Университет ИТМО

Факультет «ИКТ»

Направление подготовки «программирование в инфокоммуникационных системах»

Лабораторная работа №1

«Введение в web-разработку»

Выполнил:

Сысовский В. С.

Группа №33201

Проверила:

Марченко Е. В.

Санкт-Петербург

2023

**Цель работы**: Познакомиться с основами создания web-страниц использующих HTML, стили и код JavaScript и не требующих обмена данными с сервером.

**Ход работы**:

**Упражнение 1.**

1. Установлен браузер Mozilla и редактор Notepad ++.

2. Набран код из описания к лабораторной работе, сохранён в файл и запущен через Firefox (рис. 1-2):

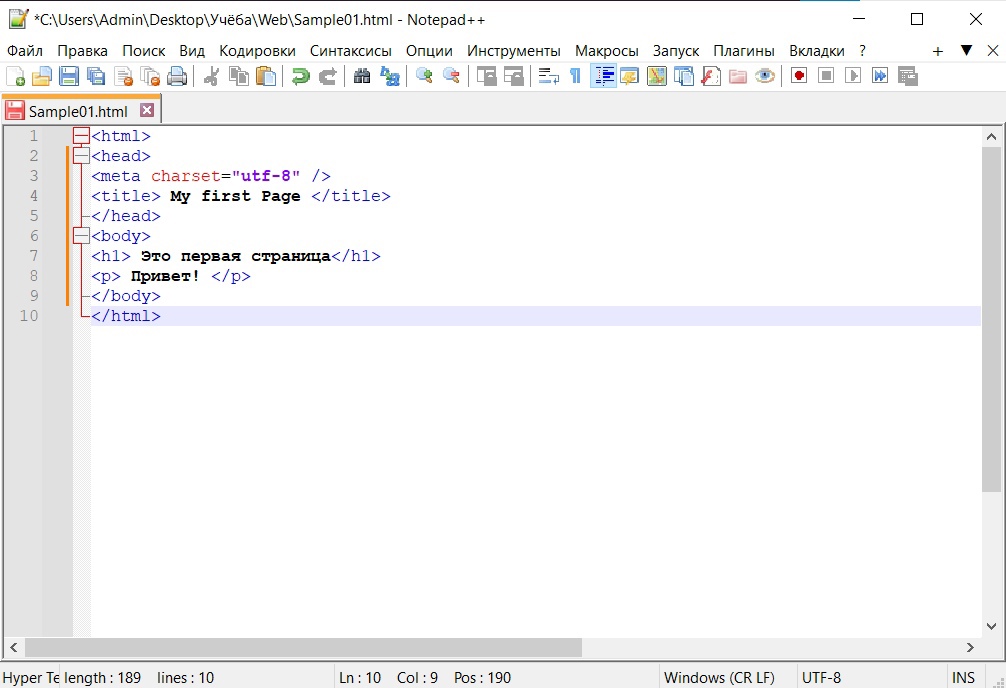


Рисунок 1 - код

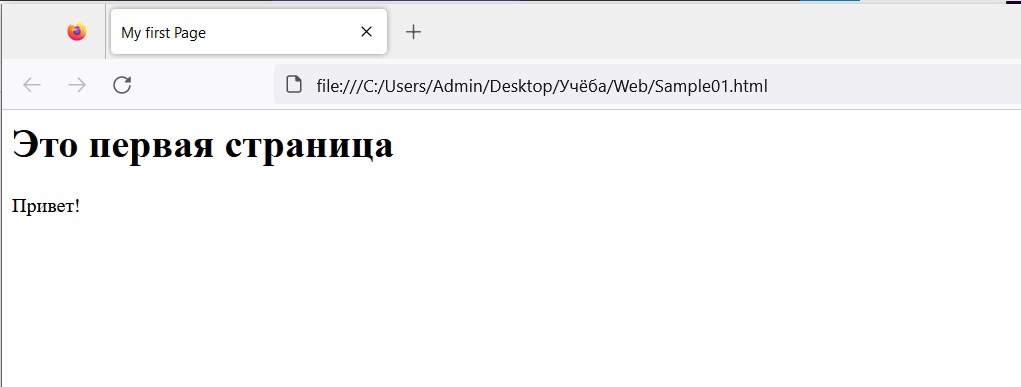


Рисунок 2 - результат запуска

**Упражнение 2.**

1. Добавлен тег style, в котором задан цвет страницы и цвет вводимого текста.

