

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

**РАЗРАБОТКА АДАПТИВНОЙ ВЁРСТКИ ВЕБ-САЙТА
ПРИ ПОМОЩИ ТЕХНОЛОГИЙ HTML5 И CSS3**

Лабораторный практикум

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

г. Тирасполь
2021 г.

Лабораторная работа №1

Создание HTML страницы. Основная разметка документа в стандарте HTML5.

Цель работы: ознакомиться со структурой страницы и научиться создавать основу веб-сайта согласно стандарту HTML5.

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 8/8.1/10).

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 включает 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.

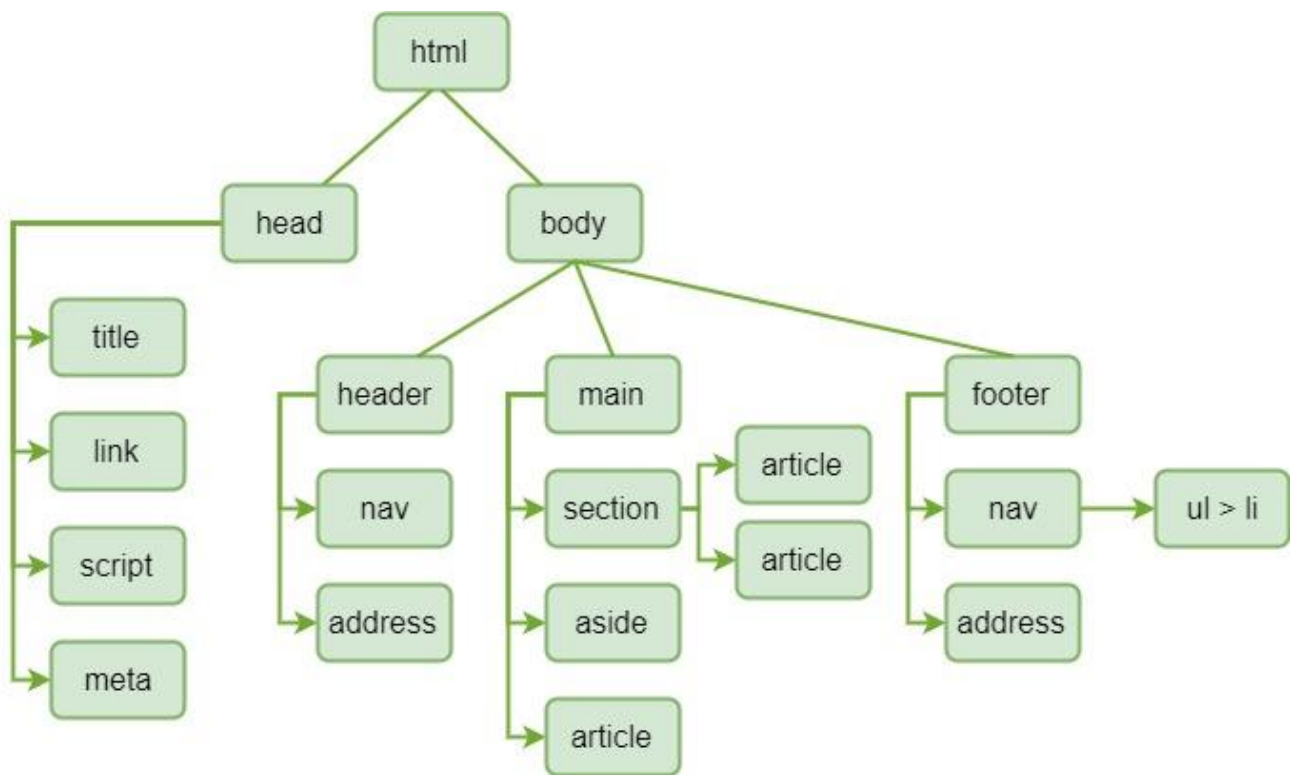


Рис. 1 – Простейшая структура веб-страницы

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 ``.

