Министерство образования Ставропольского края

Государственное бюджетное профессиональное образовательное учреждение «Ставропольский региональный многопрофильный колледж»

****

**Мастер-класс по вёрстке сайтов**

Ответственный:  
Дечева Анжелика Витальевна

Ставрополь, 2021г.

ВВЕДЕНИЕ

***Цели и задачи мероприятия:***

1. Создание простейшего одностраничного сайта
2. Добавление таблицы стилей
3. Тестирование работы сайта в браузере

***Оборудование и ПО:***VisualStudioCode, ПК, проектор, Figma.

***Форма проведения:*** мастер-класс.

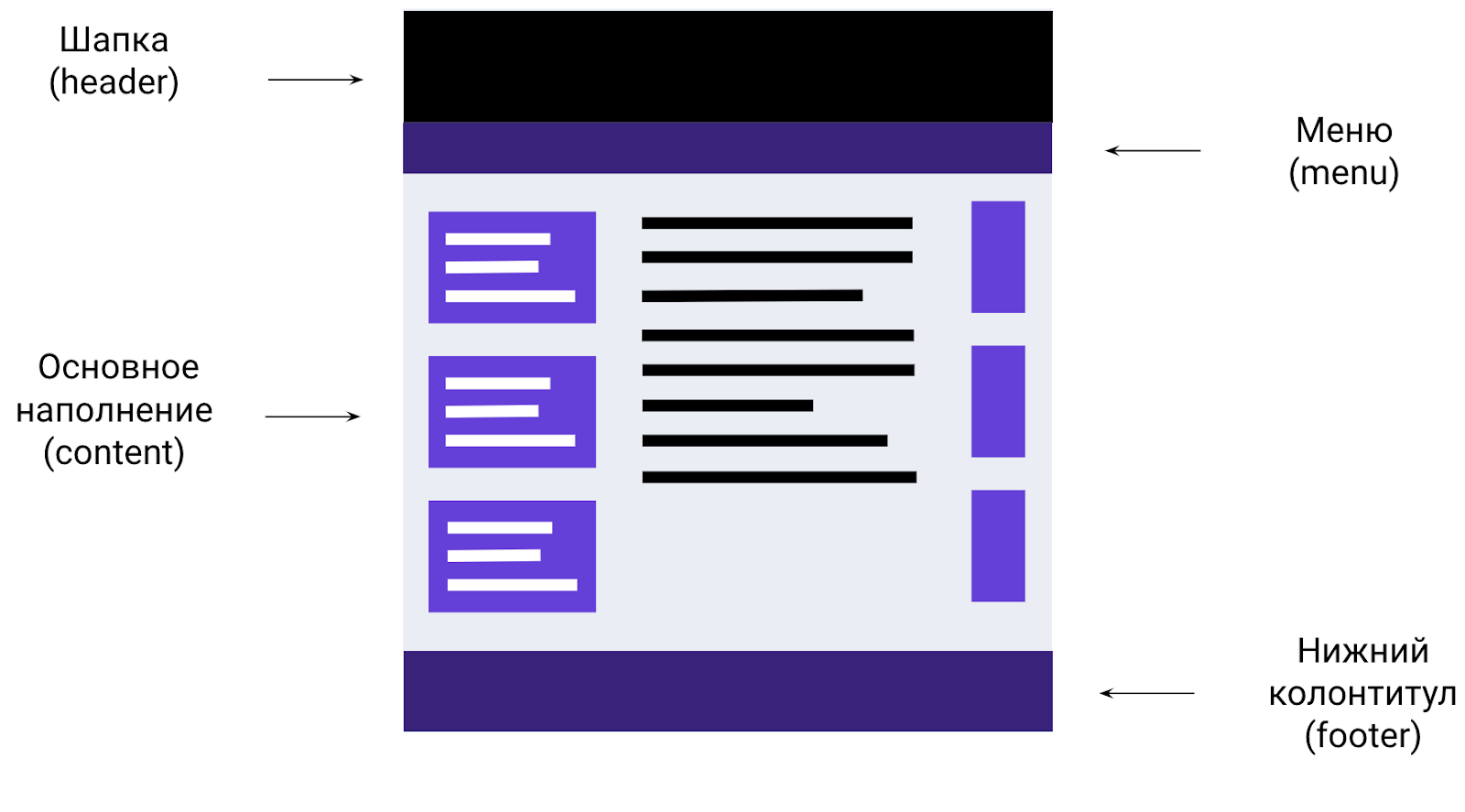
***Участники:***вмастер-классе принимают участие студенты кафедры ПО и ИТ.