2. Файл сохранён и запущен (рис. 3-4):

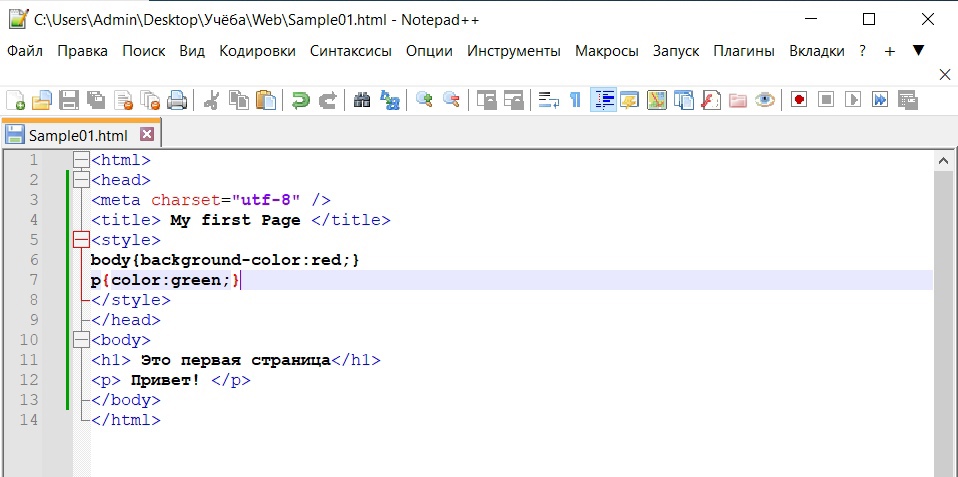


Рисунок 3 - код

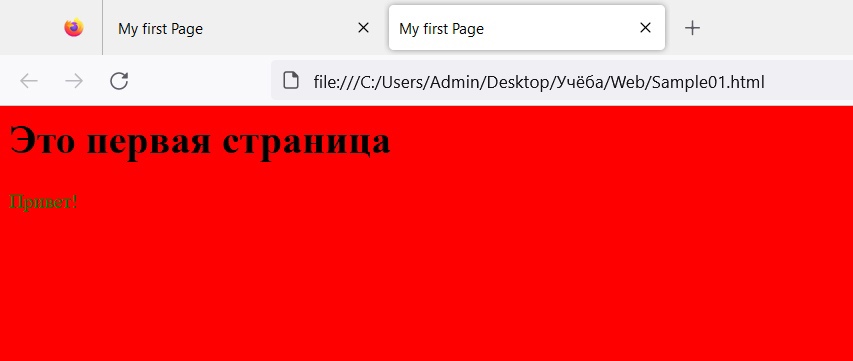


Рисунок 4 - результат запуска

**Упражнение 3.**

1. Между тегами <body> и </body> после закрывающегося тега </style> добавлен тег <script>.

2. После тега <script> добавлен метод alert(), который отображает сообщение со строкой Привет мир!

3. Файл сохранён и запущен (рис. 5-7):

