

Приднестровский государственный университет им. Т.Г. Шевченко
Физико-технический институт

**ОСНОВЫ ВЕБ-РАЗРАБОТКИ С ПРИМЕНЕНИЕМ
ИИ-АССИСТЕНТОВ**

Лабораторный практикум

Разработал:
ст. преподаватель
кафедры ИТ
Бричаг Д.В.

г. Тирасполь
2025 г.

Лабораторная работа №1

Введение в веб. Структура HTML-страницы и базовое оформление.

Цель работы:

- Понять, из чего состоит веб-страница.
- Разобраться, как работает браузер при отображении HTML и CSS.
- Освоить базовые теги HTML и стили CSS.
- Научиться использовать ИИ как помощника для генерации шаблона кода.
- Закрепить работу с git и GitHub.

Теоретическая справка

HTML (HyperText Markup Language) представляет язык разметки гипертекста, используемый преимущественно для создания документов в сети интернет. HTML начал свой путь в начале 90-х годов как примитивный язык для создания веб-страниц, и в настоящий момент уже трудно представить себе интернет без HTML. Подавляющее большинство сайтов так или иначе используют HTML.

В 2014 году официально была завершена работа над новым стандартом - HTML5, который фактически произвел революцию, привнес в HTML много нового:

- HTML5 определяет новый алгоритм парсинга для создания структуры DOM;
- добавление новых элементов и тегов, как например, элементы video, audio и ряд других;
- переопределение правил и семантики уже существовавших элементов HTML.

Фактически с добавлением новых функций HTML5 стал не просто новой версией языка разметки для создания веб-страниц, но и фактически платформой для создания приложений, а область его использования вышла далеко за пределы веб-среды интернет: HTML5 применяется также для создания мобильных приложений под Android, iOS, ~~Windows Mobile~~ и даже для создания десктопных приложений для обычных компьютеров (в частности, в ОС Windows 10/11).

В итоге, как правило, HTML5 применяется преимущественно в двух значениях:

- HTML5 как обновленный язык разметки гипертекста, некоторое развитие предыдущей версии HTML4
- HTML5 как мощная платформа для создания веб-приложений, которая включает не только непосредственно язык разметки гипертекста, обновленный HTML, но и язык программирования JavaScript и каскадные таблицы стилей CSS3.

Структура HTML-документа

В верхней части документа необходимо указать браузеру, какая версия HTML используется на странице. HTML является развивающимся языком и регулярно обновляется. Большинство современных браузеров поддерживают новейшую спецификацию HTML5. Однако старые веб-сайты могут использовать предыдущие версии языка.

Информация о версии HTML сообщается браузеру через тег `<!DOCTYPE ...>` в начале документа, где вместо ... устанавливается версия HTML. Для HTML5 используется запись `<!DOCTYPE html>`. Запись `!` и `DOCTYPE` в верхнем регистре важны для старых браузеров, тогда как запись `HTML` не чувствительна к регистру.

Остальная часть HTML-кода должна быть заключена в HTML-теги. Открывающий `<html>` находится непосредственно под строкой `<!DOCTYPE html>`, а закрывающий `</html>` - в конце страницы.

Элементы, находящиеся внутри тега `<html>`, образуют дерево документа, так называемую объектную модель документа, DOM (document object model). При этом элемент `<html>` является корневым элементом. Общий вид дерева HTML документа изображён на рисунке 1.

Чтобы разобраться во взаимодействии элементов веб-страницы, необходимо рассмотреть так называемые «родственные отношения» между элементами.

Отношения между множественными вложенными элементами подразделяются на родительские, дочерние и сестринские.

Предок — элемент, который заключает в себе другие элементы. На рисунке 1 предком для всех элементов является `<html>`. В то же время элемент `<body>` является предком для всех содержащихся в нем тегов: `<h1>`, `<p>`, ``, `<nav>` и т.д.

