

Создание веб-приложения «То-До список» на HTML и JavaScript

Саяпин Никита Анатольевич

Приамурский государственный университет им.Шолом-Алейхема

Студент

Аннотация

Цель статьи — создание простого веб-приложения «То-До список» на HTML, CSS и JavaScript. Реализация позволяет пользователю добавлять, удалять и помечать выполненные задачи. Проект демонстрирует основы фронтенд-разработки: структуру HTML-документа, стилизацию через CSS и реализацию логики на JavaScript. Такой проект является отличной основой для начинающих веб-разработчиков и может использоваться в качестве учебного примера.

Ключевые слова: To-Do список, HTML, CSS, JavaScript, веб-разработка, фронтенд.

Creating a "To-Do list" web application in HTML and JavaScript

Sayapin Nikita Anatolyevich

Sholom-Aleichem Priamursky State University

Student

Abstract

The purpose of this article is to create a simple To-Do list web application using HTML, CSS, and JavaScript. The implementation allows users to add, delete, and mark tasks as completed. The project demonstrates the basics of front-end development: HTML structure, CSS styling, and JavaScript logic. This project is ideal for beginners in web development and can serve as a basic learning example.

Keywords: To-Do list, HTML, CSS, JavaScript, web development, frontend.

1. Введение

1.1 Актуальность

Приложения «То-До список» являются важным инструментом в повседневной жизни, позволяя пользователю эффективно планировать задачи и управлять временем. Использование технологий HTML, CSS и JavaScript позволяет создать кроссплатформенное решение, работающее в любом современном браузере. В процессе разработки изучаются ключевые принципы клиентской веб-разработки: работа с DOM, обработка событий, стилизация и хранение данных в браузере.

1.2 Обзор исследований

В работе Дроздова С.В. рассмотрены базовые приёмы использования JavaScript в браузере, подчёркивается значимость манипуляций с DOM и событийной модели [1]. Лисовский С.А. и Сенькин М.В. демонстрируют создание простых веб-приложений с применением HTML5 и JavaScript, уделяя внимание организации интерфейсов [2]. Минин И.Ю. описывает методы реализации интерактивных элементов сайта и их роль в повышении удобства пользователя [3]. А. Громов рассматривает проектирование веб-интерфейсов на чистом JavaScript без использования сторонних библиотек [4]. А.С. Романенко делает акцент на использовании событий и работе с DOM при создании клиентских интерфейсов, что напрямую связано с функционалом To-Do приложений [5].

1.3 Цель исследования

Целью данного исследования является разработка простого веб-приложения «То-До список», реализующего основные функции управления задачами: добавление, удаление и отметка выполнения. В процессе создания используется HTML для структуры, CSS для оформления и JavaScript для логики взаимодействия.

2 Материалы и методы

В качестве основной среды разработки использовался текстовый редактор Visual Studio Code. Создание проекта началось с формирования структуры файлов, включающей HTML-документ, таблицу стилей и скрипт JavaScript. Каждый из файлов выполняет определённую функцию: HTML отвечает за структуру, CSS — за визуальное представление, а JavaScript — за логику приложения.

2.2 Подготовка

Для начала создается папка проекта с названием ToDoApp (рис.1). Внутри неё размещаются следующие файлы:

- index.html — основной HTML-документ
- style.css — таблица стилей для оформления интерфейса
- script.js — скрипт JavaScript, реализующий функционал приложения

Рекомендуется использовать современный браузер (например, Google Chrome) для тестирования приложения и оперативного просмотра изменений.

