

ОСОБЕННОСТИ СОВРЕМЕННЫХ МЕТОДОВ СОЗДАНИЯ И ОФОРМЛЕНИЯ САЙТОВ

Д.В. Малахов, генеральный директор, эксперт в создании сайтов на основе WordPress

WEB STUDIO DM LLC

(Россия, г. Люберцы)

DOI:10.24412/2500-1000-2024-11-3-194-200

Аннотация. В статье анализируются современные методы создания и оформления сайтов. Рассматриваются технологии веб-разработки, используемые для создания клиентской и серверной сторон сайта. Приводятся передовые технологии и инструменты веб-разработки, позволяющие создавать высокопроизводительные ресурсы. Отмечается, что в части оформления веб-сайтов ключевой тенденцией является создание адаптивных платформ. Приводится ряд элементов и функций, внедряемых веб-дизайнерами для улучшения пользовательского опыта.

Ключевые слова: веб-сайт, разработка, язык программирования, веб-дизайн, адаптивность.

Корпоративный сайт сегодня является ключевым элементом успешного бизнеса, представляя собой виртуальное отражение компании. Он используется для установления, поддержания и управления отношениями с клиентами [1]. Веб-сайт компании выступает в роли многофункциональной цифровой платформы, которая одновременно служит инструментом для формирования бренда и площадкой для взаимодействия. Он выполняет широкий спектр задач: от предоставления информации до реализации маркетинговых и рекламных стратегий. Основной задачей такого ресурса является установление взаимовыгодных коммерческих связей с клиентами и деловыми партнерами. Чтобы эффективно использовать потенциал сайта, владельцам бизнеса важно учитывать современные тренды в веб-разработке и дизайне, привлекая к работе квалифицированных специалистов, которые обеспечат создание и поддержку интернет-площадки.

Целью данного исследования является анализ современных методов разработки и оформления сайтов. Для достижения этой цели были проведены изучение и систематизация материалов, посвященных созданию веб-ресурсов, а также применен системно-структурный подход, позволяющий рассмотреть ключевые аспекты поставленной задачи.

Веб-разработка включает совокупность инструментов, которые используются для создания сайтов [2]. Все технологии разделяются на две основные категории:

Клиентская часть (фронтенд). Сюда входят элементы, с которыми пользователь взаимодействует, такие как структура страниц, их внешний вид (дизайн), управление, навигация и анимация. Основная задача фронтенд-разработчика заключается в создании удобного и интуитивно понятного интерфейса. Серверная часть (бэкенд). Она отвечает за внутреннюю работу сайта, включая обслуживание серверов, управление базами данных и организацию архитектуры ресурса. Бэкенд-разработчики несут ответственность за стабильное функционирование сайта и защиту данных.

Ключевые технологии, которые применяются для конструирования интерфейсов для пользователя и оформления веб-сайтов, включают HTML и CSS [3]. HTML, являясь стандартным языком гипертекстовой разметки, используется для структурирования и организации содержимого страницы (рис. 1). С помощью HTML определяется, как на экране будут выглядеть такие элементы, как заголовки, текст, ссылки, изображения и другие компоненты.



Рис. 1. Пример HTML-кода

CSS представляет собой формальный язык, предназначенный для определения внешнего оформления документа и отвечающий за его визуальную составляющую (рис. 2). С помощью CSS задаются ключевые параметры внешнего вида, включая размеры шрифтов, цветовые решения, отступы, интервалы и другие визуальные характеристики.

В современной веб-разработке широко используются классические CSS-фреймворки, такие как Bootstrap и Foundation. Эти инстру-

менты являются стандартом для создания адаптивного и функционального дизайна. Однако в последние годы всё больше популярность начинают пользоваться утилитарные CSS-фреймворки, такие как Tailwind CSS. Они предоставляют более удобный и модульный подход к конструированию интерфейсов, что позволяет разработчикам легко адаптировать стиль под конкретные требования проекта.

