

## ОБ АВТОРЕ

---

*Баланов Антон Николаевич* имеет большой опыт руководства и консультирования в сфере ИТ-технологий. Работал топ-менеджером в крупных компаниях — таких, как Industrial and Commercial Bank of China (КНР), Caravan portal (ОАЭ), Банк ВТБ, Сбербанк России, VK; руководил разработками сервиса Gosuslugi.ru. Имеет степень MBA IT (CIA) и сертификации Microsoft, CompTIA, ISACA, PMI, SHRM, ПВА, HRCI, ISO, Six Sigma (Master Black Belt). Преподавал в следующих вузах и учебных центрах: Российском университете дружбы народов, СберУниверситете, Институте бизнеса и делового администрирования и Центре подготовки руководителей и команд цифровой трансформации (на базе Высшей школы государственного управления РАНХиГС). Автор десятков книг и научно-практических публикаций в профессиональных изданиях. Является советником Российской академии естественных наук.

Широкая эрудиция и глубокие профессиональные компетенции автора в сфере ИТ-технологий позволили ему создать книжную серию «Айтишный университет», один из выпусков которой находится перед вами.



# ОГЛАВЛЕНИЕ

---

<b>Глава 1. Введение в веб-разработку . . . . .</b>	10
Введение . . . . .	10
Основные принципы и концепции веб-разработки . . . . .	11
Различия между статическими и динамическими веб-сайтами . . . . .	14
Роль веб-технологий и их влияние на современный Интернет . . . . .	17
Заключение . . . . .	20
<b>Глава 2. Инструменты и технологии веб-разработки . . . . .</b>	22
Введение . . . . .	22
Обзор популярных языков программирования для веб-разработки (например, HTML, CSS, JavaScript) . . . . .	23
Введение в фреймворки и библиотеки для разработки веб-приложений . . . . .	26
Использование интегрированных сред разработки (IDE) и других инструментов для повышения эффективности работы . . . . .	28
Заключение . . . . .	31
<b>Глава 3. Управление жизненным циклом веб-приложений . . . . .</b>	33
Введение . . . . .	33
Основные этапы жизненного цикла веб-приложений . . . . .	34
Планирование и анализ требований . . . . .	38
Разработка, тестирование и развертывание веб-приложений . . . . .	40
Поддержка и обновление веб-приложений . . . . .	44

## **8 Оглавление**

---

Заключение .....	45
<b>Глава 4 Обеспечение безопасности в веб-разработке .....</b>	<b>47</b>
Введение .....	47
Определение основных угроз безопасности веб-приложений .....	48
Использование методов аутентификации и авторизации для защиты данных .....	51
Применение механизмов шифрования и защиты от атак .....	53
Обзор инструментов для обнаружения и предотвращения уязвимостей веб-приложений .....	55
Заключение .....	58
<b>Глава 5. Оптимизация веб-разработки .....</b>	<b>60</b>
Введение .....	60
Основные принципы оптимизации производительности веб-приложений .....	61
Оптимизация загрузки и отображения веб-страниц .....	63
Улучшение производительности серверной и клиентской частей приложения .....	66
Методы кэширования и сжатия данных для улучшения работы веб-приложений .....	68
Заключение .....	72
<b>Глава 6. Разработка мобильных версий веб-приложений .....</b>	<b>73</b>
Введение .....	73
Введение в разработку мобильных версий веб-приложений .....	74
Адаптивный и отзывчивый дизайн для мобильных устройств .....	76
Использование мобильных фреймворков и инструментов для создания мобильных веб-приложений .....	79
Заключение .....	82
<b>Глава 7. SEO и маркетинг веб-приложений .....</b>	<b>84</b>
Введение .....	84
Оптимизация веб-приложений для поисковых систем (SEO) .....	85

