Муниципальное бюджетное общеобразовательное учреждение

«Средняя общеобразовательная школа № 56 г. Владивостока»

**Проект**

**«Основы создания сайта на языке HTML».**



Выполнил

ученик 8 класса В

Ложкин Николай

Руководитель

Паранько К. Е

Учитель Информатики

Владивосток

2024 год

## ****Содержание****

## [**Что такое HTML?**](https://gendalf.ru/news/marketing/sozdanie-sayta-s-ispolzovaniem-html/#first)

## [**Основные теги HTML и их функциональность**](https://gendalf.ru/news/marketing/sozdanie-sayta-s-ispolzovaniem-html/#second)

## [**Выбор текстового редактора или интегрированной среды разработки (IDE)**](https://gendalf.ru/news/marketing/sozdanie-sayta-s-ispolzovaniem-html/#third)

## [**Создание базовой структуры HTML-документа**](https://gendalf.ru/news/marketing/sozdanie-sayta-s-ispolzovaniem-html/#forth)

## [**Сохранение и запуск первой веб-страницы**](https://gendalf.ru/news/marketing/sozdanie-sayta-s-ispolzovaniem-html/#fifth)

## [**Работа с текстом и изображениями**](https://gendalf.ru/news/marketing/sozdanie-sayta-s-ispolzovaniem-html/#sixth)

## [**Семантическая разметка**](https://gendalf.ru/news/marketing/sozdanie-sayta-s-ispolzovaniem-html/#seventh)

## [**Работа с мультимедиа**](https://gendalf.ru/news/marketing/sozdanie-sayta-s-ispolzovaniem-html/#eighth)

## [**Адаптивный дизайн и мобильная совместимость**](https://gendalf.ru/news/marketing/sozdanie-sayta-s-ispolzovaniem-html/#ninth)

## [**Оптимизация производительности**](https://gendalf.ru/news/marketing/sozdanie-sayta-s-ispolzovaniem-html/#tenth)

## **CSS**

12.Использование CSS

13.Правила построения CSS

## ****Введение****

## **В современном цифровом обществе веб-сайты являются визитной карточкой для компаний, предоставляют информацию и услуги пользователю, позволяют развивать онлайн-бизнес, а также служат местом для выражения творческих идей и обмена знаний. Это незаменимый инструмент для коммуникации и взаимодействия с миром.**

## **Цель - предоставить начинающим разработчикам основные знания и инструкции, необходимые для создания собственных веб-страниц с использованием языка разметки гипертекста (HTML). Мы пошагово рассмотрим основы HTML, структуру HTML-документа и ключевые теги, которые позволят вам создать простые, но функциональные веб-страницы.**

## **Вы получите необходимую базу, чтобы приступить к созданию своего первого веб-сайта.**

## ****Цель проекта****

## **Целью данного проекта является изучения** языка программирования HTML **и создание сайта.**

## **При изучении материала я хочу познакомится со структурой html-документа, познакомится с CSS, научится создавать и применять стил.**

## [****Что такое HTML?****](https://gendalf.ru/news/marketing/sozdanie-sayta-s-ispolzovaniem-html/#first)

## **HTML, или язык разметки гипертекста (HyperTextMarkupLanguage), является стандартным языком для создания веб-страниц. Он используется для структурирования контента на странице, определения элементов, их функций и взаимосвязей. HTML использует различные теги и атрибуты, которые сообщают браузеру, как отображать содержимое веб-страницы.**

## **Каждая веб-страница состоит из HTML-документа, который является основным строительным блоком сайта. Структура HTML-документа состоит из следующих элементов:**

## **<!DOCTYPE html>. Объявление типа документа, указывающее на версию HTML.**

## **<html>. Корневой элемент документа, обозначающий начало и конец HTML-кода.**

## **<head>. Этот элемент содержит метаинформацию о документе, такую как заголовок страницы, подключение CSS или JavaScript.**

## **<body>. Основное содержимое веб-страницы располагается в этом элементе, включая текст, изображения, ссылки и другие элементы.**

## ****Основные теги HTML и их функциональность****

## **HTML предоставляет множество тегов, каждый из которых выполняет свою функцию. Вот несколько основных тегов HTML и их функциональность.**

## **<h1>, <h2>, ..., <h6>. Заголовки разного уровня, используются для структурирования заголовков страницы.**

## **<p>. Параграф, используется для отображения текстового контента.**

## **<a>. Гиперссылка, создает ссылки на другие веб-страницы или ресурсы.**

## **<img>. Вставка изображений на веб-страницу.**

## **<ul>, <ol>. Ненумерованные и нумерованные списки соответственно.**

## **<table>. Создание таблиц для представления данных в упорядоченной форме.**

## **<form>. Используется для создания форм, которые позволяют пользователям отправлять данные на сервер.**

## **Это лишь небольшой набор тегов, и существует множество других тегов, которые позволяют реализовать различные функции и структуры на веб-страницах.**