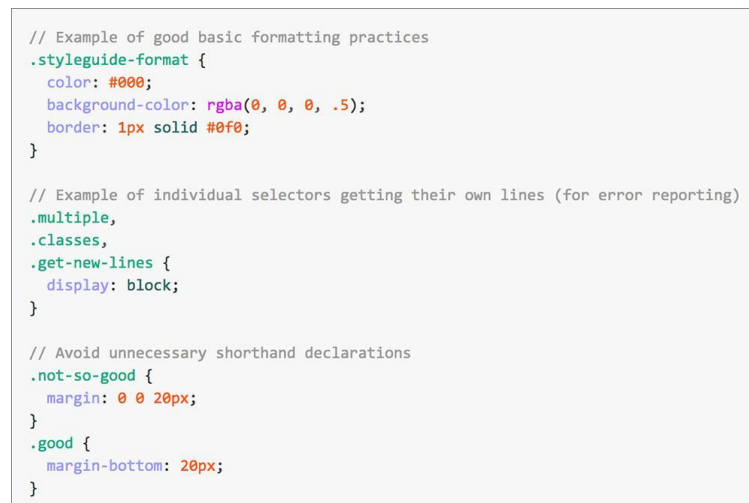


Рис. 2. Пример CSS-кода

HTML и CSS составляют фундамент любого веб-сайта и являются основными инструментами фронтенд-разработчиков. Наряду с ними, в эту категорию входит JavaScript, который используется для добавления интерактивности на веб-страницы [4].

В отличие от фронтенд-разработчиков, бэкенд-специалисты применяют более широкий спектр технологий. Одним из самых старых, но по-прежнему востребованных языков веб-разработки является PHP, на основе которого функционирует около 76% сайтов [5]. Этот язык обладает динамической типизацией, он

отличается узкой специализацией и подходит для конструирования сайтов различной сложности и масштаба.

Для работы с базами данных используется такой популярный и широкоизвестный язык, как SQL. Это язык структурированных запросов, который обеспечивает управление данными. Разработка приложений для серверной части часто выполняется на Python, универсальном языке с динамической типизацией. Его ключевыми преимуществами являются лаконичный синтаксис и обширный набор фреймворков и библиотек, таких как Django,

Pyramid, Web2py, Flask и Bottle. Для создания бэкенда высоконагруженных систем, где критически важна безопасность, применяется Java – компилируемый язык с богатым набором инструментов (например, Spring, Spark, Wicket).

Для исполнения JavaScript на серверной стороне используется Node.js – среда, которая позволяет вести разработку фронтенда и бэкенда на одном языке.

В сфере электронной коммерции и разработки стартапов, где требуется реализация сложных решений, активно применяется язык Ruby с лаконичным синтаксисом и динамической типизацией. Как правило, он используется вместе с фреймворком Ruby on Rails, специально созданным для разработки веб-приложений.

Современная архитектура Jamstack, предназначенная для создания масштабируемых, быстрых и безопасных сайтов, объединяет статические сайты, JavaScript и API для серверной функциональности [6].

При выборе технологии для создания интерфейса в первую очередь, необходимо учитывать цели и задачи проекта, а также целевую аудиторию. Каждый проект уникален, и выбор технологий должен соответствовать требованиям пользователей. Например, для создания адаптивного интерфейса, который будет хорошо отображаться на разных устройствах, необходимо использовать CSS медиа-запросы и фреймворки, такие как Bootstrap или Foundation.

Использование CSS-препроцессоров, таких как SASS (Syntactically Awesome Style Sheets) или LESS, может значительно улучшить разработку стилей. Они добавляют возможности,

такие как вложенность, переменные и миксины, что делает код более организованным и удобным в поддержке. Преимущества:

- Упрощенные стили: позволяют писать более чистый и поддерживаемый код.

- Повторное использование кода: миксины и переменные помогают избежать дублирования.