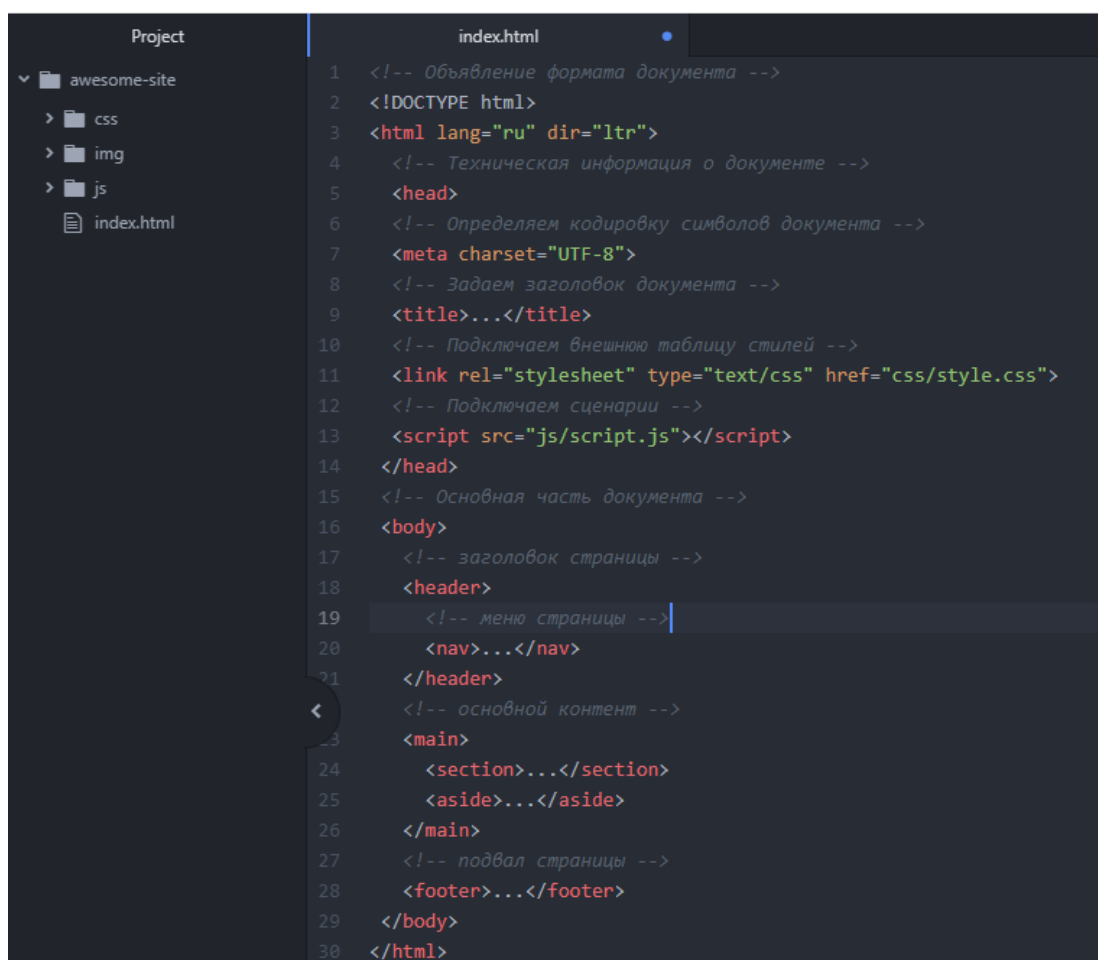
Дочерний элемент — элемент, непосредственно подчиненный другому элементу более высокого уровня. На рисунке 1 только элементы `<h1>`, `<h2>`, `<p>` и `<nav>` являются дочерними по отношению к `<body>`.

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.

Практическая часть

Для работы с HTML5 необходим текстовый редактор, чтобы набирать текст веб-страниц на html. К популярным редакторам относятся Notepad++, Atom и Sublime Text. Их преимуществом перед полновесными средами разработки является небольшой вес программы, скорость запуска и удобство редактирования одиночных файлов. Также первые две программы распространяются бесплатно. В курсе лабораторных работ примеры будут приводиться в интерфейсе редактора Atom.

