Geekbrains

Специальность: Frontend-программист Цифровые профессии

Тема дипломного проекта: «Инструкция по HTML и СSS»

ФИО студента: Дьячков Антон Геннадьевич

г. Северодвинск

2023

Содержание дипломного проекта

Введение 4

1. HTML 5

1.1 Основные понятия HTML 5

1.2 Работа с атрибутами 8

1.3 Работа с текстом 9

1.4 Работа со ссылками 11

1.5 Работа с изображениями 14

1.6 Работа с таблицами 15

1.7 Работа со списками 18

1.8 Полезные ссылки 19

2. CSS 20

2.1 Основы CSS 20

2.2 Работа с текстом

2.3 Блочные и строчные элементы

2.4 Работа с полями

2.5 Позиционирование

2.6 Работа с display: flex

2.7 Работа с CSS Grid

3. Заключение

4. Список использованной литературы и ресурсов

5. Приложения

6. Презентация к дипломному проекту

Введение

Данный дипломный проект - небольшая практическая работа, в которой предполагается использовать все полученные сведения и закрепить теорию практикой верстки небольшого сайта. Защита проекта является обязательной итоговой аттестацией, после прохождения планируемых технологических специализаций (Frontend-программист. Цифровые профессии) для получения итогового документа об обучении после его окончания.

Тема проекта была выбрана автором самостоятельно с целью объединить знания о самых основных и минимально необходимых инструментах верстки сайтов, полученных на курсе, интересы и некоторый практический опыт верстки сайтов с соблюдением следующих условий: тема актуальна, реализуема и практична.

Используемые инструменты - HTML, CSS, Visual Studio Code.

Проект представляет из себя сжатую инструкцию по использованию основных инструментов, необходимых для верстки сайтов – HTML и CSS для непосредственной верстки сайта.

Цель выбранной темы - дать самое общее представление о вышеуказанных инструментах и возможностях использования в практической работе имеющихся сведений для верстки сайтов.

Задачи данного дипломного проекта:

Во-первых, рассмотреть основные понятия HTML, возможности работы с атрибутами, с текстом, со ссылками, с изображениями, с таблицами, списками и формами.

Во-вторых, рассмотреть основные понятия CSS, возможности работы с текстом, блочными и строчными элементами, с полями, позиционирование и работа с «display: flex» и CSS Grid.

Если рассматривать каждый инструмент обстоятельно и подробно, то общий объем данного проекта выйдет далеко за свои рамки. Для более глубокого и подробного понимания предлагается использовать список литературы и источников, приведенных в Разделе 4.

1. HTML

1.1. Понятия HTML

HTML (HyperText Markup Language) - это основной или стандартный язык разметки документов сети INTERNET. В данном разделе используем правила языка HTML, описание стандартной структуры страницы HTML, отношения в общей структуре документа между его элементами. Данный язык «читается» браузерами, а затем полученная и обработанная ими информация выдается на экран монитора ПК, ноутбука, планшета или смартфона.

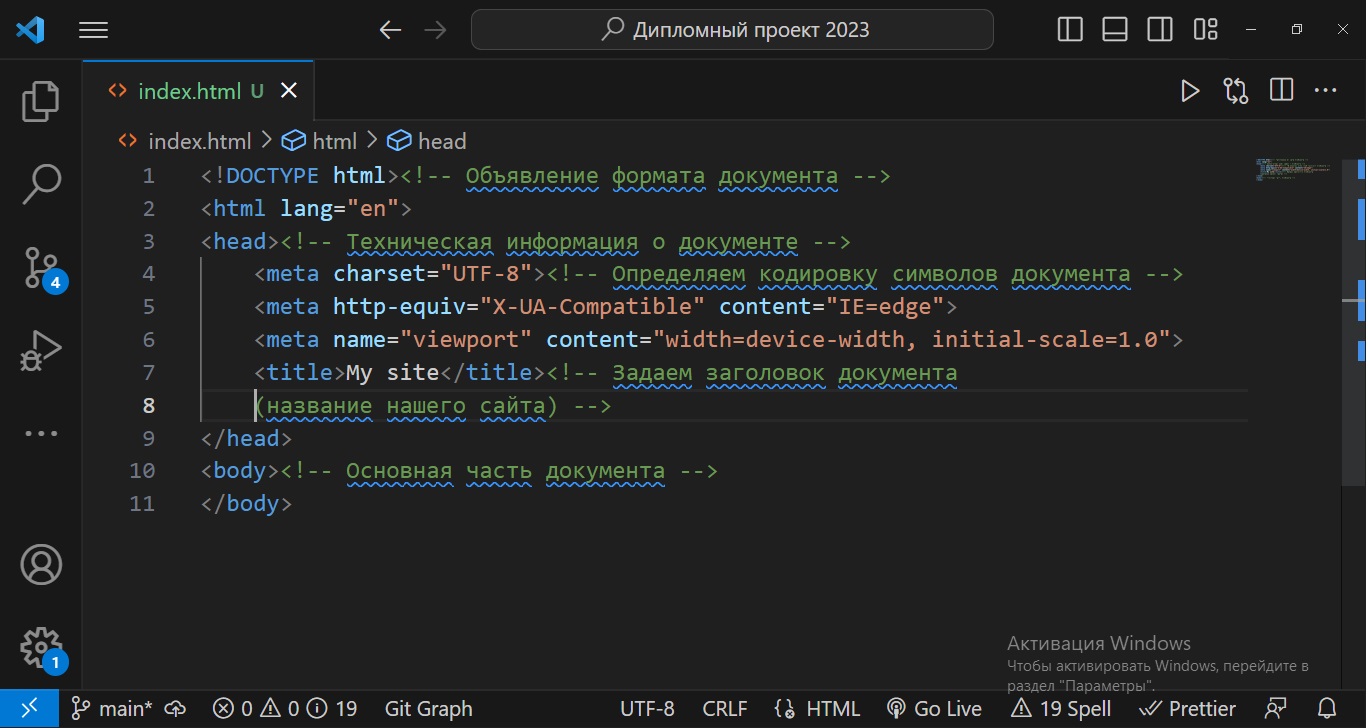
Гипертекст – система текстовых страниц, соединенных между собой ссылками, а сам документ в формате HTML – просто текстовый документ, который можно создать в любом обычном текстовом редакторе (Блокнот, Word) или в редакторе кода (Notepad++, Visual Studio Code и т.п.). Документ формата HTML имеет расширение «.html» в своем названии. Создав на своем ПК, ноутбуке, или планшете файл с таким расширением, можно открыть его через любой веб браузер, просто кликнув на него как на обычный файл на своем устройстве.