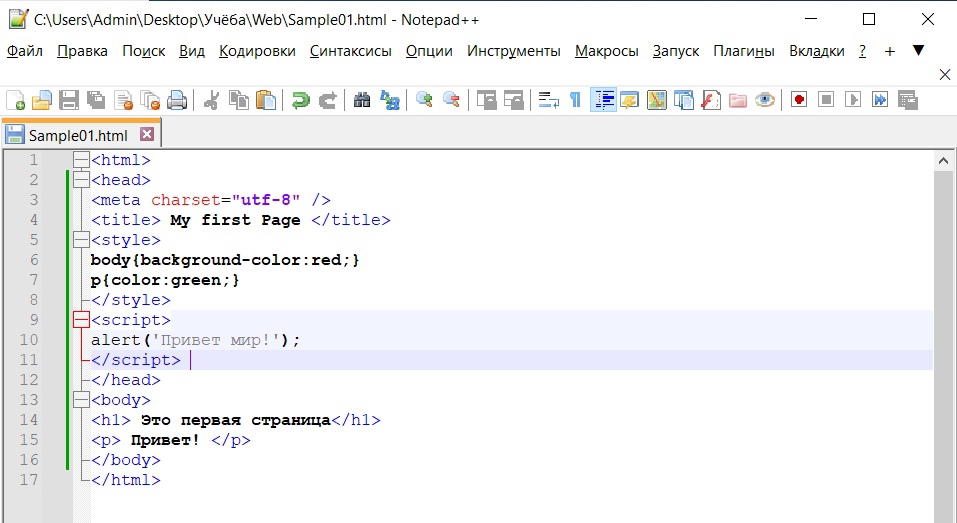


Рисунок 5 - код

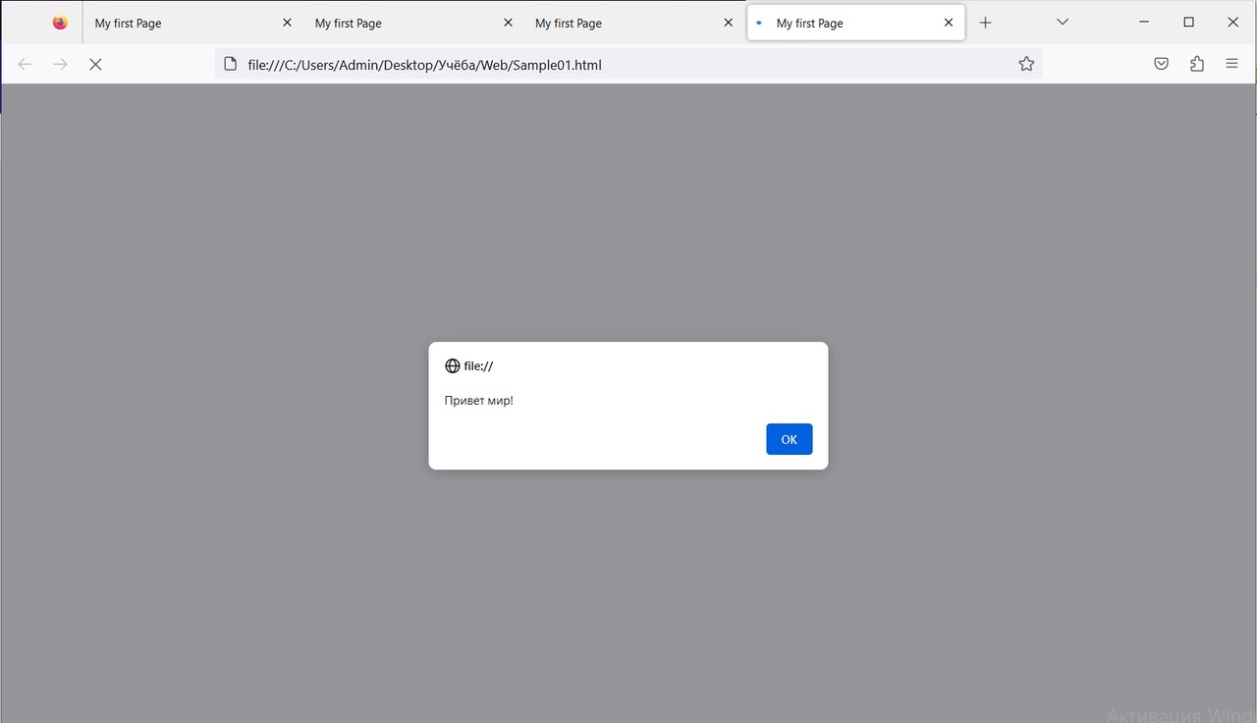


Рисунок 6 - результат запуска

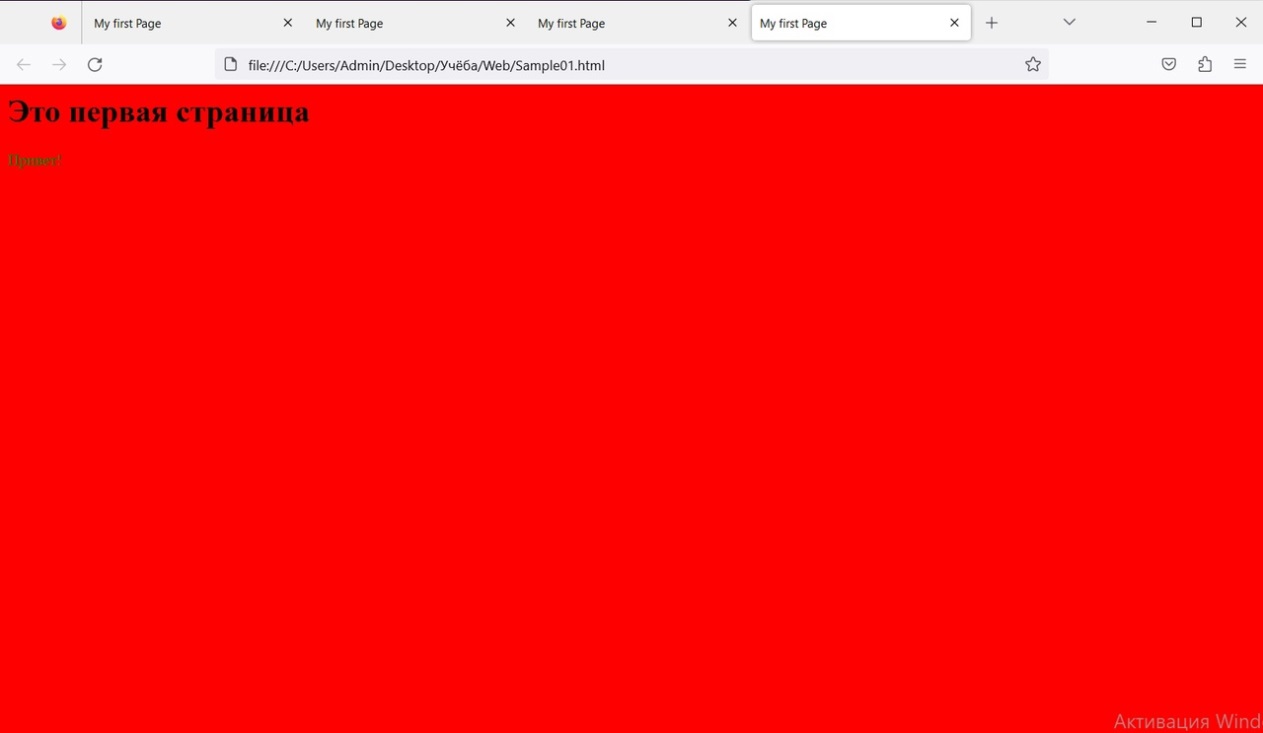


Рисунок 7 - результат запуска

**Вывод**: Изучены основы создания web-страниц использующих HTML, стили и код JavaScript и не требующих обмена данными с сервером.

**Ответы на вопросы:**

1) Какие объекты могут содержаться внутри документа HTML?

Текстовые блоки; текстовое форматирование; табличные формы; вставка картинок, аудио, текстовых и видеофайлов; гиперссылки.

2) Чем отличаются парные и непарные теги?

Парные состоят из двух тегов, открывающего и закрывающего, а непарные только из одного (<br> перевод строки, непарный).

3) Основные правила записи тегов и их атрибутов.

- Согласно спецификации HTML все значения атрибутов тегов следует указывать в двойных ("пример") или одинарных кавычках ('пример'). Отсутствие кавычек не приведет к ошибкам, браузеры во многих случаях достаточно корректно обрабатывают код и без кавычек, за исключением текста, содержащего пробелы.

- Теги можно писать как прописными, так и строчными символами.

- Внутри тега между его атрибутами допустимо ставить перенос строк.

- Если какой-либо тег или его атрибут был написан неверно, то браузер проигнорирует подобный тег и будет отображать текст так, словно тега и не было.

- Не забывать закрывать парные теги.

4) Какие теги определяют служебную и содержательную области документа HTML?

К служебным тегам относятся:

html

Элемент html — это первый служебный тег, который пишет разработчик, создавая HTML-документ. Между открывающим и закрывающим тегами располагается вся информация о веб-странице.

head

