**САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО**

**Дисциплина:** Web-программирование

Отчет

Лабораторная работа № 1

Выполнил:

Райнус А.К.

Группа K3322

Проверил:

Марченко Е.В.

Санкт-Петербург

2024

**Оглавление**

[**Введение** 3](#_Toc182693977)

[**1.** **Структура HTML-кода** 4](#_Toc182693978)

[**2. Основы логического форматирования контента** 11](#_Toc182693980)

[**3. Гиперссылки** 19](#_Toc182693981)

[**4. Таблицы** 22](#_Toc182693982)

[**5. Изображения и медиаконтент** 24](#_Toc182693983)

[**6. Формы** 27](#_Toc182693984)

[**7. Основы работы с блочной структурой** 30](#_Toc182693985)

[**Вывод** 31](#_Toc182693986)

# **Введение**

В данной работе мы рассмотрим основные концепции и элементы HTML, их назначение и правильное использование. В качестве среды для написания кода была выбрана программа Notepad++.

Цель: изучение основ языков HTML и CSS для создания веб-страниц.

## **Структура HTML-кода**

**Упражнение 1. Создание первой HTML-страницы**

Создадим файл с заданным кодом и сохраним его как ex1.html (Рисунок 1)

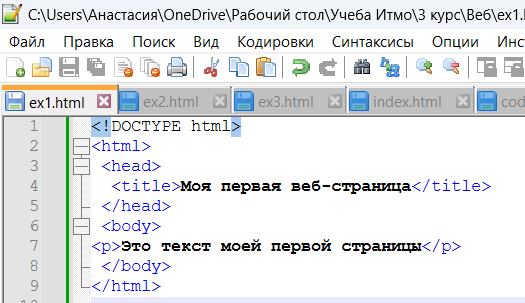


Рисунок 1 - Файл ex1

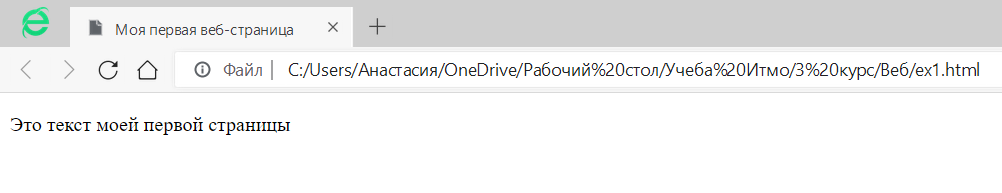
Откроем файл с помощью браузера, чтобы посмотреть результат (Рисунок 2)

Рисунок 2 - Результат работы

Посмотрим код отображаемой в браузере страницы через браузер (Рисунок 3)

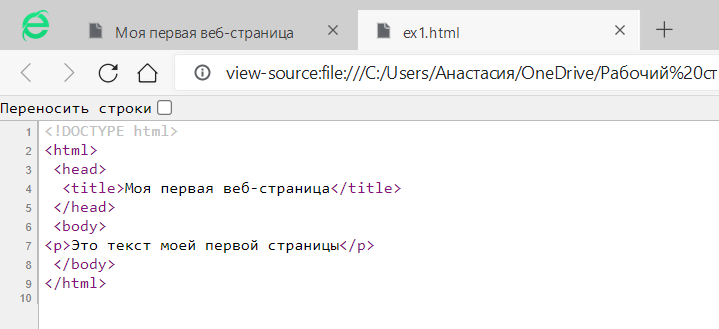


Рисунок 3 - Исходный код страницы

**Упражнение 2. Теги верхнего уровня и заголовка документа**

Откроем файл ex1.html в редакторе, добавим туда комментарии и тэг заголовка документа <meta> (Рисунок 4)

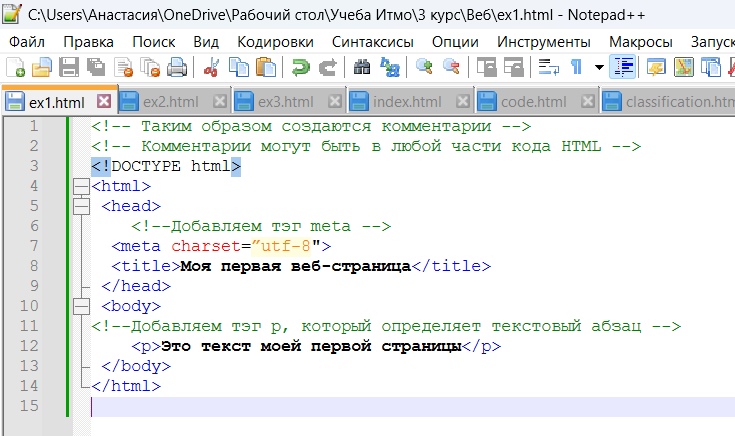
При открытии файла с помощью браузера видимых изменений не наблюдается, при этом код отображаемой в браузере страницы также поменялся (рисунок 5)

Рисунок 4 - Файл ex1 с тегами и комментариями

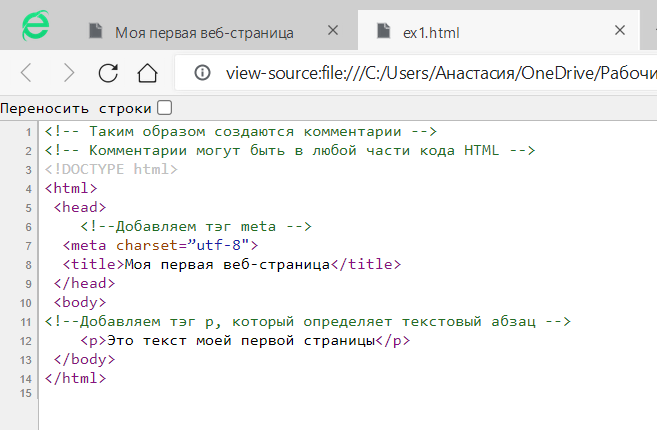


Рисунок 5 - Исходный код страницы

**Упражнение 3. Атрибуты HTML-тэгов**

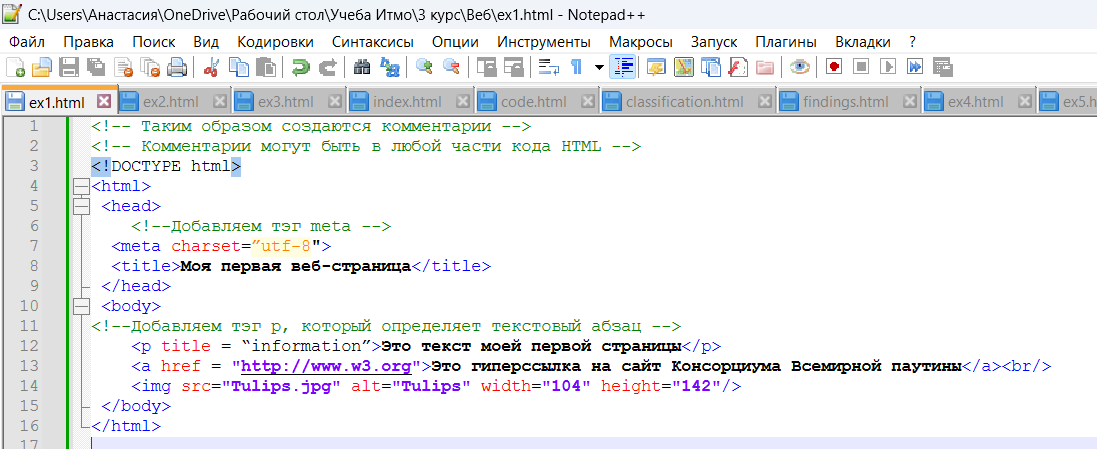
Добавим в файл ex1 тег a, с атрибутом href, содержащим ссылку <http://www.w3.org>, c информационным текстом. После загрузим в папку проекта изображение и добавим в код страницы в тег img. Чтобы перенести картинку ниже гиперссылки используем тег <br/>. В конце был добавлен заголовок для тега p, содержащего верхнюю надпись (Рисунок 6)

Рисунок 6 - Файл ex1 с гиперссылкой и картинкой

Посмотрим результат в браузере (Рисунок 7)

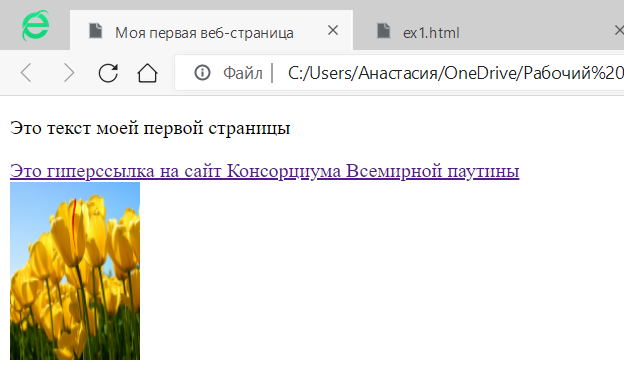


Рисунок 7 - Результат работы

**Упражнение 4. Устаревшие HTML-атрибуты**

Создадим файл ex2.html и скопируем в него код страницы ex1.html. Удалим в коде новой страницы все комментарии и атрибут title тэга <p>. Создадим еще один файл с именем ex3.html и скопируем в него получившейся код (рисунок Изображение выглядит как текст, снимок экрана, программное обеспечение, линия

Автоматически созданное описание8).

Рисунок 8 - Файл ex3

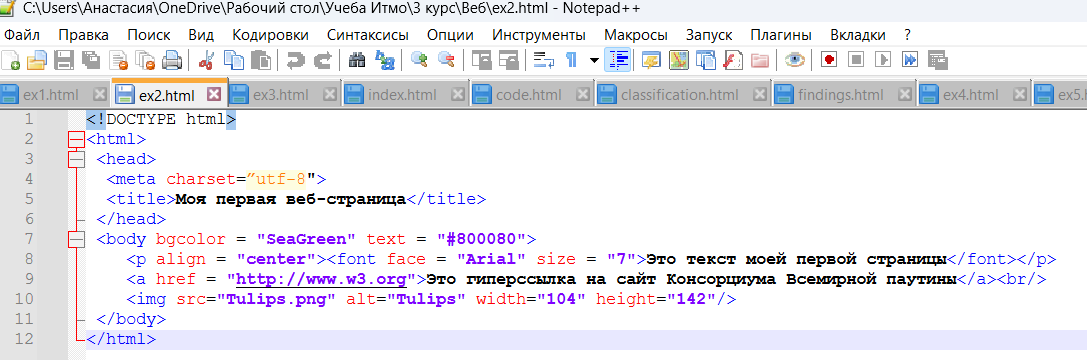
Далее используем элементы и атрибуты, отвечающие за визуальное отображение данных на странице, например <font>, align=” center” в теге <p>, bgcolor и text в <body> (Рисунок 9)