В рамках данного дипломного проекта для написания кода будет использоваться редактор кода VSC (Visual Studio Code) как наиболее подходящий по своим возможностям.

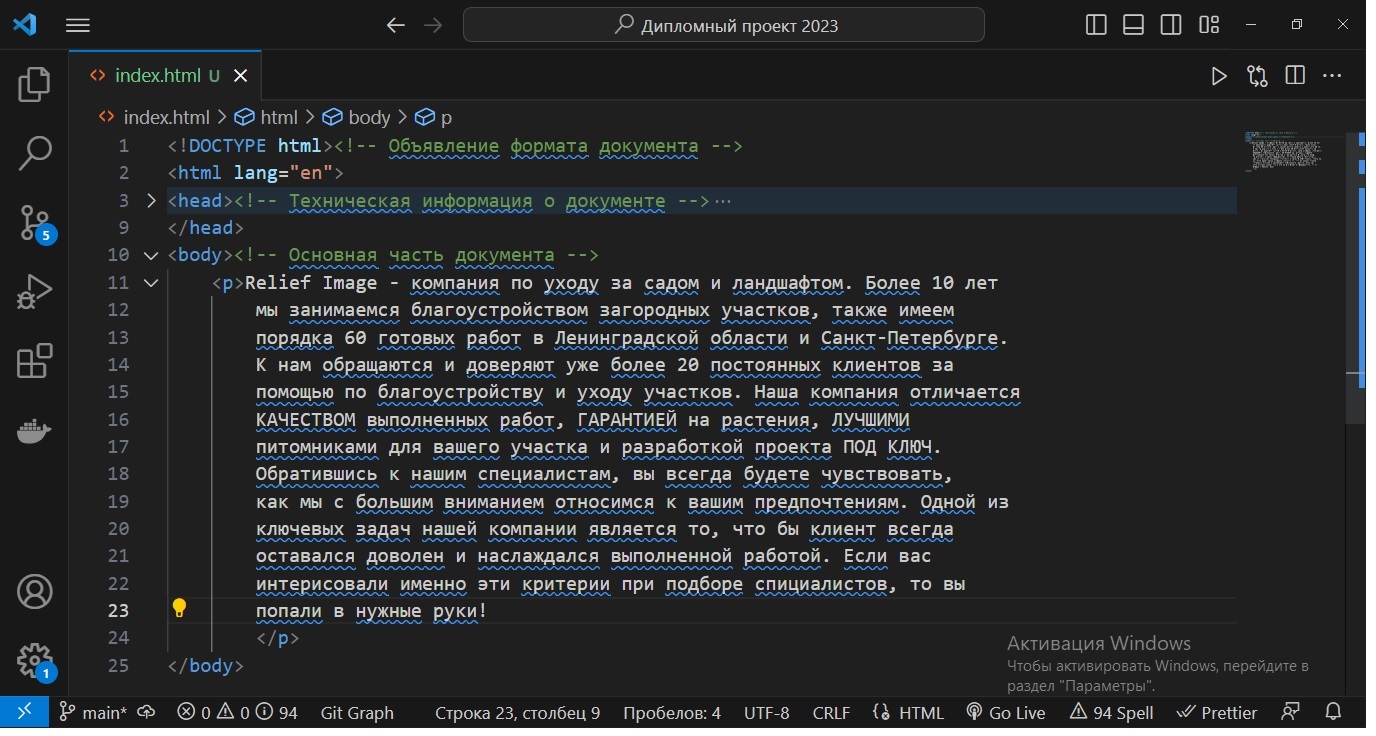
Создадим файл с рабочим названием «index.html» (название файла до наименования его расширения может быть любым) в данном редакторе кода и сформируем самую базовую структуру html-документа нашего нового сайта как представлено на Рисунке 1.1.

Любая запись в представленном коде, начинающая с символа «<!—» и заканчивающаяся на «-->», является комментарием и не отображается непосредственно на странице веб сайта и служит пояснением для самого программиста или других разработчиков.

Созданный нами HTML-документ состоит из дерева множества HTML-элементов и текста или иным другим содержимым. А каждый элемент обозначается в исходном документе начальным (открывающим) и конечным (закрывающим) тегами (с исключениями, когда закрывающийся тег не требуется).

Рисунок 1.1

На Рисунке 1.2 начальный тег «<p>» показывает, где начинается элемент, конечный «</p>» - где заканчивается. Браузер при этом выведет только все содержимое между двумя тегами.

Рисунок 1.2

На Рисунке 1.2 содержимым является текст. В качестве контента в зависимости от наименования тэга будет также название компании (продукта), заголовок статьи или раздела сайта, картинка, видео, аудиозапись и т.д.

Именно элементы, находящиеся внутри элемента «<html>», образуют дерево (модель) документа. При этом сам элемент «<html>» является корневым элементом.

Как видно из представленных рисунков почти все элементы можно вкладывать друг в друга и чаще всего именно такое вкладывание необходимо. Браузер просматривает («читает») код документа, выстраивая его структуру и отображая ее в соответствии со инструкциями стилей, включенными в этот файл (т.е. CSS, который будет рассмотрен позднее). Если разметка верная, то в окне браузера будет отображена страница сайта, содержащая HTML-элементы – текст, заголовки, таблицы, изображения и т.д.

