**1. Создание HTML страницы**

Данная практическая работа посвящена созданию самой просто web-страницы.

1. Структура HTML-документы

Файлы html имеют расширение **.htm** или **.html**.

Содержат один тип управляющих конструкций – *теги* (tags) – *дескрипторы*. Все теги  заключены в угловые скобки < >.

Обычно теги парные: начальный и конечный теги. Начальный – <>, конечный </>, например <h1> и </h1>

Парный тег называют еще ***контейнерный***.

Некоторые теги не имеют обязательного парного тега, такой тег называют ***одиночный***. Закрывается такой тег написанием правого слеша перед закрывающей угловой скобкой, например <br />

Структура html-документа выглядит следующим образом (рисунок 1).

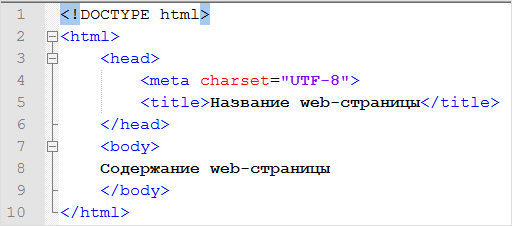


Рисунок 1 — Структура HTML документы

Пояснение к листингу на рисунке 1.

В первой строке конструкция DOCTYPE означает один из стандартов верстки web-страницы.

Далее весь HTML-документ заключен между парным тегом **<html>** и **</html>.** Это обязательный тег.

Между тегами **<head>** и **</head>** содержится заголовочная часть. В нее входит конструкция **<title>** и **</title>.** Это название при загрузке документа появляется в строке заголовка браузера (рисунок 2).

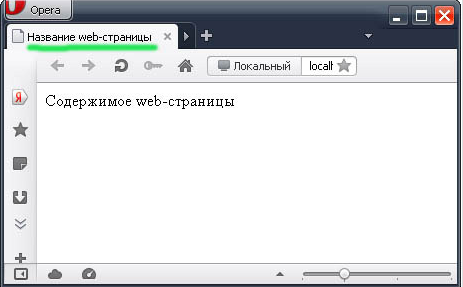


Рисунок 2 —Название HTML страницы

Запись **<meta charset=UTF-8 />**означает кодировку кириллицы, чтобы буквы корректно отображались в браузере. Если кодировку не прописывать, шрифты иногда могут отображаться следующим образом (рисунок 3).

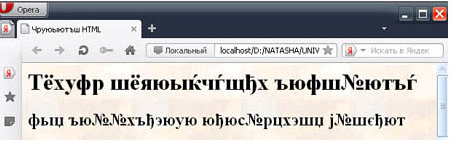


Рисунок 3 — Пример отображения шрифта без прописывания кодировки

Та же страничка после вставки кодировки (рисунок 4)

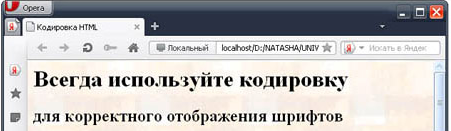


Рисунок 4 — Пример отображения шрифта с прописанной кодировкой

**Кодировка *charset="UTF-8"*позволяет создавать мультиязычные сайты, так как все существующие в мире символы в ней присутствуют. В этом случае в Notepad++ необходимо до верстки и сохранения html-файла выбрать из верхнего меню *Кодировки – Кодировка в UTF-8 с BOM***

**Практическое задание 1**

1. Создайте папку, назовите ее **myhouse**. Имена всех папок и файлов необходимо создавать латинскими строчными буквами, **русские названия использовать нельзя**. Задавайте имена короткими и смысловыми.

Код web-страницы будем писать в Блокноте. Лучше пользоваться не тем Блокнотом, который идет в поставке Windows, а более «продвинутым», например Notepad++. Дистрибутив **Notepad++** есть в папке **CD/distr**.

2. Откройте Notepad++. Убедитесь, что установлена кодировка UTF-8 с BOM для этого в меню в пункте **Кодировки**, установите позицию **Кодировка UTF-8 с BOM**.

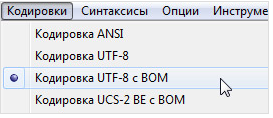


Рисунок 5—Выбор кодировки (в Notepad++)

3. Наберите в Notepad++ код из листинга на рисунке 1.

