

## Практическая работа № 2. Структура web-страницы и основные теги разметки. Файловая структура web-сайта

*Цель работы:* познакомиться с основными инструментами, применяемыми для верстки веб-страниц, изучить базовые html-теги разметки

### 1 Как спроектировать сайт-портфолио

#### Файловая структура сайта-портфолио

Как создать сайт-портфолио? Ответ на этот вопрос — взять текстовый редактор и с нуля написать сайт! Звучит просто, *не правда ли?* Для вас уже подготовлен стартовый шаблон сайта-портфолио. Начиная с этого домашнего задания, будем потихоньку его усовершенствовать и наполнять контентом. На первом занятии мы познакомимся с шаблоном сайта-портфолио и его структурой.

Для начала скачайте архив с сайтом [отсюда](#) и распакуйте (рис.1).

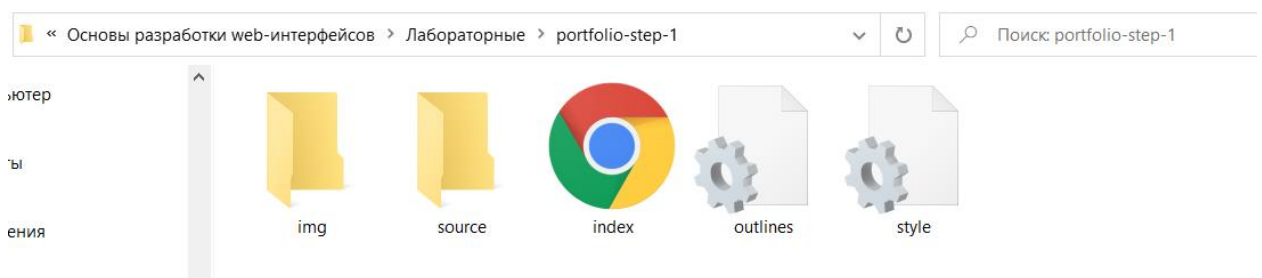


Рисунок 1. Вот что находится в распакованном архиве сайта

Вот что шаблон содержит:

- главную страницу-лендинг `index.html` с информацией о вас, ваших навыках и проектах;
- стили для этой страницы `style.css` (файл пока что пустой, мы будем наполнять его в следующем разделе);
- папку `img` с интерфейсными картинками для портфолио: фоном, иконками и декоративными элементами;
- папку `files` с контентными картинками для портфолио: фотографией и скриншотами проектов;
- папку `source` с примерами выполненных вами проектов.

На этом этапе мы включили в файлы сайта ещё и стили для прототипирования страницы `outlines.css`, они помогают наглядно видеть структуру страницы (на их содержимое можете пока что не обращать внимание).

## Задание 1. Цель задания, научиться самостоятельно создавать файловую структуру сайта (выполняется в аудитории)

1. Создаем на любом из жестких дисков новую папку.
2. Далее заходим в эту папку, щелкаем правой кнопкой мыши, выбираем пункт «создать текстовый документ» с расширением «.txt».
3. Переименовываем этот файл, даем ему любое имя, а расширение меняем на «.html»
4. Щелкаем по файлу правой кнопкой мыши, выбираем пункт «открыть с помощью», выбираем текстовый редактор и открываем.
5. Напишем теперь первое приветствие миру. Переходим в текстовый редактор и набираем:

I'm here

Чтобы посмотреть результат в браузере — дважды кликаем по этому файлу.

## 2. Следующий этап – разметка веб-документа для размещения контента на страницах сайта

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>Тренажёры — HTML Academy</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 метайнформацию) 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.

## Задание 2. Разработка базовой структуры индивидуального web-проекта

Перейти по ссылке, зарегистрировать свой аккаунт на платформе <https://htmlacademy.ru> и выполнить первые 12 заданий по теме “Основы языка разметки HTML5”. Результат прикрепить в отчет по работе в виде скриншота, на котором виден аккаунт и статистика прохождения этого раздела в тренажерах. Свой id на сайте [htmlacademy.ru](https://htmlacademy.ru) также разместить в качестве подписи к скриншоту

## Структура страницы сайта-портфолио

С файловой структурой определились, теперь давайте разберёмся со структурой страницы.

Откроем файл `index.html` в текстовом редакторе (рис.2).