Процесс «чтения» начинается прежде, чем веб-страница полностью загружена в браузер. Браузеры интерпретируют документы HTML последовательно, с самого начала, обрабатывая CSS и соотнося таблицы стилей с элементами самой страницы. Любой HTML-документ состоит из двух разделов: заголовок (содержимое элемента <head>) и содержательная часть (содержимое элемента <body>).

Взаимодействие всех элементов между собой в нашей веб-странице, происходит в отношениях между элементами, представленных в Таблице 1.1.

Таблица 1.1

|  |  |  |
| --- | --- | --- |
| Наименование  элемента | Описание взаимоотношений | Пример  представления |
| Предок | Элемент, включающий в себя другие элементы | «<html>» для «<body>» |
| Потомок | Элемент, сам находящийся внутри одного или более элементов | «<body>» для «<html>» |
| Родитель | Элемент, связанный с другими элементами более низкого уровня, и находящийся в дереве документа выше | «<html>» для «<head>» и «<body>» |
| Дочерний элемент | Элемент, подчиненный другому элементу более высокого уровня | «<р>» для «<body>» |
| Сестринский элемент | Элемент, имеющий общего родителя с рассматриваемым элементом (то есть элементы одного уровня | «<head>» и «<body>» для «<html>» |

Таким образом кроме элемента «html» в нашем (и любом другом) HTML-документе обязательно присутствуют:

- «head» («<head>…</head>») – здесь расположена техническая информация о нашей странице, и она не отображается в окне браузера, но указывает браузеру, как следует интерпретировать страницу;

- «title» («<title>…</title>») – здесь отображается текст в строке заголовка веб-браузера и содержится максимально полное описание содержимого веб-страницы т.е. по сути это название страницы сайта, на котором мы находимся (каждая страница сайта должна иметь собственное название, которое и отображается в данных тегах);

- «meta» («<meta>») – здесь содержится описание содержимого страницы и ключевые слова для поисковых машин, автора документа и прочие свойства метаданных и при этом может содержаться несколько подобных элементов, поскольку в зависимости от используемых атрибутов они несут различную информацию т.е. может иметь множество различных атрибутов;

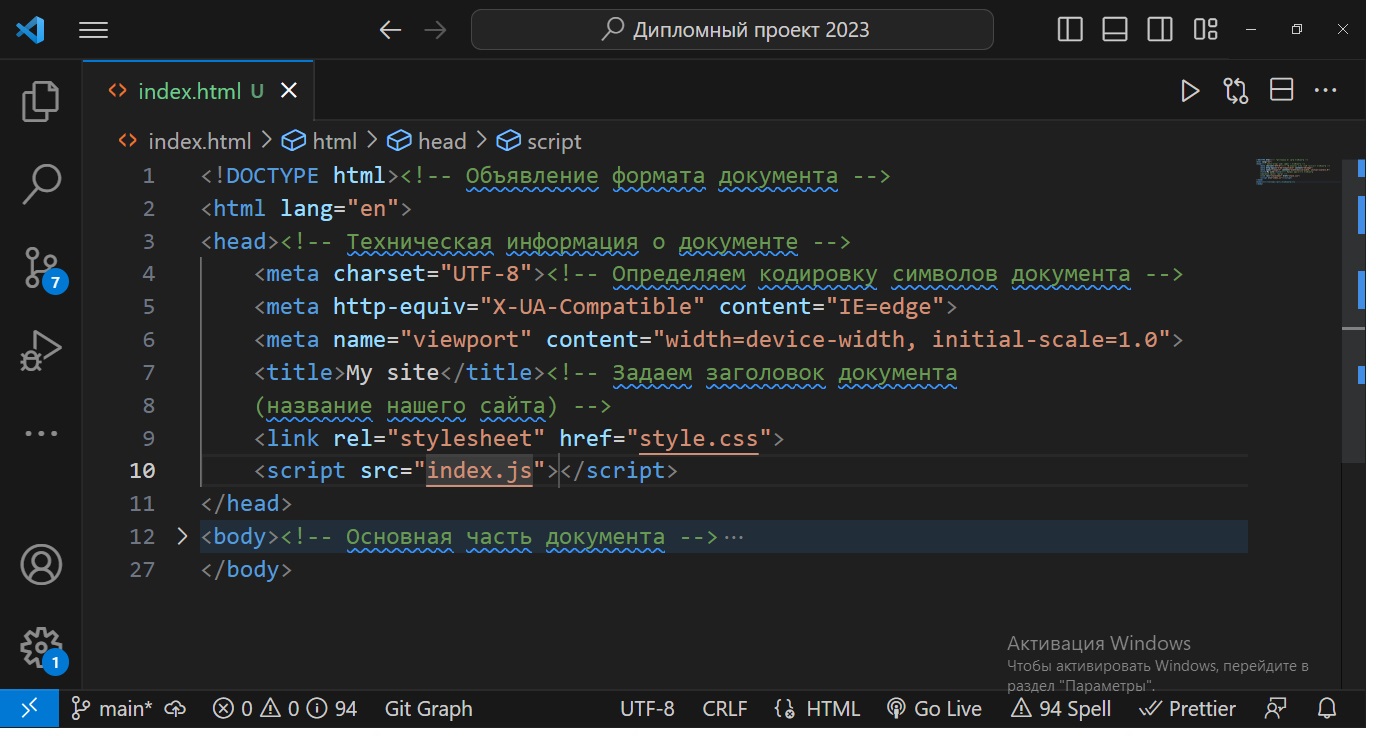
- «style («<style>…</style>») - здесь находятся стили, которые будут использоваться на странице, но в HTML-документе лучше всего использовать все же внешний подключаемый файл CSS (с помощью элемента, указанного ниже);

- «link» («<link>») - подключение к html-документу других исполняемых файлов – файл со стилями style.css, где href является основным атрибутом элемента, где в качестве значения выступает путь к файлу со стилями;

- «script» («<script>») - присоединяет к документу различные сценарии, написанные на языке программирования. Текст сценария может располагаться по аналогии с CSS либо внутри этого элемента, либо во внешнем файле;

- «body» («<body>…</body>») - здесь располагается все основное содержимое документа или его контент, ради чего и создается страница.