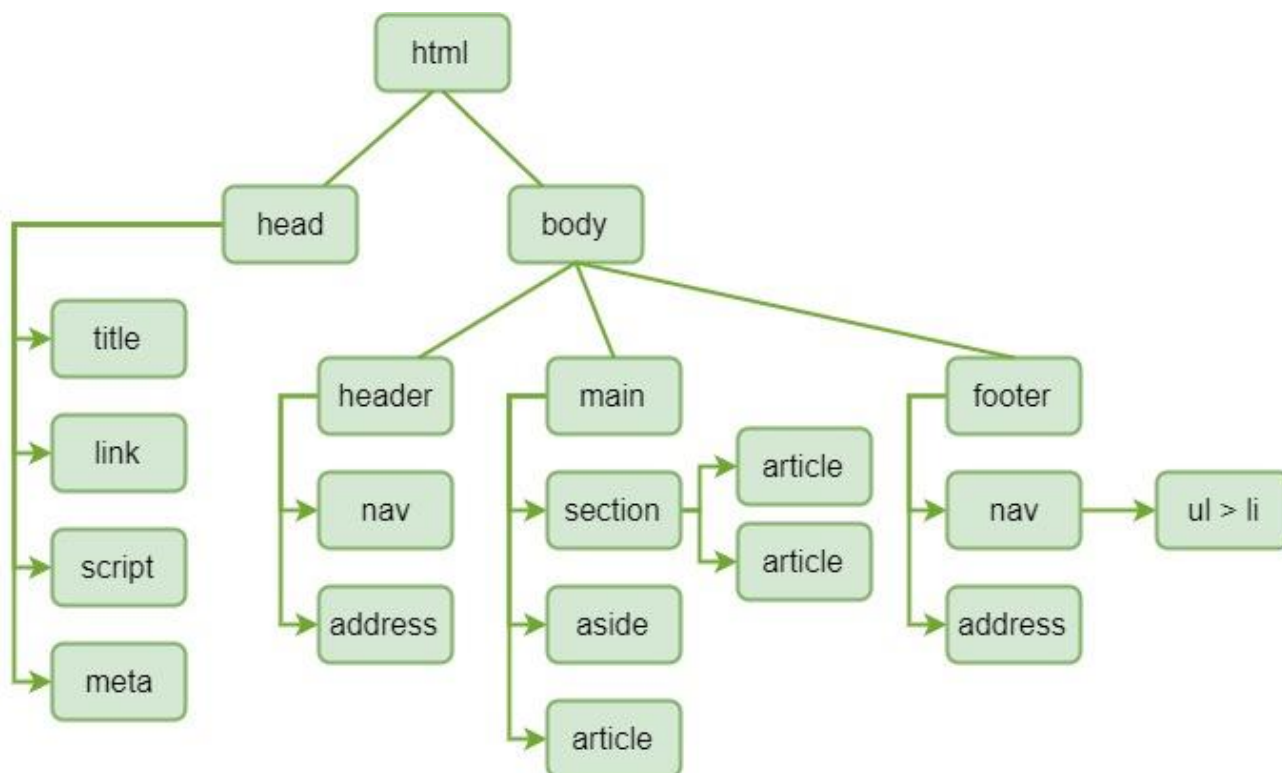


Рис. 1 – Простейшая структура веб-страницы

Потомок — элемент, расположенный внутри одного или более типов элементов. Например, `<body>` является потомком `<html>`, а элемент `<p>` является потомком одновременно для `<body>` и `<html>`.