Элемент head — это служебный тег, который содержит в себе всю техническую информацию о веб-странице, а также задает ее заголовок. Эта информация не видна обычному пользователю (кроме title — заголовка окна веб-страницы), но помогает браузерам и поисковым системам работать с данными страницы.

title

Элемент title отвечает за заголовок страницы, который отображается в шапке браузера. Иными словами, title задает текст, который будет написан на самой вкладке браузера.

meta

Служебный тег meta предназначен для хранения технической информации, которая используется браузерами и поисковыми роботами. Например, это могут быть ключевые слова, описание сайта, кодировка документа и др.

Элемент meta не имеет закрывающего тега. Он располагается в контейнере head и может содержать внутри себя несколько атрибутов одновременно:

style

Служебный тег style определяет стили элементов страницы. Например, стиль H1 (его размер, шрифт, цвет). Задается данный тег в контейнере head и может применяться неоднократно (для разных элементов).

body

Элемент body — это служебный тег, содержащий в себе все данные, которые отображаются на веб-странице (текст, медиафайлы, таблицы и т. д.). Иными словами, body отвечает за отображение контента документа.

Внутри контейнера body используются все элементы, предназначенные для дизайна страницы. Но иногда там могут быть расположены и обработчики событий. Например, onload, который выполняется после того, как документ завершит загрузку.

Внутри открывающего тега body можно разместить атрибуты, задающие определенные значения параметров для всей веб-страницы целиком. Например, ее фон, цвет текста, цвет гиперссылок и т. д.

script

Элемент script — служебный тег, предназначенный для описания скриптов. Он может содержать в себе ссылку на саму программу или ее текст, написанный на другом языке. Данный тег позволяет HTML-документам работать с программами, написанными на языке JavaScript и др.