***Ход мероприятия:***

1. Теоретическая часть (30 мин)
2. Практическая часть (1 час)

ТЕОРЕТИЧЕСКАЯ ЧАСТЬ

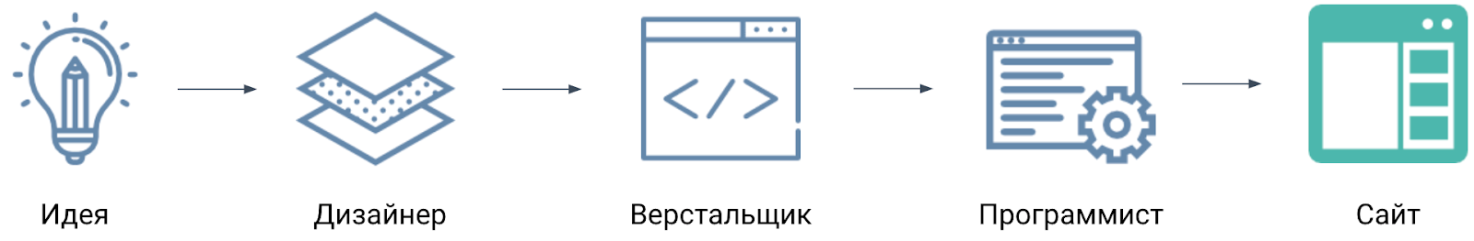
## Что представляет собой веб-страница



Веб-страница сайта состоит из различных блоков:

* Шапка (header), в которой могут размещаться логотип компании, название сайта, телефоны организации и проч.
* Вертикальная и/или горизонтальная навигация по сайту (меню).
* Основное содержимое (content).
* Нижний колонтитул (footer), где размещается дополнительная информация: автор сайта, счетчики посещаемости.

## Процесс разработки сайта



На схеме представлен процесс разработки сайта на примере веб-студии, где определенные функции выполняет определенный человек.

Когда заказчик приходит со своей идеей сайта в веб-студию, с ним начинает работать менеджер проекта. Очень часто бывает, что идея у заказчика есть, но она настолько размыта, что тяжело представить на этом этапе, что в итоге хочет получить клиент.

В разных организациях используется свой подход к выяснению потребностей заказчика. Один из вариантов – специальная анкета, после составления которой можно уже представить, как будет выглядеть будущий сайт и каким функционалом обладать.

Далее менеджер составляет два технических задания: одно дизайнеру на составление макета сайта (дизайна), а другое – программисту с указанием, какой функционал необходимо реализовать в этом проекте.

После этого верстальщику отправляется макет сайта от дизайнера и различные модули от программиста, и после того, как сайт будет сверстан, копирайтер наполняет его контентом, т.е. заполняет товарами, пишет статьи и т.д. В это же время seo-специалист начинает заниматься продвижением сайта.

## Гипертекст

Гипертекст – термин, введенный Тедом Нельсоном в 1965 году для обозначения «текста ветвящегося или выполняющего действия по запросу». Обычно гипертекст представляется набором текстов, содержащих узлы перехода между ними, которые позволяют избирать читаемые сведения или последовательность чтения.

Общеизвестным и ярким примером гипертекста служат веб-страницы – документы HTML (язык разметки гипертекста), размещенные в сети. В более широком понимании термина гипертекстом является любая повесть, словарь или энциклопедия, где встречаются отсылки к другим частям данного текста, имеющим отношение к данному термину. В компьютерной терминологии гипертекст – текст, сформированный с помощью языка разметки, потенциально содержащий в себе гиперссылки.

## Теги и атрибуты

Язык HTML – язык тегов. Теги описывают структуру HTML-документа. Они оформляются угловыми скобками <имя тега>, между которыми прописывается имя тега. Теги HTML-документа предназначены для управления конструкциями разметки: заголовками, абзацами, списками, таблицами и картинками.

Теги бывают парные и одиночные.

Парные теги сначала открываются, потом закрываются. Все, что описывают парные теги, находится внутри них. Так, тег <b></b> делает текст полужирным. Вот как это выглядит:

|  |
| --- |
| <b>Какое прекрасное утро</b> |

Одиночные теги стоят сами по себе и, как правило, не меняют контент вокруг себя.