Рисунок 9 - Файл ex2

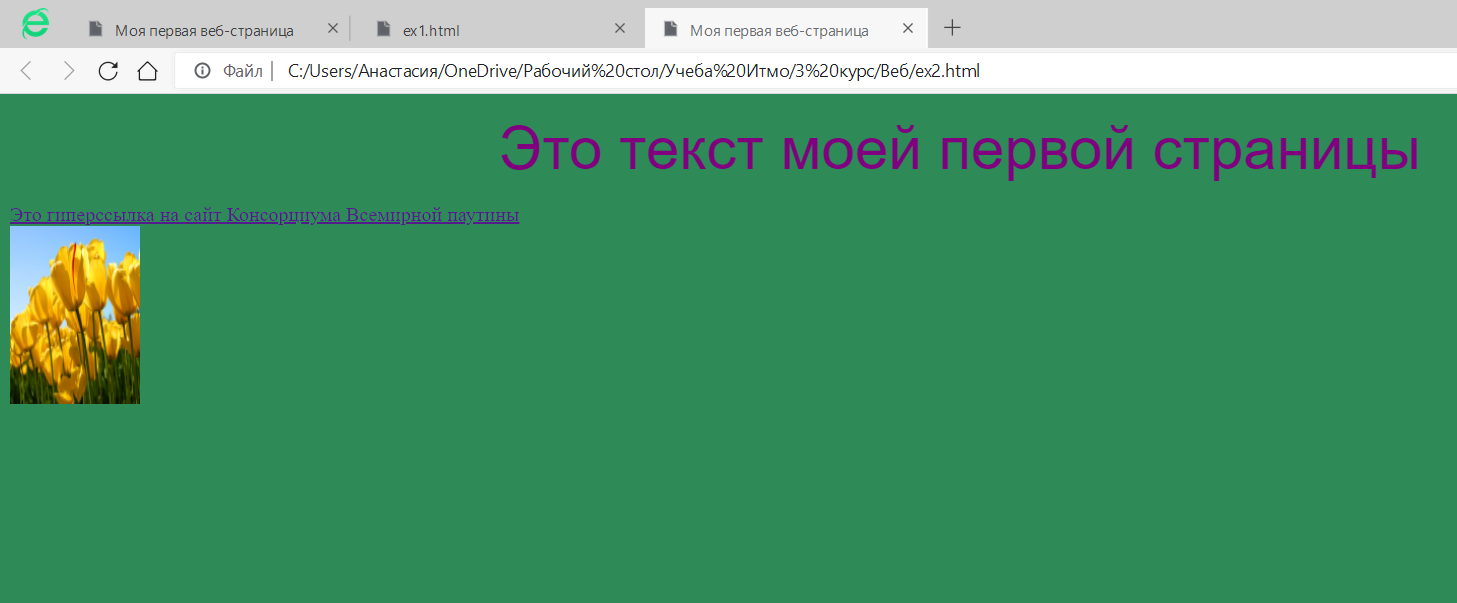
В результате изменился шрифт, цвет текста, выравнивание (Рисунок 10)

Рисунок 10 - Результат работы

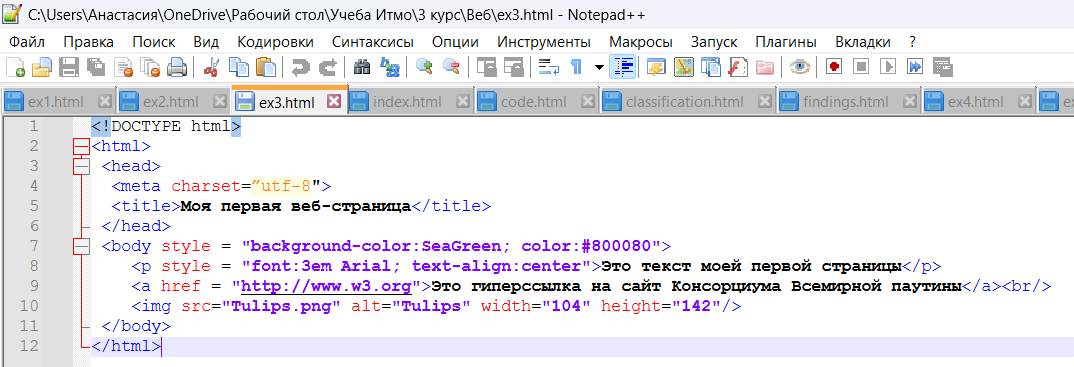
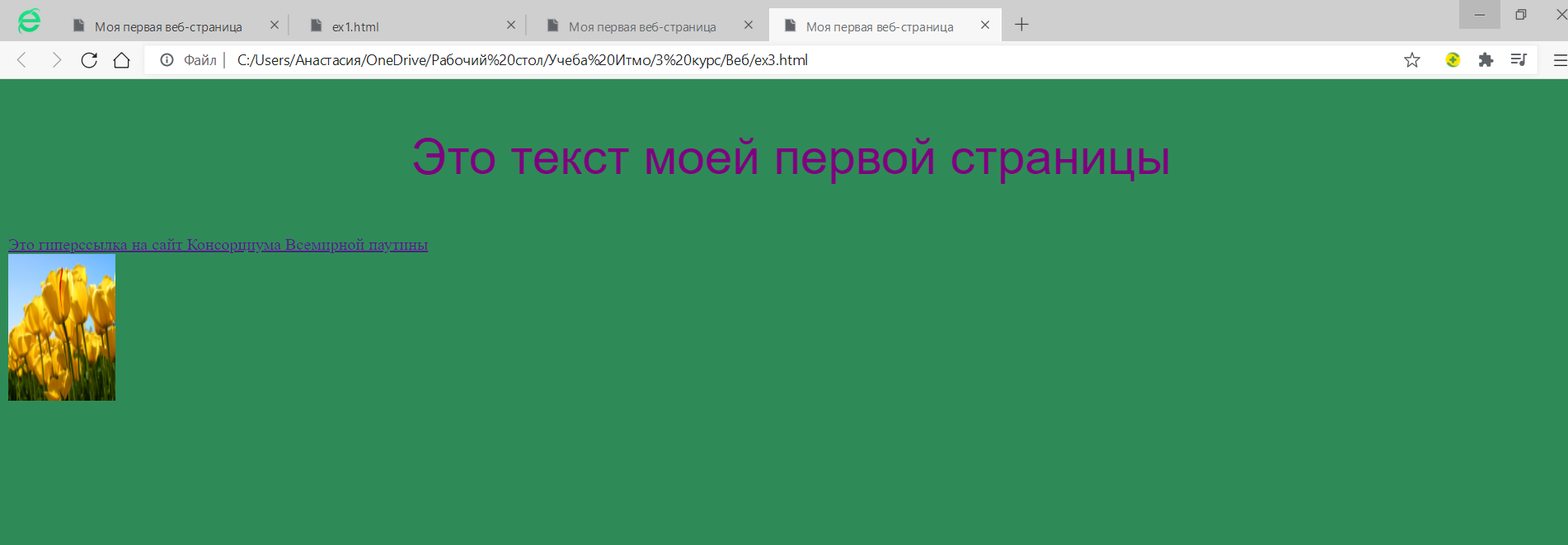
Далее такой же результат был достигнут с помощью стилевого оформления CSS (Рисунок 11-12)

Рисунок 12 - Результат работы

Рисунок 11 - Файл ex3

**Упражнение 5. Основные особенности работы с текстом в HTML**

Вернемся к файлу ex1. В данном задании будем работать с тегами <p>, <pre> для работы с текстом. Тег <p> воспринимает все пробельные символы как один, не реагирует на перенос строк клавишей Enter. Тег <pre> все это считывает и отображает текст так, как он есть в исходном коде страницы.

Рассмотрим код с тегом <p> (Рисунок 13-14)

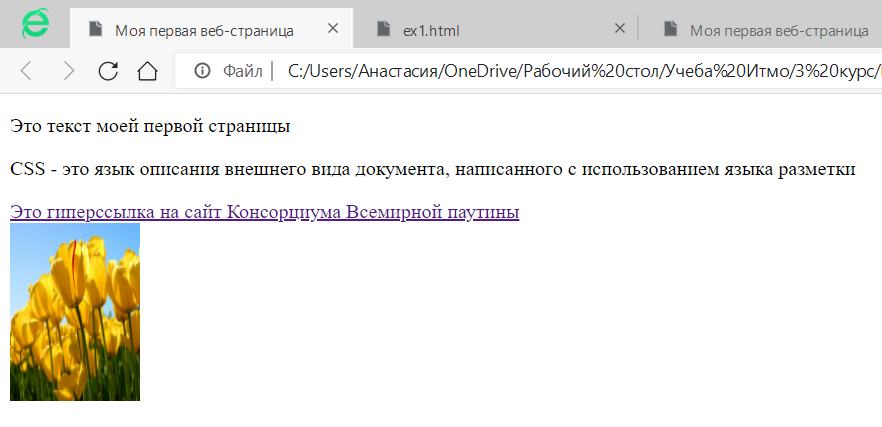
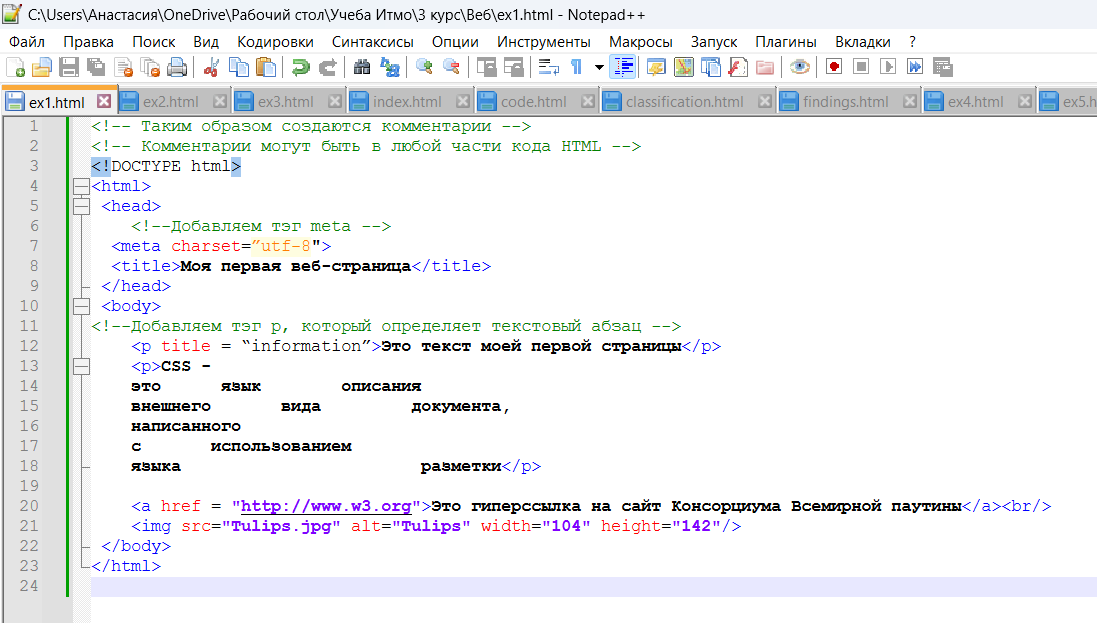
 