Все вышеперечисленные элементы представлены на Рисунке 1.3

Рисунок 1.3

В «body» все содержимое разделяется ка правило на «header» («шапка» сайта, в котором находятся элементы навигации по самому сайту, поисковый сервис, контент с названием организации и логотипа), «footer» («подвал» сайта, в котором могут располагаться контактные данные организации) и основное, видимое в браузере, содержимое.

Это лишь часть основных элементов стандартной html-страницы.

Благодаря использованию в настоящем дипломном проекте Visual Studio Code можно не прописывать все элементы в разделе «head», а развернуть автоматически («!» + Tab) и начать проектировать наш сайт в разделе «body».

1.2 Работа с атрибутами

HTML-элементы на нашем сайте могут (и должны) иметь атрибуты (применяемые вообще для всех HTML-элементов, и собственные). Атрибуты помогают изменять свойства и поведение элемента, для которого они заданы.

Многие из них повторяются в различных тегах («title» - добавляет всплывающую подсказку и используется в большинстве HTML тегов).

HTML-атрибуты — специальные слова, управляющие поведением элемента, к которому они прописаны. Они добавляют дополнительную функциональность, либо меняют поведение элемента по умолчанию. Атрибуты элемента выражаются внутри начального тега элемента.

Некоторые из самых необходимых атрибутов, которые будут нужны в нашем проекте:

- «class» («class="название класса"») - список классов элемента с учетом регистра (позволяют CSS и Javascript выбирать и получать доступ к отдельным элементам с помощью селекторов классов или функций);

- «id» («id="название идентификатора"») - уникальный идентификатор элемента для идентификации элемента;

- «lang» («lang="en "») - указывает основной язык для содержимого элемента и для любого из атрибутов элемента, содержащих текст;

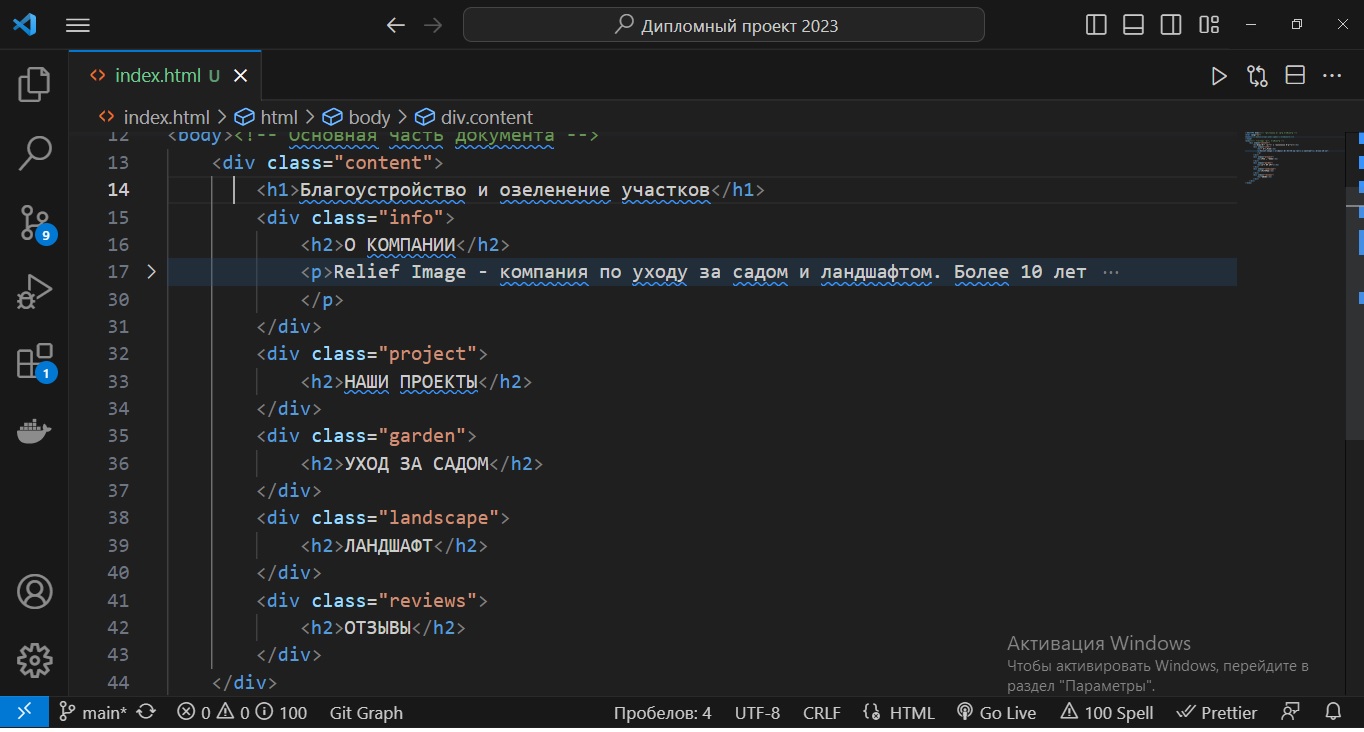
- «style» («style="color: blue; background: transparent"») - содержит объявления стилей CSS, применяемые к конкретному элементу (имеется также тег style, который задает стиль для всей html страницы);

- «title» («title="Hypertext Transport Protocol"») - содержит дополнительную информацию об элементе, задавая всплывающую подсказку для страницы.