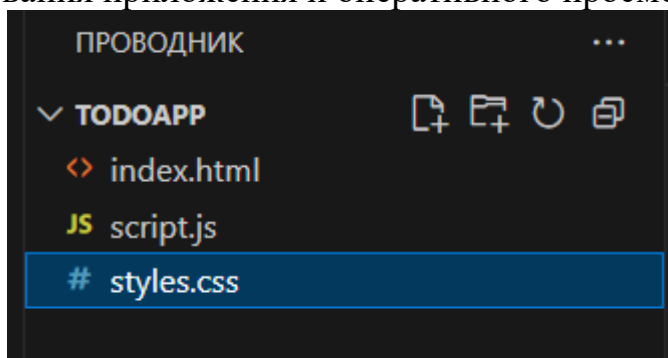
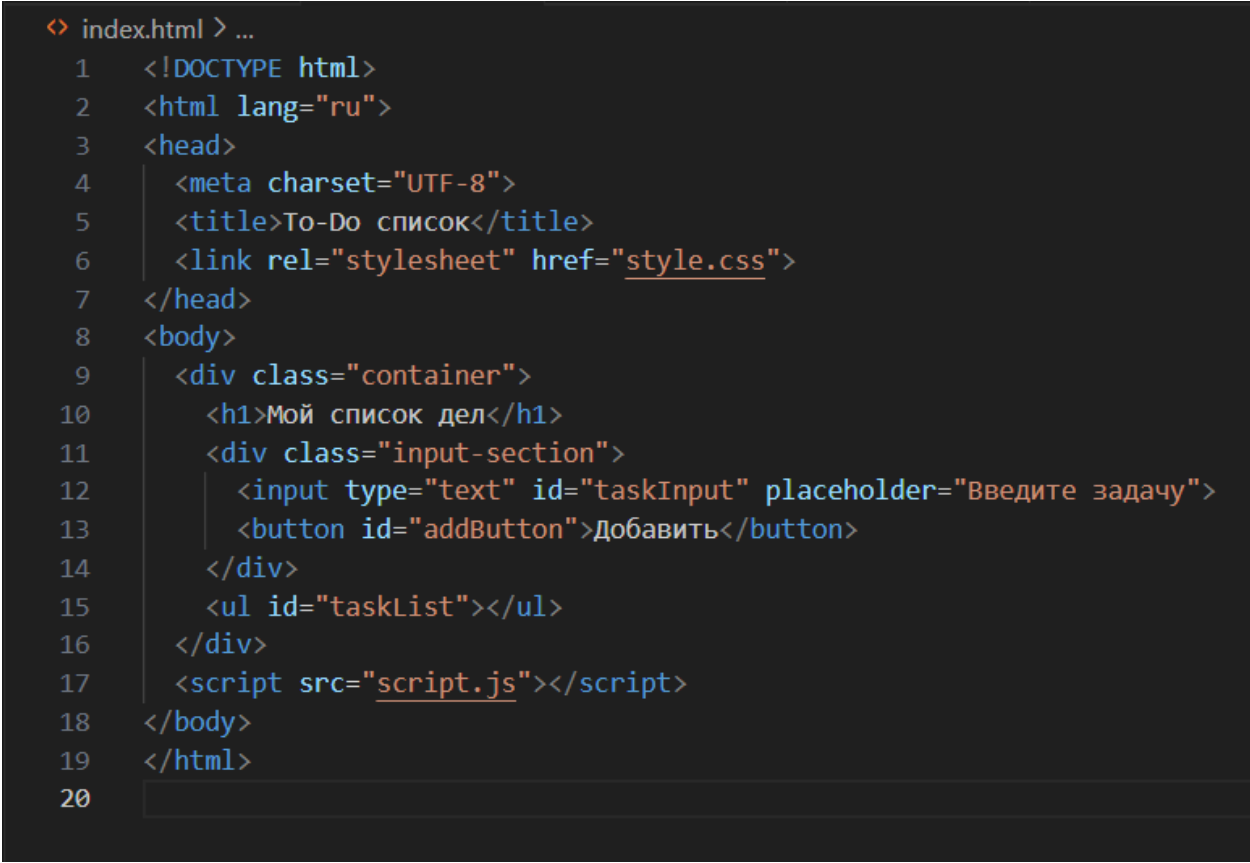


Рисунок 1 – Создание папки ToDoApp

3. Результаты

HTML-документ является основой веб-страницы и описывает её структуру. В данной реализации он содержит заголовок, поле для ввода новой задачи, кнопку добавления и список, в котором будут отображаться все задачи. (Рис.2).



```
index.html > ...
1  <!DOCTYPE html>
2  <html lang="ru">
3  <head>
4    <meta charset="UTF-8">
5    <title>To-Do список</title>
6    <link rel="stylesheet" href="style.css">
7  </head>
8  <body>
9    <div class="container">
10     <h1>Мой список дел</h1>
11     <div class="input-section">
12       <input type="text" id="taskInput" placeholder="Введите задачу">
13       <button id="addButton">Добавить</button>
14     </div>
15     <ul id="taskList"></ul>
16   </div>
17   <script src="script.js"></script>
18 </body>
19 </html>
20
```