И также потребуется веб-браузер для запуска и проверки написанных веб-страничек. В качестве веб-браузера можно взять 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.



```
1 <!-- Объявление формата документа -->
2 <!DOCTYPE html>
3 <html lang="ru" dir="ltr">
4 <!-- Техническая информация о документе -->
5 <head>
6 <!-- Определяем кодировку символов документа -->
7 <meta charset="UTF-8">
8 <!-- Задаем заголовок документа -->
9 <title>...</title>
10 <!-- Подключаем внешнюю таблицу стилей -->
11 <link rel="stylesheet" type="text/css" href="css/style.css">
12 <!-- Подключаем сценарии -->
13 <script src="js/script.js"></script>
14 </head>
15 <!-- Основная часть документа -->
16 <body>
17 <!-- заголовок страницы -->
18 <header>
19 <!-- меню страницы -->
20 <nav>...</nav>
21 </header>
22 <!-- основной контент -->
23 <main>
24 <section>...</section>
25 <aside>...</aside>
26 </main>
27 <!-- подвал страницы -->
28 <footer>...</footer>
29 </body>
30 </html>
```

Рис. 2 – Структура веб-страницы в редакторе

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

Приступим к созданию будущего проекта. Создадим в проводнике каталог с названием сайта, например, awesome-site.

Внутри каталога awesome-site создадим ещё несколько каталогов с названиями css, img и js. Пока они будут пустыми, но их содержимое наполнится в следующих лабораторных работах. Сейчас они просто будут находиться в папке проекта, чтобы к ним привыкнуть. Дело в том, что большое количество корневых каталогов веб-сайтов в интернете выглядят именно так.

Теперь откроем редактор Atom, выберем пункт File -> Open Folder... -> Открываем нашу папку с проектом -> Выбор папки.

Результат ваших действий должен выглядеть как на рисунке 3.

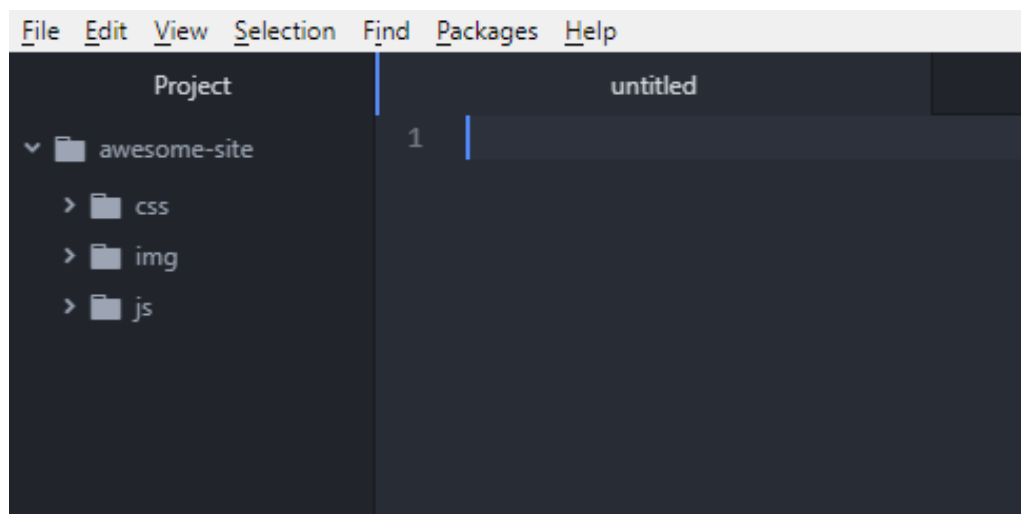


Рис. 3 – Открытый проект

Теперь создадим файл с HTML вёрсткой: правый клик по корневому каталогу проекта, в контекстном меню выбираем «New Folder». В модальном окне вводим название и расширение файла: index.html. Название должно быть именно таким. К нему по умолчанию обращается браузер, пытаясь прочитать каталог вашего сайта.

Если вы работаете в редакторе Atom, начните набирать корневой тег `html` (VSCode предлагает несколько вариантов на выбор, нас интересует `html5`) и нажмите Ввод, редактор создаст для вас заготовку страницы – рисунок 4.