Рисунок 13 - Файл ex1 с тегом <p>

Рисунок 14 - Результат работы

Рассмотрим код с тегом <pre> (Рисунок 15-16)

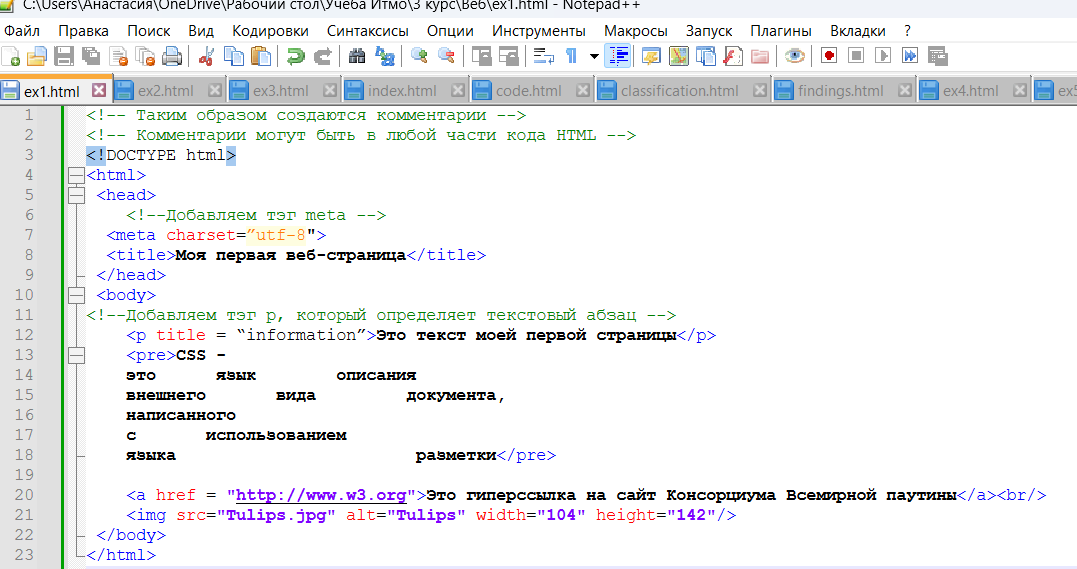
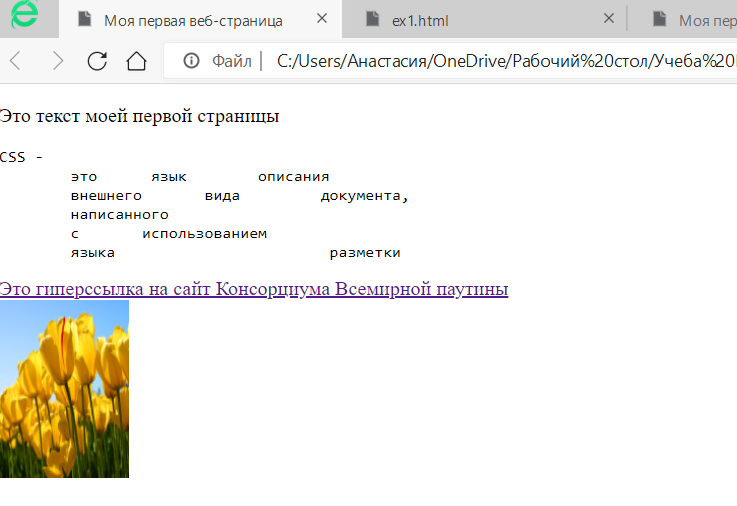


Рисунок 15 - Файл ex1 с тегом <pre>



### 

### 

Рисунок 16 - Результат работы

**Упражнение 6. Использование спецсимволов**

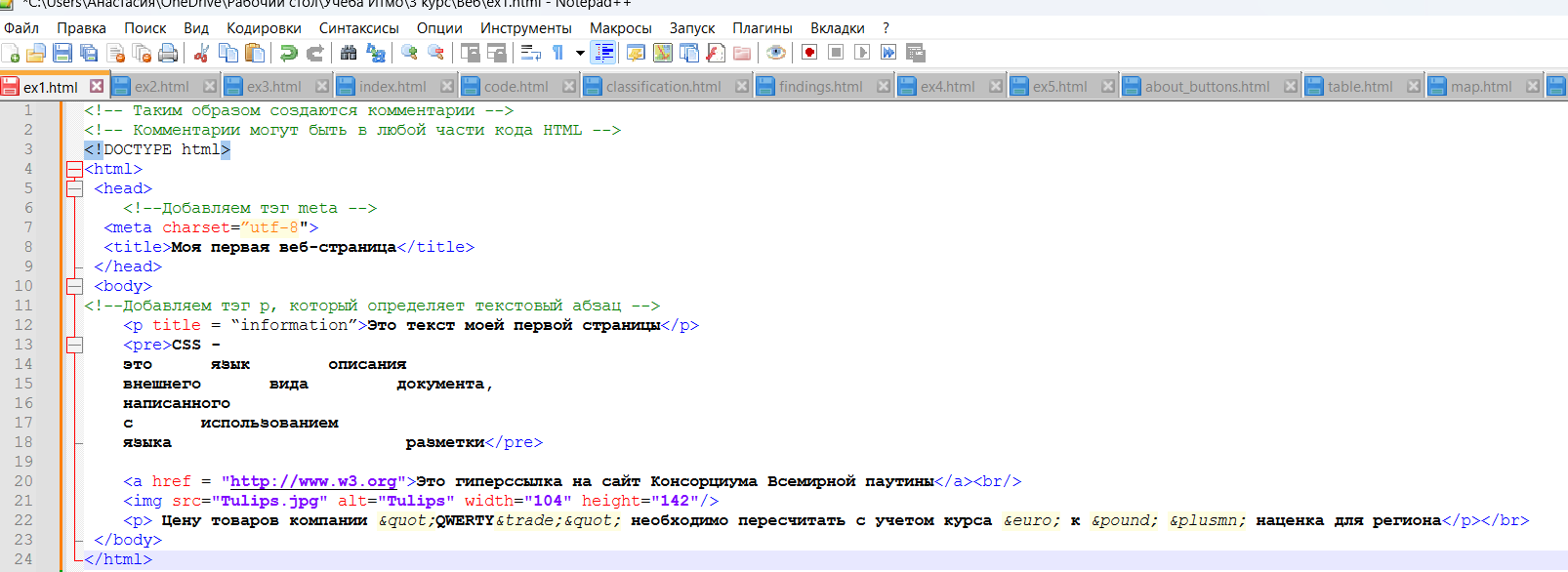
В файл ex1 добавим строку, которая содержит спецсимволы (Рисунок 17)

Рисунок 17 - Фaйл ex1 со спецсимволами

Посмотрим результат работы в браузере (Рисунок 18)

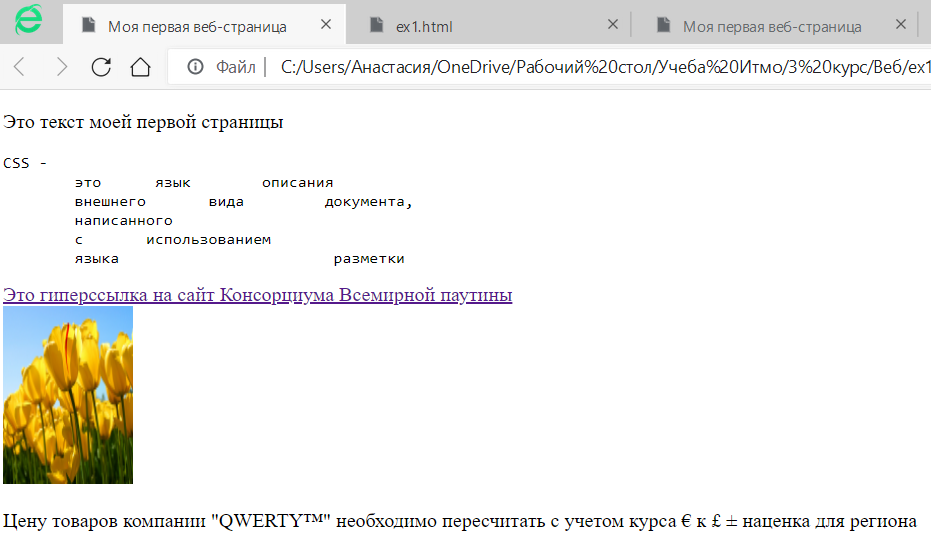


Рисунок 18 - Результат работы

## **2. Основы логического форматирования контента**

**Упражнение** **1. Использование заголовков**

Создадим файл с заданным кодом и сохраним его как index.html (Рисунок 19)

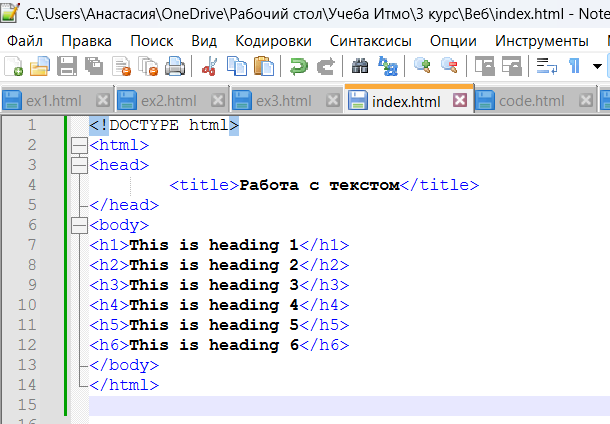


Рисунок 19 - Файл index

Посмотрим результат работы в браузере (Рисунок 20)

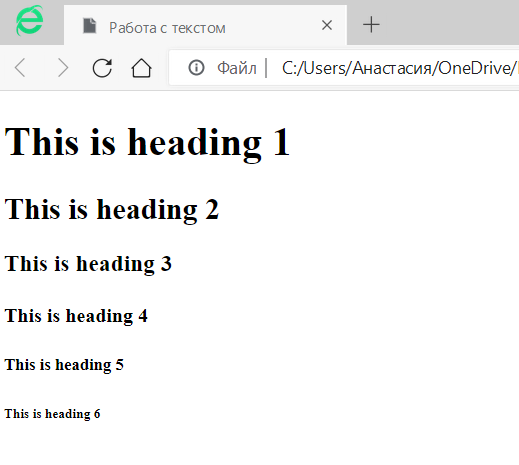


Рисунок 20 - Результат работы

**Упражнение 2. Возможности логического форматирования текста**

В данном задании будем работать с тегами для форматирования текста: тег <i> - курсив, тег <b> - жирный, тег <del> - зачеркнутый, тег <ins> - подчеркнутый и тд