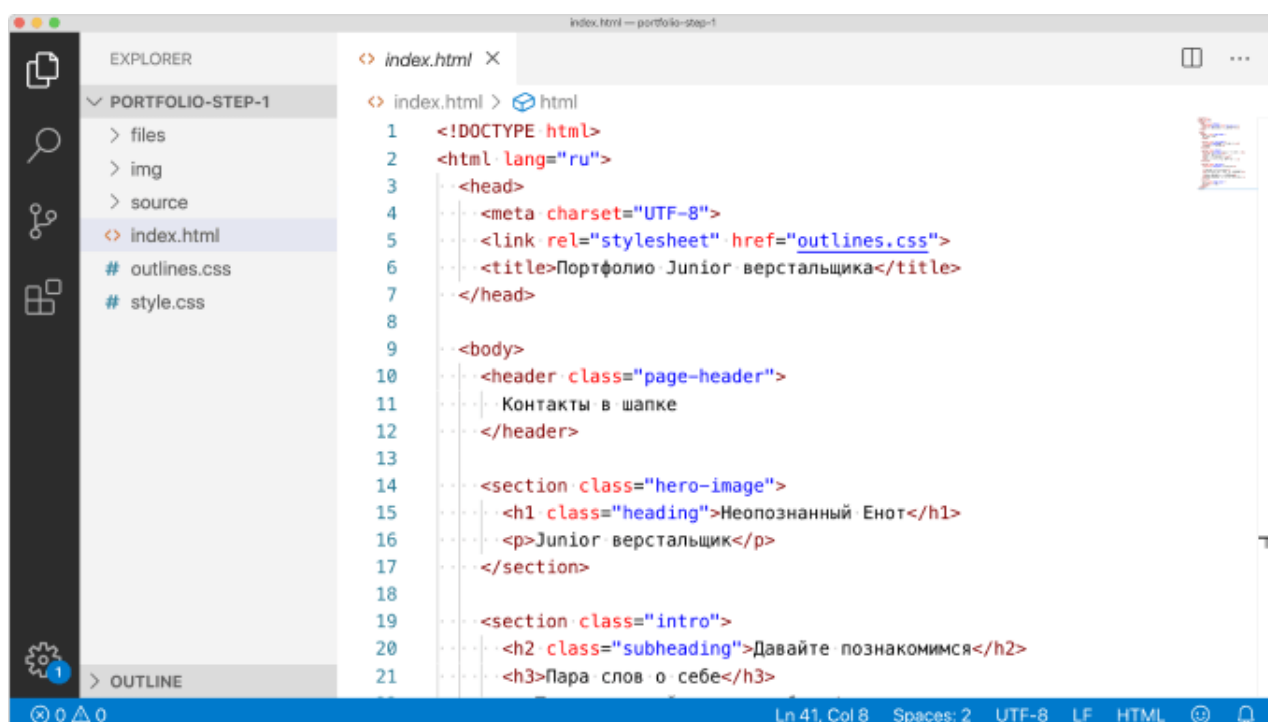


Рисунок 2 – Код главной страницы в редакторе

В разметке уже созданы информационные разделы, которые вы можете заполнить по вашему усмотрению:

- шапка сайта — тут можно разместить ваши контакты;
- вступительный раздел — здесь написаны ваше имя и профессия;
- текстовый раздел — тут вы можете развернуто рассказать о себе, здесь же позже подключим вашу фотографию;
- ещё один текстовый раздел — здесь будут размещены примеры ваших работ с коротким описанием и картинками;
- подвал сайта — тут тоже могут быть ваши контакты.

Сфокусируйтесь пока что на текстовом контенте сайта, оформление и картинки мы подключим позднее.

### **Задание 3. Создание страницы одного проекта для портфолио**

1. Установить на домашнем компьютере вышеуказанные инструменты
2. Создать в удобном для вас месте на жестком диске папку, где будет храниться ваш проект (сайт-портфолио)
3. Выбрать тематику сайта
4. Создать html-страницу элемента каталога, по вашей тематике
5. На странице должны присутствовать:
  - 5.1. структура HTML документа
  - 5.2. название элемента каталога проектов
  - 5.3. заголовок «Описание проекта»
    - 5.3.1. картинка элемента каталога
    - 5.3.2. текст краткого описания элемента каталога
  - 5.4. Заголовок «Характеристики проекта»
    - 5.4.1. Текст характеристик проекта, который вы выбрали
  - 5.5. Заголовок «Подробное описание проекта»
    - 5.5.1. Текст подробного описания проекта
  - 5.6. Горизонтальная черта
  - 5.7. Текст «Все права защищены»
6. Выучить комбинации горячих клавиш
7. Добавить произвольные подразделы описания элемента каталога, например, описание интерфейса, или особенностей проекта
8. **\*Добавить 2 страницы элементов каталога по вашей тематике.**

Задачи со \* предназначены для продвинутых студентов, которым мало сделать обычное ДЗ.

Задание размещается на github (создайте специальный репозиторий для домашних заданий и проектов по данному курсу) или в вашем облачном хранилище. Результат прикрепляется в виде ссылки на эту работу, доступной для просмотра для всех

### **Вспомогательные материалы**

[Irina-64/web-interfaces: материалы курса "Основы web-интерфейсов" \(github.com\)](#)

<https://htmlacademy.ru/courses/297>

The h1-h6 elements — <https://w3c.github.io/html/sections.html#the-h1-h2-h3-h4-h5-and-h6-elements>

Common idioms — <https://w3c.github.io/html/common-idioms-without-dedicated-elements.html#common-idioms-without-dedicated-elements>

Hiding DOM elements — <https://allyjs.io/tutorials/hiding-elements.html>

Курс «Разметка текста» — <https://htmlacademy.ru/courses/38>

<https://www3schools.com>

<https://htmlacademy.ru/blog/boost/tools/git-and-github-glossary>

<https://htmlacademy.ru/blog/boost/frontend/git-console>

<https://htmlacademy.ru/blog/boost/frontend/first-aid-git>

<https://htmlacademy.ru/blog/boost/frontend/doctype>

<https://htmlacademy.ru/blog/boost/frontend/short-7>