| Boosty | https://boosty.to/dmitry_ko |
| # Номер урока | 33 |

💡 Задание к этому уроку - написать приложение своими руками согласно видео. Можно попытаться расширить функционал приложения. Также в 36 уроке напишем продвинутое **todo** приложение.

```
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <script src="script.js" defer></script>
  <link rel="stylesheet" href="styles.css">
  <title>todo app</title>
</head>

<body>
  <section class="todo">
    <div class="text-field-wrapper">
      <input type="text" data-text-field>
      <button data-add-todo-btn>add todo</button>
    </div>

    <hr>

    <div class="container" data-todo-container></div>
  </section>
</body>

</html>
```

```

* {
  box-sizing: border-box;
}

body {
  margin: 0;
  background-color: rgb(153, 179, 198);
}

.todo {
  margin: 50px auto;
  padding: 0.5em;
  width: 500px;
  border: 1px solid black;
  background-color: rgb(210, 216, 206);
}

.text-field-wrapper {
  display: flex;
  justify-content: space-between;
  align-items: center;
  gap: 10px;
}

.text-field-wrapper input {
  flex-grow: 3;
  height: 30px;
}

.text-field-wrapper button {
  flex-grow: 1;
  height: 30px;
}

.container {
  width: 100%;
  display: flex;
  flex-direction: column;
  gap: 10px;
}

.todo-item {
  width: 100%;
  padding: 0.5em;
  display: flex;
  justify-content: space-between;
  align-items: center;
  gap: 10px;
  border: 1px solid rgb(20, 85, 117);
  background-color: burlywood;
}

```

```

const input = document.querySelector("[data-text-field]");
const addTodoBtn = document.querySelector("[data-add-todo-btn]");
const container = document.querySelector("[data-todo-container]");

const todoList = JSON.parse(localStorage.getItem("todos")) || [];

```

```

const saveToLocalStorage = (key = "todos") => {
  localStorage.setItem(key, JSON.stringify(todoList));
}

addTodoBtn.addEventListener("click", () => {
  if(input.value.trim()) {
    todoList.push(input.value);
    input.value = "";

    saveToLocalStorage();
    render();
  }
})

const createElement = (tagName, textContent) => {
  const element = document.createElement(tagName);
  element.textContent = textContent;

  return element;
}

const removeTodo = (index) => {
  todoList.splice(index, 1);
  saveToLocalStorage();
  render();
}

const render = () => {
  container.innerHTML = "";
  todoList.forEach((todo, index) => {
    const todoElement = createElement("div", todo);
    const removeBtn = createElement("button", "✖");

    removeBtn.addEventListener("click", () => removeTodo(index))

    todoElement.classList.add("todo-item")
    todoElement.append(removeBtn);

    container.append(todoElement);
  })
}

render();

```