Если есть необходимость ускорить процесс разработки интерфейса, лучше рассмотреть возможность использования CSS-фреймворков. Наиболее популярные из них:

- Bootstrap: предлагает набор готовых компонентов и шаблонов, что позволяет быстро разрабатывать адаптивные интерфейсы.

- Tailwind CSS: это утилитарный CSS-фреймворк, который позволяет создавать уникальные дизайны, используя классы для стилизации непосредственно в HTML.

Преимущества:

- Скорость разработки: готовые компоненты позволяют сократить время на верстку.

- Адаптивность: большинство фреймворков уже оптимизированы для мобильных устройств.

Среди передовых технологий, применяемых в разработке высокопроизводительных веб-приложений, особое место занимает WebAssembly (Wasm). Это низкоуровневый двоичный формат, позволяющий исполнять код в веб-браузерах с производительностью, близкой к нативной. WebAssembly предоставляет разработчикам возможность использовать такие языки программирования, как C, C++ и Rust, для написания высокопроизводительного кода. Этот код компилируется в модули Wasm, которые затем могут выполняться в браузере наряду с JavaScript (рис. 3).

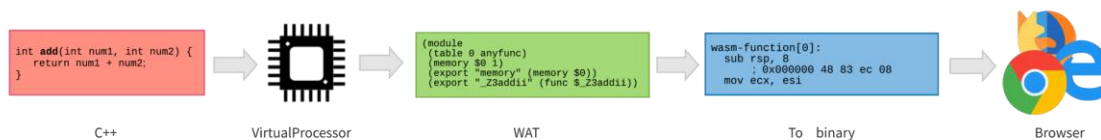


Рис. 3. Процесс преобразования функции для сложения чисел, написанной на C++, в бинарный формат с использованием WebAssembly

Современная веб-разработка активно внедряет серверно-ориентированные пользовательские интерфейсы. Этот подход предполагает динамическую генерацию макетов и элементов интерфейса серверной логикой, осно-

ванной на данных и контексте пользователя. Такой метод способствует более тесной интеграции внутреннего и внешнего интерфейсов, что упрощает процесс разработки и повышает

удобство взаимодействия для конечных пользователей.

Одной из ведущих тенденций в дизайне веб-сайтов продолжает быть создание платформ с адаптивным и отзывчивым дизайном (рис. 4) [7, 8].

Отзывчивый веб-дизайн представляет собой подход к созданию контента, который корректно отображается на устройствах с любым разрешением экрана. Для этого используются контрольные точки области просмотра, что позволяет автоматически адаптировать макет для планшетов, смартфонов и настольных компьютеров. В таком дизайне правила изменения структуры и отображения содер-

жимого зависят от размеров экрана пользователя.

В свою очередь, адаптивный дизайн предлагает одинаковый контент и функциональность для всех устройств, но использует разные макеты. Этот подход является эволюцией отзывчивого дизайна, поскольку учитывает не только характеристики устройства, но и контекст его использования. Это позволяет создавать контекстно-зависимые взаимодействия, благодаря которым функциональность и содержание веб-приложения могут отличаться от десктопной версии, что повышает удобство использования и делает дизайн более гибким.

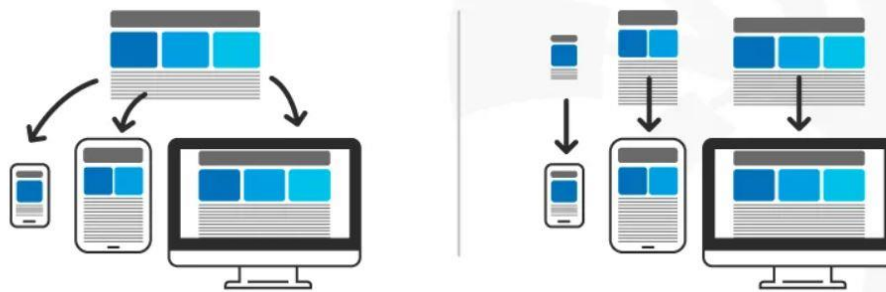


Рис. 4. Различия между адаптивным (справа) и отзывчивым (слева) дизайном

Современный веб-дизайн направлен на улучшение пользовательского опыта и активно использует новейшие технологии и подходы. Помимо адаптивности, дизайнеры внедряют в работу разнообразные элементы и функции, которые делают взаимодействие с веб-ресурсами более увлекательным и удобным [9].