Кроме того, редактор допишет в тег `html` вспомогательные атрибуты `lang` и `dir`. Они не являются обязательными, но необходимы для соответствия стандарту HTML5 так, как браузер использует их значения для правильного отображения некоторых символов. Атрибут `lang` (language) нужно установить значение «`ru`», а для `dir` (direction – направление текста) – «`ltr`».

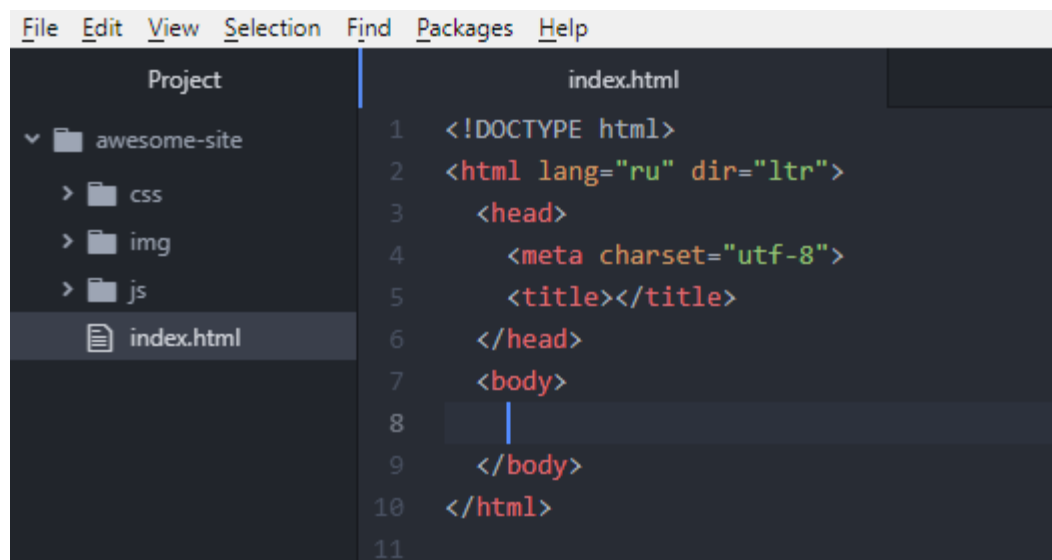


Рис. 4 – Заготовка страницы

Допишите в файл `index.html` остальную разметку из рисунка 2. Комментарии записывать не обязательно, но они помогут вам первое время ориентироваться в документе. В теге `title` вместо многоточия запишите свой заголовок документа. А в теге `section`, по доброй традиции – Hello world! Посмотрим, что получилось. Сохраните ваши изменения: `File -> Save` или горячие клавиши `ctrl + S`. Перейдите к проекту в проводнике и откройте файл `index.html`. Обычно в операционной системе установлена ассоциация с файлами `html` с каким-либо браузером.

ром. В ином случае вы можете открыть свою страницу прямо из браузера: откройте новую вкладку и вызовите вашу страницу, вписав её полный путь на жёстком диске в адресную строку:

file:///C:/D/E:[путь к проекту]/awesome-site/index.html

Результат должен быть, примерно, как на рисунке 5. Введённый вами заголовок страницы отображается в названии вкладки, а введённый внутри тега `body` текст, на основной странице.