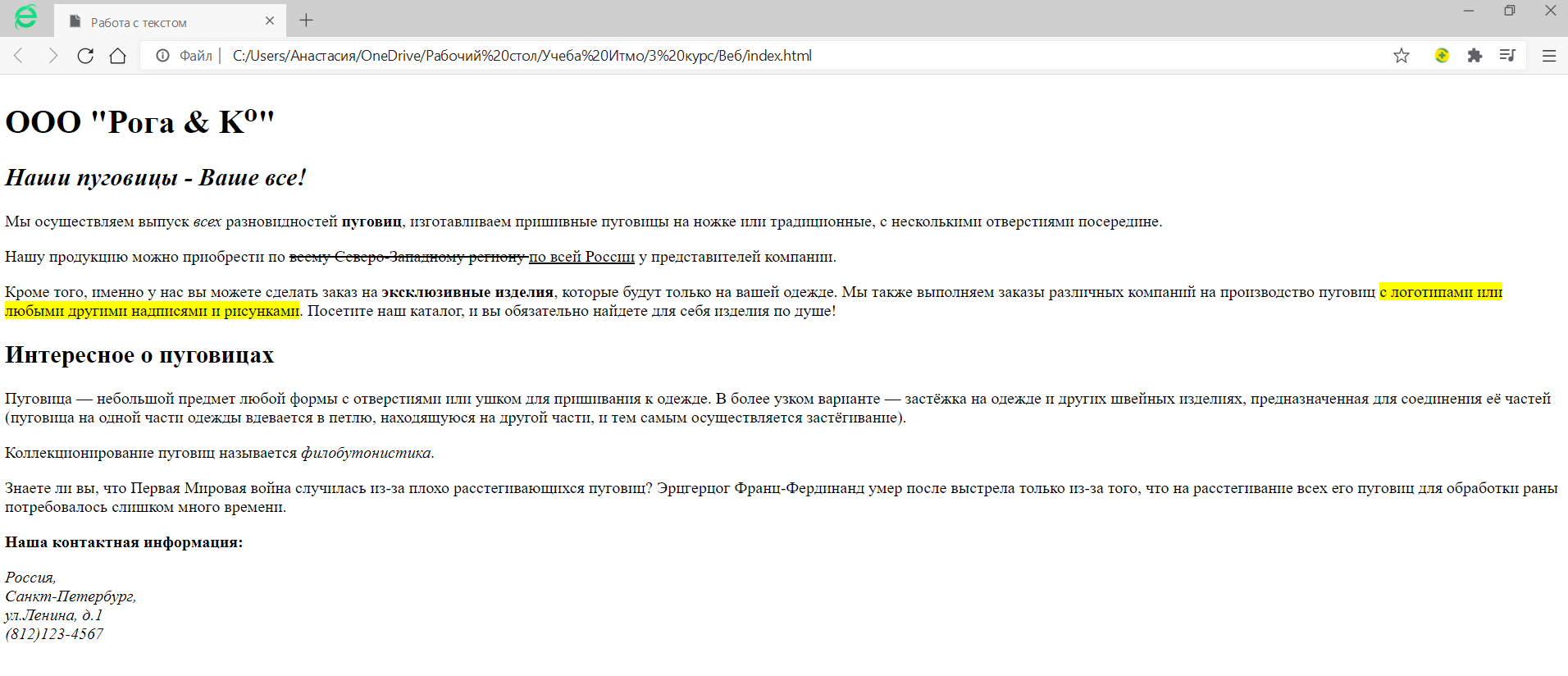
****Посмотрим результат работы в браузере (Рисунок 21)

Рисунок 21 - Файл index с тегами для форматирования текста

**Упражнение 3. Ссылки, цитаты, определения**

Рисунок 21 - Результат работы

В данном задании используем теги <dfn>, <cite>, <blockquote>, <address> (Рисунок 22)

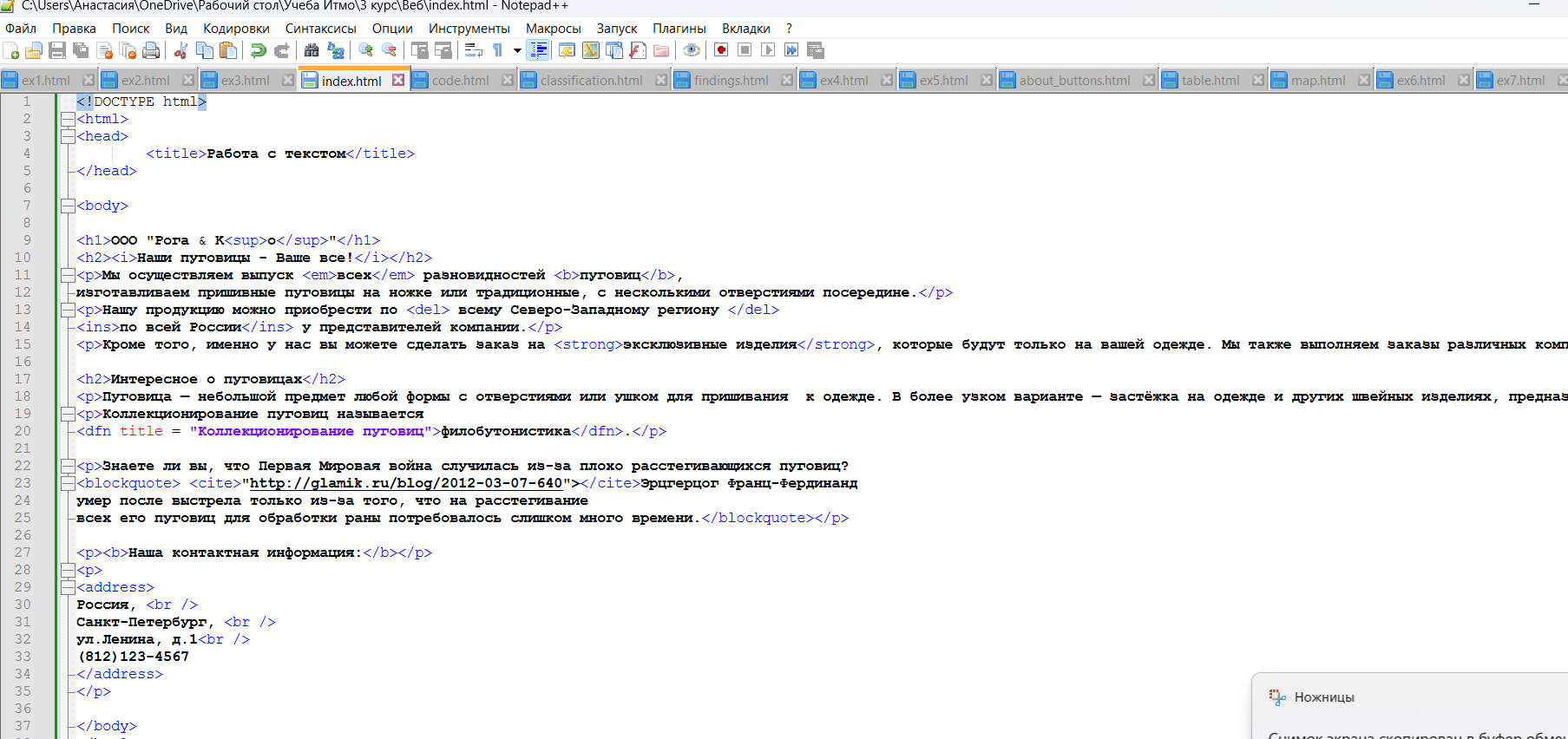
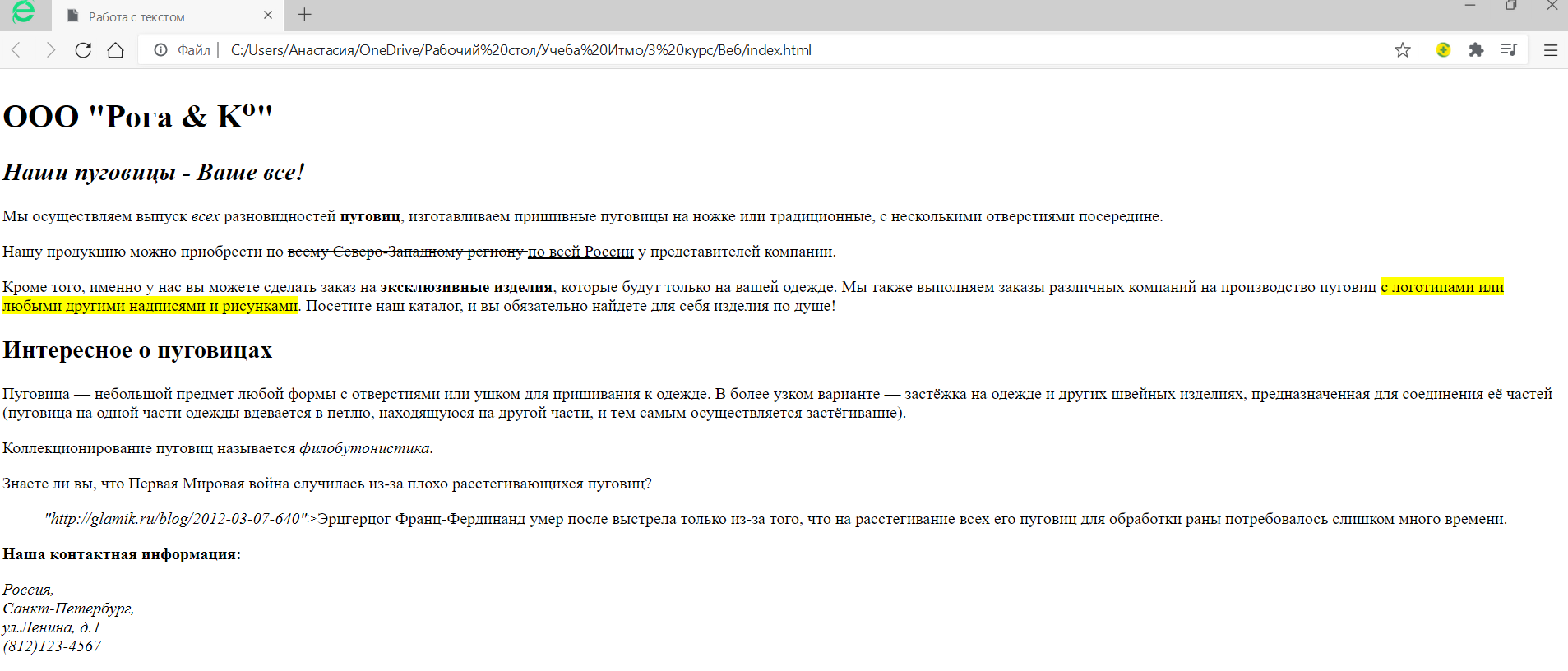
Посмотрим результат работы в браузере (Рисунок 23)

Рисунок 23 - Результат работы

Рисунок 22 - Файл index с тегами ссылок, цитат, определений

**Упражнение 4. Элементы компьютерного кода**

Создадим файл, самостоятельно определим его структуру и сохраним его как code.html. Для работы используем следующие теги: <kbd> - для обозначения ввода с клавиатуры, <samp> - для обозначения вывода компьютерной программы, <code> - для блока кода, <var> - для обозначения переменной (Рисунок 24)

