Муниципальное бюджетное общеобразовательное учреждение «Средняя общеобразовательная школа №8 имени Героя Советского Союза, летчика-космонавта СССР В.М. Афанасьева г.Брянска

ИНДИВИДУАЛЬНЫЙ ПРОЕКТ

Тема: "Создание Web-сайта"

Автор проекта:

Учащийся 10 класса

Удалый Павел Витальевич

Ф.И.О.

Руководитель проекта:

учитель

Киселева Людмила Васильевна

Ф.И.О

.

г.Брянск 2025г.

[Введение 3](#_Toc197368806)

[ГЛАВА I.ТЕОРЕТИЧЕСКАЯ ЧАСТЬ 5](#_Toc197368807)

[1.1. Понятие сайта.Первый сайт и появление WWW 5](#_Toc197368808)

[1.2 Способы создания веб-сайтов 6](#_Toc197368809)

[1.3 Этапы создания веб-сайтов 7](#_Toc197368810)

[1.4. Основы построения веб-страниц и сайтов 8](#_Toc197368811)

[1.5. Основы веб-разработки: HTML и CSS 9](#_Toc197368812)

[ГЛАВА II. ПРАКТИЧЕСКАЯ ЧАСТЬ. 11](#_Toc197368813)

[1.6. Изучение способов создания сайтов. 11](#_Toc197368814)

[1.7. Содержание сайта. 12](#_Toc197368815)

[1.8. Описание функционала веб-страницы 12](#_Toc197368816)

[1.9. Публикация сайта в интернет 13](#_Toc197368817)

[2.0. Заключение: 13](#_Toc197368818)

[Список литературы 14](#_Toc197368819)

# Введение

В современном мире сайты стали такой же необходимостью, как визитная карточка или магазинная витрина. Они открывают безграничные возможности для самовыражения, бизнеса и обмена информацией. Собственный сайт - это ваше персональное пространство в интернете, где вы можете представить себя, своё творчество или бизнес именно так, как захотите, без ограничений соцсетей. Для компаний сайт работает круглосуточно, привлекая клиентов и продавая товары даже ночью. Для специалистов — это мощное портфолио, которое говорит за вас. В обучении создание сайтов развивает логику, дизайн-мышление и технические навыки, востребованные в цифровую эпоху. Проще говоря, сайт сегодня это не просто страница в интернете, а инструмент, который открывает двери в мир новых возможностей.

Цель работы:

1. Изучение фундаментальных технологий: HTML для структуры страницы, СЅЅ для визуального оформления и JavaScript для интерактивности.

2. Развитие дизайнерских навыков: создание удобного и эстетичного интерфейса с учетом принципов UX/UI.

3. Приобретение практического опыта: от идеи до реализации и публикации готового продукта.

4. Формирование портфолио: созданный сайт может стать первым проектом в вашем профессиональном портфолио.

Этот проект особенно важен, так как веб-разработка - одно из самых востребованных направлений в ІТ, а полученные навыки пригодятся в любой цифровой профессии.

Задачи проекта:

1. Анализ и проектирование:

Определение целевой аудитории сайта

Разработка структуры и навигации

Создание пользовательских сценариев

1. Визуальное оформление:

Подбор цветовой схемы (не более 3-4 основных цветов)

Выбор шрифтовых пар (заголовки и основной текст)

Создание логотипа и фирменного стиля

1. Техническая реализация:

Семантическая верстка с использованием HTML5 Стилизация c

применением современных возможностей СЅЅЗ

Реализация адаптивности через медиазапросы

Добавление интерактивных элементов на JavaScript

1. Тестирование и публикация:

Кросс-браузерное тестирование Проверка на разных устройствах Оптимизация скорости загрузки Размещение на хостинге

# ГЛАВА I.ТЕОРЕТИЧЕСКАЯ ЧАСТЬ

## 1.1. Понятие сайта. Первый сайт и появление WWW

Веб-сайт - это совокупность веб-страниц и связанного с ними контента, которые идентифицируются общим доменным именем и публикуются по крайней мере на одном веб-сервере.