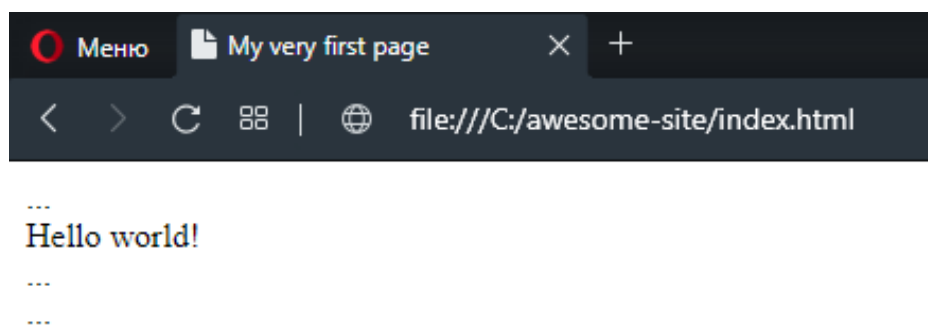


Рис. 5 – Вывод контента страницы в окне браузера

Любой веб-браузер имеет набор стандартных стилей, которые он применяет к html документу. Если вы не стилизовали свою страницу, то к ней применяется стандартный набор. Именно поэтому текст имеет определенный размер, стиль и отступы. Times New Roman на странице выглядит слегка скучно. Чтобы заменить его, мы можем воспользоваться двумя решениями: добавить к нашему проекту файл шрифтов или подключить набор шрифтов из внешнего источника в сети. У каждого подхода есть свои преимущества и недостатки, поэтому рассмотрим оба варианта.

Создадим в нашем проекте ещё один каталог `fonts`. Скопируем в него файлы с наборами шрифтов из приложения к лабораторной работе.

В папку `css` скопируем подготовленный файл `fonts.css`. В нём прописаны правила подключения шрифтов к списку стилей сайта. Теперь необходимо подключить данный файл к нашей странице: добавим в тег `head` тег следующего содержания `<link rel="stylesheet" type="text/css" href="css/fonts.css">`.

Ниже добавим тег `style`, внутри него можно задавать стиль для html элементов при помощи CSS. Наберём:

```
body { font-family: "OpenSans-Light", Helvetica, Arial, sans-serif; }
```

На рисунке 6 изображено как выглядит наш проект внутри. Сохраните изменения и обновите страницу. В результате текст принял более утончённый вид.

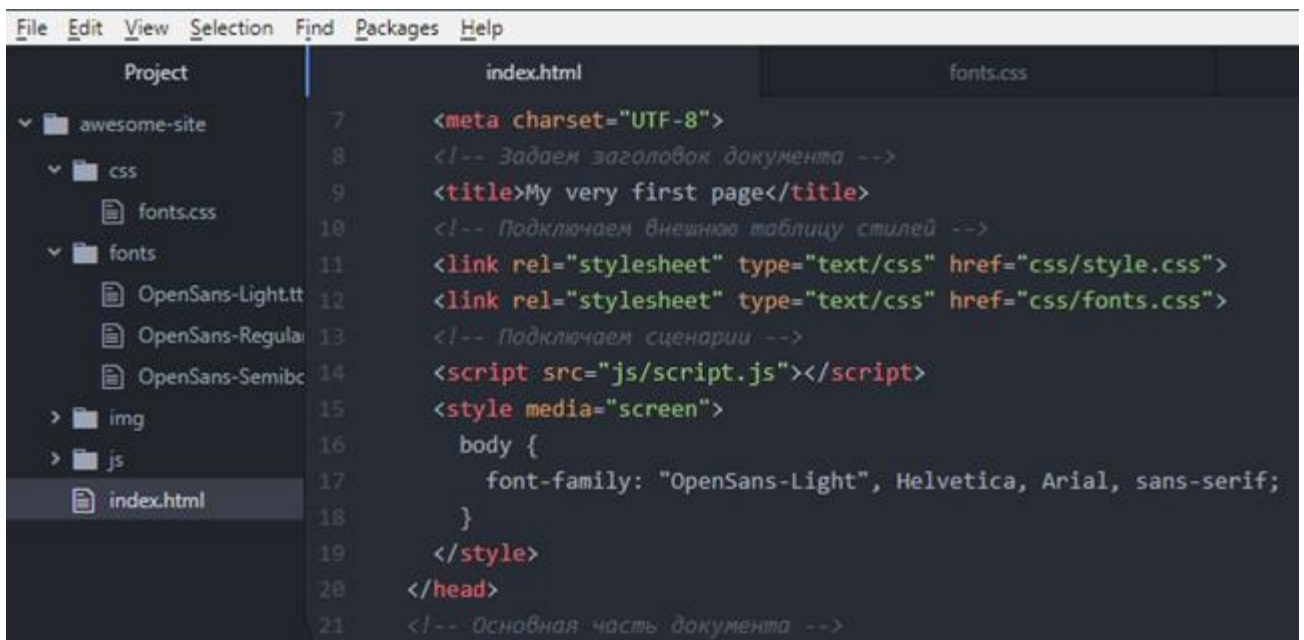


Рис. 6 – Подключение шрифтов и создание стиля текста

Второй способ проще, но он зависит от сервера, на котором расположены наборы шрифтов. Пока они не загрузятся, посетитель вашей страницы увидит текст в более привычном виде.