Код в листинге на рисунке 1 в дальнейшем будет служить нам шаблоном, чтобы каждый раз не набирать структуру html-документа.

4. Сохраните файл под именем **shablon.html**в папке **myhouse**, при этом в поле **Тип файла** установите **All types**(рисунок 6), иначе Ваша web-страничка потом не откроется в браузере.

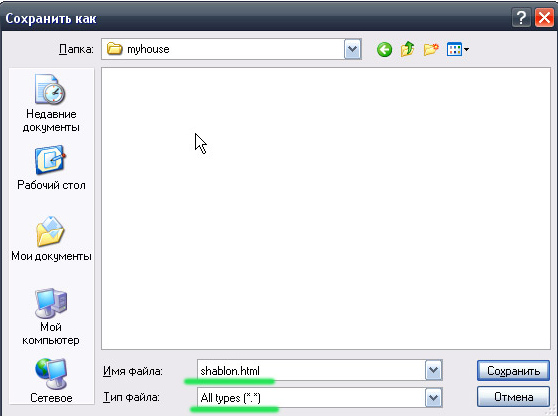


Рисунок 6—Создание и сохранение шаблона страницы

5. После сохранения запустите **shablon.html** двойным щелчком. В результате Ваш файл будет выглядеть следующим образом (рисунок 7).

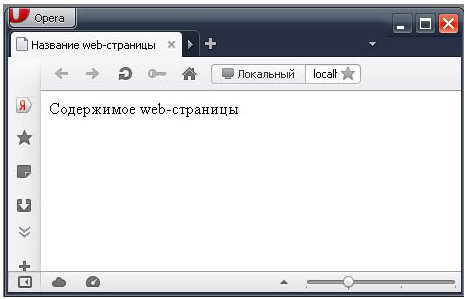


Рисунок 7

Если у Вас не получился результат, как на рисунке выше, значит, в коде скорей всего есть ошибка. Достаточно одного неверного символа и web-страница не будет отображаться корректно. Еще раз сверьте код с листингом на рисунке 2.

6. Внутри папки **myhouse** создайте папку **public\_html**. Таким именем обычно называется папка, в которой хранится сайт при размещении на настоящем хостинге (также эта папка может называться www).

7. Сохраните файл **shablon.html** в папке **public\_html** под новым именем **main.html**.

8. Из папки **CD/html\_css\_1**откройте файл **text\_main.txt** в Notepad++, а также откройте файл, сохраненный под именем **main.html**

9. Скопируйте весь текст из файла **text\_main.txt**и вставьте в файл **main.html**вместо фразы «Содержимое web-страницы». В теге <title> напишите слово «Главная». Вот так **<title>Главная</title>.**

10. Сохраните изменения и просмотрите файл **main.html**в браузере. Вы увидите неотформатированный текст. Даже переносы строк, которые есть в исходном тексте, браузер не делает (рисунок 8).

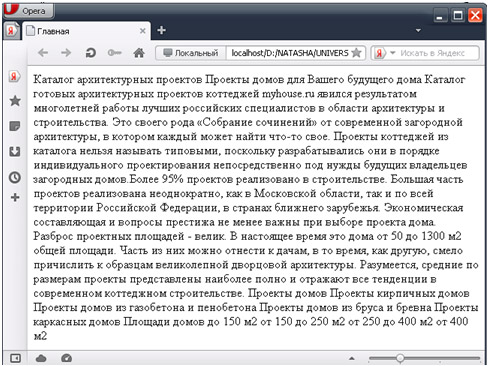


Рисунок 8

**2. Форматирование web-страницы тегами HTML**

В данном уроке «Как создать HTML-страницу» рассматриваются основные теги, которые используются для разметки веб-страницы.

Теги предназначены для форматирования текста web-страницы. Список тегов более подробно можно посмотреть на ресурсе **http://htmlbook.ru/**

Рассмотрим некоторые из тегов.

***Элементы h1, h2, h3, h4, h5, h6***

Структурирование тела документа выполняется внутри элемента **<body>** с помощью заголовков, задаваемых элементами h1, h2, h3, h4, h5, h6.