Введение в маркетинговые стратегии для продвижения веб-приложений .....	87
Использование аналитики и метрик для оценки эффективности и улучшения видимости приложения .....	90
Заключение .....	93
<b>Глава 8. Работа с базами данных в веб-разработке .....</b>	<b>95</b>
Введение .....	95
Введение в базы данных и их роль в веб-приложениях .....	96
Использование языка запросов SQL для управления данными .....	98
Разработка и оптимизация моделей баз данных для веб-приложений .....	101
Заключение .....	103
<b>Глава 9. Развертывание и управление веб-приложениями в облаке .....</b>	<b>105</b>
Введение .....	105
Преимущества и методы развертывания веб-приложений в облачной среде .....	106
Управление масштабированием и балансировкой нагрузки в облаке .....	108
Обзор платформы и инструментов для управления веб-приложениями в облаке .....	112
Заключение .....	115
<b>Глава 10. Работа с API в веб-разработке .....</b>	<b>117</b>
Введение .....	117
Введение в использование API в веб-приложениях .....	118
Интеграция сторонних сервисов и данных через API .....	120
Создание и документирование собственного API для веб-приложений .....	121
Заключение .....	124

# ГЛАВА 1

## ВВЕДЕНИЕ В ВЕБ-РАЗРАБОТКУ

---

### ВВЕДЕНИЕ

Веб-разработка является одной из наиболее динамично развивающихся областей в информационных технологиях. Сегодня практически каждая компания или организация нуждается в присутствии в Интернете, и веб-разработка играет важную роль в создании и поддержке веб-сайтов и приложений. Глава 1 вводит нас в мир веб-разработки, рассматривая основные принципы, концепции и влияние веб-технологий на современный Интернет.

Основные принципы и концепции веб-разработки представляют нам фундаментальные знания, необходимые для понимания и создания веб-сайтов и приложений. В этой главе мы рассмотрим архитектуру веб-приложений, включая клиентскую и серверную стороны, а также основные языки программирования и инструменты, используемые в веб-разработке. Мы также изучим принципы пользовательского интерфейса и пользовательского опыта, которые играют важную роль в создании удобных и эффективных веб-сайтов.

Различия между статическими и динамическими веб-сайтами являются ключевыми для понимания разнообразия возможностей и функциональности, которые можно реализовать в веб-разработке. Статические веб-сайты состоят из предопределенных файлов, которые отображаются на всех страницах сайта без изменений. Динамические веб-сайты, напротив, могут генерировать контент на основе данных, пользовательского взаимодействия и других факторов. Мы изучим различные подходы к созданию статических и динамических веб-сайтов, а также их преимущества и недостатки.

Роль веб-технологий в современном Интернете невозможно переоценить. Веб-технологии предоставляют нам инстру-

менты и ресурсы для создания, развертывания и поддержки веб-сайтов и приложений. HTML, CSS и JavaScript являются основными языками, используемыми в веб-разработке, и они обеспечивают возможности для создания интерактивного контента, стилизации и программирования на клиентской стороне. Кроме того, серверные языки программирования, такие как PHP, Python и Ruby, позволяют нам создавать динамические веб-сайты и взаимодействовать с базами данных.

В этой главе мы введемся в увлекательный мир веб-разработки, исследуя основные принципы и концепции, различия между статическими и динамическими веб-сайтами, а также роль веб-технологий в современном Интернете. Это будет основой для более глубокого изучения веб-разработки в следующих главах, где мы рассмотрим более специфические аспекты и инструменты этой увлекательной области.

## ОСНОВНЫЕ ПРИНЦИПЫ И КОНЦЕПЦИИ ВЕБ-РАЗРАБОТКИ

Основные принципы и концепции веб-разработки играют важную роль в создании эффективных и качественных веб-сайтов.

### 1. Таблица основных принципов и концепций веб-разработки.

Таблица 1.1

Принцип или концепция	Описание
Responsive Design	Создание веб-сайтов, которые адаптируются к различным устройствам и экранам, обеспечивая оптимальный пользовательский опыт.
Usability	Ориентированность на удобство использования веб-сайта для пользователей, включая легкость навигации, понятность контента и удобство взаимодействия.