Для того, чтобы на странице не отображался обычный Times New Roman, в стиле `font-family` после названия подключенного шрифта указано через запятую ещё несколько, которые обычно установлены в операционной системе. Если ни один из них браузер не найдёт, то он откроет любой шрифт из семейства `sans-serif`, то есть шрифт без засечек.

Самым популярным хранилищем шрифтов на сегодняшний день является <https://fonts.google.com/>. О том, как выбрать и сконфигурировать шрифт можно

подробно прочитать в одной из множества инструкций в сети, пока просто скопируем в наш проект тег `<link href="https://fonts.googleapis.com/css?family=Roboto:300,400,500&display=swap&subset=cyrillic" rel="stylesheet">`.

Добавим на страницу список меню и применим наш новый шрифт. На рисунке 7 показана разметка верхней части страницы.

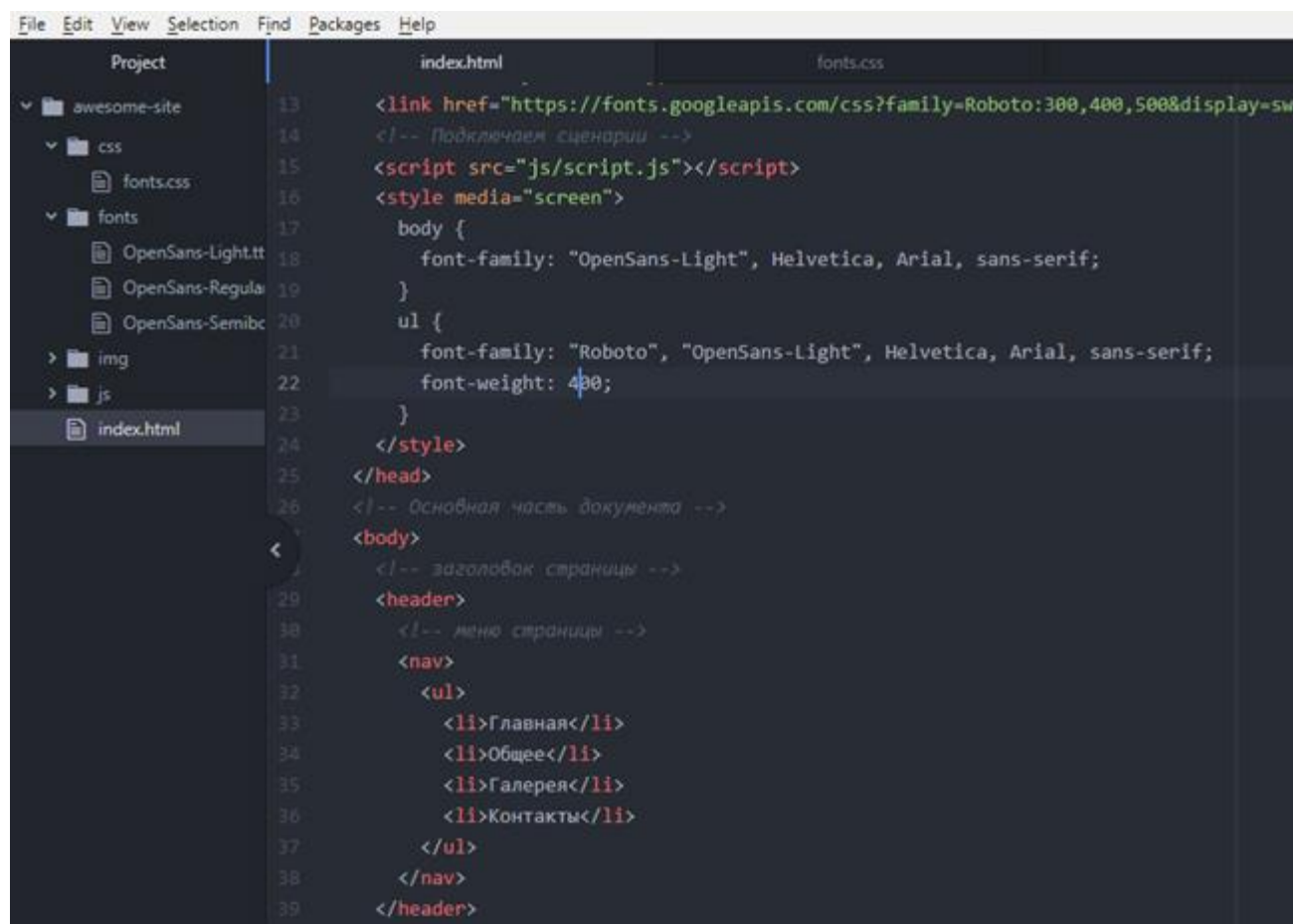


Рис. 7 – Подключение внешних шрифтов и создание меню

Обновите страницу, и вы заметите, что меню и приветствие немного отличаются. Добавим ещё немного контента, чтобы увидеть разницу между шрифтами.

Внутри тега `section` добавим тег заголовка `<h1>` и какой-то текст. Например, `<h1>Самый большой заголовок</h1>`. Под ним добавим тег `<p>` – это тег

параграфа для обычного текста. Чтобы наполнить его текстом можно ввести ключевое слово `lorem` и нажать клавишу `Tab`. Большинство редакторов введут несколько предложений из слов на латыни. На самом деле это не латынь, а «текст-рыба» – бессмысленный набор слов для визуализации вёрстки: то, что нам нужно.

Повторите эти же действия для тега `aside`, только в нём создайте в качестве заголовка тег `h3`.

```
35      <!-- основной контент сайта -->
36      <main>
37          <section>
38              <h1>Самый большой заголовок</h1>
39              <p>Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor
40          </section>
41          <aside>
42              <h3>Заголовок боковой панели</h3>
43              <p>Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor
44          </aside>
45      </main>
46      <!-- подвал страницы -->
47      <footer>...</footer>
```