Элементы заголовков являются парными, поэтому должны имеет открывающий **<h1>**и закрывающий **</h1>** теги.

HTML располагает шестью уровнями заголовков: h1 (самый верхний), h2, h3, h4, h5 и h6 (самый нижний). Функции элементов заголовков подобны обычным стилям заголовков в текстовых редакторах.

Действие этих шести тегов представлено на рисунках ниже. На одном рисунке исходный код (рисунок 9), на другом – вид в браузере (рисунок 10).

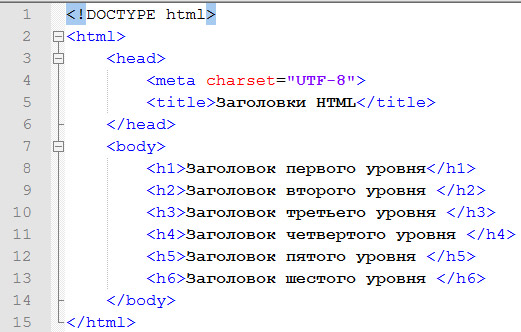


Рисунок 9

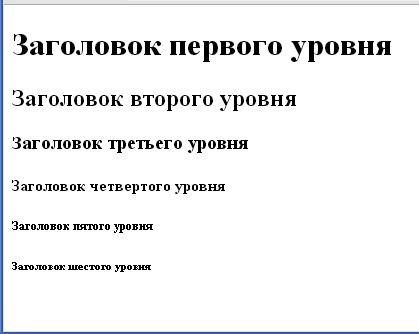


Рисунок 10

***Разделение текста на абзацы***

Тег **<p>** задает начало абзаца и вставляет сверху абзаца расстояние – отступ для отделения этого абзаца от предыдущего.

***Принудительный разрыв строки***

Тег **<br />** позволяет выполнить перенос оставшейся части текста абзаца на следующую строку. Это непарный тег и в отличие от тега абзаца не увеличивает интервал между строками.

**Практическое задание 2**

1. Отформатируйте заголовок «Каталог архитектурных проектов» с помощью тегов <h1> и </h1>.

2. Отформатируйте заголовок «Проекты для Вашего будущего дома» с помощью тегов <h2> и </h2>.

3. Отформатируйте заголовки «Проекты домов» и «Площади домов» с помощью тегов <h3> и </h3>.

4. Основной текст разделите на абзацы с помощью тега <p>. Внутри первых двух абзацев используйте тег <br /> для переноса строки. В результате Ваш код должен выглядеть следующим образом (рисунок 11).

5. Просмотрите результат в браузере.

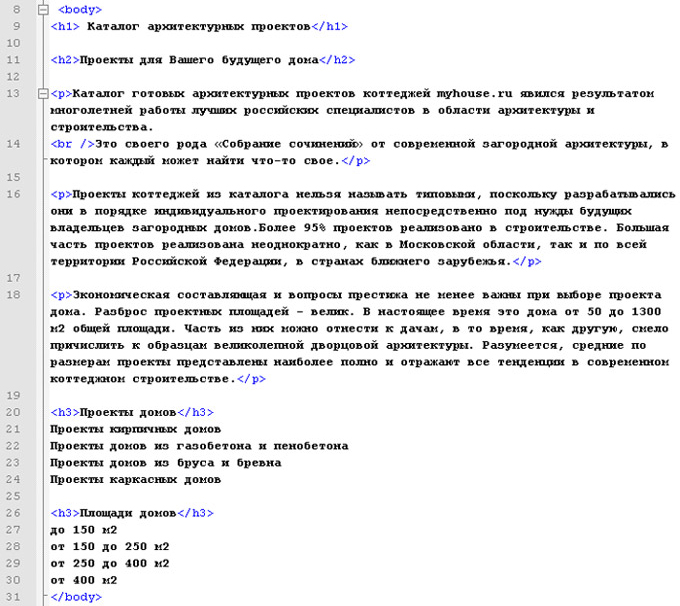


Рисунок 11

***Маркированные и нумерованные списки***

Средствами HTML можно создавать любые списки: нумерованные (цифровые и буквенные) и маркированные с разными типами маркеров.

Тег **<ul>…</ul>** формирует маркированный список.