## ****Выбор текстового редактора или интегрированной среды разработки (IDE)****

## **Прежде чем мы начнем создавать веб-страницы с помощью HTML, вам потребуется текстовый редактор или интегрированная среда разработки (IDE). Выбор инструмента для написания кода - важный шаг, который влияет на вашу производительность и удобство работы.**

## **Существует множество текстовых редакторов и IDE, и выбор зависит от ваших предпочтений. Некоторые популярные текстовые редакторы, которые отлично подходят для начинающих, включают.**

## **VisualStudioCode. Бесплатный и мощный текстовый редактор с широкими возможностями и поддержкой различных языков программирования.**

## **Atom. Еще один бесплатный и легкий в использовании редактор, разработанный GitHub.**

## **SublimeText. Легкий, быстрый и хорошо настраиваемый редактор.**

## **Notepad++. Бесплатный редактор для операционных систем Windows, поддерживающий различные языки.**

## **Выберите тот, который соответствует вашим потребностям и скачайте его с официального сайта.**

## ****4.Создание базовой структуры HTML-документа****

## **Когда у вас есть текстовый редактор или IDE, можно приступать к созданию базовой структуры HTML-документа. Вся веб-страница будет находиться внутри этой структуры. Вот простейший шаблон HTML.**

## **[Скриншот шаблона для создания сайта на HTML в момент разработки](https://gendalf.ru/upload/medialibrary/0e5/w4c3gjewl94md800qwing4bp274fl8m4/html.png)**

## **Обратите внимание на следующие элементы.**

## **<!DOCTYPE html>. Это объявление типа документа и указывает на версию HTML, которую мы собираемся использовать (в данном случае HTML5).**

## **<html>. Открывающий тег корневого элемента документа, указывающий начало HTML-кода.**

## **<head>. Этот элемент содержит метаинформацию о документе, такую как заголовок страницы, подключение CSS или JavaScript.**

## **<title>. Заголовок вкладки браузера, который отображается, когда вы открываете веб-страницу.**

## **<body>. Основное содержимое веб-страницы располагается в этом элементе. Ваш контент, такой как текст, изображения, ссылки и другие элементы, будет размещаться внутри этого тега.**

## ****5.Сохранение и запуск первой веб-страницы****

## **Теперь, когда у вас есть базовая структура HTML-документа, сохраним ее с расширением .html. Например, index.html - хорошее имя для вашего первого файла. Убедитесь, что файл сохранен в формате «Веб-страница, HTML» или аналогичном в вашем текстовом редакторе.**

## **Чтобы запустить вашу первую веб-страницу, просто дважды кликните на файл index.html, и браузер откроет его. Вы увидите пустую страницу с заголовком, который вы указали в элементе <title>. Поздравляю, вы только что создали свою первую веб-страницу с использованием HTML!**

## ****6.Работа с текстом и изображениями****

## **A. Форматирование текста с помощью HTML-тегов**

## **Теперь, когда вы освоили базовую структуру HTML-документа, пришло время заполнить вашу страницу контентом. В HTML есть много тегов для форматирования текста. Вот несколько примеров.**

## **[Схема основных инструментов создания успешного сайта на HTML](https://gendalf.ru/upload/medialibrary/d02/9i3q3x2l1h876idy8jvikkq0vhirpe2n/html.png)**

## **B. Вставка изображений на веб-страницу**

## **Изображения могут улучшить внешний вид вашей веб-страницы. Для вставки изображения используется тег <img> с атрибутом src, который указывает путь к файлу изображения. Вот пример.**

## **[Изображение версии сайта на HTML для разных типов устройств](https://gendalf.ru/upload/medialibrary/794/nnx56sfu6q17sczx97ibs6ekn1ticy4a/html.png)**

## **Создание гиперссылок**

## **Гиперссылки позволяют переходить с одной веб-страницы на другую. Для создания ссылки используется тег <a> с атрибутом href, который указывает целевой URL или путь к другой странице. Вот пример.**

## **[Пример веб-страницы, созданной на HTML с помощью пошаговой инструкции](https://gendalf.ru/upload/medialibrary/efd/050ec10mxw3d75a8a3a4zdf96om6hgmp/html.png)**

## **HTML предоставляет разнообразные элементы управления для форм. Некоторые из них.**

## **<input type="text">. Поле для ввода текста.**

## **<input type="password">. Поле для ввода пароля (текст отображается замаскированным).**

## **<input type="checkbox">. Флажок для множественного выбора.**

## **<input type="radio">. Радиокнопка для выбора одного варианта из нескольких.**

## **<input type="submit">. Кнопка для отправки данных формы.**

## **<textarea>. Многострочное текстовое поле.**

## **<select> и <option>. Выпадающий список с выбором одного варианта.**

## ****7.Семантическая разметка****

## **Семантическая разметка в HTML обозначает значение и функциональность элементов на веб-странице. Это не только помогает поисковым системам лучше понимать контент вашего сайта, но и повышает доступность для пользователей с ограниченными возможностями.**

## **HTML5 представил ряд семантических тегов, которые помогают более ясно определить структуру веб-страницы. Некоторые из них.**