Рисунок 2 – Реализация файла index.html

В блоке `<div class="container">` размещаются все элементы управления. Поле ввода предназначено для ввода новой задачи, кнопка «Добавить» инициирует добавление задачи в список. Список `<ul>` с идентификатором `taskList` будет заполняться элементами задач при работе скрипта.

Для улучшения визуального восприятия создаётся таблица стилей `style.css`. Она определяет внешний вид элементов, таких как фон страницы, контейнер задач, кнопки и текстовые поля (Рис.3).

```
# styles.css > ...
1  body {
2      font-family: Arial, sans-serif;
3      background-color: #f2f2f2;
4      display: flex;
5      justify-content: center;
6      padding-top: 50px;
7  }
8  .container {
9      background: white;
10     padding: 20px;
11     border-radius: 10px;
12     width: 400px;
13     box-shadow: 0 4px 6px rgba(0,0,0,0.1);
14 }
15
```

Рисунок 3 – Работа над стилем сайта

Эти стили делают интерфейс современным и удобным. Контейнер задач выравнивается по центру экрана, а элементы внутри него располагаются с отступами (Рис.4).

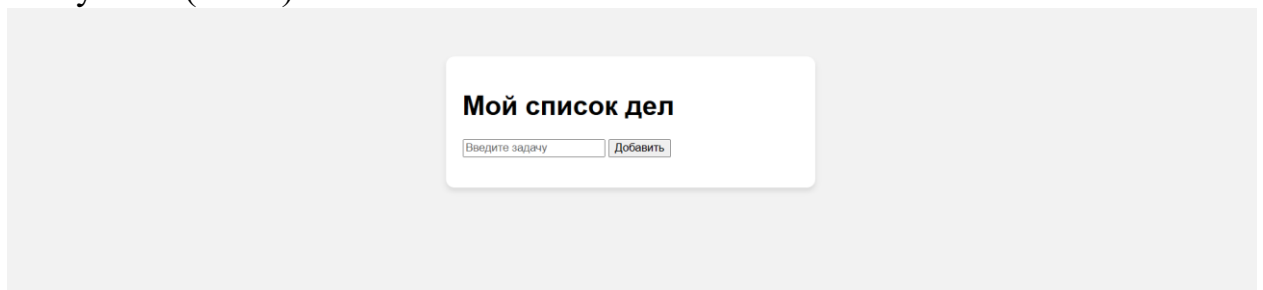


Рисунок 4 – Сайт с минималистичным стилем

JavaScript-файл script.js содержит код, который отвечает за добавление задач, их удаление и отметку о выполнении. Основная задача скрипта — реагировать на действия пользователя и изменять содержимое DOM (Рис.5).

```
JS script.js
1  const taskInput = document.getElementById('taskInput');
2  const addButton = document.getElementById('addButton');
3  const taskList = document.getElementById('taskList');
4
5  addButton.addEventListener('click', () => {
6    const taskText = taskInput.value.trim();
7    if (taskText !== '') {
8      const li = document.createElement('li');
9      li.textContent = taskText;
10
11     li.addEventListener('click', () => {
12       li.classList.toggle('completed');
13     });
14
15     const deleteBtn = document.createElement('button');
16     deleteBtn.textContent = 'Удалить';
17     deleteBtn.addEventListener('click', () => {
18       taskList.removeChild(li);
19     });
20
21     li.appendChild(deleteBtn);
22     taskList.appendChild(li);
23     taskInput.value = '';
24   }
25 });
26
```

Рисунок 5 – Реализация скриптов приложения

Код создаёт новую задачу при нажатии кнопки «Добавить». Каждая задача — это элемент `<li>` с текстом и кнопкой удаления. По нажатию на саму задачу она помечается как выполненная, при этом применяется класс `completed`, который можно стилизовать в CSS. Для каждого элемента предусмотрена функция удаления через кнопку (Рис.6).