Пример:

|  |
| --- |
| <br> -- тег переноса строки <hr> -- тег горизонтальной линии |

Атрибуты тегов определяют какие-либо дополнительные, уточняющие параметры того или иного тега. Вы можете указать внутри тега столько атрибутов, сколько вам необходимо. Атрибуты разделяются пробелами.

<название\_тега атрибут1="значение\_атрибута1"> текст внутри тега </название\_тега>

Пример:

|  |
| --- |
| <ahref="contacts.html">Контакты</a> <imgsrc="book.png"alt="Книга"> |

## Структура HTML-документа

Каждая страница в HTML-документе состоит из трех обязательных элементов:

1. Тип документа.
2. Раздел <head></head> с технической информацией о странице: заголовок, описание, ключевые слова для поисковых машин, кодировка. Введенная в нем информация в основном не отображается в окне браузера, однако содержит данные, которые указывают браузеру, как следует обрабатывать страницу.
3. Раздел <body></body>, где располагаются все элементы, которые видит пользователь. Именно с этим разделом мы в основном и будем работать.

DOCTYPE отвечает за корректное отображение веб-страницы браузером. DOCTYPE определяет не только версию HTML (например, html), но и соответствующий DTD-файл в интернете.

Элементы, находящиеся внутри тега <html>, образуют дерево документа, так называемую объектную модель документа, DOM (documentobjectmodel). При этом элемент <html> является корневым.

Пример структуры HTML 5:

|  |
| --- |
| <!doctype html> <html> <head>     <title>Hello HTML</title>     <metacharset="UTF-8"> </head> <body> ... </body> </html> |

## Основные теги оформления текста

*Заголовки*

Как и в газетах и журналах, в HTML-странице любая статья или новость должна начинаться с заголовка. Предусмотрено 6 уровней заголовков: первый из них будет отображен самым крупным шрифтом, а далее чем выше уровень, тем меньше будет размер шрифта.

Для отображения заголовков существует тег <h> и указывается цифра от 1 до 6, которая соответствует уровню заголовка. Тег заголовка – парный, не забудьте его закрыть.

<h1>Заголовок первого уровня</h1>

<h2>Заголовок второго уровня</h2>

<h3>Заголовок третьего уровня</h3>

<h4>Заголовок четвертого уровня</h4>

<h5>Заголовок пятого уровня</h5>

<h6>Заголовок шестого уровня</h6>

*Параграфы*

После заголовка обычно следует какой-нибудь текст, который необходимо заключать в параграфы, или абзацы. При составлении документа выделяйте блоки текста в отдельные параграфы, как это сделано в книгах – в противном случае может получиться сплошной текст, который очень трудно будет читать посетителю сайта.

В HTML для параграфов используется парный тег <p>, и внутри него помещается тот текст, который нужно отобразить.

Пример:

|  |
| --- |
| <p>Здесь мы напишем первый параграф</p> |

## Теги выделения текста

Иногда необходимо выделить слово, словосочетание, предложение или целый участок текста, чтобы привлечь внимание читателя или поискового робота. Чтобы это сделать, можно выделить фрагмент жирным, курсивом или подчеркнуть.

Аккуратнее с подчеркиванием текста: по стандартам принято, что подчеркнуты гиперссылки, и многие пользователи уже настолько к этому привыкли, что, видя подчеркнутый текст, хотят «кликнуть» по этому участку текста, ожидая перехода на другую страницу.

Теги <b> и <strong> внешне делают одно и то же: выделяют текст полужирным. Разница в том, что <strong> указывает на важность текста, а <b> просто делает текст полужирным. Так, текст в тегах <strong> устройство для чтения текста вслух будет выделять интонацией, а <b> – нет.

Аналогично с тегами <em> и <i>, делающими текст курсивным: <em> указывает на важность текста, а <i> нет.

<small> уменьшает размер шрифта на единицу по отношению к обычному тексту.

<sub> используется для создания нижних индексов. Сдвигает текст ниже уровня строки, уменьшая его размер.

<sup> используется для создания степеней и верхних индексов. Сдвигает текст выше уровня строки, уменьшая его размер.

<ins> выделяет текст в новой версии документа, подчеркивая его.

<del> перечеркивает текст. Используется для выделения текста, удаленного из документа.

<code> служит для выделения фрагментов программного кода. Отображается моноширинным шрифтом.

<pre> позволяет вывести текст на экран, сохранив изначальное форматирование. Пробелы и переносы строк при этом не удаляются.