Тег **<ol>…</ol>** формирует нумерованный список.

Отдельный элемент списка как в  <ul>, так и в <ol> формируется с помощью тега **<li>**.

**Практическое задание 3**

1. Создайте нумерованный список под заголовком «Проекты домов».

2. Создайте маркированный список под заголовком «Площади домов». Ваш код будет выглядеть следующим образом (рисунок 12).

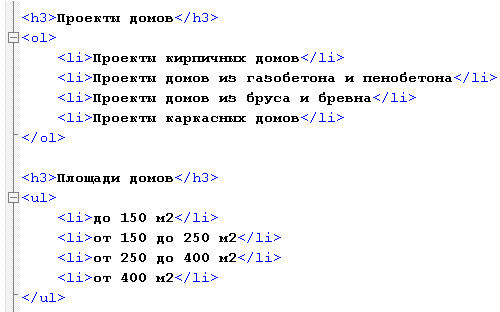


Рисунок 12

3. Просмотрите страницу в браузере.

Списки можно вкладывать друг в друга, используя при этом различные маркеры.

Пример вложенного списка приведен на рис. 13

**Практическое задание 4**

1. Реализуйте в новом файле код, приведенный на рисунке 13.
2. Сохраните файл в папке **myhouse** под именем **spisok\_vlozh.html.**Результат на рис. 13

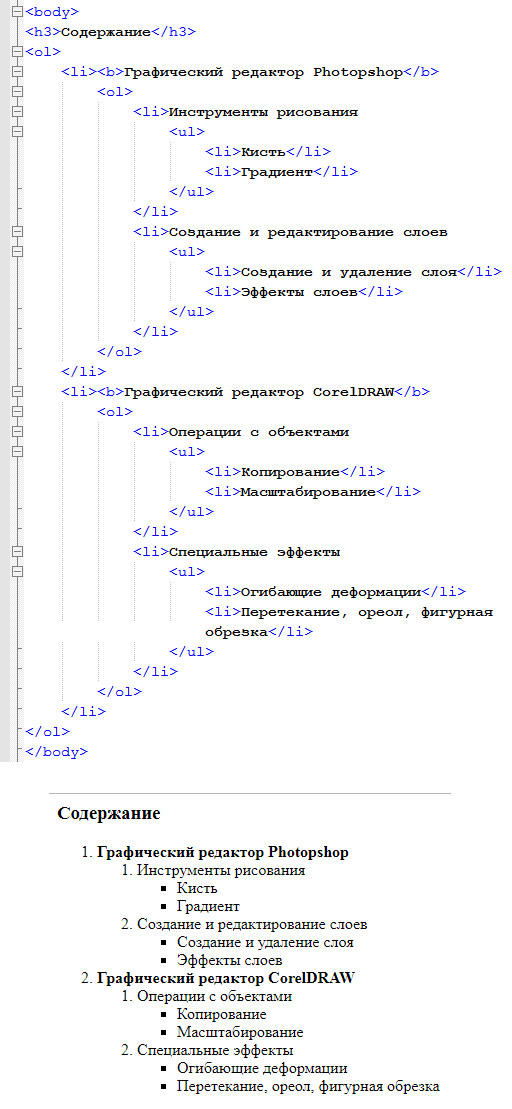


Рисунок 13. Пример вложенного списка

***Начертания шрифтов***

Тег **<b>…</b>** – позволяет отобразить текст полужирным шрифтом.

Тег **<i>…</i>** – позволяет отобразить текст в курсивном начертании.

Тег **<u>…</u>** – отображает подчеркнутый текст.

*Например:*

В этом случае текст будет отображен <b><i>***жирным курсивом***</i></b>, но не подчеркнутым.

А в этом случае текст будет написан <b><i><u>***жирным подчеркнутым курсивом***</u></i></b>.

***Нижние и верхние индексы***

Тег **<sub>** и **</sub>** позволяет опустить текст на полстроки ниже обычного текста.

Тег **<sup>** и **</sup>** позволяет поднять текст на полстроки выше обычного текста. Текст, расположенный между этими тегами, будет изображаться меньшим шрифтом, по сравнению с обычным текстом.