В 1991 году произошло знаковое событие, которое перевернуло

представление о взаимодействии с интернетом. Тим Бернерс-Ли, ученый в ЦЕРНе, создал первый веб-сайт и представил язык разметки HTML, отмечая начало эры легкости и доступности создания веб-страниц.

Тим Бернерс-Ли разработал World Wide Web (WWW) с целью облегчения обмена информацией между учеными. В 1991 году он создал первый веб- сайт, который описывал проект World Wide Web и предоставлял информацию об основах работы этой новой системы.



Рис.1 Первый сайт World Wide Web и его создатель

HTML давал создавать структурированные документы, содержащие гипертекстовые ссылки для связи между страницами. Это был ключевой инструмент, позволяющий пользователям не только читать информацию, но и взаимодействовать с ней. Развитие HTML предоставило простой и эффективный способ создания веб- страниц, a гипертекстовые ссылки обеспечивали навигацию между различными документами. Это привело к тому, что создание и публикация собственных веб-страниц стало доступным для широкой аудитории, что в свою очередь способствовало экспоненциальному росту веб-пространства. Рождение первых веб-сайтов и языка HTML стало переломным моментом в истории интернета, открывая двери для миллионов людей, желающих делиться информацией и создавать контент. В следующих частях мы рассмотрим, как эти инновации привели к формированию современного веба и изменениям в нашем восприятии информации.

## 1.2 Способы создания веб-сайтов

1. Использование конструкторов сайтов

Этот метод считается одним из самых бюджетных. Многие конструкторы предлагают бесплатные тарифы, а также встроенные услуги хостинга, онлайн-платежей и других полезных функций.

Плюсы:

* Простота использования
* Минимальные усилия при настройке
* Благодаря множеству готовых шаблонов и встроенному хостингу можно быстро запустить сайт - даже без специальных знаний. За дополнительную плату доступны расширенные возможности, которые помогут масштабировать проект.

Минусы:

В бесплатных версиях часто ограничены настройки, а доступ к исходному коду отсутствует (что усложняет перенос сайта на другой хостинг).

2. Готовые системы управления (СМЅ)

СМЅ – это универсальный инструмент, который позволяет быстро и недорого запустить сайт с базовым функционалом. По сути, это программная платформа, включающая движок и стандартные инструменты для управления

контентом.

Плюсы:

* Удобный интерфейс (понятный даже новичкам)
* Гибкость и широкие возможности настройки – с помощью CMS можно создать сайт практически любой структуры
* Оптимизация под поисковые системы — такие ресурсы обычно уже адаптированы для ЅЕО

Минусы:

* Платные версии могут стоить дорого, а бесплатные иногда имеют уязвимости в коде.

3. Ручная разработка на чистом коде

Создание сайта с нуля с использованием языков программирования — самый гибкий, но и самый сложный вариант.

Плюсы:

* Чистый и качественный код без лишних элементов или ошибок
* Высокий уровень безопасности — ручная разработка исключает риски, связанные с шаблонными решениями
* Полная свобода в реализации функций и интеграции сторонних сервисов

Минусы:

* Такой подход требует значительных временных и финансовых затрат. Для сложных проектов понадобится команда опытных разработчиков и большой бюджет.

## 1.3 Этапы создания веб-сайтов

1. Анализ и постановка целей

Перед началом разработки важно четко определить, для кого и зачем создается сайт. Без правильно сформулированных целей есть риск получить результат, который не будет соответствовать ожиданиям. На этом этапе проводят маркетинговые исследования, анализируют целевую аудиторию и конкурентов.