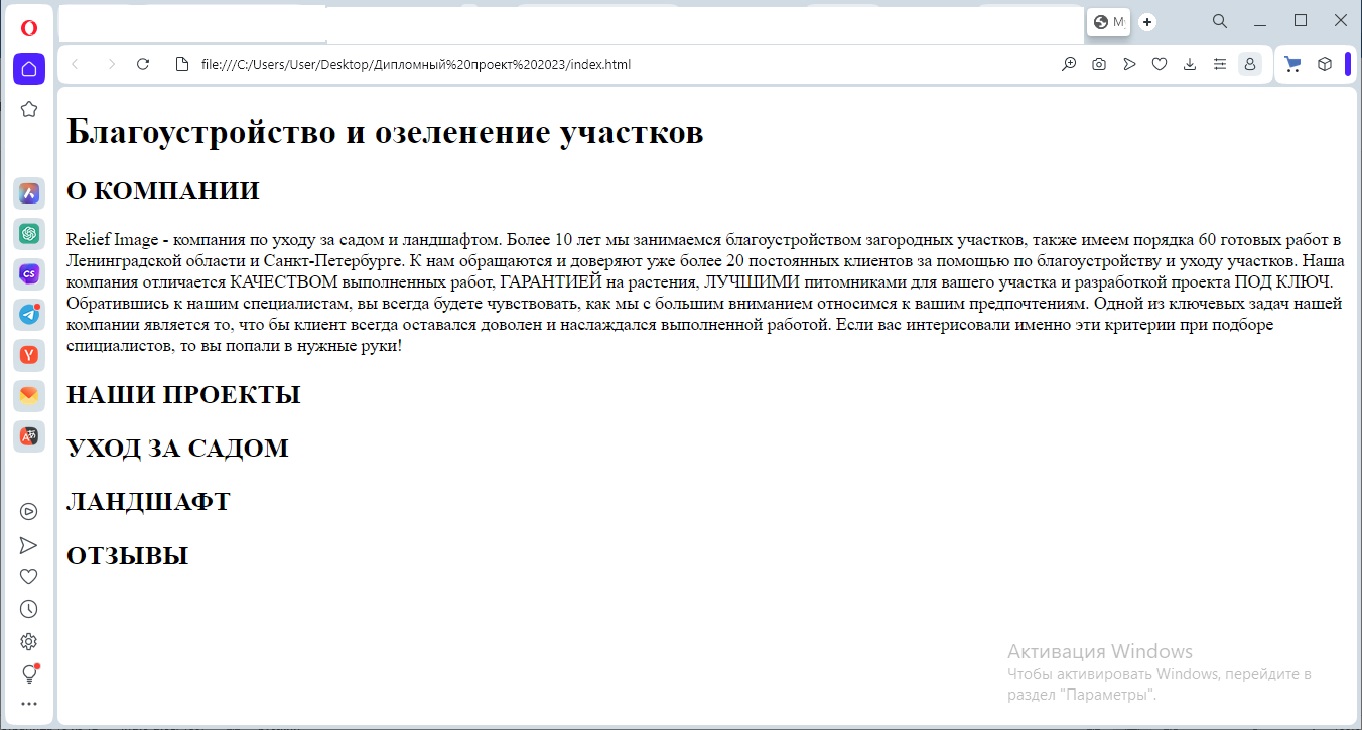
1.3 Работа с текстом

Текст является одним из основных типов контента (содержимого) сайта и сделать его ярче и лучше для восприятия можно за счет добавления к нему специальных HTML тегов.

В HTML можно создавать заголовки за счет тегов «h1», «h2», «h3», «h4», «h5», «h6». Они дают возможность создать заголовки разных размеров (от большего к меньшему). Добавим к нашему проекту один элемент «h1» (должен быть один на странице) и несколько элементов «h2» (элементов «h2» - «h6» может быть сколько необходимо) как показано на Рисунке 1.4.

Рисунок 1.4

Визуальное же отображение представленного кода непосредственно в браузере выглядит как на Рисунке 1.5.

Рисунок 1.5

Заголовки упорядочивают текстовую часть страницы, формируя его визуальную структуру. Элементы «h1» - «h6» должны использоваться только для выделения заголовков нового раздела или подраздела.

Основным тегом для разделения текста является тег абзаца «<p>». В него можно поместить любое количество текста и все будет выделено в формате абзаца с отступами.

Дополнительно в HTML можно делать текст более разнообразным при помощи специальных тегов (жирный текст при помощи тега «<b>», курсивом при помощи тега «<em>»). Но в рамках данного проекта и в целом исходя из сложившейся практики такие элементы при форматировании текста несут лишь смысловую нагрузку и задают для текста, заключенного внутрь, стилевое оформление, выделяяя текст жирным начертанием или отображают его шрифтом другого семейства (свойство font-family).

Все HTML-элементы для работы с текстом представлены в Таблице 1.2.

Таблица 1.2

|  |  |
| --- | --- |
| Наименование элемента | Описание элемента |
| Элементы для форматирования текста | |
| <b> | Задаётся полужирное начертания шрифта текста. Выделяется текст без акцента на его важность |
| <em> | Отображается шрифт текста курсивом, придавая тексту значимость |
| <i> | Отображается шрифт курсивом |
| <small> | Уменьшается размер шрифта на единицу по отношению к обычному тексту |
| <strong> | Задаётся полужирное начертание шрифта, относится к элементам логической разметки, указывая браузеру на важность текста |
| <sub> | Используется для создания нижних индексов, сдвигая текст ниже уровня строки и уменьшая его размер |
| <sup> | Используется для создания степеней, сдвигая текст выше уровня строки и уменьшая его размер |
| <ins> | Выделяется текст в новой версии документа, подчёркивая его |
| <del> | Перечёркивается текст. Используется для выделения текста, удаленного из документа |
| <mark> | Применяется для выделения фрагментов текста в справочных целях, окрашивая блок символов желтым цветом |
| Абзацы, средства переноса текста | |
| <p> | Разбивается текст на отдельные абзацы, отделяя друг от друга пустой строкой |
| <br> | Переносится текст на следующую строку, создавая разрыв строки |
| <hr> | Используется для разделения контента на веб-странице. Отображается в виде горизонтальной линии |

Из всего представленного списка элементов для работы с текстом понадобился лишь элемент «<p>».