**Практическое задание 5**

1. Отформатируйте в первом абзаце название myhouse.ru полужирным шрифтом (рисунок 14).

http://web-verstka.ru/webverstka/kurs_html_css/images_html_css/1.14.jpg

Рисунок 14

2. Отформатируйте во втором абзаце фразу «Более 95% проектов» подчеркнутым курсивом (рисунок 15).

http://web-verstka.ru/webverstka/kurs_html_css/images_html_css/1.15.jpg

Рисунок 15

3. Отформатируйте верхние индексы в тех местах, где используются квадратные метры (рисунок 16).

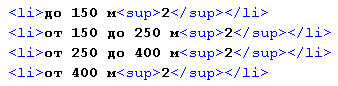


Рисунок 16

4. Сохраните файл. Просмотрите через браузер. Web-страница должна выглядеть следующим образом (рисунок 17).

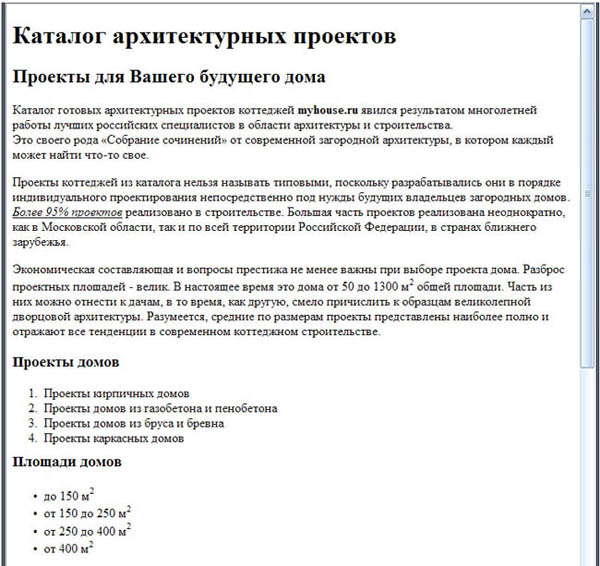


Рисунок 17

**3. Вставка изображений**

В нашей статье «Как создать HTML-страницу» мы рассмотрим, как размещать на веб-странице изображения.

Для вставки изображений используется тег **<img>.** Обязательным для этого тега является атрибут **src** (от английского SouRCe – источник). Он определяет путь до графического файла, изображение которого должно быть выведено на web-странице.

Для вставки изображения используется команда **<img src="Имя файла" />**

Например: **<img src="image1.jpg" alt="Картинка" />**

Атрибут alt необходим для того, чтобы при просмотре web-страницы в режиме отключенных изображений, вместо отсутствующей картинки была надпись, которая прописана в атрибуте alt. Также alt-тексты рекомендуется использовать всегда, т.к. поисковые машины анализируют их как ключевые слова при поиске изображений.

Графические файлы могут быть в формате **jpg, gif, png и обязательно в цветовой модели RGB.**

**Практическое задание 6**

1. После списка площадей домов вставьте в web-страницу изображения проектов коттеджей **project\_1.jpg** и **project\_2.jpg**из папки **CD/ html\_css\_1**. Для этого сначала скопируйте два этих изображения в ту папку, в которой расположен файл main.html. Код вставки изображений будет выглядеть следующим образом (рисунок 18).

http://web-verstka.ru/webverstka/kurs_html_css/images_html_css/1.18.jpg

Рисунок 18

2. Просмотрите результат в браузере (рисунок 19).