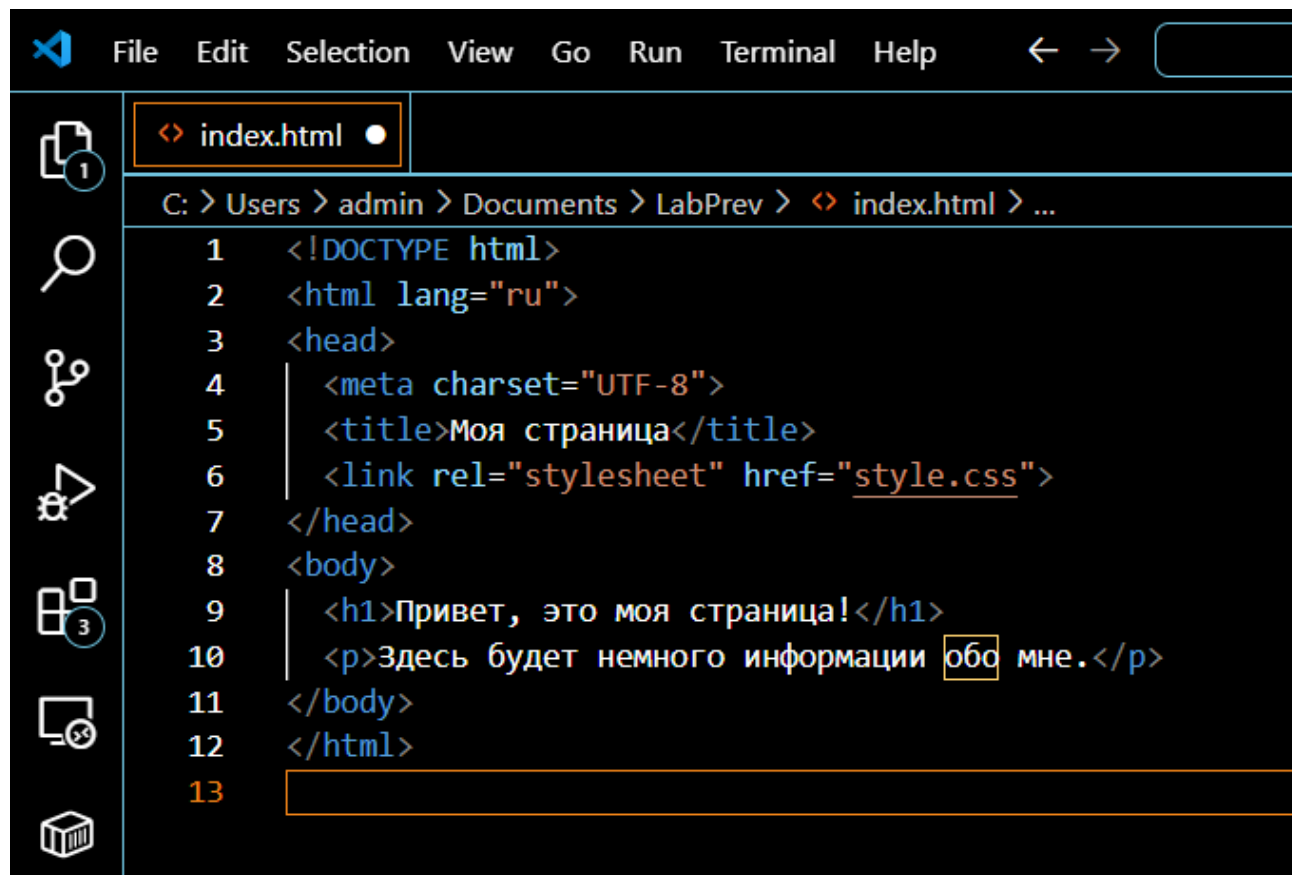
Родительский элемент — элемент, связанный с другими элементами более низкого уровня, и находящийся на дереве выше их. На рисунке 1 `<html>` является родительским только для `<head>` и `<body>`. Тег `<p>` является родительским только для ``.

Дочерний элемент — элемент, непосредственно подчиненный другому элементу более высокого уровня. На рисунке 1 только элементы `<h1>`, `<h2>`, `<p>` и `<nav>` являются дочерними по отношению к `<body>`.

Сестринский элемент — элемент, имеющий общий родительский элемент с рассматриваемым, так называемые элементы одного уровня. На рисунке 1 `<head>` и `<body>` — элементы одного уровня, так же, как и элементы `<h1>`, `<h2>` и `<p>` являются между собой сестринскими.

На рисунке 2 изображён скриншот с кодом общей структуры дерева документа из предыдущего раздела, набранного в текстовом редакторе.

Здесь представлено те же самые теги в такой же иерархической последовательности. Обратите внимание на открытые и закрытые теги, а также на то, каким образом дочерние элементы находятся внутри своих родителей.