1.4 Работа со ссылками

Ссылка - перенаправление пользователя сайта не только на новые страницы, но и на определенное место на текущей странице. Ссылка будет являться активной и отображаться в формате ссылки только при условии прописанного атрибута «href». Если не добавить значения в данный атрибут, то при клике на ссылку страница будет перезагружаться.

Ссылки создаются с помощью элемента «<a>». Внутрь помещается текст, который будет отображаться на веб-странице. Текст ссылки отображается в браузере с подчёркиванием, цвет шрифта - синий, при наведении на ссылку курсор мыши меняет вид.

Обязательным параметром элемента «<a>» является уже упомянутый атрибут href, который задает URl-адрес веб-страницы.

Ссылка состоит из двух частей - указателя («Мы в социальной сети ВК») и адресной части (https://vk.com).

Ссылки бывают:

- относительные;

- абсолютные;

- якоря.

Относительные ссылки используются для перемещения внутри документа или сайта (по страницам сайта).

Абсолютные ссылки используются для перехода на страницы внешнего сайта в сети Интернет. Для этого в значении атрибута href нужно указать полный путь до той страницы, на которую мы хотим перейти, включая и тип протокола.

Когда же в ссылке указывается только имя файла, браузер предполагает, что файл находится в той же папке, что и документ, содержащий ссылку. То это ссылка относительная. Нередко веб-сайты содержат множество документов, которые размещают в отдельные папки, чтобы ими было легче управлять. Создадим отдельный файл Чтобы создать ссылку на файл, находящийся вне папки, содержащей текущий документ, необходимо указать расположение файла или путь: абсолютный или относительный.

Абсолютный путь указывает точное местоположение файла в пределах всей структуры папок на компьютере (сервере). Он даёт доступ к файлу со сторонних ресурсов и содержит различные компоненты (протокол, домен, папка, файл).

Когда необходимо сослаться на страницу на другом сайте, то можно использовать только полный абсолютный путь.

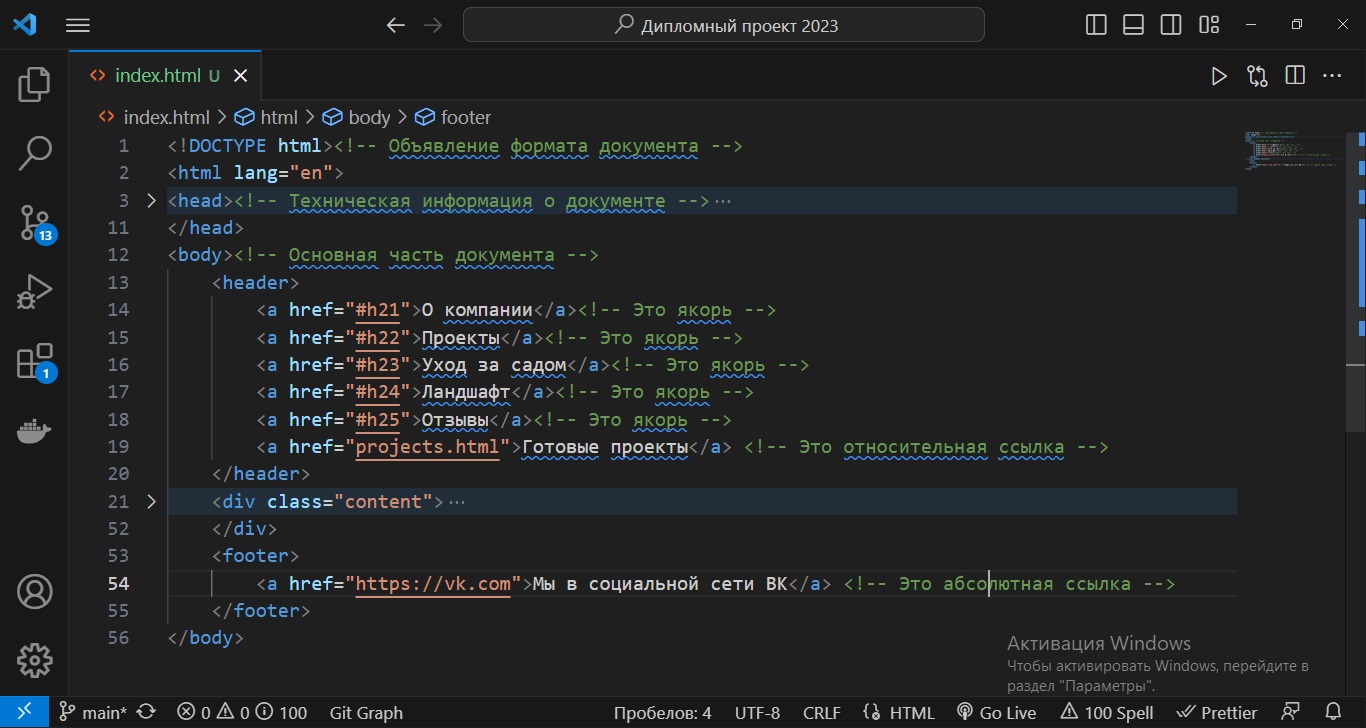
Если файл находится в корневой папке, то путь к нему - http://mysite.ru/index.html.