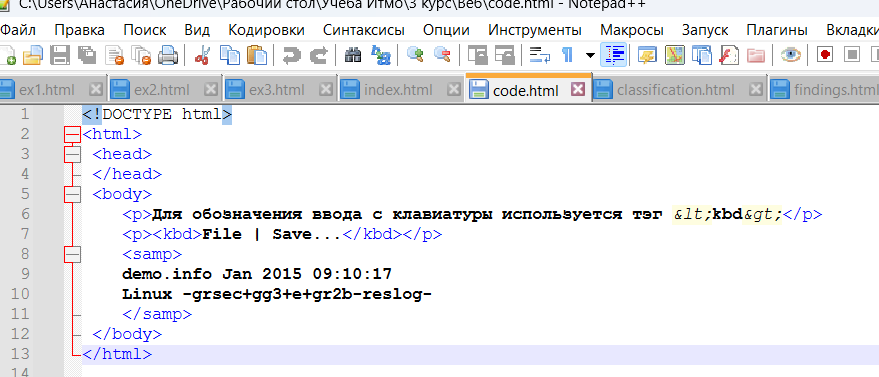
Посмотрим результат работы в браузере (Рисунок 25)

Рисунок 24 - Файл code

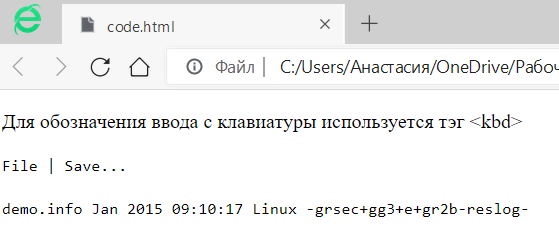


Рисунок 25 - Результат работы

**Упражнение 5. Маркированные списки**

Создадим файл, самостоятельно определим его структуру и сохраним его как classification.html. С использованием тегов <ul>, <li> был реализован маркированный список. Далее с помощью CSS меняем как список целиком, так и отдельные элементы списка (Рисунок 26)

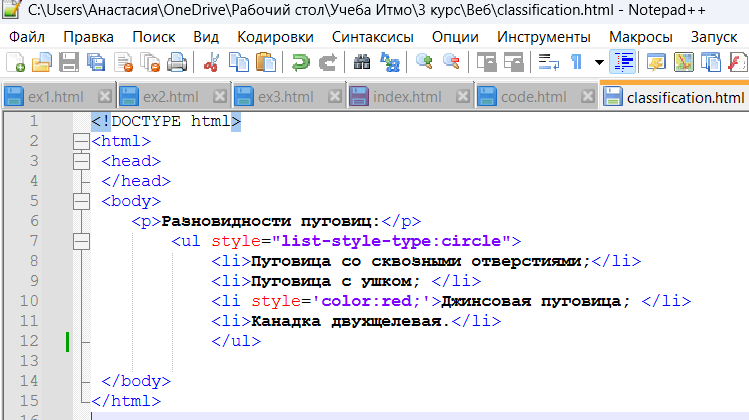
****

Рисунок 26 - Файл classification

Посмотрим результат работы в браузере (Рисунок 27)

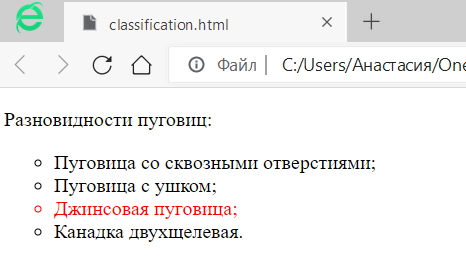


Рисунок 27 - Результат работы

**Упражнение 6. Нумерованные списки**

В файле classification.html с помощью тегов <ol>, <li> создадим нумерованный список. В тег <ol> были добавлены атрибуты start=”5” и type=”i” (Рисунок 28)

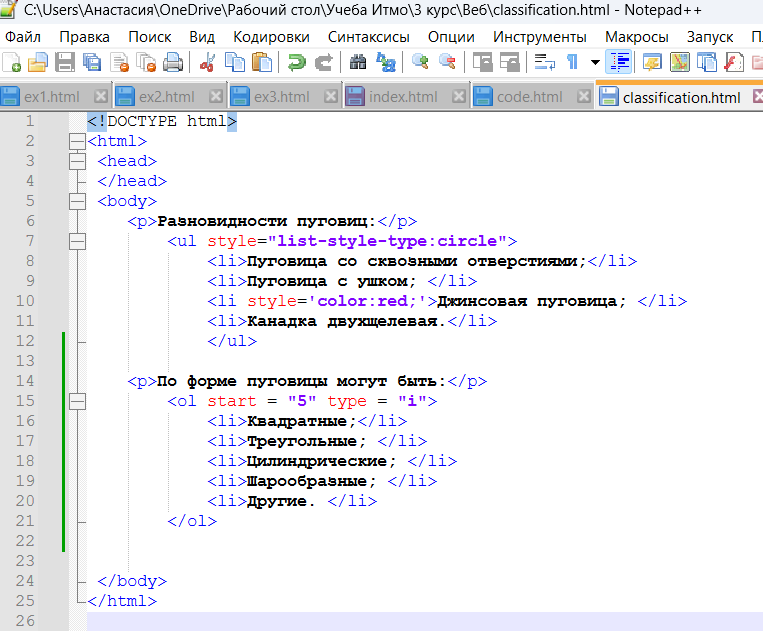
Посмотрим результат работы в браузере (Рисунок 29)

Рисунок 28 - Файл classification c нумерованным списком

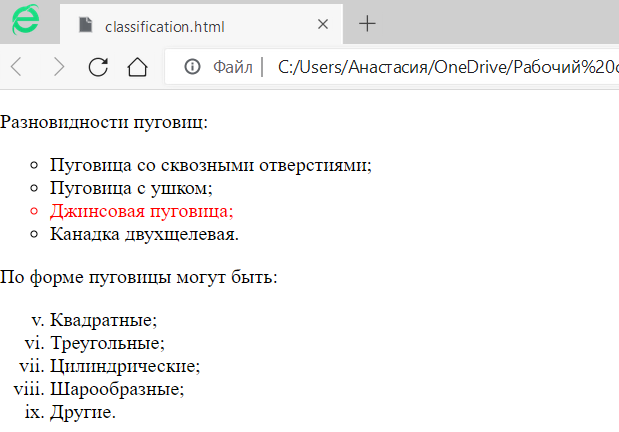


Рисунок 29 - Результат работы

**Упражнение 7. Список определений**

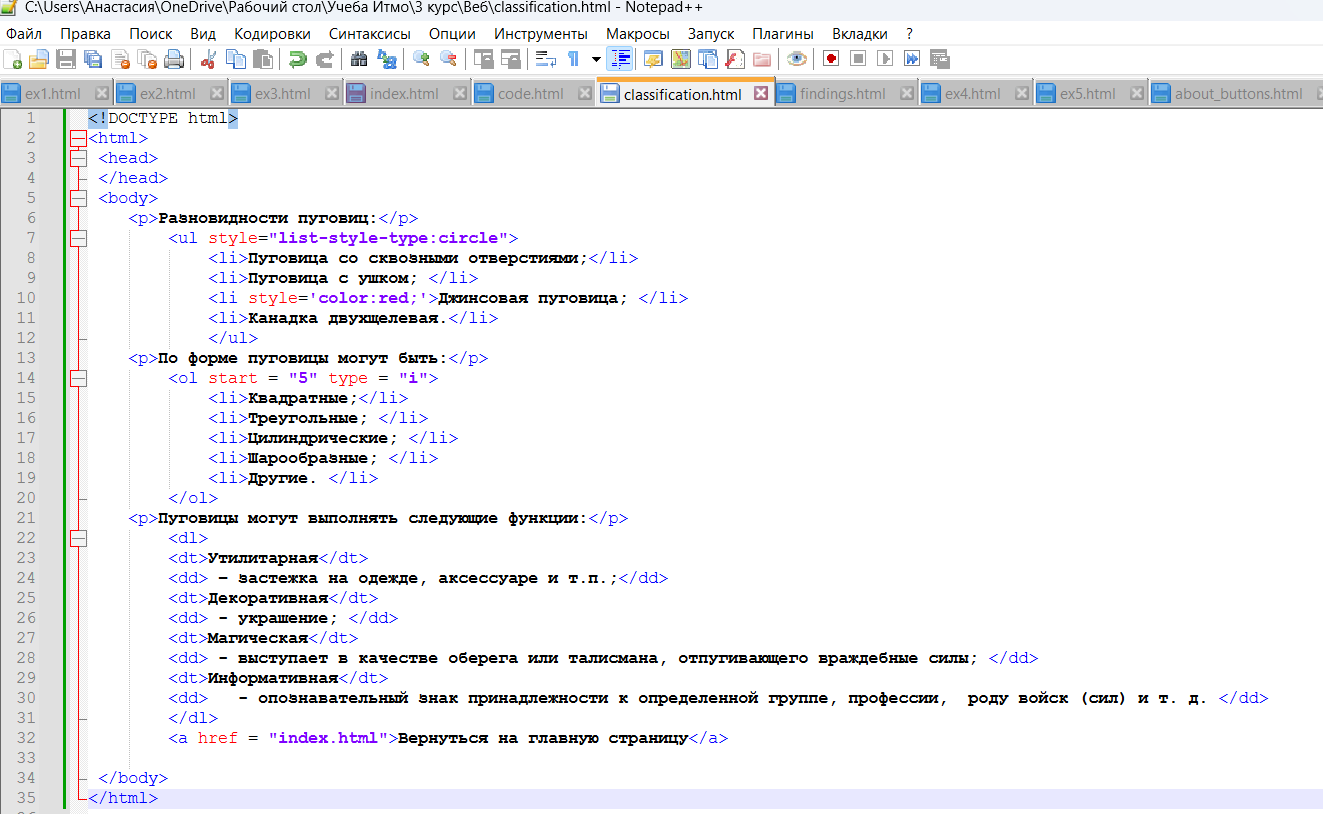
В файле classification.html с помощью тегов <dl>, <dt>, <dd> создадим список определений (Рисунок 30)

Рисунок 30 - Файл classification со списком определений

Посмотрим результат работы в браузере (Рисунок 31)