## **<header>. Заголовочная часть страницы, обычно содержит логотип и навигацию.**

## **<nav>. Элемент для группировки навигационных ссылок.**

## **<main>. Основное содержимое страницы.**

## **<article>. Самостоятельный контент, который может быть переиспользован.**

## **<section>. Логическая секция контента.**

## **<aside>. Контент, связанный с основным содержимым страницы.**

## **<footer>. Нижняя часть страницы, содержит информацию об авторе, копирайт и другие.**

## **Применение семантических тегов делает код более читаемым и понятным для разработчиков и браузеров, а также улучшает опыт пользователей на вашем сайте.**

## **В этой части статьи мы рассмотрели создание таблиц, списков, форм и важность семантической разметки в HTML. Научившись использовать эти элементы, вы сможете создавать более удобные и информативные веб-страницы. В следующей части статьи мы погрузимся в работу с мультимедиа и стилизацией веб-страниц с помощью CSS.**

## ****8.Работа с мультимедиа****

## **Веб-страницы становятся более интересными и привлекательными благодаря мультимедийным элементам, таким как видео и аудио. Для вставки видео используется тег <video>, а для аудио - тег <audio>.**

## **Использование элемента canvas для рисования на странице**

## **Элемент <canvas> позволяет создавать рисунки, графику и анимацию непосредственно на веб-странице с помощью JavaScript.**

## **Стилизация с помощью CSS**

## **CSS (CascadingStyleSheets) - язык стилей, используемый для оформления и внешнего вида веб-страниц. Он позволяет изменять цвета, шрифты, размеры, расположение элементов и многое другое, что делает дизайн веб-страниц более привлекательным и современным.**

## **Стили CSS можно добавить в HTML-документ через встроенные стили <style>, внешние стили с помощью тега <link> или встроенные стили в элементы HTML с помощью атрибута style.**

## ****9.Адаптивный дизайн и мобильная совместимость****

## **Современные веб-сайты должны быть адаптивными и хорошо выглядеть на разных устройствах, таких как компьютеры, планшеты и смартфоны. Адаптивный дизайн позволяет автоматически изменять макет и размеры элементов для оптимального отображения на различных экранах.**

## **Медиа-запросы позволяют применять разные стили CSS для разных типов устройств. Например, вы можете использовать медиа-запросы для изменения размеров шрифтов или перестроения макета в зависимости от ширины экрана устройства.**

## ****10.Оптимизация производительности****

## **Оптимизация изображений - это важная часть создания быстрых и эффективных веб-страниц. Вы можете использовать сжатие изображений и выбор правильных форматов, таких как JPEG для фотографий и PNG для изображений с прозрачностью.**

## **Минимизируйте CSS и JavaScript файлы, чтобы уменьшить их размер. Также следите за порядком загрузки ресурсов на странице - сначала загружайте критически важные элементы.**

## **Веб-разработка - это постоянно развивающаяся область, и есть много возможностей для улучшения навыков и изучения новых технологий. Продолжайте практиковаться, изучать новые инструменты и техники, и вы станете более опытным веб-разработчиком.**

## ****11**.**CSS****

CSS (aнгл. *Cascading Style* Sheets «каскадные таблицы стилей») — формальный язык декорирования и описания внешнего вида документа (веб-страницы), написанного с использованием языка разметки (чаще всего HTML или XHTML). Также может применяться к любым XML-документам, например, к SVG или XUL.

**12.Использование CSS**

CSS используется создателями веб-страниц для задания цветов, шрифтов, стилей, расположения отдельных блоков и других аспектов представления внешнего вида этих веб-страниц. Основной целью разработки CSS является ограждение и отделение описания логической структуры веб-страницы (которое производится с помощью HTML или других языков разметки) от описания внешнего вида этой веб-страницы (которое теперь производится с помощью формального языка CSS). Такое разделение может увеличить доступность документа, предоставить большую гибкость и возможность управления его представлением, а также уменьшить сложность и повторяемость в структурном содержимом.

**13.Правила построения CSS**

В первых трёх случаях подключения стилей CSS к документу (см. выше) каждое правило CSS из файла имеет две основные части — селектор и блок объявлений. Селектор, расположенный в левой части правила до знака «{», определяет, на какие части документа (возможно, специально обозначенные) распространяется правило. Блок объявлений располагается в правой части правила. Он помещается в фигурные скобки, и, в свою очередь, состоит из одного или более объявлений, разделённых знаком «;». Каждое объявление представляет собой сочетание свойства CSS и значения, разделённых знаком «:». Селекторы могут группироваться в одной строке через запятую. В таком случае свойство применяется к каждому из них.

**селектор**, **селектор** {

свойство: значение;

свойство: значение;

свойство: значение;

}

В четвёртом случае подключения CSS к документу (см. список) правило CSS, являющееся значением атрибута style элемента, к которому он применяется, представляет собой перечень объявлений («свойство CSS : значение»), разделённых знаком «;».

## ****Заключение****

## **Я научился создавать сайты на языке программирования html. Что вероятно пригодится мне в дальнейшем…**