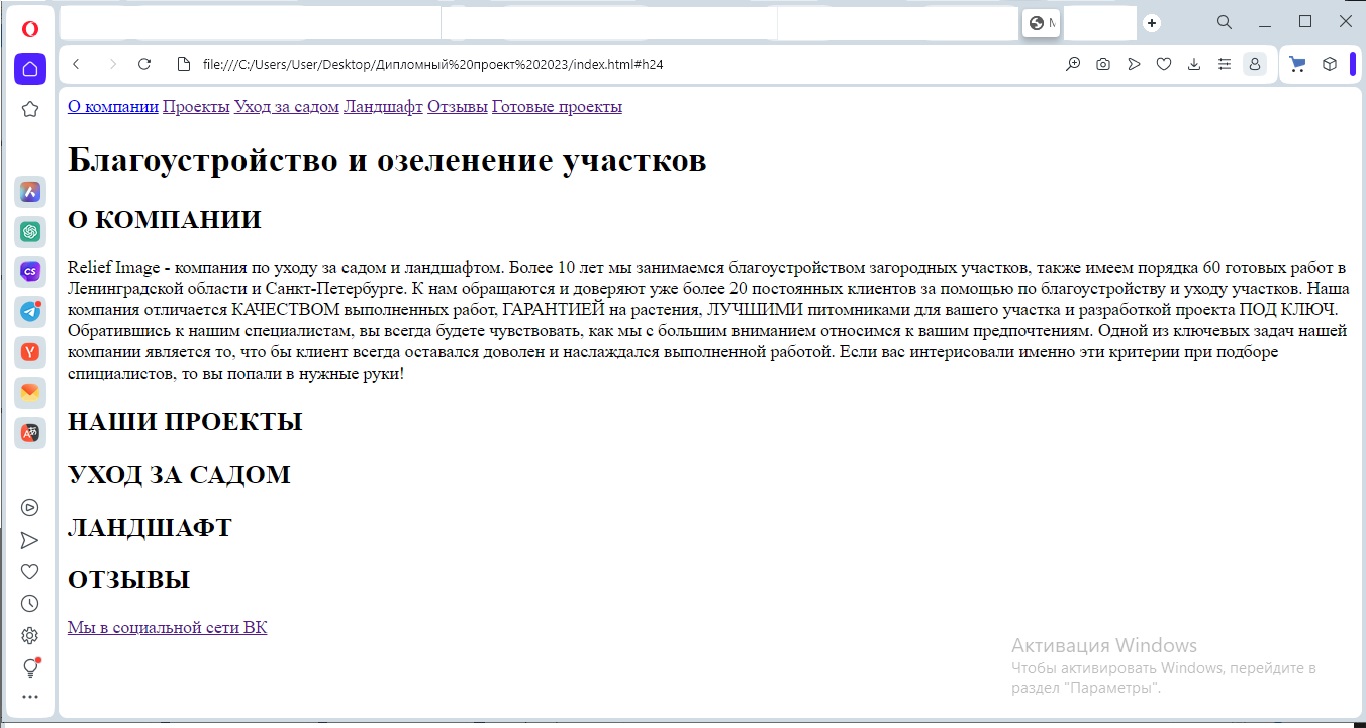
Обычно в качестве индексного файла выступает документ с именем index.html.

Относительный путь описывает путь к указанному документу относительно текущего. Он определяется с учётом местоположения веб-страницы, на которой находится ссылка. Относительные ссылки используются при создании ссылок на другие документы на одном и том же сайте.

Главное отличие относительного пути от абсолютного в том, что относительный путь не содержит имени корневой папки и родительских папок, что делает адрес короче, и в случае переезда с одного домена на другой не нужно прописывать новый абсолютный адрес. Но если сторонний ресурс будет ссылаться на файлы (изображения, видео, другой контент) с относительными адресами, то они не будут отображаться на другом сайте.

Якорь - это внутренняя ссылка, создающая переход на различные разделы текущей веб-страницы, позволяя быстро перемещаться между разделами. Это очень удобно в случае, когда на странице слишком много текста. Внутренние ссылки также создаются при помощи элемента <a> с разницей в том, что атрибут href содержит имя указателя, а не URl-адрес и перед именем указателя всегда ставится знак «#» как показано на Рисунке 1.7, где также представлены ссылки относительная и абсолютная. При нажатии на любую созданную ссылку можно сразу перейти к нужному разделу сайта или другой страницы сайта, или сторонний ресурс. Визуально страница сайта браузером отображается как представлено на Рисунке 1.8.

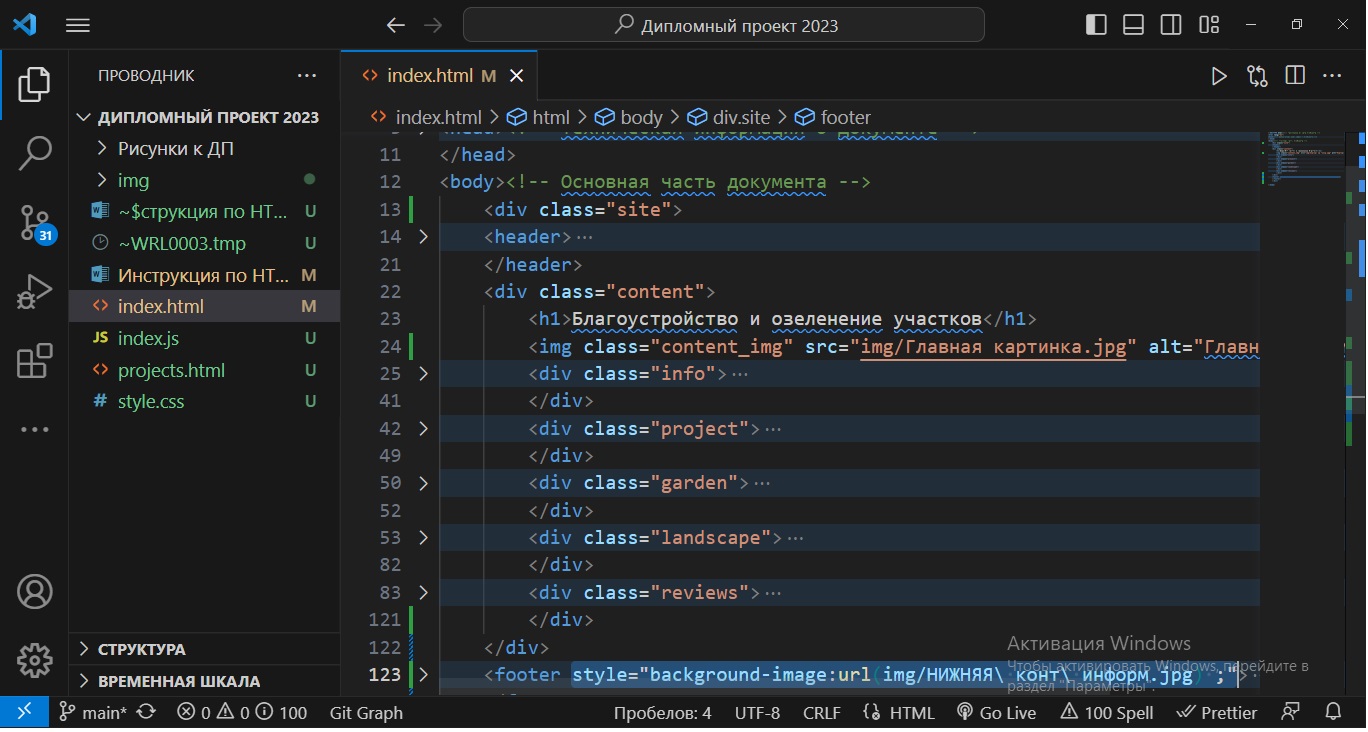
Рисунок 1.7.