Один и тот же скрипт может храниться во внешнем файле и использоваться одновременно для нескольких страниц. Это позволяет выполнять его быстрее при последующих загрузках, так как при первой загрузке файл кэшируется.

5) Для чего служит тег <IMG>? Почему атрибут SRC этого тега является обязательным?

Тег <img> используется для вставки графического изображения (картинки) в HTML документ. HTML тег <img> имеет два обязательных атрибута: src - адрес файла картинки и alt - альтернативный текст, который будет отображен, если картинка не может быть загружена.

6) META-теги и их атрибуты. Приведите примеры значений атрибутов МЕТА-тегов.

Мета-теги можно разделить на две основные группы — это NAME и HTTP-EQUIV. Группа NAME отвечает за текстовую информацию о веб-документе, его авторе, а также — формирует рекомендации для поисковых роботов. Мета-теги, относящиеся к группе HTTP-EQUIV фактически эквивалентны гипертекстовым заголовкам, они формируют заголовок веб-страницы и определяют его обработку, а также управляют действиями браузеров и используются для формирования информации, выдаваемой обычными заголовками.

Элемент meta принимает следующие атрибуты: content, http-equiv, name, charset и scheme.

|  |  |
| --- | --- |
| Name | Имя метатега, также косвенно устанавливает его предназначение. Примеры: include, keywords, description, author, revised, generator и др. |
| content | Устанавливает значение атрибута, заданного с помощью name или http-equiv. |
| scheme (устарел) | Указывает полезную информацию о схеме или название самой схемы, которая должна быть использована для интерпретации значения свойства (то есть значения атрибута «content»). Не применяется в HTML5. |
| charset | Новый атрибут, показывает кодировку документа в HTML5. Пример: <meta charset="utf-8"> |
| http-equiv | Формирует заголовок страницы и определяет его обработку. Как правило, управляет действиями браузеров и используется для формирования информации, выдаваемой обычными заголовками. Например HTTP-EQUIV может использоваться для управления кэшированием, обновлением страницы, автоматической загрузки другой страницы. |

http и https

«HTTP, или Hyper Text Transfer Protocol, — это протокол передачи гипертекстовой разметки, которая используется для передачи данных в интернете».

HTTPS — это не совсем протокол. Это расширение HTTP-протокола — объединение двух протоколов: HTTP и SSL или HTTP и TLS.

Протоколы TLS (Transport Layer Security) и SSL (Secure Socket Layer) — криптографические. Это значит, что они позволяют шифровать данные, в нашем случае те, что передаются между браузером и сервером. Расшифровать эти данные могут только сервер и браузер, для всех остальных это будет набор нечитаемых символов.

Примечание: TLS основан на SSL, но второй уже устарел, и вместо него используют TLS.

HTTP — универсальный протокол. Он может передавать любые данные: страницы, музыку, видео, PDF-файлы. Но его минус в том, что он открытый: данные, которые передает протокол, никак не защищены.

По пути между браузером и сервером его легко перехватить, прочитать данные, например, пароли или данные кредитной карты. Все равно, что отправлять посылки и письма по почте без конвертов. Кроме того, HTTP-ответ можно подменить или добавить в него свои данные.

HTTPS тоже можно перехватить, но толку от этого мало — данные зашифрованы, а секретного ключа для расшифровки нет. Здесь есть не только конверты или упаковка, но и сургучная печать. HTTPS шифрует данные.

HTTPS шифрует данные открытым ключом, а затем получатель расшифровывает его. Открытый ключ лежит на сервере и входит в SSL-сертификат. Сертификаты, в свою очередь, криптографически подписываются центром сертификации (ЦС). Каждый браузер имеет список доверенных ЦС. Любой подписанный центром сертификации сертификат, входящий в список доверенных, получает в адресной строке значок в виде зеленого замка.

В чем разница между HTTP и HTTPS

* Кратко перечислим основные различия протоколов:
* По-разному записывается URL-адрес. Для HTTP — http://, а для HTTPS — [https://](NULL).
* HTTP является не защищенным, а HTTPS защищенным.
* HTTP посылает данные через порт 80, а HTTPS использует порт 443.
* HTTP работает на уровне приложений, а HTTPS — на транспортном уровне.
* Для HTTP не нужны SSL-сертификаты; для HTTPS нужен SSL-сертификат, подписанный центром сертификации.
* HTTP не требует проверки домена, тогда как HTTPS требует хотя бы проверки домена, а для некоторых сертификатов даже проверки юридического документа.
* В HTTP нет шифрования, HTTPS шифрует данные перед отправкой.