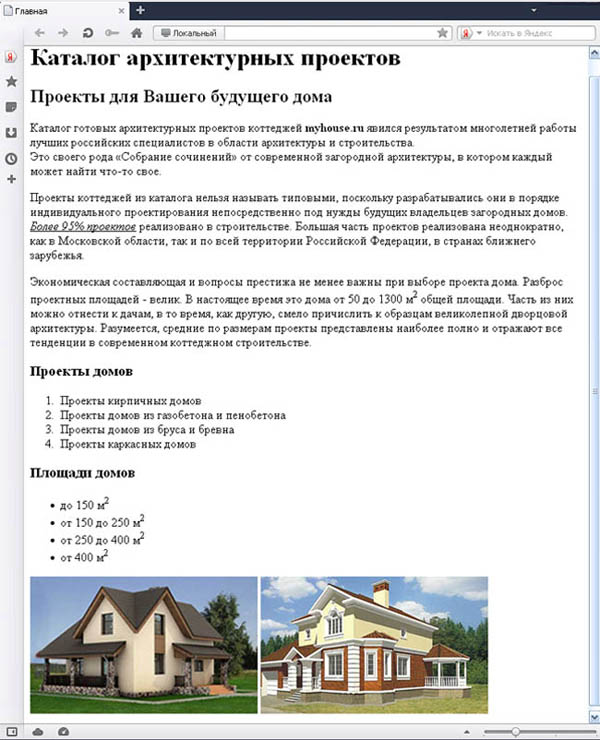


Рисунок 19

**В основном принцип использования тегов HTML должен быть понятен. Они используются по одному и тому же принципу: если тег контейнерный, то есть открывающий и закрывающий тег. Если тег одиночный, то закрывающего тега нет, в этом случает правый слэш пишется до закрывающей угловой скобки, например <br />. Является ли тег контейнерным или одиночным Вы всегда можете посмотреть в спецификации html401\_ru.chm в разделе «элементы».**

**4. Адресация внутри сайта**

В нашем первом уроке «Как создать HTML-страницу» мы разберем понятия адресации в HTML.

Рассмотрим варианты адресации, когда в html-файле надо разместить изображения, которые могут находиться в разных папках сайта.

Существует два вида адресации:

* абсолютная;
* относительная.

***Абсолютная адресация*** (с использованием названий дисков компьютера) **не используется**(рисунок 20)

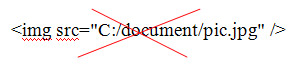


Рисунок 20

Используется ***относительная адресация*** – адресация в пределах документа или совокупности документов на одном сервере. Чтобы сослаться на файл внутри сайта, нужно указать браузеру, какой путь он должен проделать, что бы прийти к нужному файлу.

***Пример 1***

Изображение расположено в одной папке с HTML-документом (рисунок 21). То есть изображение и HTML-файл находятся на одном уровне в файловой системе и HTML-документ может сразу присоединить изображение. В этом случае вставка изображения в web-страницу будет выглядеть следующим образом.

**<img src="pic.jpg" />**



Рисунок 21

**Практическое задание 7**

Реализуйте пример 1 (создайте папку, файл doc.html, возьмите любое изображение). Изображение должно быть вставлено в web-страницу doc.html.

***Пример 2***

Изображение **pic.jpg** находится в папке **folder\_1**. HTML-документ расположен за пределами папки folder\_1. Т.е. в файловой системе сайта HTML-документ расположен на один уровень выше, чем изображение (рисунок 22). Необходимо войти в папку folder\_1, затем присоединить изображение

**<img src="folder\_1/pic.jpg" />**

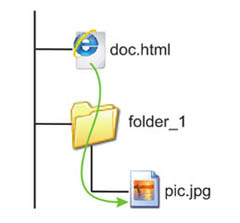


Рисунок 22

**Практическое задание 8**

Реализуйте пример 2 (создайте папку, файл doc.html, возьмите любое изображение). Изображение должно быть вставлено в web-страницу doc.html.

***Пример 3***

Изображение **pic.jpg** находится в папке **folder\_1**, которая в свою очередь лежит в папке **folder\_2**. HTML-документ расположен за пределами этих папок (рисунок 23). Таким образом, HTML-документ расположен на два уровня выше, чем изображение. Необходимо:

* войти в папку folder\_2,
* затем войти в папку folder\_1,
* затем присоединить изображение.

**<img src="folder\_2/folder\_1/pic.jpg" />**

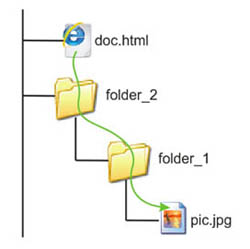


Рисунок 23

**Практическое задание 9**

Реализуйте пример 3 (создайте папки, файл doc.html, возьмите любое изображение). Изображение должно быть вставлено в web-страницу doc.html.

***Пример 4***

HTML-документ находится в папке **folder\_1**. Изображение находится за пределами папки folder\_1. Т.е. HTML-документ расположен на один уровень ниже, чем изображение (рисунок 24). Необходимо выйти из папки folder\_1, затем присоединить изображение. Выход из папки обозначается конструкцией **../**(две точки и слэш вправо).