Рисунок 1.8

1.5. Работа с изображениями

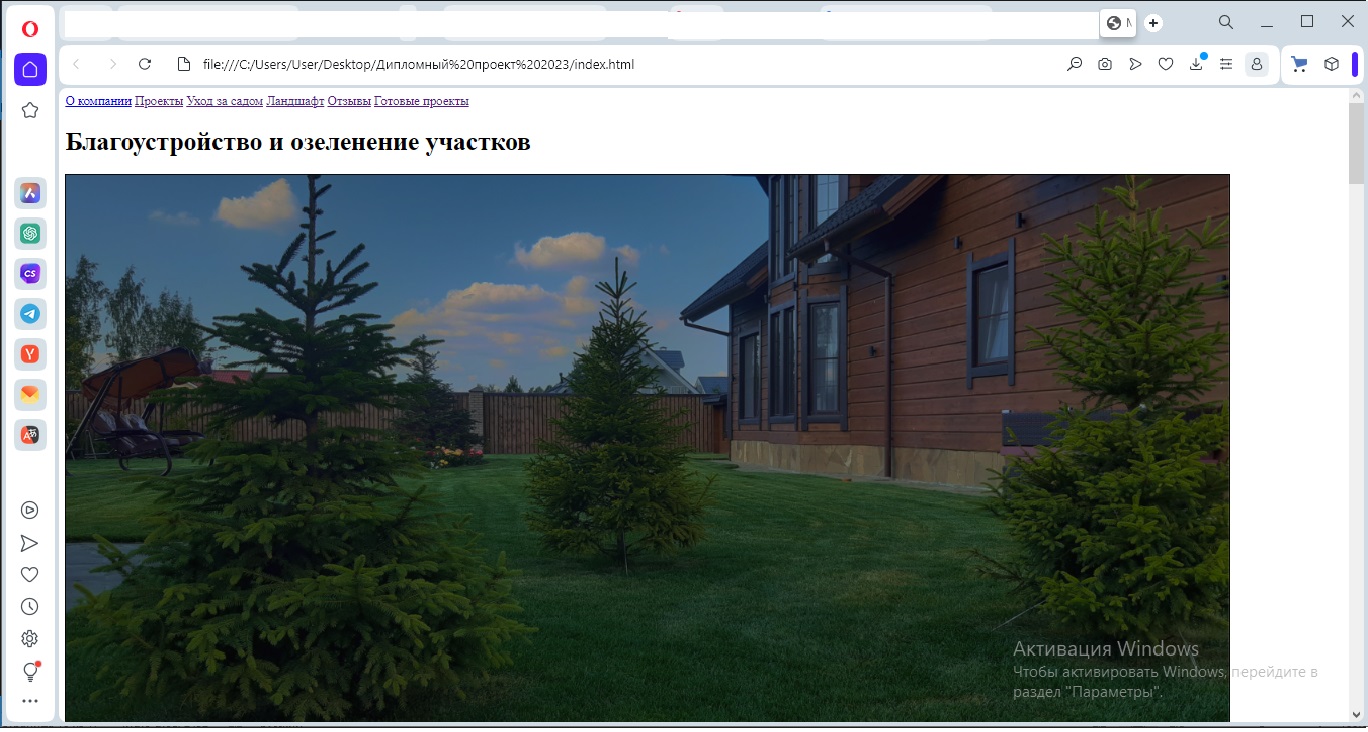
Изображения добавляются на страницы сайта с помощью одинарного тега «<img>» (не требует закрывающего тега).

Вообще использование какой-либо графики делает веб-страницы значительно привлекательнее и позволяет лучше передать суть и содержание веб-документа. Добавим в код нашего проекта несколько элементов «<img>» как представлено на Рисунке 1.9.

Рисунок 1.9

Как видно для тега «<img>» доступны основные атрибуты «src» (задает путь к изображению либо полный к картинке, либо относительный, что находится на сервере или же в одной папке с файлом index.html.) и «alt» (добавляется альтернативный текст для изображения и выводится на месте появления изображения до его загрузки или при отключенной графике). Все изображения любых форматов необходимо размещать в отдельной папке файлов проекта (например, создадим папку img) и уже с помощью атрибута «src» элемента «img» добавлять в код.

Внешний вид веб-страницы при обработке браузером представлен на Рисунке 1.10.

 Рисунок 1.10

Кроме вышеперечисленных используются и другие атрибуты для задания высоты изображения, его размера и ширины.

Возможные форматы изображений для их загрузки на сайт – JPEG (подходит для больших изображений, фотографий), GIF (анимация), SVG (походит для изображений – иконок социальных сетей, почтовых сервисов и других мелких изображений) и многие другие.