<i>Принцип или концепция</i>	<i>Описание</i>
Accessibility	Создание доступных веб-сайтов для пользователей с ограниченными возможностями, учитывая различные потребности и требования.
Performance	Оптимизация производительности веб-сайта, чтобы обеспечить быструю загрузку страниц, минимальное использование ресурсов и отзывчивость интерфейса.
Security	Обеспечение безопасности веб-сайта путем защиты от взлома, злоупотребления и утечки данных.
Scalability	Учет возможности масштабирования веб-сайта для обработки увеличенной нагрузки и расширения функциональности.
SEO	Оптимизация веб-сайта для поисковых систем, чтобы улучшить его видимость и рейтинг в поисковых результатах.
Code Efficiency	Создание эффективного и оптимизированного кода, который обеспечивает быструю загрузку и легкость поддержки веб-сайта.

## 2. Примеры основных принципов и концепций веб-разработки.

- *Responsive Design*. Веб-сайт электронного магазина разработан с использованием responsive design, чтобы пользователи могли комфортно просматривать его на различных устройствах, таких как компьютеры, смартфоны и планшеты. Дизайн сайта автоматически адаптируется к размеру экрана, обеспечивая оптимальный пользовательский опыт независимо от устройства.
- *Usability*. Веб-сайт новостного портала предоставляет простую и интуитивно понятную навигацию для пользователей. Основные разделы и категории новостей легко доступны, а функциональность поиска и фильтрации по-

зволяет пользователям быстро находить интересующую информацию.

- *Security*. Онлайн-банк использует шифрование данных, двухфакторную аутентификацию и защищенные протоколы связи для обеспечения безопасности пользовательской информации и финансовых транзакций. Также сайт регулярно проверяется на наличие уязвимостей и обновляется для защиты от новых угроз.

Основные принципы и концепции веб-разработки играют важную роль в создании современных и эффективных веб-сайтов. Responsive design обеспечивает адаптивность сайта к различным устройствам и экранам, что становится особенно важным в эпоху мобильных устройств. Удобство использования (usability) является ключевым фактором для привлечения и удержания пользователей. Понятная навигация, интуитивный интерфейс и удобство взаимодействия с сайтом — все это способствует положительному пользовательскому опыту.

*Accessibility* обеспечивает доступность веб-сайта для всех пользователей, включая людей с ограниченными возможностями. Учитывая различные потребности пользователей, веб-разработчики создают сайты с улучшенной читабельностью, управляемостью и доступностью с помощью инструментов, таких как подписи к изображениям, альтернативные тексты и возможность увеличения шрифта.

*Performance* является одним из ключевых аспектов веб-разработки. Оптимизация загрузки страниц, минимизация запросов к серверу и оптимизированный код способствуют быстрому и отзывчивому взаимодействию с сайтом. Безопасность (security) играет важную роль в современной веб-разработке, особенно с учетом угроз хакеров, киберпреступности и утечек данных. Защита пользовательской информации и обеспечение безопасности финансовых транзакций являются основными задачами веб-разработчиков.

*Scalability* относится к способности веб-сайта масштабироваться и справляться с увеличенной нагрузкой при росте поль-

зовательской базы и объема данных. Оптимизация архитектуры и выбор подходящих технологий помогают обеспечить масштабируемость веб-сайта.

SEO является важным аспектом веб-разработки, поскольку веб-сайты должны быть оптимизированы для поисковых систем. Хорошая оптимизация SEO помогает улучшить видимость веб-сайта и привлечь больше органического трафика.

*Code efficiency* относится к созданию чистого, эффективного и оптимизированного кода. Хорошо структурированный код облегчает поддержку, расширение и сопровождение веб-сайта, а также способствует его быстрой загрузке и отзывчивости.

В целом, эти принципы и концепции являются основой успешной веб-разработки. Комбинирование их в процессе создания веб-сайтов помогает обеспечить высокую функциональность, удобство использования, безопасность и производительность, что в свою очередь создает приятный пользовательский опыт и способствует достижению целей бизнеса.

## РАЗЛИЧИЯ МЕЖДУ СТАТИЧЕСКИМИ И ДИНАМИЧЕСКИМИ ВЕБ-САЙТАМИ

### Статический веб-сайт

Статический веб-сайт — это сайт, содержимое которого остается неизменным без вмешательства разработчика. Каждая страница статического сайта является отдельным файлом HTML, который уже был создан и сохранен. Основные характеристики статического веб-сайта включают.