2. Составление технического задания (ТЗ)

Качественное ТЗ

фундамент успешного проекта. Упущенные детали (например, отсутствие важного модуля) могут привести к дополнительным затратам или отказу исполнителя от доработок. В ТЗ включают:

* Целевую аудиторию и задачи сайта
* Структуру (разделы, подразделы, примерное количество страниц)
* Функционал (формы обратной связи, фильтры, интерактивные элементы)
* Дизайн-требования (цвета, шрифты, расположение блоков, примеры референсов)
* Технические аспекты (CMS, мобильная адаптация, используемые технологии)
* Контент-стратегию (источники и процесс наполнения текстами/ изображениями)
* Прочие условия (хостинг, безопасность, интеграции)

3. Разработка дизайн-концепции

* Дизайнер создает визуальный стиль, учитывая бренд-айдентику компании: Цветовая палитра
* Типографика
* Анимации и интерактивные элементы
* Узнаваемость (логотип, фирменные элементы)
* Готовый макет согласовывается с заказчиком перед передачей в работу.

4. Верстка

Утвержденный дизайн преобразуют в HTML-код, обеспечивая корректное отображение на всех устройствах (ПК, смартфоны, планшеты) и браузерах. Это этап создания "статичной" версии сайта.

5. Программирование

Разработчики "оживляют" сайт, добавляя функционал: Динамические элементы (формы, pop-up окна)

* Интеграции с внешними сервисами
* Административную панель (для СМЅ)
* Иногда верстка и программирование идут параллельно это зависит

от сложности проекта.

6. Наполнение контентом

На сайт загружают тексты, изображения, видео и другие материалы. Важно учитывать:

SEO-оптимизацию (для продвижения в поисковиках)

Регулярное обновление (иначе сайт теряет позиции в выдаче)

7. Тестирование

Проверяют:

* Работу всех модулей
* Устойчивость к нагрузкам
* Безопасность (уязвимости, вирусы)
* Кроссбраузерность и адаптивность

После исправления ошибок сайт переносят на хостинг.

8. Поддержка и развитие

После запуска необходимы:

* Мониторинг работы
* Обновление контента • Исправление багов
* Добавление нового функционала

## 1.4. Основы построения веб-страниц и сайтов

Веб-страница по своей структуре представляет собой обычный текстовый файл с расширением HTML. Так как данный формат использует стандартную кодировку ASCII, для создания таких документов подойдет даже простейший текстовый редактор.

При обращении к сайту по его адресу (URL) браузер автоматически ищет главную страницу, которая традиционно должна называться либо index.html, либо welcome.html. Все остальные страницы ресурса могут иметь любые имена, однако при их создании следует соблюдать важные правила:

* использовать только буквы латинского алфавита
* избегать специальных символов, особенно решетки (#)

Эти требования обусловлены особенностями работы веб-серверов. Основу любой HTML-страницы составляют специальные теги. Главный контейнер <html> обрамляет весь документ, сообщая браузеру, что перед ним веб-страница, которую нужно обработать особым образом. Внутри этого тега обычно размещаются только скрипты, если они предусмотрены в проекте.

Стандартная структура веб-страницы включает два обязательных раздела:

1. Секция <head> - содержит служебную информацию
2. Секция <body> - включает основной контент

Минимальная рабочая структура документа выглядит следующим образом:

<html> <head> <title>Заголовок страницы</title> </head> <body> Основное содержимое </body> </html>

Особое внимание стоит уделить тегу <title>, который определяет название страницы, отображаемое во вкладке браузера. Этот элемент обязательно должен находиться внутри секции <head>.

## 1.5. Основы веб-разработки: HTML и CSS

HTML (HyperText Markup Language) - это фундаментальный язык разметки, который лежит в основе всех веб-страниц. С его помощью создается структура и содержимое сайта от текстовых блоков до интерактивных элементов (кнопок, форм, медиа-контента). Основной строительный блок HTML - это теги, которые определяют тип и назначение каждого элемента на странице.

СЅЅ (Cascading Style Sheets) - это мощный инструмент для оформления веб-страниц. В отличие от HTML, CSS не является самостоятельным

языком программирования, а работает исключительно в связке с разметкой. Его основная задача преобразование "сырой" HTML- структуры в эстетически привлекательный и удобный для пользователя интерфейс.