Рис. 8 – Заполнение основного контента страницы

Сохраните изменения и обновите страницу. Как вы видите текст отображается в различном виде. Так примерно начинал свой путь HTML. Мы задали лишь стили шрифта, но не регулировали отступы, толщину и размер текста. Однако, браузер это сделал за нас. Всё потому, что в браузере уже заложены стили для заголовков, параграфов и других стандартных элементов HTML. Что касается тегов `section`, `aside`, `main` или `footer`, то они внешне не создают никакой разницы. Это семантические теги и их «понимают» роботы поисковых машин, что индексируют страницы в сети.

- Главная
- Услуги
- Галерея
- Контакты

Самый большой заголовок

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Заголовок боковой панели

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Рис. 9 – Внешний вид контента, стилизованного браузером

Заголовки в стандарте HTML следуют от h1 до h6. Где h1 – это самый большой и главный заголовок, а h6 самый маленький и наименее важный. Важность заголовков определяет веб-дизайнер или СЕО-специалист. Но хорошим тоном считается использовать только один заголовок h1 для страницы, который описывает её главную тему, для всех остальных заголовков использовать другие теги.

Неплохое начало. На социальную сеть ещё не похоже, но для портфолио на сайте для фрилансеров уже почти годится. Осталось добавить фото и несколько персональных стилей. О них поговорим в следующей лабораторной работе.

Задание на контроль

1. Повторить пример их практической части.
2. Для страницы примера должен быть создан каталог проекта и добавлен файл главной страницы.
3. Объяснить, как были подключены шрифты к проекту.
4. Добавить несколько пунктов в основное меню страницы.
5. Объяснить, как был стилизован шрифт на странице.
6. Объяснить, как стилизованы заголовки и параграфы на странице веб-браузера.
7. Лабораторная работа считается защищенной, если студент показал в окне браузера страницу из практической части с меню и ответил правильно на все заданные контрольные вопросы (следующая лабораторная работа не подлежит защите до тех пор, пока не будет защищена предыдущая).