- Анимация прокрутки. При выполнении пользователем прокрутки активируются визуальные эффекты и анимации, добавляющие динамичности.

- Микровзаимодействия. Небольшие анимации, которые дают пользователям тонкую обратную связь при взаимодействии с элементами интерфейса.

- Микроанимация. Используется для демонстрации продуктов, предоставляя покупателям более наглядное и привлекательное представление.

- Динамические курсоры. При взаимодействии с сайтом курсор изменяет свою форму или запускает анимационные эффекты, что делает пользовательский опыт более интерактивным.

- Интерактивные 3D-элементы. Трехмерный контент, с которым пользователи могут взаимодействовать, создавая эффект погружения.

- Геймифицированный дизайн. Включает игровые элементы, которые мотивируют пользователя проводить больше времени на сайте, изучая информацию о компании.

- Материальный дизайн. Основан на использовании цветов, теней и текстур, имитирующих физический мир, что придает интерфейсу реалистичности и глубины.

Одним из подходов, активно применяемых для разработки современных интерфейсов, является атомарный веб-дизайн [10]. Этот метод заключается в разбиении дизайна на мельчайшие независимые элементы, называемые атомами. Атомы комбинируются в молекулы, молекулы объединяются в организмы. Из организмов формируются шаблоны, которые составляют каркас веб-страницы. На заключительном этапе в шаблон добавляется контент, что позволяет получить полноценную страницу (рис. 5).

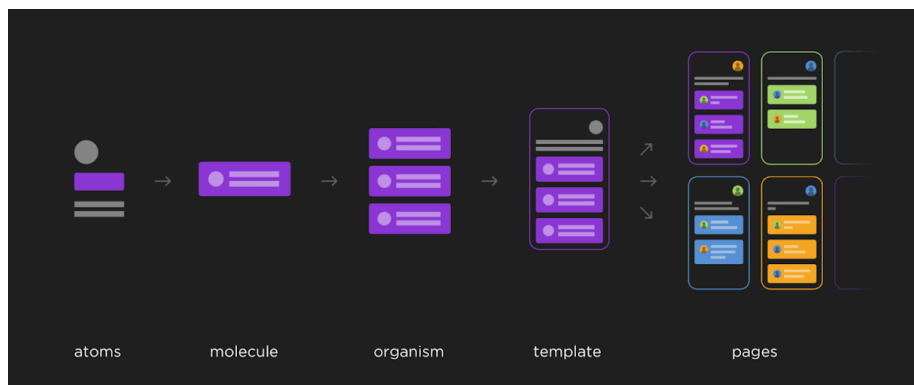


Рис. 5. Компоненты атомарного веб-дизайна

Атомарный дизайн ярко отражает современные тенденции в веб-разработке. Все большее распространение получают подходы, методы и инструменты, отличающиеся высокой гибкостью и масштабируемостью. Эти технологии позволяют оптимально распределять ресурсы, повышать продуктивность командной работы, а также значительно упрощают поддержку, сопровождение и обеспечение доступности веб-ресурсов. Такой подход отвечает требованиям динамично развивающегося цифрового мира, делая процессы разработки более эффективными и адаптируемыми под разнообразные задачи.

Современный интернет – это динамичная и многообразная среда, где пользователи взаимодействуют с множеством веб-сайтов через различные браузеры. Однако проблема совместимости браузеров остается одной из наиболее распространенных и сложных задач для веб-разработчиков. Причины проблемы совместимости:

1. Разнообразие браузеров: Существует множество браузеров – Google Chrome, Mozilla Firefox, Safari, Microsoft Edge и другие. Каждый из них использует свои движки для обработки HTML, CSS и JavaScript, что может приводить к различиям в отображении и функционировании веб-страниц.