<q> используется для выделения коротких цитат. Браузерами заключается в кавычки.

## Вложенные теги

Очень часто при верстке веб-страниц необходимо вкладывать одни теги в другие. Эта вложенность может достигать несколько уровней. Поэтому следует запомнить одно простое правило: закрывать теги необходимо именно в той последовательности, в которой вы их открывали. Первый открытый тег закрывается последним.

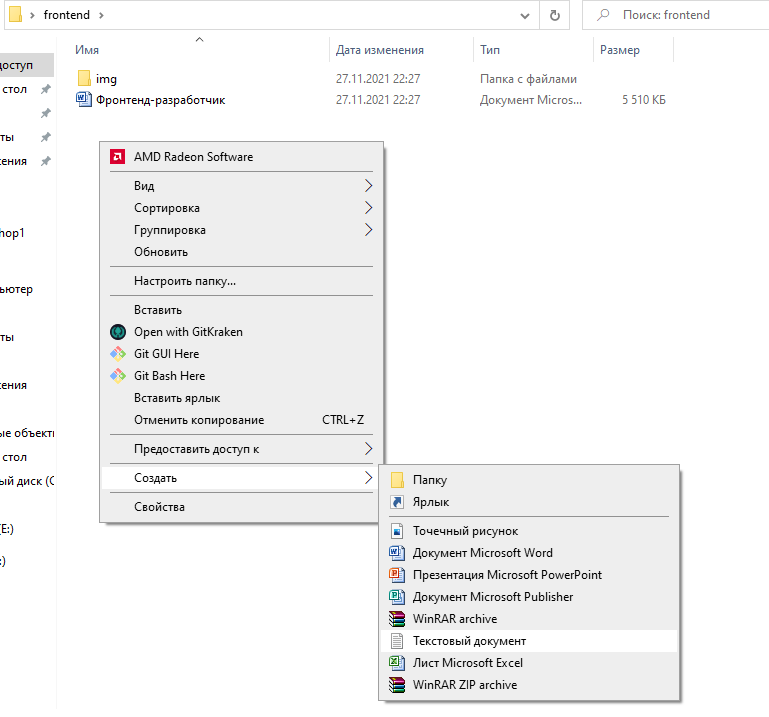
|  |
| --- |
| <p>Использовать вложенность тегов     <b>легко             <i>и просто</i>    </b> </p> |

Тег <i> был открыт последним – закрываем его первым, а далее уже все остальные теги по очереди.

ПРАКТИЧЕСКАЯ ЧАСТЬ

**Оформляем html-документ**

1. Открываем папку frontendи создаем в ней текстовый документ (ПКМ – Создать - Текстовый документ).



2. Открываем созданный текстовый документ, записываем «скелет» нашего сайта.(Записанный ниже код на картинке можно скопировать)

<!DOCTYPEhtml>

<htmllang="en">

<head>

    <metacharset="UTF-8">

    <metahttp-equiv="X-UA-Compatible"content="IE=edge">

    <metaname="viewport"content="width=device-width, initial-scale=1.0">

    <linkhref="https://fonts.googleapis.com/css2?family=Josefin+Sans&display=swap"rel="stylesheet">