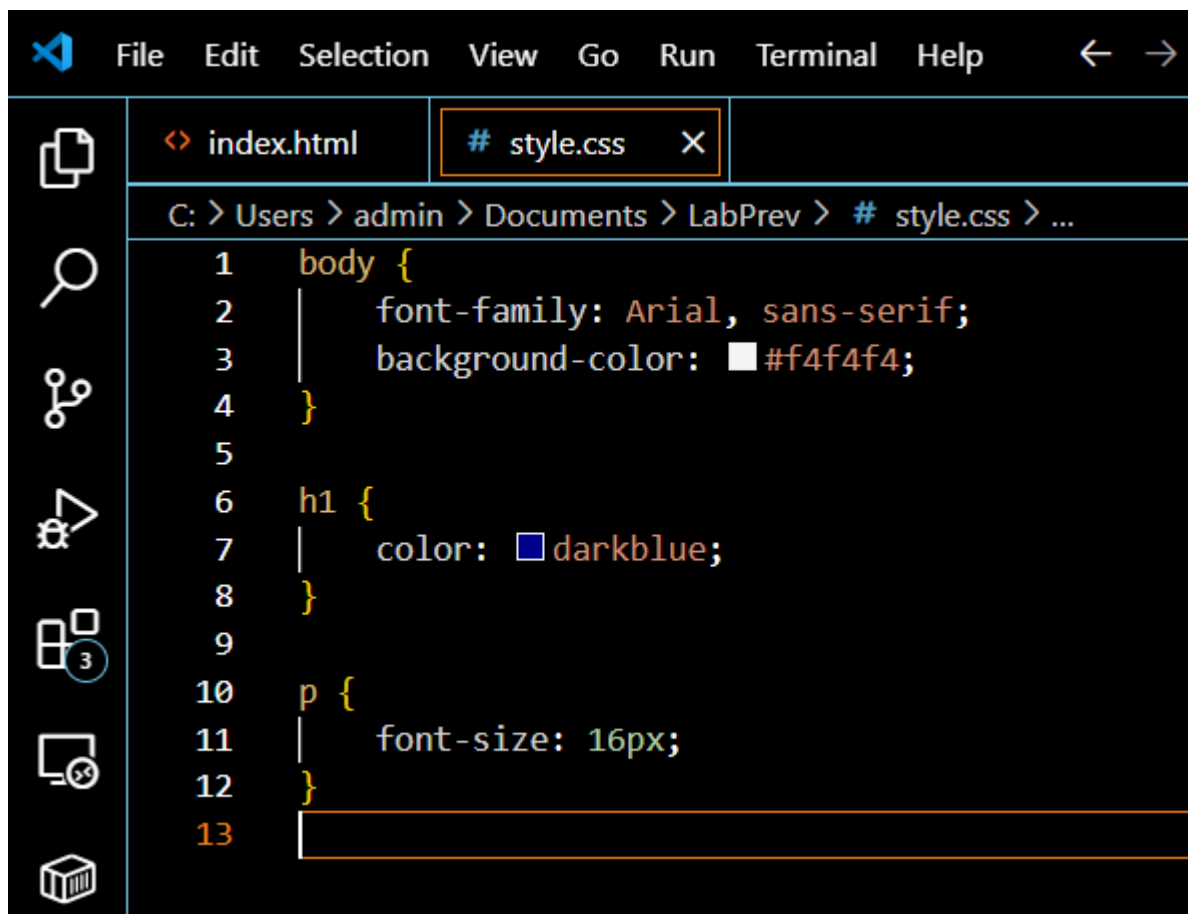


```
1 <!DOCTYPE html>
2 <html lang="ru">
3 <head>
4   <meta charset="UTF-8">
5   <title>Моя страница</title>
6   <link rel="stylesheet" href="style.css">
7 </head>
8 <body>
9   <h1>Привет, это моя страница!</h1>
10  <p>Здесь будет немного информации обо мне.</p>
11 </body>
12 </html>
13
```

Рис. 2 – Структура веб-страницы в редакторе VS Code

Однако не все теги требуется закрывать. В примере это теги `link` и `meta`. Кроме них существует ещё ряд тегов, о которых пойдёт речь по ходу курса.

