

УДК 004.4

СОЗДАНИЕ ВЕБ-ПРИЛОЖЕНИЯ СРЕДСТВАМИ HTML, CSS, JAVASCRIPT

Домбровский Я.А.

старший преподаватель

*Калужский государственный университет им. К.Э. Циолковского,
Калуга, Россия*

Ткачев Е.В.

магистрант

*Калужский государственный университет им. К.Э. Циолковского,
Калуга, Россия*

Аннотация.

Создание веб-приложений с использованием HTML, CSS и JavaScript является основой современной веб-разработки. Эти три технологии позволяют создавать интерактивные, функциональные и визуально привлекательные веб-страницы и приложения. В данной статье представлен обзор основных этапов создания веб-приложения средствами HTML, CSS и JavaScript, а также приводятся примеры реализации различных элементов интерфейса и функциональности. Особое внимание уделено использованию современных практик и библиотек, которые помогают ускорить процесс разработки и улучшить качество готового продукта.

Ключевые слова: HTML, CSS, JavaScript, веб-приложения, веб-разработка, структуры веб-страниц, веб-дизайн.

CREATING A WEB APPLICATION USING HTML, CSS, JAVASCRIPT

Dombrovsky Y.A.

Senior Lecturer

*Kaluga State University named after K.E. Tsiolkovsky,
Kaluga, Russia*

Tkachev E.V.

Master's student

Kaluga State University named after K.E. Tsiolkovsky,

Kaluga, Russia

Annotation.

Creating web applications using HTML, CSS, and JavaScript is the foundation of modern web development. These three technologies allow you to create interactive, functional, and visually appealing web pages and applications. This article provides an overview of the main stages of creating a web application using HTML, CSS, and JavaScript, as well as examples of the implementation of various interface elements and functionality. Special attention is paid to the use of modern practices and libraries that help speed up the development process and improve the quality of the finished product.

Keywords: HTML, CSS, JavaScript, web applications, web development, web page structures, web design.

Сегодня веб-приложения являются неотъемлемой частью нашей повседневной жизни. Они используются для самых разных целей, от простых статичных веб-страниц до сложных многофункциональных систем, таких как корпоративные порталы, онлайн-магазины и социальные сети. Создание веб-приложений требует глубокого понимания трех основных технологий: HTML, CSS и JavaScript. Веб-разработчики используют HTML для разметки структуры страницы, CSS для оформления внешнего вида и JavaScript для добавления динамического поведения и интерактивности. Эти технологии образуют основу для большинства веб-проектов, независимо от их сложности и назначения. В статье мы рассмотрим каждую из этих технологий, их основные особенности и способы их совместного использования для создания веб-приложений.

HTML (HyperText Markup Language) представляет собой язык разметки, служащий для формирования структуры веб-страниц. Основой HTML служат Дневник науки | www.dnevniknauki.ru | СМИ Эл № ФС 77-68405 ISSN 2541-8327

специальные теги, которые задают различные элементы страницы, такие как заголовки, абзацы, изображения, гиперссылки и таблицы [4]. HTML5, последняя версия языка, добавила новые теги и функции, такие как семантические элементы (<header>, <footer>, <nav>), мультимедийные элементы (<video>, <audio>) и улучшенные формы (рисунок 1).

```
1 | <!DOCTYPE html>
2 | <html lang="ru">
3 | <head>
4 |     <meta charset="UTF-8">
5 |     <title>Моя первая веб-страница</title>
6 | </head>
7 | <body>
8 |     <h1>Привет, мир!</h1>
9 |     <p>Это моя первая веб-страница, созданная с использованием HTML.</p>
10 | </body>
11 | </html>
```

Рис.1 – Пример простого HTML-документа (составлено авторами)

Семантическая разметка – это практика использования HTML-элементов, которые точно описывают назначение их содержимого. Это помогает поисковикам и другим программам лучше понимать структуру документа [2]. Примеры семантических элементов: <article>, <section>, <aside>.

HTML5 предоставил новые возможности для создания форм и обработки данных, включая проверку введенных значений и улучшенную поддержку различных типов полей ввода [3].

CSS (Cascading Style Sheets) – это язык, применяемый для стилизации веб-страниц. С его помощью разработчики могут управлять внешним видом элементов HTML, определяя такие параметры, как шрифт, цвет фона, размеры блоков, расположение контента и многое другое [4]. CSS может быть встроен непосредственно в HTML-код, включен в отдельные файлы или загружен с внешних источников (рисунок 2).

```
1  body {  
2      font-family: Arial, sans-serif;  
3      background-color: #f0f0f0;  
4  }  
5  
6  h1 {  
7      color: blue;  
8      text-align: center;  
9  }  
10  
11 p {  
12     margin: 20px;  
13 }
```

Рис.2 – Пример использования CSS (составлено авторами)

Одним из ключевых понятий в CSS является селектор, который определяет, к какому элементу применяются стили. Существует несколько видов селекторов: идентификаторы, классы, теги и псевдоклассы. Каскадность означает, что стили применяются последовательно, начиная с самого общего стиля и заканчивая самыми специфическими правилами.

