



# Frontender[1.0] JavaScript - TODO приложение ООП

<a href="#">YouTube</a>	<a href="https://youtu.be/gxVzSA_U7sQ">https://youtu.be/gxVzSA_U7sQ</a>
<a href="#">Telegram</a>	<a href="https://t.me/Dmitry_Kolotilshikov">https://t.me/Dmitry_Kolotilshikov</a>
<a href="#">Github</a>	<a href="https://github.com/DmitryKolotilshikov/">https://github.com/DmitryKolotilshikov/</a>
<a href="#">Boosty</a>	<a href="https://boosty.to/dmitry_ko">https://boosty.to/dmitry_ko</a>
# Номер урока	45

💡 Ссылка на апку в github: <https://github.com/DmitryKolotilshikov/todo-app-oop>

💡 дизайн приложения

TODO APP

продолжать делать контент

хорошо отдохнуть

Удалить

восстановить силы

Удалить

радоваться жизни

Удалить

больше положительных эмоций

Удалить

заряд на успех

Удалить

```

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

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

<body>
  <h1>TODO APP</h1>

  <div class="container">
    <input type="text" class="search-input" placeholder="добавьте новую задачу" data-add-
todo-input>

    <div class="todos-container" data-todos-container></div>
  </div>
</body>

</html>

```

```

:root {
  --neon-inner: #35beaf;
  --neon-outer: #35b9aa;
  --completed-todo: #2ea396;
}

* {
  box-sizing: border-box;
}

body {
  margin: 0;
  font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;
  min-height: 100vh;
  background: linear-gradient(180deg, #4d6f93 0%, #a84545a6 100%);
}

h1 {
  text-align: center;
  margin: 0;
  padding: 20px 0 0 0;
  color: #e1e1e1;
}

.container {
  padding: 50px 10px 0 10px;
  margin: 0 auto;
  width: 100%;
  max-width: 400px;
  display: flex;
  flex-direction: column;
  justify-content: center;
}

```

```

    align-items: center;
    gap: 24px;
}

.search-input {
    padding: 0 10px;
    width: 100%;
    height: 45px;
    font-size: 20px;
    background: none;
    border: none;
    border-radius: 3px;
    outline: 2px solid #21303f;
    transition: .22s box-shadow, .22s color;
}

.search-input:focus-visible {
    box-shadow: 0 0 15px 15px var(--neon-outer), inset 0 0 25px 10px var(--neon-inner);
    color: #fff;
}

.search-input::placeholder {
    color: #3c3c3c;
    font-size: 16px;
    font-style: italic;
}

.todos-container {
    width: 100%;
    display: flex;
    flex-direction: column;
    justify-content: center;
    align-items: stretch;
    gap: 8px;
}

.todo-item {
    padding: 0 10px;
    height: 45px;
    display: flex;
    justify-content: space-between;
    align-items: center;
    gap: 24px;
    border: 1px solid black;
}

.todo-item span {
    pointer-events: none;
    user-select: none;
}

.todo-item.completed {
    background-color: var(--completed-todo);
}

.todo-item.completed span {
    text-decoration: line-through;
}

```

```

.todo-item:hover {
  border: 1px solid var(--neon-inner);
  box-shadow: 0 0 10px 5px var(--neon-outer), inset 0 0 10px 5px var(--neon-inner);
  color: #fff;
}

```

```

class LocalStorage {
  #keyName;

  constructor(keyName) {
    this.#keyName = keyName;
  }

  GetItem() {
    const items = localStorage.getItem(this.#keyName);
    return items ? JSON.parse(items) : [];
  }

  SetItem(itemsList) {
    localStorage.setItem(this.#keyName, JSON.stringify(itemsList));
  }
}

class DOM {
  query(selector) {
    return document.querySelector(selector);
  }

  create(type, textContent, ...classNames) {
    const item = document.createElement(type);
    item.textContent = textContent;
    // item.classList.add(...classNames);
    classNames.length && (item.className = classNames.join(" "));

    return item;
  }
}

class Item {
  constructor(id, text) {
    this.id = id;
    this.text = text;
  }
}

class TodoItem extends Item {
  constructor(id, text, completed = false) {
    super(id, text);
    this.completed = completed;
  }
}

class TodoApp {
  constructor() {
    this.dom = new DOM();
    this.todosStorage = new LocalStorage("todos");
  }
}

```

```

    this.todoList = this.todosStorage.GetItem();
    this.todoInput = this.dom.query("[data-add-todo-input]");
    this.todoContainer = this.dom.query("[data-todos-container]");

    this.bindEvents();
    this.render();
}

addTodo(text) {
    const newTodo = new TodoItem(Date.now(), text);

    this.todoList.push(newTodo);
    this.todosStorage.SetItem(this.todoList);
    this.render();
}

removeTodo(id) {
    this.todoList = this.todoList.filter(todo => todo.id !== id);
    this.todosStorage.SetItem(this.todoList);
    this.render();
}

toggleTodo(id) {
    const todo = this.todoList.find(todo => todo.id === id);
    if (todo) {
        todo.completed = !todo.completed;
        this.todosStorage.SetItem(this.todoList);
        this.render();
    }
}

bindEvents() {
    this.todoInput.addEventListener("keypress", (e) => {
        if (e.key === "Enter" && e.target.value.trim()) {
            this.addTodo(e.target.value.trim());
            this.todoInput.value = "";
        }
    });

    this.todoContainer.addEventListener("click", (e) => {
        const el = e.target;

        if (el.classList.contains("remove-btn")) {
            const id = Number(el.dataset.id);
            this.removeTodo(id);
        } else if (el.classList.contains("todo-item")) {
            const id = Number(el.dataset.id);
            this.toggleTodo(id);
        }
    });
}

render() {
    this.todoContainer.innerHTML = "";

    this.todoList.forEach(todo => {
        const todoItem = this.dom.create("div", "", "todo-item", todo.completed ? "comple

```

```

ted" : "");
    todoItem.dataset.id = todo.id;
    const todoItemText = this.dom.create("span", todo.text);

    const removeBtn = this.dom.create("button", "Удалить", "remove-btn");
    removeBtn.dataset.id = todo.id;
    removeBtn.disabled = !todo.completed;

    todoItem.appendChild(todoItemText);
    todoItem.appendChild(removeBtn);
    this.todoContainer.appendChild(todoItem);
  })
}
}

new TodoApp();

```