На рисунке 3 – скриншот файла с CSS кодом, описывающим внешний вид тегов, указанных в файле `html`.



```
1  body {
2      font-family: Arial, sans-serif;
3      background-color: #f4f4f4;
4  }
5
6  h1 {
7      color: darkblue;
8  }
9
10 p {
11     font-size: 16px;
12 }
13
```

Практическая часть

Для работы с HTML5 необходим текстовый редактор, чтобы набирать текст веб-страниц на html. В принципе для работы с HTML разметкой достаточно базового блокнота в ОС Windows или Notepad++. Но, так как мы планируем развивать веб-страницу до полноценного сайта гораздо продуктивнее и удобнее будет работа в специализированном приложении. К таким относятся бесплатные Visual Studio Code и Atom или платные WebStorm и Sublime Text. Их преимуществом перед полновесными средами разработки является небольшой вес программы, скорость запуска и удобство редактирования одиночных файлов. В курсе лабораторных работ скриншоты из примеров будут приводиться в интерфейсе редактора VS Code.

Также потребуется веб-браузер для просмотра созданных веб-страниц. Рекомендуется использовать последнюю версию любого из распространенных браузеров - Google Chrome, Mozilla Firefox, Microsoft Edge, Opera/Vivaldi/Яндекс.Браузер и т.п. на основе движка Blink, используемого в открытом браузере Chromium.

Приступим к созданию будущего проекта. Подготовка рабочей среды:

Создайте папку **lab1**.

В ней создайте файлы:

- index.html
- style.css
- README.md (для отчёта)

Использование ИИ для генерации шаблона.

ИИ-инструменты позволяют значительно ускорить процесс создания HTML/CSS шаблонов для веб-страниц. Вы можете использовать их для быстрого прототипирования и получения базовой структуры проекта.

Доступные инструменты:

ChatGPT – OpenAI – <https://chatgpt.com/>

Grok – xAI – <https://grok.com/>

Gemini – Google – <https://gemini.google.com/>

Claude – Anthropic – <https://claude.ai/>

DeepSeek - <https://chat.deepseek.com/>

Примеры запросов (пром프트в) для ИИ:

Сгенерируй простую HTML-страницу-визитку со стилями в отдельном CSS-файле.
Страница должна содержать заголовок с моим именем, блок "Обо мне", список моих хобби и контакты.
Сделай аккуратный дизайн.

Напиши HTML+CSS для личной страницы. Добавь фото (плейсхолдер), имя, короткий текст обо мне и таблицу с моими любимыми книгами.

Создай HTML-страницу-визитку с тёмной темой. Используй flexbox для расположения элементов.

Сделай HTML-страницу, где есть:

- приветственный заголовок,
- параграф обо мне,
- список хобби,
- ссылки на соцсети.

Стили вынеси в отдельный CSS.

Сгенерируй HTML и CSS для портфолио-страницы: секция "Обо мне", блок "Проекты", контакты.
Сделай современный минималистичный дизайн.

Результат скопируйте в соответствующие файлы вашего проекта.

Редактирование вручную

- Замените текст на свой (имя, хобби, контакты).
- Измените цвета, фон и шрифты в style.css.
- Добавьте свой аватар/фото (можно из интернета или placeholder).

Просмотр в браузере

- Откройте index.html двойным кликом.
- Проверьте, что страница отображается корректно.
- При необходимости сгенерируйте новый запрос или попросите инструмент исправить ошибки.

Работа с git и GitHub

1. Инициализируйте репозиторий:

```
git init
git add .
git commit -m "Lab1: initial commit"
```

2. Создайте новый репозиторий на GitHub.

3. Свяжите локальный проект с GitHub:

```
git remote add origin https://github.com/<ваш_логин>/<имя_репозитория>.git
git branch -M main
git push -u origin main
```

Отчёт (в README.md)

В README.md добавить:

- Заголовок: *Лабораторная работа №1*
- Ответы на вопросы:
 - Что удалось сделать легко?

- Где помог ИИ?
- Где пришлось разбираться самому?

Результаты работы

В итоге у вас должно быть:

1. Страница-визитка в папке lab1.
2. HTML и CSS-файлы, изменённые вручную.
3. Репозиторий на GitHub с загруженными файлами.
4. Отчёт в README.md.

Критерии оценки

- Страница корректно отображается в браузере (2 балла).
- Использован шаблон, сгенерированный ИИ (2 балла).
- Внесены собственные изменения в текст и стили (2 балла).
- Код выложен на GitHub (2 балла).
- README.md содержит отчёт (2 балла).