Ключевые особенности:

* HTML отвечает за содержание и структуру страницы
* СЅЅ контролирует внешний вид и визуальное представление
* Вместе они образуют идеальный дуэт для создания современных веб-сайтов

Дополнительные возможности:

1. HTML5 (последняя версия) поддерживает:

* о Мультимедийные элементы (аудио, видео)
* Семантическую разметку (header, footer, article)
* Встроенную графику (canvas, SVG)

1. СЅЅ3 предлагает:

* Сложные анимации и переходы
* Адаптивный дизайн (медиа-запросы)
* Продвинутые эффекты (тени, градиенты)

Практическое применение:

* HTML определяет где расположен элемент
* СЅЅ задает как он выглядит (цвет, размер, положение)
* Совместно они обеспечивают кросс-браузерную совместимость

Важность правильного использования:

Грамотное сочетание HTML и CSS позволяет создавать:

* Быстрые и легкие страницы
* Удобные для пользователя интерфейсы
* Сайты, корректно отображающиеся на любых устройствах

# ГЛАВА II. ПРАКТИЧЕСКАЯ ЧАСТЬ.

## 1.6. Изучение способов создания сайтов.

Также мне помогли полезные заметки, касающиеся навигации по сайту и вставки в него графических объектов и небольших Java-скриптов, из главы «Мир творчества: курс начинающего Web-мастера» книги В.Леонтьева «Новейшая энциклопедия Интернет». Из книги В.Леонтьева «Новейшая энциклопедия компьютера» я узнал об основных методах создания Web-страниц и о специальных программах, помогающих объединить их в целый сайт, администрировать и изменять их содержимое.

В книге С.В.Симоновича «Интернет у вас дома» (9) представлены общие понятия и размышления о роли Интернета в нашей жизни и возможности его практического

использования.

Общие представления о Web-дизайне я почерпнул из книги «Кухня Web-мастера Сидорова» А.А.Дуванова, в которой автор старается оградить читателя от распространенных ошибок начинающих «творцов».

Автор «Самоучителя HTML», А.Гончаров, научил меня правильно использовать особенности синтаксиса HTML и работать с формами на Web-сайтах.

План работы над созданием сайта:

Первым шагом в разработке сайта стало изучение различных подходов к созданию веб-страниц. Я ознакомился с книгой В.Леонтьева «Новейшая энциклопедия компьютера», и узнал о новых методах создания Web-страниц

Поскольку я выбрал в качестве инструмента язык разметки HTML, не требующий сложного программного обеспечения (достаточно обычного текстового редактора), мне потребовалось глубоко изучить его основы и возможности.

Начал я с изучения учебных материалов по веб-разработке. Одним из первых источников стала книга по информатике, где подробно объяснялись принципы построения HTML-страниц.

Также я ознакомился с обзорами специализированных программ, которые помогают объединять отдельные страницы в единый сайт, управлять контентом и вносить изменения.

Помимо технических аспектов, я изучил материалы о роли интернета в современном мире и его практическом применении. Это помогло мне лучше понять, каким должен быть полезный и удобный сайт.

Чтобы избежать типичных ошибок новичков, я прочитал несколько советов по веб-дизайну. Особенно полезными оказались рекомендации по правильному оформлению страниц и работе с HTML-формами.

Таким образом, перед тем как приступить к непосредственному созданию сайта, я провел тщательную подготовку, изучив как техническую сторону вопроса, так и принципы удобного дизайна.

## 1.7. Содержание сайта.

Я не стал подбирать информацию, так как основная задача, просто посмотреть функционал HTML, CSS и JavaScript, поэтому я просто разместил основные блоки на лицевой странице, которые чаще всего используются для создания сайтов: текст и картинки. Для обработки визуальной части макета страницы я использовал программу Figma.