**<img src="../pic.jpg" />**

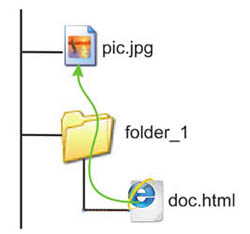


Рисунок 24

**Практическое задание 10**

Реализуйте пример 4 (создайте папку, файл doc.html, возьмите любое изображение). Изображение должно быть вставлено в web-страницу doc.html.

***Пример 5***

HTML-документ находится в папке **folder\_1**, которая в свою очередь находится в папке **folder\_2**. Изображение находится за пределами этих папок. Таким образом, HTML-документ расположен на два уровня ниже, чем изображение (рисунок 25). Необходимо:

* выйти из папки folder\_1,
* выйти из папки folder\_2,
* присоединить изображение.

Так как надо выйти два раза, то и конструкция **../** повторяется два раза.

**<img src="../../pic.jpg" />**

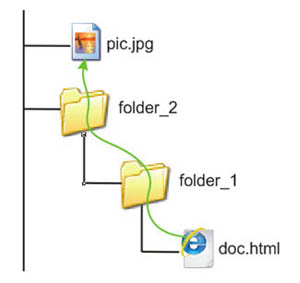


Рисунок 25

**Практическое задание 11**

Реализуйте пример 5 (создайте папки, файл doc.html, возьмите любое изображение). Изображение должно быть вставлено в web-страницу doc.html.

***Пример 6*** (рисунок 26)

Необходимо:

* выйти из папки folder\_1,
* выйти из папки folder\_2,
* зайти в папку folder\_3,
* зайти в папку folder\_4,
* присоединить изображение pic.jpg

**<img src="../../folder\_3/folder\_4/pic.jpg" />**

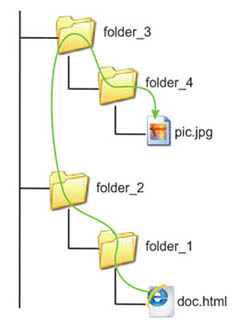


Рисунок 26

**Практическое задание 12**

Реализуйте пример 6 (создайте папки, файл doc.html, возьмите любое изображение). Изображение

должно быть вставлено в web-страницу doc.html.

***Пример 7*** (рисунок 27)

Необходимо:

* выйти из папки folder\_1,
* выйти из папки folder\_2,
* выйти из папки folder\_3,
* войти в папку folder\_4,
* присоединить изображение pic.jpg.

Так как выйти надо три раза, то и конструкция **../** повторяется три раза.

**<img src="../../../folder\_4/pic.jpg" />**

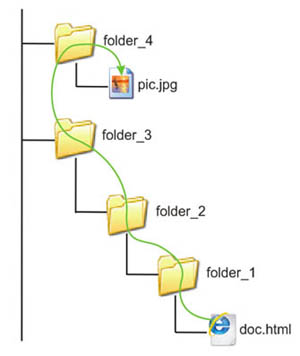


Рисунок 27

**Практическое задание 13**

Реализуйте пример 7 (создайте папки, файл doc.html, возьмите любое изображение). Изображение

должно быть вставлено в web-страницу doc.html.

**Таким образом, сколько раз надо выйти, столько раз ставим конструкцию ../, а если заходим, то перечисляем имена встречающихся на пути папок.**

**Практическое задание 14**

1. Создайте html-документ и вставьте в него изображение, чтобы путь к изображению был следующим.

**<img src="../../../../../../folder\_1/folder\_2/folder\_3/image.jpg" />**

2. Разработайте собственный пример вставки изображения в html-документ, чтобы он содержал и ***выход*** из папок, и ***вход*** в папки. Пример должен отличаться от всех вышеприведенных примеров.

**В результате выполнения этой темы, у Вас должны быть созданы следующие файлы:**

* **shablon.html**
* **main.html**
* **spisok\_vlozh.html**
* **внутри папки myhouse должна быть папка public\_html с файлами будущего сайта**
* **семь примеров на относительную адресацию и два примера в качестве контрольных из практического задания 14**