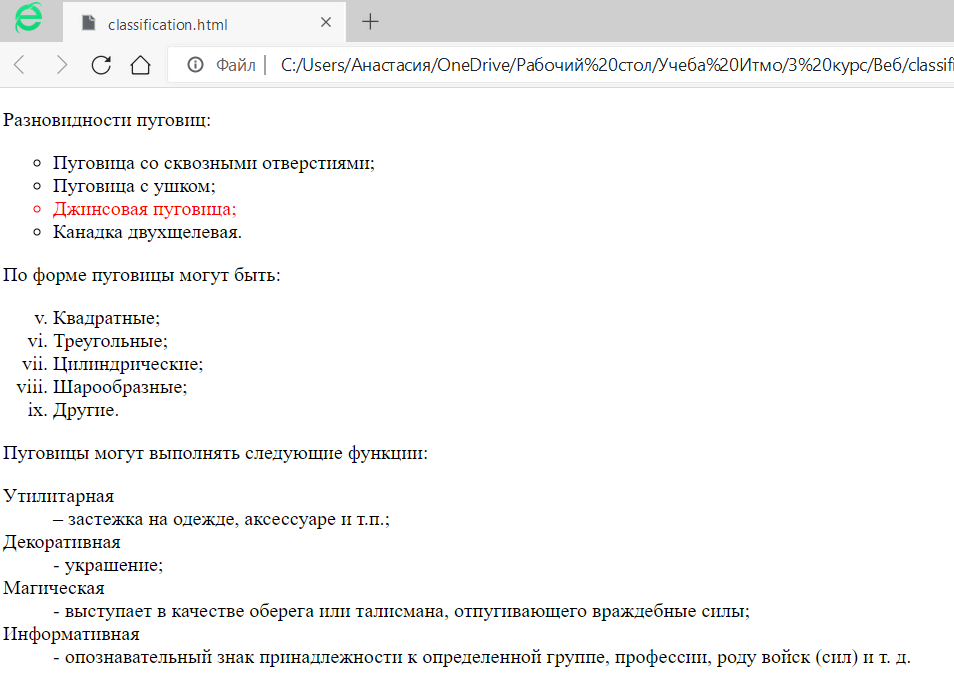


Рисунок 31 - Результат работы

**Упражнение 8. Вложенные списки**

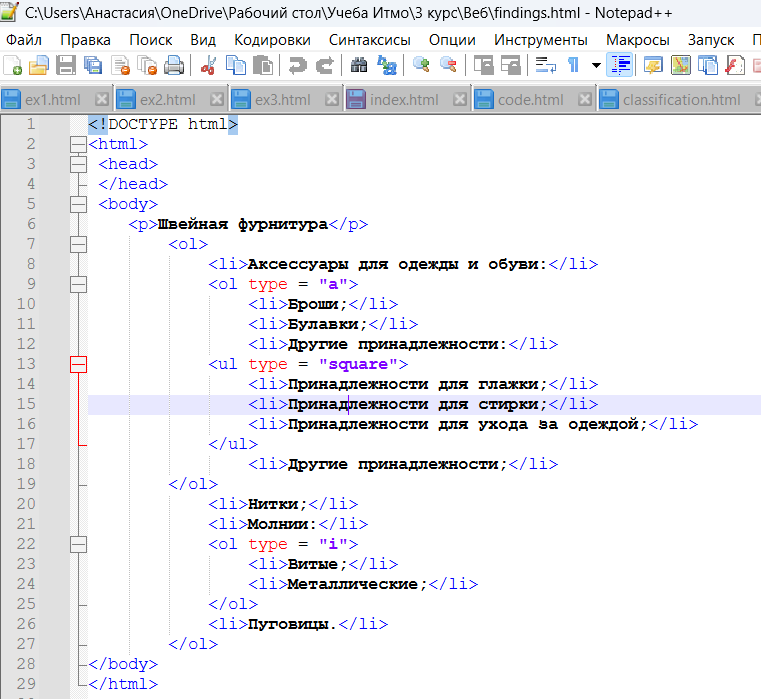
В файле findings.html создадим многоуровневый список (Рисунок 32)

Рисунок 32 - Файл fightings

Посмотрим результат работы в браузере (Рисунок 33)

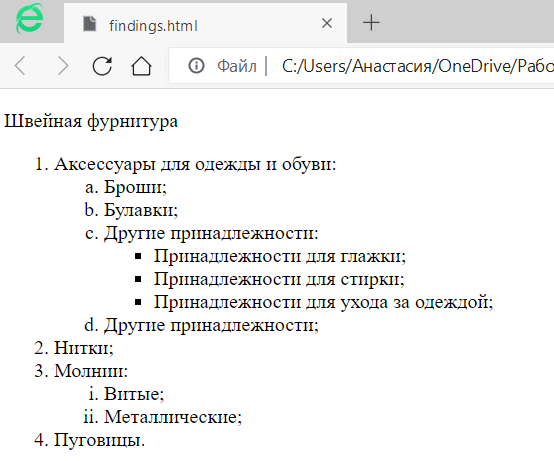


Рисунок 33 - Результат работы

## **3.** **Гиперссылки**

**Упражнение 1. Основы работы с гиперссылками**

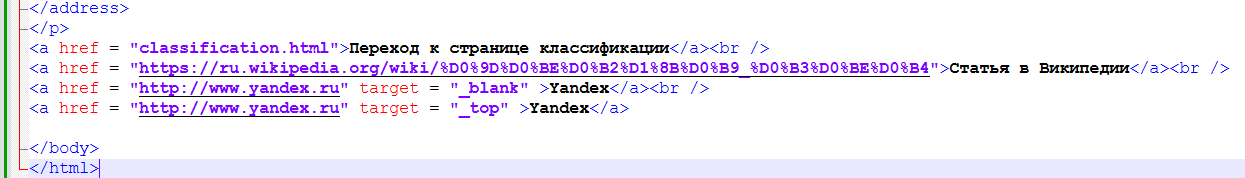
Вернемся к файлу index. Добавим несколько гиперссылок: ссылка на страницу classification.html, на внешний ресурс (Википедия), на страницу поисковой системы Яндекс – две ссылки – одна с открытием в новой вкладке, другая в этой же (атрибут target=”\_blank” или “\_top”) (Рисунок 34)

Рисунок 34 - Файл index с гиперссылками

Посмотрим результат работы в браузере (Рисунок 35)

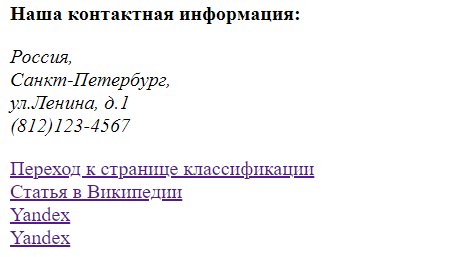


Рисунок 35 - Результат работы

**Упражнение 2. Работа с цветом гиперссылок**

Создадим файл с заданным кодом и сохраним его как ex4.html. Зададим цвет для активной и посещенной ссылки с помощью атрибутов тега <body> (alink, vlink). После этого с помощью CSS в теге <style> установим цвет для трех состояний ссылки visited(посещенная), active(активная), hover (при наведении) (Рисунок 36)

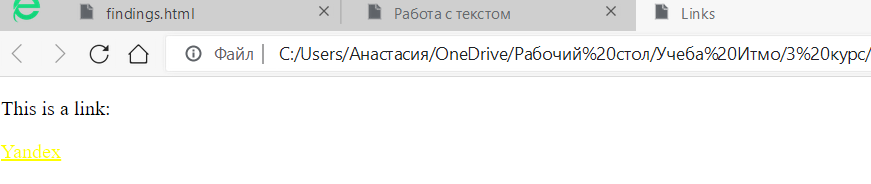
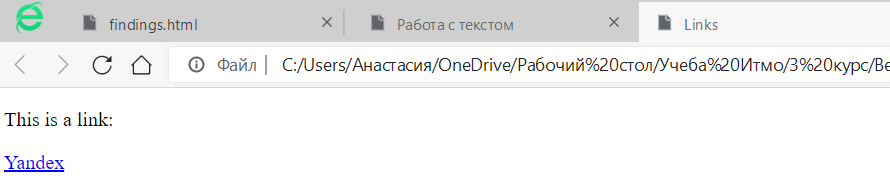
Посмотрим результат работы в браузере (Рисунок 37-38)

Рисунок 38 - Результат работы для ссылки при наведении

Рисунок 37 - Результат работы для посещенной ссылки

Рисунок 36 - Файл ex4



**Упражнение 3. Ссылка на адрес электронной почты**

Добавим в файл ex4 ссылку на адрес электронной почты с помощью mailto. В результате при нажатии на ссылку открывается приложение почта с окном нового письма на указанный адрес (Рисунок 39)