    <linkrel="stylesheet"href="main.css">

    <title>Document</title>

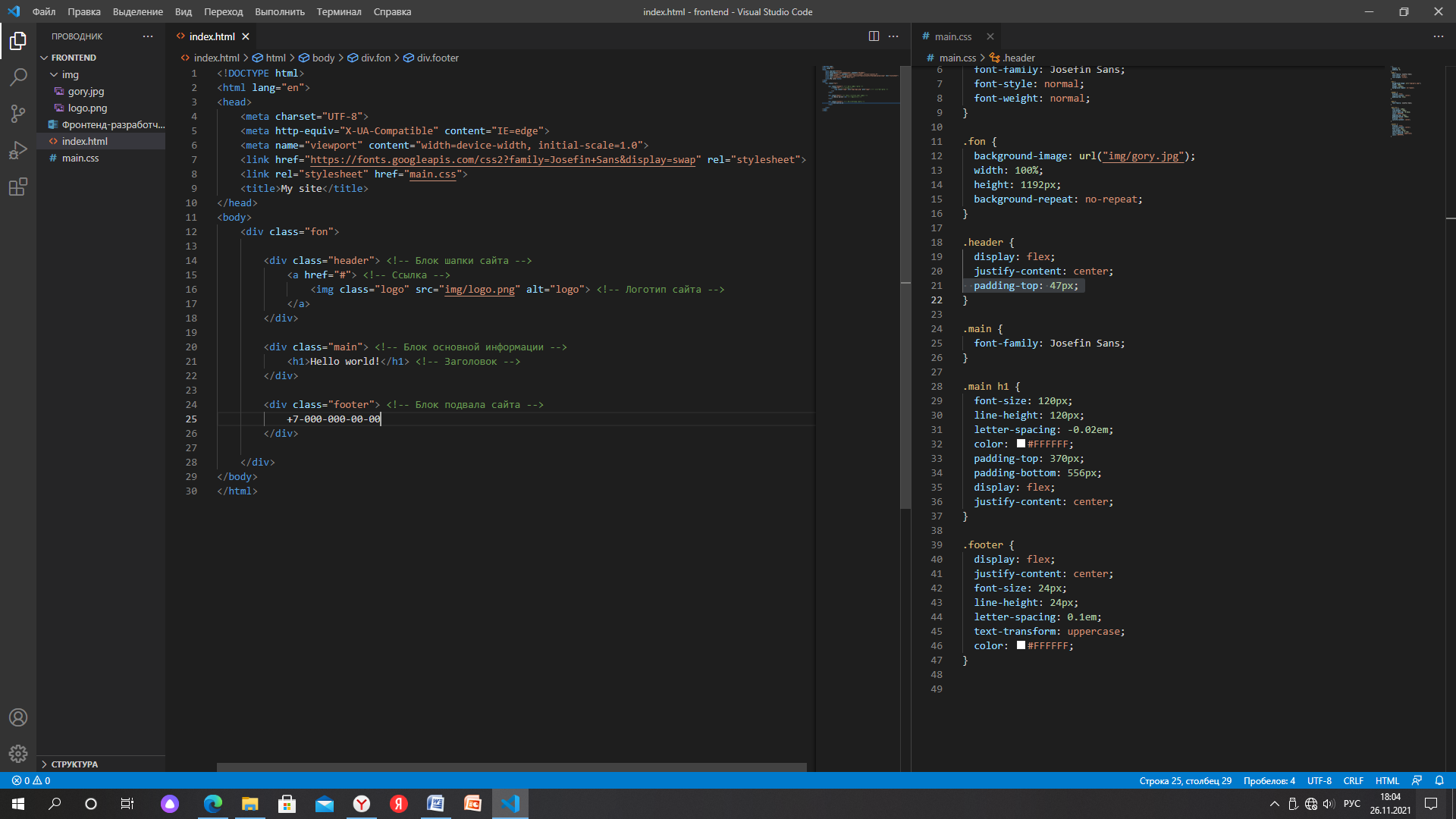
</head>

<body>

</body>

</html>

3. Изменяем имя страницы, которое будет отображаться в браузере на вкладке, для этого между тегами <title></title>введите своё название страницы.

4. Между тегами <body></body>, которые являются основой сайтов,начинаем писать код, который будет отображаться в браузере в виде странички (всё, что написано зеленым, можно не писать – это комментарии для вас).

### Обратите внимание!

### В этой строке, в атрибуте src необходимо правильно указать путь к изображению.

<imgclass="logo"src="img/logo.png"alt="logo">

### Сайт делится на контейнеры <div>, которые нужны для группировки элементов сайта. Данный сайт можно разделить на несколько блоков:

### Основной блок, в который входят еще несколько блоков

### 

### Блок шапки сайта

### 

### Блок основной информации

### 

### Блок подвала сайта

### 5. Сохраняем текстовый файл (Файл –Сохранить как.., задаем имя файлу index.html, выбираем тип файла Все файлы, сохраняем).

### 

### Оформление стиля

### 6.Открываем папку frontend и создаем в ней текстовый документ (ПКМ – Создать - Текстовый документ).

### 

### 7. В созданном текстовом документе пишем код.

### 

### Желтым цветом выделены классы (объект, которому нужно задать стиль), синем их атрибуты (непосредственно стиль, расположение и другие параметры).

### 8. Сохраняем текстовый файл (Файл –Сохранить как.., задаем имя файлу main.css, выбираем тип файла Все файлы, сохраняем).

### 

### Обратите внимание!

### Чтобы файл таблицы стилей заработал, его нужно подключить к html-документу, для этого заново пропишите эту строчку в файле index.html

### 9. Открываем папку frontendи запускаем текстовый документ index.htmlв браузере.