

Кейс 1: Основы веб-разработки

Описание проблемы: вы только приступаете к изучению веб-разработки. На данном курсе вы будете изучать часть веб-разработки, которую называют фронтендом, т.е. визуальной частью.

В ходе выполнения данного кейса вам необходимо создать структуру будущего проекта, развернуть среду для разработки, а в конце создать свой первый header.

Краткая инструкция к выполнению кейса:

1. Создайте на рабочем столе папку **myfirstproject.com**
2. Внутри папки проекта создайте следующие папки:

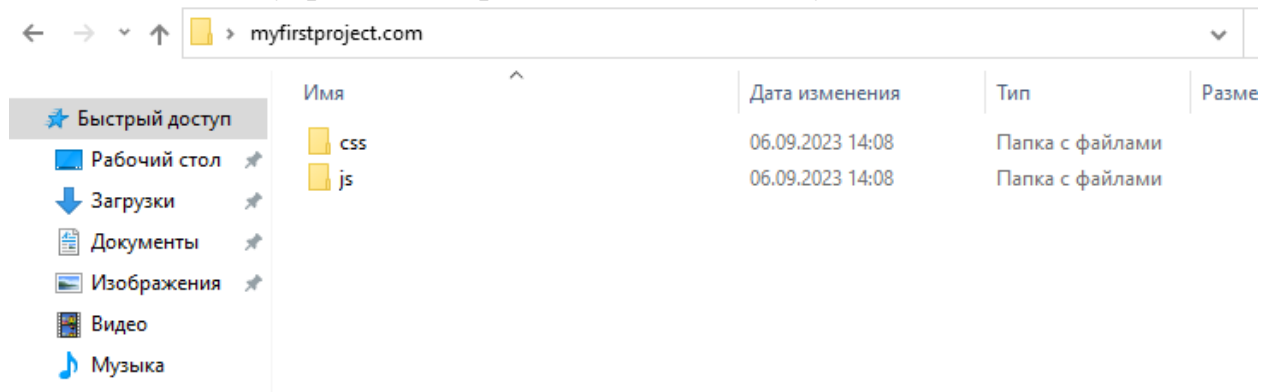


Рисунок 1- первичный вариант структуры проекта

3. В папке **css** мы будем хранить всё, что связано со стилями. Там же, как правило, создаются папки **img** для хранения картинок, **fonts** для хранения шрифтов. **Создайте эти папки самостоятельно внутри папки css.** В папке **js** хранятся все созданные скрипты, написанные на языке JavaScript. Они позволяют оживить интерфейс, сделав его динамическим, а также реализовать более сложные анимации или взаимодействия с интерфейсом.
4. Откройте **Visual Studio Code**. Для начала работы с веб-разработкой необходимо настроить данную программу, установив плагины, ускоряющие разработку. Изучите следующие элементы интерфейса, находящиеся в левой части программы и найдите кнопку, отвечающую за Extensions (плагины, дополнения):

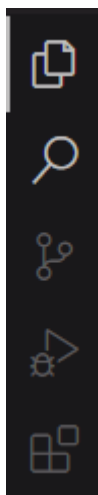


Рисунок 2 – элементы интерфейса программы VS Code

5. Установите следующие плагины:
 - Live Server
 - CSS Peek
 - VS code Icons
 - Prettier
 - GitHub Pull Request
6. Изучите, что делает каждый из плагинов.
7. Перейдите в **Explorer**, либо нажмите **File – Open Folder**. В качестве проекта выберите папку, которую создали на рабочем столе. В саму папку при выборе заходить не нужно. Обратите внимание, что для каждого действия есть горячие клавиши, рекомендуется их выучить для ускорения разработки.
8. Создайте новый пустой файл. Для этого нажмите горячие клавиши **CTRL+N**. Назовите его **index.html**.
9. Название **index.html** является общепринятым в сообществе разработчиков, что означает – данный файл является главной(стартовой) страницей вашего будущего веб-приложения.
10. Внутри файла введите **!** и нажмите **Enter**. Программа за вас сгенерирует основную структуру HTML документа. Внимательно изучите и запомните её.

```

5 index.html > ...
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <title>Document</title>
7  </head>
8  <body>
9
10 </body>
11 </html>
12

```

Рисунок 3 – Основная структура HTML-документа

Каждый HTML-документ начинается с декларации типа документа или «доктайпа». Тип документа необходим, чтобы браузер мог определить версию HTML и правильно отобразить страницу.

```
<!DOCTYPE html>
```

Простейшая HTML-страница состоит как минимум из трёх тегов: `<html>`, `<head>` и `<body>`. Тег `<head>` обычно содержит заголовок, ключевые слова, описание страницы и другие служебные данные. Также внутри него подключаются внешние ресурсы, например, стили. Содержимое этого тега не отображается на странице напрямую. А в теге `<body>` хранится содержание страницы, которое отображается в окне браузера.