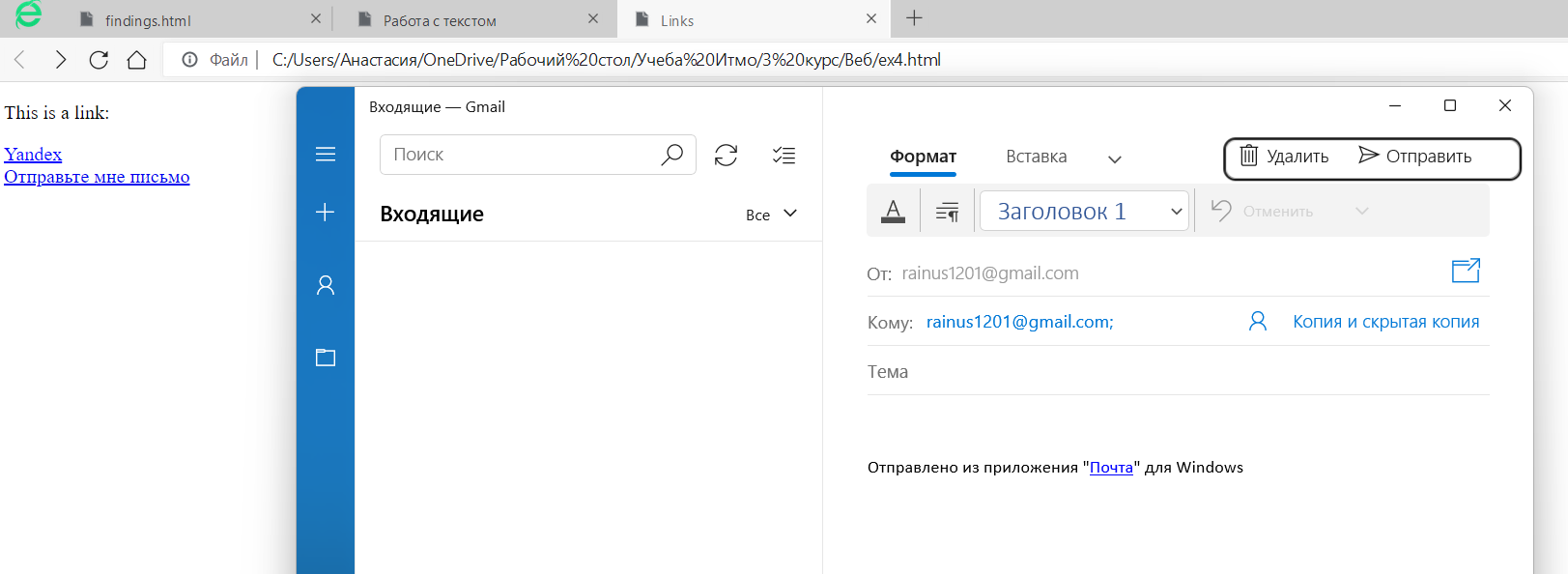


Рисунок 39 - Результат работы

**Упражнение 4. Внутренние ссылки документа**

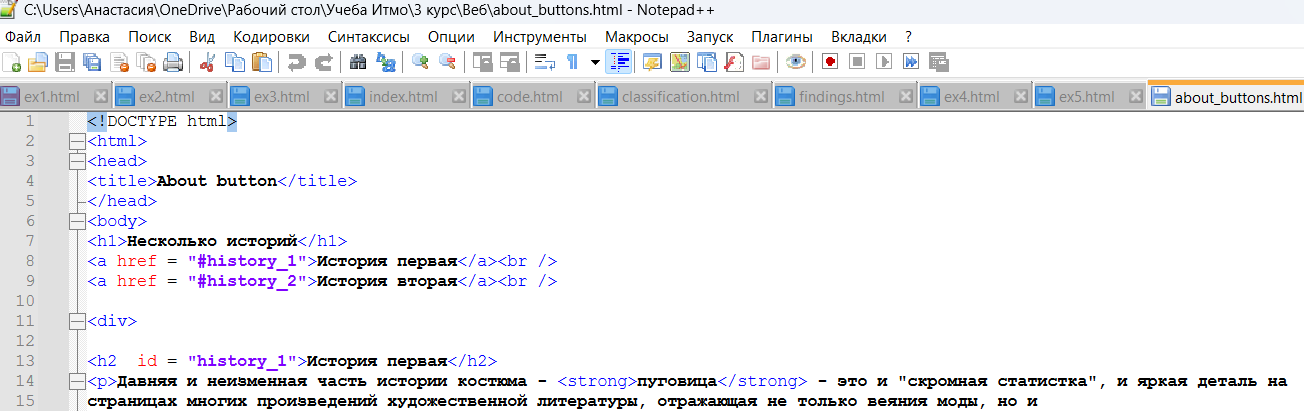
Откроем файл about\_buttons.html. Добавим внутренние ссылки на заголовки второго уровня для упрощения навигации по странице (Рисунок 40)

Рисунок 40 - Файл about\_buttons с внутренними ссылками

Посмотрим результат работы в браузере (Рисунок 41)

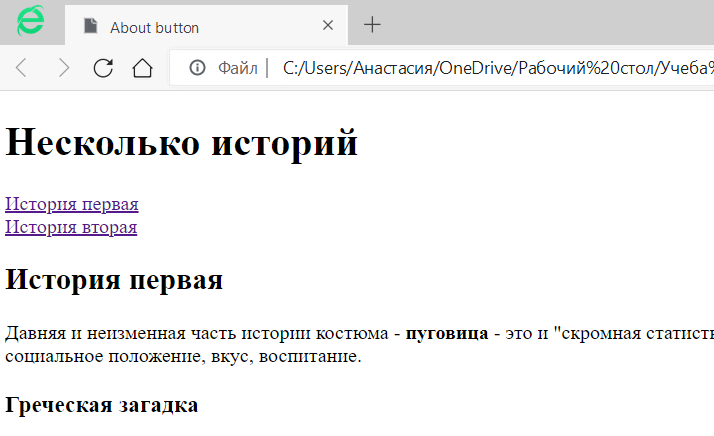


Рисунок 41 - Результат работы

\*тут я узнала, что не надо так подробно расписывать\*

## **4. Таблицы**

Данный раздел был посвящен работе с таблицами, их созданию и визуальным изменениям.   
Сначала мы работали с файлом ex5.html. Была создана таблица с помощью тегов <table> - таблица, <tr> - ряд таблицы, <td> - значение в столбце. Далее были добавлены границы для таблицы (атрибут border), а также стиль style="width:80%", определяющий ширину данного блока. В следующем упражнении мы выделили первую строку таблицы как шапку таблицы с помощью тегов <th> (вместо <td>). Также в таблице было добавлено название тегом <caption>. В конце мы использовали атрибуты colspan и rowspan для объединения ячеек в нескольких колонках или рядах соответственно.

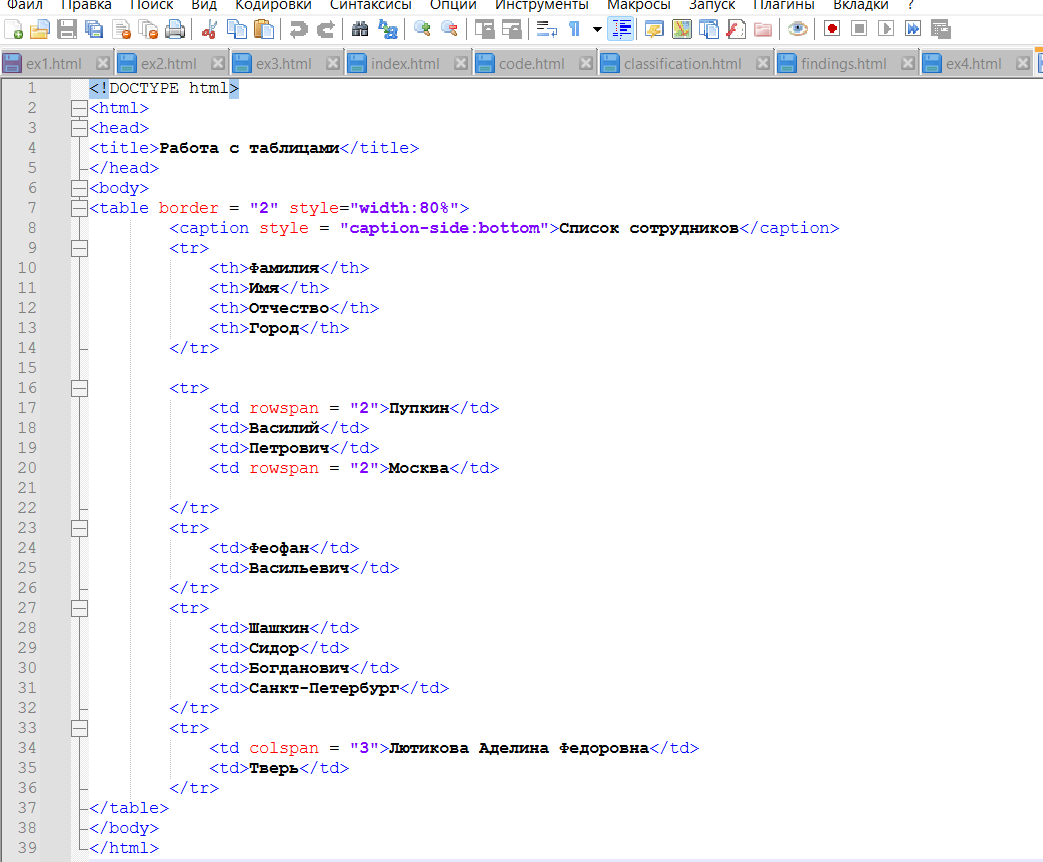
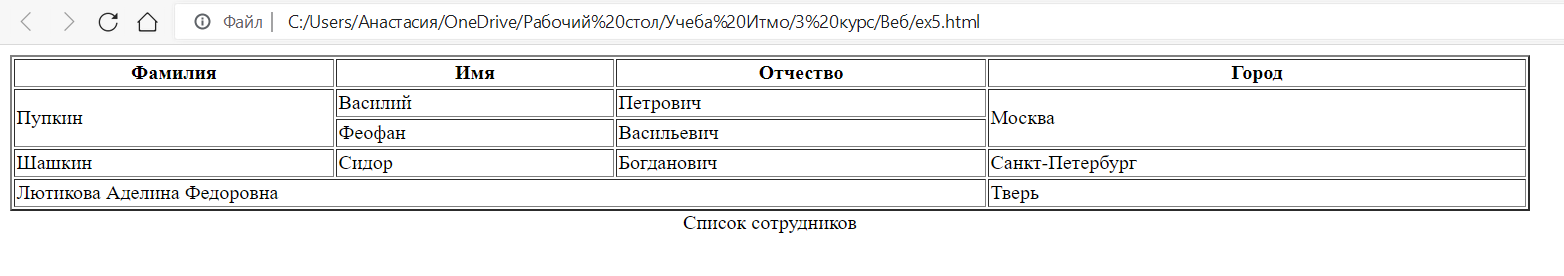
Итоговый код представлен на рисунке 42:

Рисунок 42 - Файл ex5

Посмотрим результат работы в браузере (Рисунок 43)

Рисунок 43 - Результат работы

Далее мы работали с файлом table.html. Создали таблицу с использованием тэгов <tbody>, <thead>, <tfoot>. Текже, были добавлены стили CSS, использован тег <colgroup> в комбинации с тегом <col> для указания стилей сразу нескольких колонок.

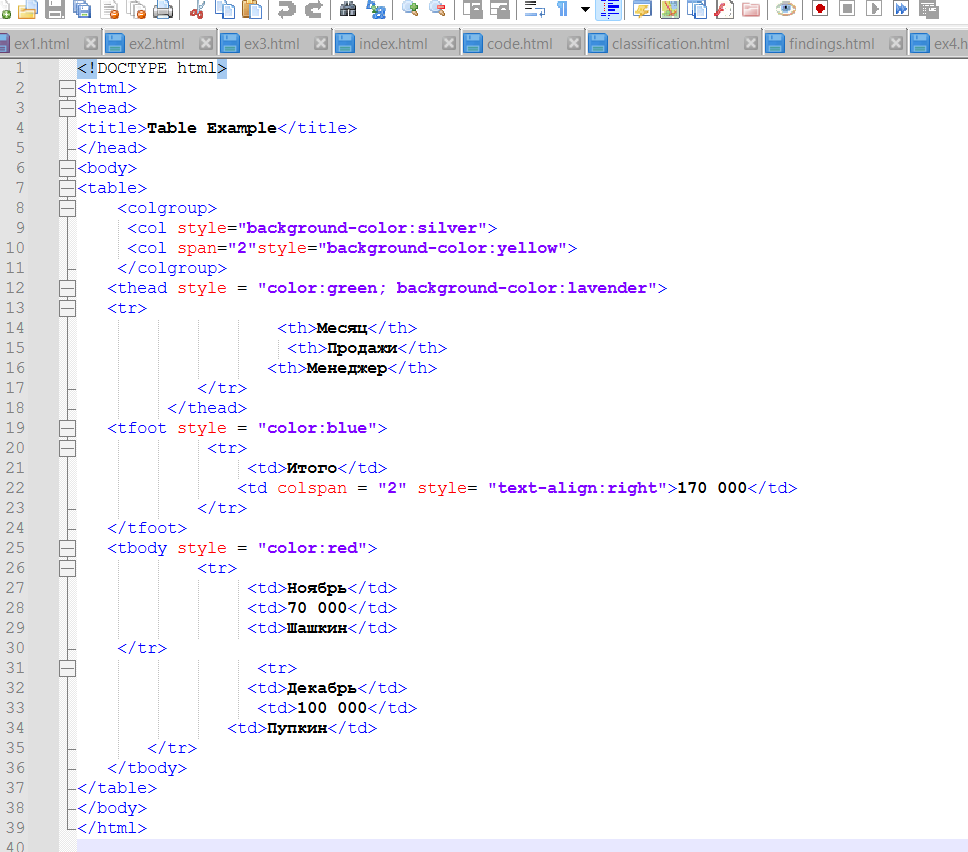
Итоговый код представлен на рисунке 44:

Рисунок 44 - Файл table

Посмотрим результат работы в браузере (Рисунок 45)

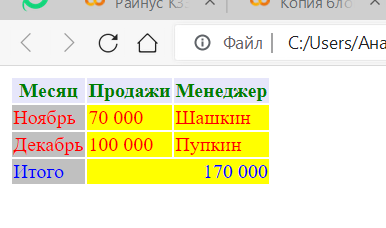


Рисунок 45 - Результат работы

## **5. Изображения и медиаконтент**

Данный раздел был посвящен работе с изображениями в качестве кнопок, гиперссылок.