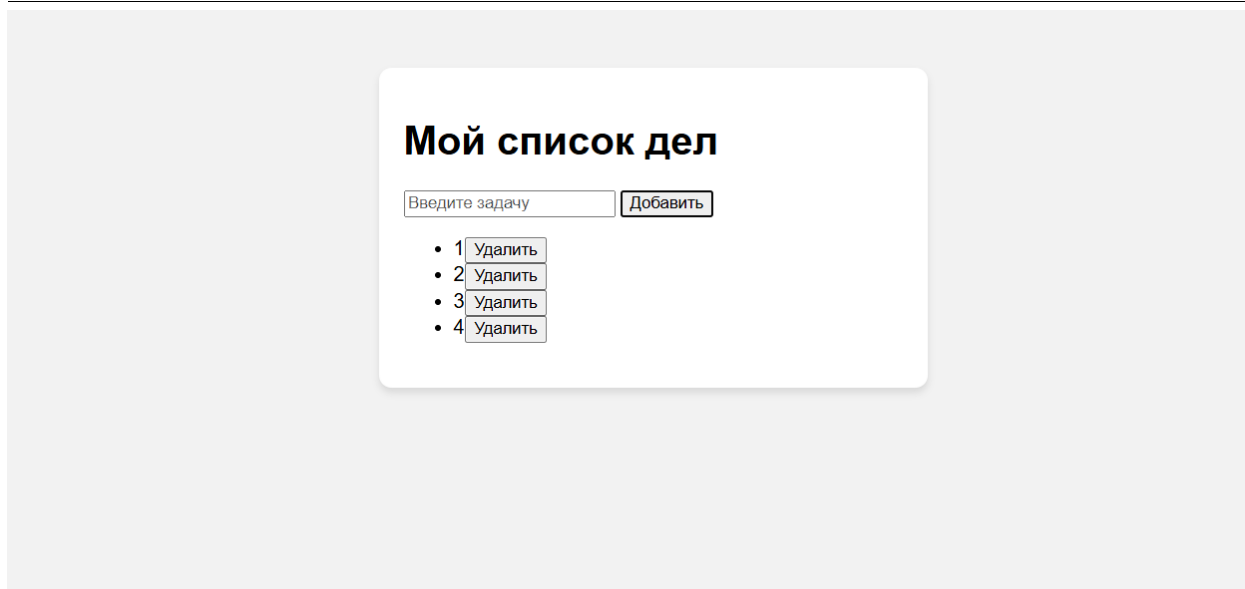


Рисунок 6 – Пример готового веб приложения

4 Выводы

В результате проделанной работы было разработано простое, но функциональное веб-приложение «To-Do список», реализованное с применением HTML, CSS и JavaScript. В процессе разработки были изучены основы клиентской веб-разработки, включая создание структуры HTML-документа, стилизацию с использованием CSS и реализацию логики взаимодействия с пользователем через JavaScript. Приложение позволяет пользователю создавать задачи, отмечать их выполнение и удалять. Созданный проект может служить базовой основой для изучения фронтенд-технологий и расширяться дополнительными функциями, такими как сохранение данных в localStorage, использование библиотек или интеграция с серверной частью. Таким образом, задача, поставленная в рамках исследования, была успешно выполнена, а полученный результат демонстрирует возможности начального уровня веб-программирования.

Библиографический список

1. Дроздов С.В. Основы JavaScript. Руководство по началу веб-программирования // Современные наукоемкие технологии. 2021. № 6-1. С. 45–48.
2. Лисовский С.А., Сенькин М.В. Разработка веб-приложений средствами HTML5 и JavaScript // Молодой ученый. 2020. № 22. С. 107–110.
3. Минин И.Ю. Применение JavaScript при создании интерактивных элементов веб-сайта // Информационные технологии и нанотехника. 2022. № 2. С. 56–59.
4. Громов А. Создание интерфейсов пользователя с использованием чистого JavaScript // Интернет-журнал «Программирование». 2023. № 4. С. 22–28.
5. Романенко А.С. Использование DOM и событий в JavaScript при разработке веб-приложений // Информационные технологии. 2021. № 9. С. 81–85.