1. *Статичное содержимое.* Статический веб-сайт содержит постоянное и неизменное содержимое. Это может быть текст, изображения и другие элементы, которые не подвержены изменениям без вмешательства в исходный код сайта.

2. *Простота и быстрота разработки.* Создание статического веб-сайта обычно проще и быстрее, поскольку нет необходимости в настройке серверных скриптов или баз данных.

3. *Отсутствие динамического функционала.* Статические веб-сайты не предоставляют возможность взаимодействия

пользователя с сайтом. Они не могут выполнять динамические операции, такие как поиск, фильтрация данных или авторизация пользователей.

4. *Ограниченнная масштабируемость.* При необходимости добавления нового контента или внесения изменений на статический веб-сайт требуется вмешательство разработчика для редактирования HTML-файлов и повторной загрузки на сервер.

### **Динамический веб-сайт**

*Динамический веб-сайт* — это сайт, содержимое которого может изменяться и обновляться динамически в соответствии с запросами пользователей или событиями. Он использует серверные скрипты и базы данных для создания и доставки содержимого на основе входных данных. Основные характеристики динамического веб-сайта включают.

1. *Динамическое содержимое.* Динамический веб-сайт может генерировать содержимое на лету в зависимости от запросов пользователей или других факторов. Это может быть результатом выполнения серверных скриптов или обращения к базам данных.

2. *Расширенный функционал.* Динамический веб-сайт может предоставлять разнообразный функционал, такой как поиск, регистрация пользователей, интерактивные формы, онлайн-торговля и другие возможности взаимодействия с пользователями.

3. *Гибкость и масштабируемость.* Динамические веб-сайты обладают большей гибкостью и масштабируемостью. Изменения в дизайне или содержимом могут быть внесены с использованием шаблонов и обновленных данных без необходимости изменения каждой страницы вручную.

4. *Потребность в серверных скриптах и базах данных.* Для работы динамического веб-сайта требуется наличие серверных скриптов, таких как PHP, Ruby, Python, и баз данных, таких как MySQL, PostgreSQL, для обработки и доставки динамического контента.

*Пример таблицы.*

Таблица 1.2

Характеристика	Статический веб-сайт	Динамический веб-сайт
Содержимое	Статичное	Динамическое
Функциональность	Ограниченнaя	Расширенная
Масштабируемость	Ограниченнaя	Высокая
Разработка	Простая и быстрая	Более сложная
Взаимодействие	Отсутствует	Возможно

Статические и динамические веб-сайты имеют существенные различия в терминах функциональности, возможностей и способов разработки. Статический веб-сайт подходит для простых сайтов с постоянным содержимым, требующих быстрой и простой разработки. Однако, он ограничен в функциональности и масштабируемости.

Динамический веб-сайт, напротив, обеспечивает более широкий спектр возможностей и функциональности благодаря использованию серверных скриптов и баз данных. Он позволяет создавать динамическое содержимое, взаимодействовать с пользователями и обеспечивает более гибкую и масштабируемую разработку.

Выбор между статическим и динамическим веб-сайтом зависит от требований проекта, его целей и функциональности. Если необходим простой сайт с постоянным содержимым, статический веб-сайт может быть оптимальным выбором. Однако, если требуется более сложный функционал, взаимодействие с пользователями и возможность обновления содержимого динамически, то динамический веб-сайт будет более подходящим решением.

## РОЛЬ ВЕБ-ТЕХНОЛОГИЙ И ИХ ВЛИЯНИЕ НА СОВРЕМЕННЫЙ ИНТЕРНЕТ

Веб-технологии играют важную роль в современном Интернете и оказывают значительное влияние на различные аспекты онлайн-среды, начиная от разработки веб-страниц и веб-приложений до создания динамичного пользовательского опыта и повышения безопасности в Интернете. В этом материале мы рассмотрим несколько ключевых веб-технологий, их роль и влияние на современный Интернет.