Flexbox – это мощная модель компоновки, которая позволяет легко выстраивать элементы в ряд или столбец (рисунок 3).

```
1  .container {
2      display: flex;
3      justify-content: center;
4      align-items: center;
5      height: 100vh;
6  }
7
8  .item {
9      flex-grow: 1;
10     margin: 10px;
11     padding: 20px;
12     border: 1px solid #ccc;
13     text-align: center;
14 }
15
16 @media (max-width: 768px) {
17     .container {
18         flex-direction: column;
19     }
20 }
```

Рис.3 – Пример использования flexbox для создания адаптивного макета
(составлено авторами)

Grid Layout – это еще одна мощная модель компоновки, которая позволяет создавать сложные многостолбцовые макеты (рисунок 4).

```
1  .wrapper {
2      display: grid;
3      grid-template-columns: repeat(auto-fill, minmax(200px, 1fr));
4      gap: 20px;
5  }
6
7  .item {
8      background-color: #f9f9f9;
9      border: 1px solid #ccc;
10     padding: 20px;
11     text-align: center;
12 }
13
14 @media (max-width: 600px) {
15     .wrapper {
16         grid-template-columns: 1fr;
17     }
18 }
```

Рис.4 – Пример использования grid для создания макета главной страницы
(составлено авторами)

JavaScript – это язык программирования, который добавляет интерактивность и динамику веб-страницам. Он позволяет реагировать на действия пользователя, манипулировать элементами DOM, отправлять запросы на сервер и многое другое [5]. JavaScript может быть встроен непосредственно в HTML-код или загружен из внешних файлов (рисунок 5).

```
1 document.querySelector("button").addEventListener("click", function() {  
2     alert("Вы нажали на кнопку!");  
3 });
```

Рис.5 – Пример использования JavaScript (составлено авторами)

DOM (Document Object Model) представляет собой объектную модель документа, которая организует HTML-документ в виде иерархии объектов. Используя JavaScript, можно динамически изменять структуру и содержимое веб-страницы, манипулируя этими объектами в режиме реального времени.

AJAX (Asynchronous JavaScript and XML) – это методика, благодаря которой данные могут загружаться с сервера без необходимости полной перезагрузки страницы. Такая техника существенно повышает удобство использования веб-приложений, делая их более оперативными и плавными в работе.

В экосистеме JavaScript существует большое количество библиотек и фреймворков, которые заметно упрощают процесс разработки. Среди наиболее известных примеров можно выделить jQuery, React, Angular и Vue.js [1]. Эти инструменты предоставляют разработчикам готовые решения для часто встречающихся задач, что значительно ускоряет создание приложений и улучшает их качество.

Для создания полноценного веб-приложения необходимо объединить HTML, CSS и JavaScript. HTML определяет структуру страницы, CSS придает ей стиль, а JavaScript добавляет интерактивность и функциональность. Все три компонента работают вместе, чтобы создать законченный продукт (рисунок 6).

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>My App</title>
7   <link rel="stylesheet" href="styles.css">
8   <script src="app.js" defer</script>
9 </head>
10 <body>
11   <h1>Welcome to My App</h1>
12   <button onclick="showMessage()">Click Me</button>
13   <p id="message"></p>
14 </body>
15 </html>
```

```
1 body {
2   font-family: Arial, sans-serif;
3   background-color: #f0f0f0;
4 }
5
6 h1 {
7   text-align: center;
8 }
9
10 #message {
11   display: none;
12 }
```

```
1 function showMessage() {
2   let message = document.getElementById("message");
3   message.textContent = "Hello, World!";
4   message.style.display = "block";
5 }
```

Рис.6 – Пример простого веб-приложения (составлено авторами)

При разработке веб-приложений важно следовать лучшим практикам, чтобы обеспечить высокое качество кода и производительность. Некоторые из них включают:

– Использование семантической разметки: Применение правильных HTML-элементов помогает улучшить SEO и доступность.

– Минификация и сжатие: Сокращение размеров файлов CSS и JavaScript для уменьшения времени загрузки.

– Кэширование: Хранение статичного контента в кэше браузера для ускорения последующих посещений.

– Адаптация под мобильные устройства: Использование медиа-запросов и адаптивного дизайна для поддержки различных экранов.

– Тестирование и отладка: Регулярное тестирование кода и использование инструментов отладки для выявления и исправления ошибок.

Разработка веб-приложений с применением HTML, CSS и JavaScript представляет собой универсальный и действенный подход, позволяющий создавать современные и высокоэффективные веб-решения. Глубокое понимание основ каждой из этих технологий и умение гармонично сочетать их между собой служит залогом успеха в сфере веб-разработки. В рамках данной статьи были рассмотрены базовые концепции и практические примеры использования HTML, CSS и JavaScript, а также даны рекомендации и советы по созданию высококачественных веб-приложений.

Библиографический список:

1. Макдональд, М. Веб-разработка. Исчерпывающее руководство. — СПб.: Питер, 2017. — 640 с.
2. Фрэйн, Б. HTML5 и CSS3. Разработка сайтов для любых браузеров и устройств. — СПб.: Питер, 2017. — 272 с.
3. Хоган, Б. HTML5 и CSS3. Веб-разработка по стандартам нового поколения. 2-е изд. — СПб.: Питер, 2014. — 320 с.
4. Duckett, J. HTML and CSS: Design and Build Websites. — Hoboken, NJ: John Wiley & Sons, 2014. — 527 p.
5. Flanagan, D. JavaScript: The Definitive Guide. — Sebastopol, CA: O'Reilly Media, 2011. — 1096 p.

Оригинальность 75%