## **1.8. Описание функционала веб-страницы**

На данной странице представлен базовый каркас, демонстрирующий ключевые элементы Фронтенд-разработки: HTML, CSS и JavaScript. Основная цель — не глубокий анализ или сложная реализация, а беглый обзор стандартных блоков, которые чаще всего используются при создании современных веб-сайтов.

1. HTML: Структура и контент

HTML (HyperText Markup Language) формирует основу страницы. Здесь использованы типовые элементы:

* Текстовые блоки (заголовки, параграфы, списки) - для организации информации.
* Изображения - для визуального дополнения контента.
* Контейнеры (<div>, <section> ) — для группировки и дальнейшего стилевого оформления.

2. СЅЅ: Стили и визуальное оформление

СЅЅ (Cascading Style Sheets) отвечает за внешний вид:

* Простая сетка (flex/grid) для расположения элементов.
* Базовые стили (шрифты, цвета, отступы) — чтобы страница выглядела аккуратно.
* Адаптивность (медиа-запросы) ) — если добавлена, обеспечивает корректное отображение на разных устройствах.

3. JavaScript: Интерактивность

JЅ придает динамику:

* Обработчики событий (клики, скролл) — для реакции на действия пользователя.
* Манипуляции с DOM - изменение контента «на лету».
* Упрощенные анимации например, плавное появление элементов.

Рис 2. Внешний вид сайта

## 1.9. Публикация сайта в интернет

Чтобы выложить свой HTML-сайт в интернет, проще всего использовать GitHub Pages: создать репозиторий с названием username.github.io (где username - логин на GitHub), загрузить

туда файлы сайта (HTML, CSS, JS, PNG), и через пару минут он будет доступен по этому адресу. Если нужен красивый домен, можно подключить его в настройках GitHub Pages или использовать Netlify / Vercel — просто перетащить папку с сайтом в их интерфейс, и они дадут бесплатный URL вида sitename.netlify.app. Для более продвинутых вариантов подойдёт платный хостинг вроде Hostinger или Reg.ru, где можно залить файлы через FTP. Но я воспользовался именно GitHub-ом.

## 2.0. Заключение:

B результате изучения технологий веб-разработки был создан информационный сайт с учётом освоенных принципов веб-дизайна. Все этапы работы над проектом — от планирования до реализации 1 - выполнены в соответствии с первоначальной концепцией. Содержание ресурса структурировано и оформлено согласно разработанным требованиям к навигации и визуальному представлению информации.

# Список литературы

1. Определение сайта [Электронная публикация] // По материалам сайта MDN Web Docs (Mozilla Developer Network)
2. Появление WWW [Электронная публикация] // По материалам сайта <https://timthewebmaster.com/ru/articles/world-wide-web-history/>
3. Самоучитель HTML - Гончаров А // <https://obuchalka.org/2011042154584/samouchitel-html-goncharov-a.html?ysclid=mabg630kgd455186507>
4. Дуванов А.А. Кухня web-мастера Сидорова: Основы практического web-дизайна. // По материалам сайта [Кухня web-дизайнера Сидорова: Основы практического web-дизайна | Дуванов А.А. | download on Z-Library](https://z-library.sk/book/2969247/470cd1/%D0%9A%D1%83%D1%85%D0%BD%D1%8F-web%D0%B4%D0%B8%D0%B7%D0%B0%D0%B9%D0%BD%D0%B5%D1%80%D0%B0-%D0%A1%D0%B8%D0%B4%D0%BE%D1%80%D0%BE%D0%B2%D0%B0-%D0%9E%D1%81%D0%BD%D0%BE%D0%B2%D1%8B-%D0%BF%D1%80%D0%B0%D0%BA%D1%82%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%BE%D0%B3%D0%BE-web%D0%B4%D0%B8%D0%B7%D0%B0%D0%B9%D0%BD%D0%B0.html)
5. Леонтьев В.П. Новейшая энциклопедия Интернет. // По материалам сайта <http://www.kavserver.ru/library/computerandinternet.shtml>