### I. HTML (Hypertext Markup Language)

HTML является основным языком разметки веб-страниц и является фундаментальным элементом веб-технологий. С помощью HTML разработчики определяют структуру и содержимое веб-страницы. Он предоставляет основные элементы, такие как заголовки, параграфы, ссылки, изображения и таблицы, позволяя создавать информационные и интерактивные веб-страницы.

*Пример HTML-кода.*

Рисунок 1.1

```
<!DOCTYPE html>
<html>
<head>
    <title>Пример веб-страницы</title>
</head>
<body>
    <h1>Добро пожаловать на мою веб-страницу!</h1>
    <p>Это пример текста на веб-странице.</p>
    
    <a href="https://www.example.com">Пример ссылки</a>
</body>
</html>
```

## II. CSS (Cascading Style Sheets)

CSS является языком таблиц стилей и используется для описания внешнего вида веб-страниц. Он позволяет разработчикам определять цвета, шрифты, расположение элементов, анимации и другие аспекты дизайна веб-страницы. С помощью CSS можно создавать современные и привлекательные пользовательские интерфейсы.

*Пример CSS-кода.*

Рисунок 1.2

```
body {  
    font-family: Arial, sans-serif;  
    background-color: #f2f2f2;  
}  
  
h1 {  
    color: #333333;  
}  
  
p {  
    font-size: 16px;  
}  
  
a {  
    color: #0000ff;  
    text-decoration: none;  
}
```

## III. JavaScript

JavaScript является языком программирования, который добавляет динамическое поведение к веб-страницам. Он позволяет создавать интерактивные элементы, обрабатывать события, выполнять асинхронные запросы к серверу и многое другое. JavaScript является одной из основных технологий, используемых для разработки веб-приложений и создания богатого пользовательского опыта.

Пример JavaScript-кода.

Рисунок 1.3

```
function greet(name) {  
    console.log("Привет, " + name + "!");  
}  
  
greet("Мария"); // Вывод: Привет, Мария!
```

#### IV. AJAX (Asynchronous JavaScript and XML)

AJAX представляет собой подход к разработке веб-приложений, который позволяет обновлять содержимое веб-страницы асинхронно без необходимости полной перезагрузки страницы. AJAX использует JavaScript для выполнения асинхронных запросов к серверу и обновления содержимого страницы на основе полученных данных. Это позволяет создавать более отзывчивые и удобные в использовании веб-приложения.

#### V. Библиотеки и фреймворки

Существует множество библиотек и фреймворков, которые значительно упрощают разработку веб-приложений. Некоторые из наиболее популярных включают React, Angular и Vue.js. Эти инструменты предоставляют разработчикам готовые компоненты, функциональность и структуру, что позволяет более эффективно разрабатывать сложные веб-приложения.

Пример React-кода (рис. 1.4).

Веб-технологии, такие как HTML, CSS, JavaScript, AJAX и библиотеки/фреймворки, играют важную роль в современном Интернете. Они позволяют разработчикам создавать привлекательные и функциональные веб-страницы, разрабатывать интерактивные веб-приложения и обеспечивать более отзывчивый пользовательский опыт. Благодаря постоянному раз-

Рисунок 1.4

```
import React from 'react';

function App() {
  return (
    <div>
      <h1>Привет, мир!</h1>
      <p>Это пример React-компонента.</p>
    </div>
  );
}

export default App;
```

витию и улучшению веб-технологий, Интернет становится все более мощным и полезным инструментом для общения, работы и развлечений.

## ЗАКЛЮЧЕНИЕ

В заключение, глава 1 вводит нас в захватывающий мир веб-разработки, представляя основные принципы и концепции этой области. Веб-разработка — это процесс создания и поддержки веб-сайтов и приложений, которые играют ключевую роль в современном Интернете.

Одним из важных аспектов, рассмотренных в главе, является различие между статическими и динамическими веб-сайтами. Статический веб-сайт состоит из заранее созданных HTML-страниц, которые отправляются пользователю без изменений. В то же время, динамический веб-сайт использует

Конец ознакомительного фрагмента.

Приобрести книгу можно  
в интернет-магазине  
«Электронный универс»  
[e-Univers.ru](http://e-Univers.ru)