2. Версии браузеров: Браузеры часто обновляются, и новые версии могут вводить изменения, которые влияют на веб-стандарты. Это может приводить к тому, что сайт, корректно отображавшийся в одной версии, может работать некорректно в другой.

3. Стандарты и спецификации: Хотя существуют общепринятые стандарты, такие как W3C, не все браузеры их поддерживают в одинаковой степени. Некоторые браузеры мо-

гут поддерживать новые технологии быстрее других, в то время как устаревшие версии могут не поддерживать их вовсе.

Последствия несовместимости:

- Увеличение времени разработки: Веб-разработчикам часто приходится тестировать свои сайты в различных браузерах, что увеличивает время разработки и повышает затраты на проект.

- Плохой пользовательский опыт: Если сайт работает некорректно в каком-то браузере, пользователи могут испытывать трудности при его использовании, что может привести к потере клиентов и ухудшению репутации компании.

- SEO-проблемы: Поисковые системы учитывают пользовательский опыт при ранжировании сайтов. Если пользователи покидают страницу из-за её некорректного отображения, это может негативно повлиять на позиции сайта в результатах поиска.

Решения проблемы совместимости

1. Кроссбраузерное тестирование**: Регулярное тестирование сайтов на различных браузерах и устройствах – это ключ к выявлению и устранению проблем совместимости. Существуют инструменты и платформы (например, BrowserStack, CrossBrowserTesting), которые позволяют проверять, как ваш сайт отображается в разных условиях.

2. Использование новых технологий с осторожностью: Перед тем как использовать новые технологии (например, CSS Grid или Flexbox), важно удостовериться в их поддержке в целевых браузерах. Существует множество ресурсов, таких как Can I use, которые позволяют проверить совместимость веб-технологий.

3. Применение полифилов: Полифилы – это специальные библиотеки, которые позволяют использовать новые функции в устаревших браузерах. Например, вы можете использовать полифилы для поддержки ES6 в Internet Explorer.

4. Responsive Design: Респонсивный дизайн помогает адаптировать сайт под разные устройства и разрешения экрана, что также важный аспект совместимости, особенно с учетом растущего числа мобильных пользователей.

Еще одной проблемой является обеспечение адаптивности сайтов. Адаптивный веб-дизайн (AWD) – это подход, при котором сайт автоматически изменяет свое отображение в зависимости от размера экрана устройства, на котором он просматривается. Это значит, что один и тот же сайт будет удобен для пользователей как на компьютерах, так и на смартфонах и планшетах. С другой стороны, мобильная версия сайта – это специальная версия, оптимизированная только для мобильных устройств.

Создание адаптивного сайта требует более сложного кода и дизайна. Это может увеличить время разработки и повысить стоимость проекта. Необходимость тестирования на различных устройствах также добавляет дополнительную нагрузку на команду.

Решения и подходы:

1. Использование фреймворков

Существует множество фреймворков, таких как Bootstrap и Foundation, которые облегчают процесс создания адаптивных сайтов.

Эти инструменты предлагают готовые компоненты, которые можно использовать для быстрого и качественного дизайна.

2. Тестирование на различных устройствах

Регулярное тестирование сайта на разных устройствах и разрешениях поможет выявить и исправить проблемы с отображением, что обеспечит лучший пользовательский опыт.

3. Проектирование с учетом мобильных пользователей

Важно начинать создание сайта с мобильной версии в первую очередь (подход Mobile First). Это позволит сосредоточиться на необходимых функциях и минимализме, добавляя все остальные элементы по мере расширения для более крупных экранов.

4. Анализ пользовательского поведения

Это поможет понять, какой контент наиболее важен для мобильных пользователей, и соответственно адаптировать сайт.

