

DOM-дерево проекты

Задание 1. **ToDo List**

Как сделать?

ToDo List

Добавить новую задач

Добавить задачу

Поесть

Удалить

Важно!

Погулять

Удалить

Важно!

Очень важные дела

Сделать домашку

Удалить

Важно!

Пропылесосить

Удалить

Важно!

```
<!DOCTYPE html>
```

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

```
<head>
```

```
  <meta charset="UTF-8">
```

```
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
```

```
  <title>ToDo List</title>
```

```
  <link rel="stylesheet" href="styles.css">
```

```
</head>
```

Head

Body

```
<body>
  <div id="app" class="container">
    <h1 class="container">ToDo List</h1>
    <div class="input-container">
      <input type="text" id="taskInput" placeholder="Добавить новую задачу">
      <button id="addTaskBtn">Добавить задачу</button>
    </div>
    <ul id="taskList">
      <!-- Обычные задачи будут добавлены здесь динамически -->
    </ul>
    <h2 class="container">Очень важные дела</h2>
    <ul id="importantTaskList">
      <!-- Очень важные дела будут добавлены здесь динамически -->
    </ul>
  </div>
  <script src="script.js"></script>
</body>
</html>
```

CSS

```
body {  
  font-family: 'Arial', sans-serif;  
  margin: 0;  
  padding: 0;  
  display: flex;  
  align-items: center;  
  justify-content: center;  
  height: 100vh;  
  background-color: #f0f0f0;  
}
```

```
.container {  
  text-align: center;  
  background-color: #fff;  
  padding: 20px;  
  border-radius: 8px;  
  box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);  
}
```

```
h1 {  
    color: #3498db;  
}  
  
h2 {  
    color: #3498db;  
}  
  
.input-container {  
    margin-bottom: 20px;  
}
```

```
input {  
    padding: 8px;  
    margin-right: 8px;  
    border: 1px solid #ccc;  
    border-radius: 4px;  
}  
  
button {  
    cursor: pointer;  
    background-color: #3498db;  
    color: #fff;  
    border: none;  
    padding: 8px 12px;  
    border-radius: 4px;  
}
```

```
ul {  
  list-style: none;  
  padding: 0;  
}  
  
li {  
  display: flex;  
  justify-content: space-between;  
  align-items: center;  
  border-bottom: 1px solid #ccc;  
  padding: 10px;  
}
```

```
button.delete-btn {  
  background-color: #e74c3c;  
}  
  
.completed {  
  background-color: #c7f1c7;  
}
```

JS

```
JS script.js > document.addEventListener('DOMContentLoaded') callback > moveToImportant
1 document.addEventListener('DOMContentLoaded', () => {
2     // Получаем необходимые элементы из DOM
3     const taskInput = document.querySelector("#taskInput");
4     const addTaskBtn = document.querySelector("#addTaskBtn");
5     const taskList = document.querySelector("#taskList");
6     const importantTaskList = document.querySelector("#importantTaskList");
7     // Добавляем слушатель события для кнопки "Добавить задачу"
8     addTaskBtn.addEventListener('click', () => {
9         // Проверяем, что поле ввода не пусто
10        taskInput.value.trim() !== "" &&
11            (() => {
12                // Создаем новую задачу и добавляем ее в список общих задач
13                const newTask = createTaskElement(taskInput.value, moveToImportant);
14                taskList.appendChild(newTask);
15            })();
16        // Очищаем поле ввода
17        taskInput.value = "";
18    });
```

```
// Функция создания элемента задачи
const createTaskElement = (text, moveHandler) => {
    const newTask = document.createElement("li");
    const taskText = document.createTextNode(text);
    const deleteButton = createButton("Удалить", () => deleteTask(newTask));
    const moveToImportantButton = createButton("Важно!", () => moveHandler(newTask));
    // Добавляем текст и кнопки в элемент задачи
    newTask.appendChild(taskText);
    newTask.appendChild(deleteButton);
    newTask.appendChild(moveToImportantButton);
    // Добавляем слушатель события для выделения/снятия выделения задачи
    newTask.addEventListener('click', () => {
        newTask.classList.toggle('completed');
    });
    return newTask;
};

// Функция создания кнопки
const createButton = (text, clickHandler) => {
    const button = document.createElement("button");
    button.appendChild(document.createTextNode(text));
    button.addEventListener('click', clickHandler);
    return button;
};
```

```
// Функция перемещения задачи в список очень важных дел
const moveToImportant = (task) => {
    const importantTask = createTaskElement(task.firstChild.nodeValue, deleteTask);
    importantTaskList.appendChild(importantTask);
    taskList.removeChild(task);
};

// Функция удаления задачи
const deleteTask = (task) => {
    task.parentNode.removeChild(task);
};

});
```


Д/З

Создать 'Мессенджер' - форум
При отправке **send** сообщение
добавляется в контейнер с
сообщениями

The diagram illustrates a chat interface within a gray container. It features a white rounded rectangle containing a list of four messages and a text input area. The messages are represented as rounded rectangles with a light blue gradient, each containing a user name and a message text. The messages are ordered from top to bottom: User2 with #Текст4, User1 with #Текст3, User2 with #Текст2, and User1 with #Текст1. Below the list is a white rounded rectangle for text input, containing a vertical cursor. At the bottom of the container is a blue rounded rectangle button labeled 'SEND'.

User2
#Текст4

User1
#Текст3

User2
#Текст2

User1
#Текст1

|

SEND