Для подключения стилей к странице существует тег `<link>`. Для этого у него есть атрибут `href` в котором задаётся адрес стилевого файла, а значение `stylesheet` атрибута `rel` говорит браузеру, что мы подключаем именно стили, а не что-то другое.

```

<head>
    <link href="адрес_файла_стилей.css" rel="stylesheet">
</head>

```

Ещё один элемент, который располагается в `<head>` — это тег `<title>`. В нём задаётся заголовок страницы, который отображается во вкладках браузера. По заголовку должно быть понятно, о чём эта страница, даже когда она не открыта в браузере, а отображается в результатах поиска или в браузерных закладках.

```

<head>
    <title>Первая HTML страница</title>
</head>

```

Ещё один важный тег, располагающийся внутри `<head>` это тег `<meta>`. Он одиночный, то есть не требует парный закрывающий тег в конце. С помощью `<meta>` можно сообщать браузеру, поисковому роботу или другому устройству различную служебную информацию (или метаинформацию) о вашем сайте: кодировку текста, описание контента и так далее. Для этого используются теги `<meta>` с разными атрибутами и их значениями.

Кодировка текста HTML-страницы указывается с помощью атрибута `charset`:

```
<meta charset="название кодировки">
```

Самая распространённая современная кодировка — `utf-8`.

Перечень ключевых слов задаётся тегом `<meta>`, у которого атрибут `name` имеет значение `keywords`. Ключевые слова (самые важные слова из содержания страницы) перечисляются в атрибуте `content` через запятую:

```
<meta name="keywords" content="важные, ключевые, слова">
```

Краткое описание (или аннотация) страницы задаётся похожим образом, только значение атрибута `name` меняется на `description`:

```
<meta name="description" content="краткое описание">
```

Внутри `<body>` находятся те теги, которые отображаются на странице. Например, тег `<main>` выделяет основное содержание страницы, которое не повторяется на других страницах. И обычно на странице используется один `<main>`.

Тег `<header>` содержит вводную часть страницы, которую чаще называют «шапкой», а тег `<footer>` описывает заключительную часть страницы, или «подвал». Существует тег `<section>`, который обозначает крупный смысловой (или «логический») раздел.

Тег `<article>`, обозначает цельный, законченный и самостоятельный фрагмент информации.

Для создания логического раздела с основной навигацией предназначен тег `<nav>` (сокращение от английского «navigation»). Обычно в `<nav>` включают ссылки на другие страницы или навигацию по текущей странице.

Тег `<aside>` включает в себя дополнительное содержание, не связанное напрямую с основным. Такие блоки ещё часто называют «сайдбарами» или боковыми панелями.

Для создания основной структуры текста используют заголовки. В HTML существует целое семейство заголовочных тегов: от `<h1>` до `<h6>`. Тег `<h1>` обозначает самый важный заголовок (заголовок верхнего уровня), а тег `<h6>` обозначает подзаголовок самого нижнего уровня.

```
<h1>Спецификация HTML</h1>
```

```
<h2>Раздел 1 Введение</h2>
```

```
<h3>Раздел 1.1 Происхождение языка</h3>
```

Основную структуру текста создают с помощью заголовков, а более мелкую выстраивают с помощью параграфов (или абзацев). Для разметки параграфов предназначен тег `<p>`. По умолчанию абзацы начинаются с новой строки и отделяются от остального контента отступами сверху и снизу.

ЗАДАНИЯ:

1. Создайте в теге `body` шапку `header`.
2. Создайте в папке `css` файл `main.css`.

Чтобы подключить стили к html документу, воспользуйтесь следующей командой (вписывается внутри тега `head`):

```
<link rel="stylesheet" href="../css/main.css">
```

Где `href` – ссылка на файл стилей.

3. Для написания стиля к тегу или классу используется следующий синтаксис:

```
4.  h4 {
5.      text-align: left;
6.      font-size: 32px;
7.      margin-top: 100px;
8.      margin-left: 100px;
9.      background: aliceblue;
10.     border-radius: 10px;
11.     color: #001C30;
12.     width: 1000px;
13.     padding: 10px;
14. }
```

Указывается тег или класс в файл с расширением `.css`, после чего открываются фигурные скобки, в каждой новой строке указывается

свойство, а через двоеточие его значение. В конце строки свойства используется точка с запятой.

Список свойств можно посмотреть здесь:

<https://msiter.ru/references/css-reference>

4.Ознакомьтесь с величинами, которые используются в CSS.
(<https://learn.javascript.ru/css-units>)

5.Создайте у тега header свойства, чтобы он занимал весь экран устройства. Внутри тега header создайте заголовок, в котором напишите приветствующий слоган.

6*. Выровняйте заголовок по центру экрана.