Таким образом, создание веб-сайта – комплексная задача, которая требует технических компетенций, творческого подхода, анализа и понимания потребностей целевой аудитории. От верного выбора инструментов и технологий разработки и оформления зависит успешность привлечения внимания потенциальных покупателей и величина конверсии. Сфера веб-разработки как часть цифровой индустрии активно развивается, что требует от владельцев сайтов, разработчиков и дизайнеров высокой адаптивности, изучения и использования новых тенденций и передового опыта.

Библиографический список

1. Никитина О.О. Роль веб-сайта как главного коммуникационного канала компании в современном контексте // Экономические исследования и разработки. – 2022. – № 6. – С. 65-70.
2. Инструменты для веб-разработки // Optimal group. – 2024. – [Электронный ресурс]. – Режим доступа: <https://optimalgroup.ru/blog/instrumenty-dlya-veb-razrabotki/> (дата обращения: 09.08.2024).
3. Левитов А. ТОП-5 популярных технологий для создания сайтов: просто и понятно для новичков // Proglib. – 2021. – [Электронный ресурс]. – Режим доступа: <https://proglib.io/p/top-5-populyarnyh-tehnologiy-dlya-sozdaniya-saytov-prosto-i-ponyatno-dlya-novichkov-2021-05-19> (дата обращения: 09.08.2024).
4. Кононенко А. Технологии для разработки сайта: что это, какие наиболее популярны и как выбрать // IAMPM. – 2024. – [Электронный ресурс]. – Режим доступа: <https://iampm.club/blog/tehnologii-dlya-razrabotki-sajta-cto-eto-kakie-naibolee-populyarny-i-kak-vybrat/> (дата обращения: 09.08.2024).
5. Уминская А. Языки программирования для создания сайтов // Хекслет. – 2024. – [Электронный ресурс]. – Режим доступа: <https://ru.hexlet.io/blog/posts/yazyki-programirovaniya-dlya-sozdaniya-saytov> (дата обращения: 09.08.2024).

6. Шмидт Р. 10 главных тенденций веб-разработки на 2024 год // AppMaster. – 2023. – [Электронный ресурс]. – Режим доступа: <https://appmaster.io/ru/blog/top-10-tendentsii-na-radare-veb-gazrabotok-2023-goda> (дата обращения: 09.08.2024).

7. Soegaard M. *Adaptive vs. Responsive Design* (2020). Interaction Design Foundation. – [Электронный ресурс]. – Режим доступа: <https://www.interaction-design.org/literature/article/adaptive-vs-responsive-design> (accessed 09.08.2024).

8. Schelling K.J. *The Art of Responsive Web Design: Techniques and Best Practices for Modern Websites* (2023). Medium. – [Электронный ресурс]. – Режим доступа: <https://medium.com/@kjschelling/the-art-of-responsive-web-design-techniques-and-best-practices-for-modern-websites-fa6eeaa4070f> (accessed 09.08.2024).

9. Lara C. *24 Top Web Design Trends 2024* (2024) TheeDigital. – [Электронный ресурс]. – Режим доступа: <https://www.theedigital.com/blog/web-design-trends> (accessed 09.08.2024).

10. Malashuk Y. Atomic Design в веб-дизайне // Хабр. – 2023. – [Электронный ресурс]. – Режим доступа: <https://habr.com/ru/articles/740416/> (дата обращения: 09.08.2024).

MODERN METHODS OF CREATING AND DESIGNING WEBSITES

D.V. Malakhov, CEO, expert in creating websites based on WordPress
WEB STUDIO DM LLC
(Russia, Lyubertsy)

Abstract. *The article analyzes modern methods of creating and designing websites. The web development technologies used to create the client and server sides of the site are considered. Advanced technologies and web development tools are provided that allow you to create high-performance resources. It is noted that in terms of website design, the key trend is the creation of adaptive platforms. It provides a number of elements and functions implemented by web designers to improve the user experience.*

Keywords: *website, development, programming language, web design, adaptability.*