Сначала мы работали с файлом index.html. В первом упражнении мы добавили картинку пуговицы в теге <img>. Были заданы атрибуты alt – альтернативный текст, title – текст при наведении. Далее мы рассматривали два способа изменить размер изображения: задать атрибуты witdh и height в теге (средствами HTML) или задать стили CSS и указать размеры аналогично через них. В конце изображение пуговицы было превращено в гиперссылку. Код картинки был вставлен в тег <a>, содержащий ссылку на страницу classification.html.

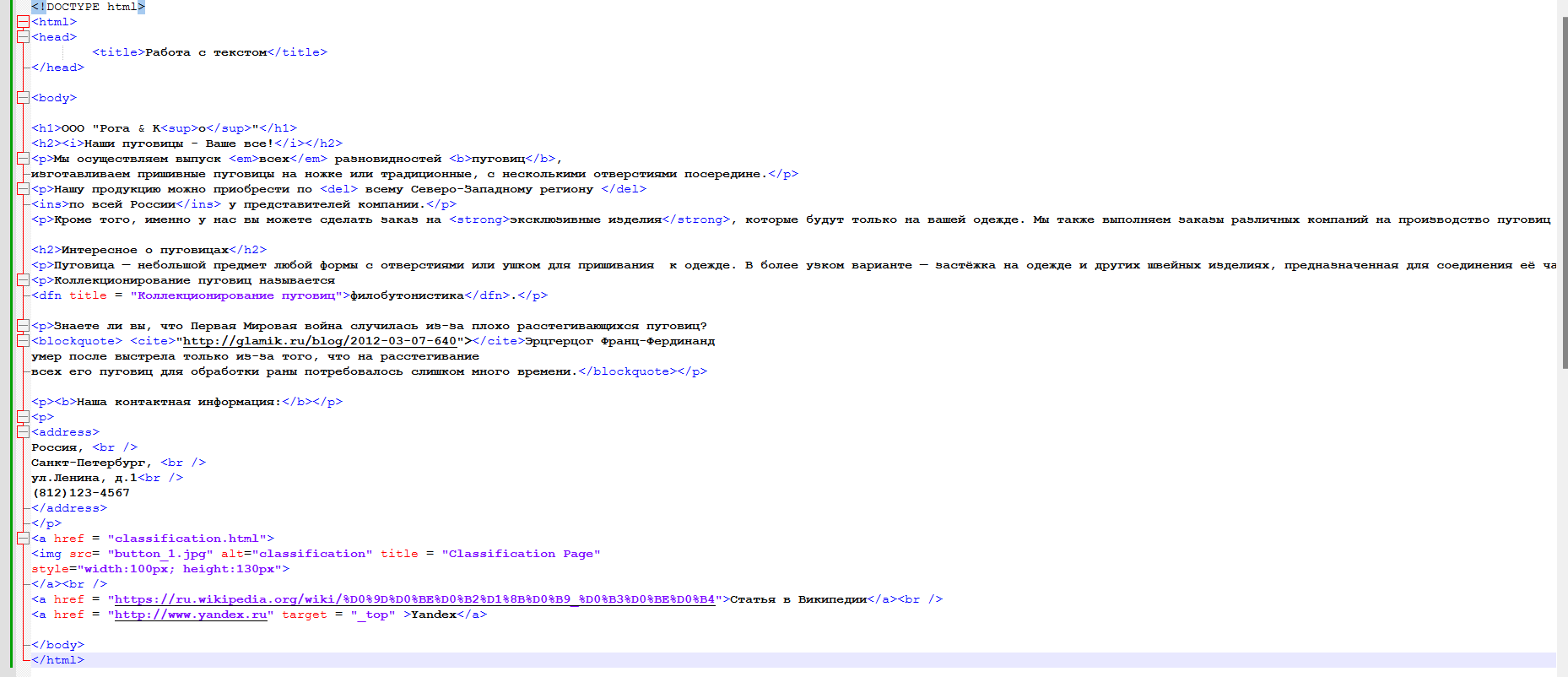
Итоговый код представлен на рисунке 46:

Рисунок 46 - Файл index

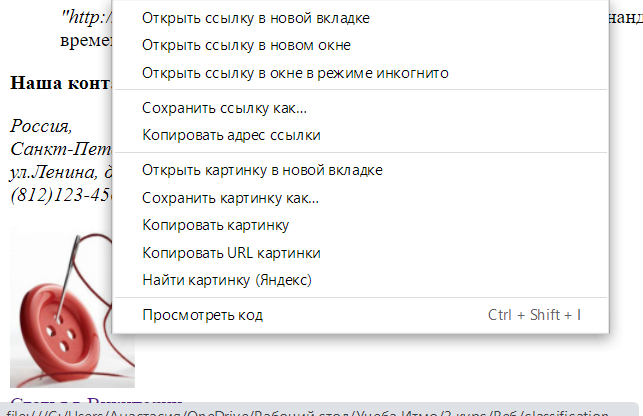
Посмотрим результат работы в браузере (Рисунок 47)

Рисунок 47 - Результат работы

Далее мы работали с файлом map.html. Мы создали область на изображении, которая стала ссылкой на внешнюю страницу. Для этого был использован тег <map> для обозначения карты изображения, с ним через атрибут usemap была связана картинка. В теге area была указана активная область (круг) на изображении.

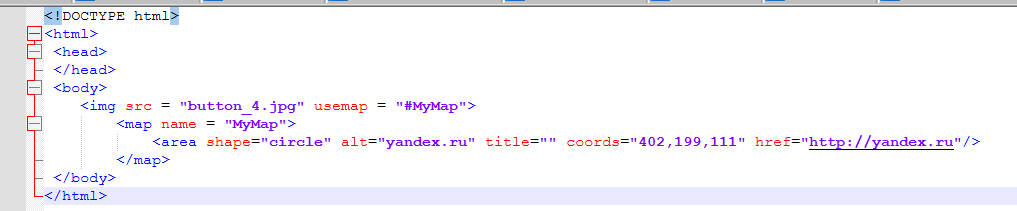
Итоговый код представлен на рисунке 48:

Рисунок 48 - Файл map

Посмотрим результат работы в браузере (Рисунок 49)

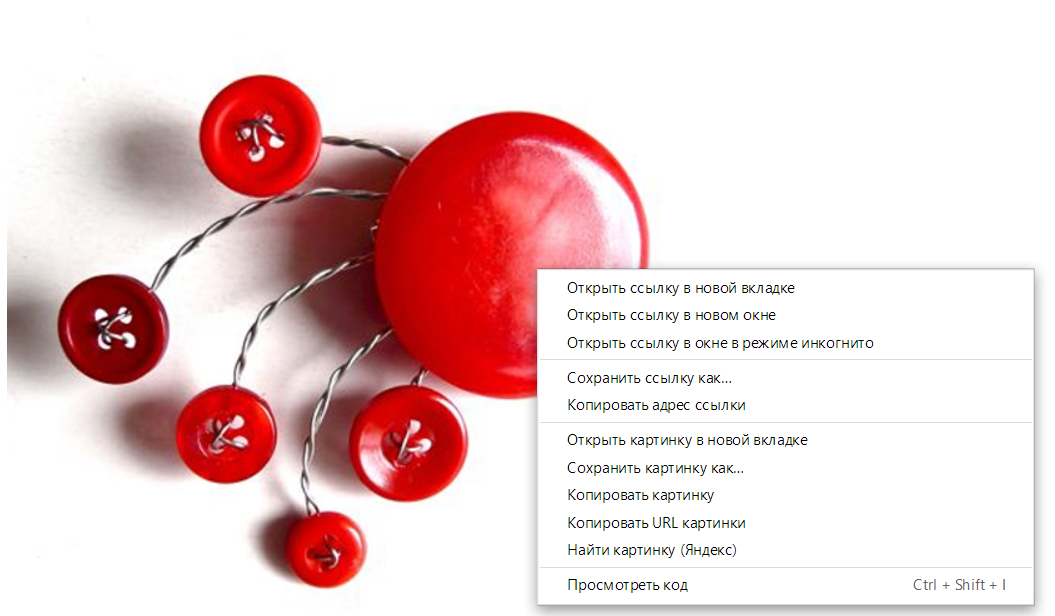


Рисунок 49 - Результат работы

В последнем упражнении мы работали с файлом ex6.html, в котором была создана веб-страница, содержащая видео в формате mp4 с помощью тега <video>. Код представлен на рисунке 50:

Рисунок 49 - Результат работы

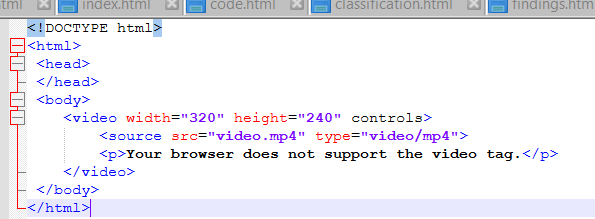


Рисунок 50 - Файл ex6

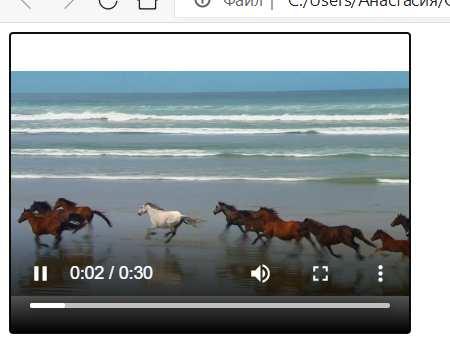
Посмотрим результат работы в браузере (Рисунок 51)

Рисунок 51 - Результат работы

## **6. Формы**

Данный раздел был посвящен работе с объектами, которые принимают данные от пользователя.

Сначала мы работали с файлом ex7.html. Сначала были изучены разные текстовые поля, которые можно использовать в HTML-формах (простое текстовое поле, поле с ограничением длины ввода (атрибут maxlenght), textarea, поле для ввода пароля (атрибуты type = "password" name = "hdnTxt"), скрытое поле (атрибут type = "hidden")). Далее были изучены разные типы полей в формах. Задать тип поля можно с помощью атрибутов type и name в теге <input>. В следующем упражнении мы изучили различные типы кнопок в HTML. Задать тип кнопки можно также в теге <input> с помощью атрибутов type, name. В новой форме были созданы кнопка сбрасывающая значение всех полей в данной форме (type = "reset"), кнопка отправления результатов (type = "submit"), простая кнопка (type = "button"), кнопка-картинка (тег <button>).

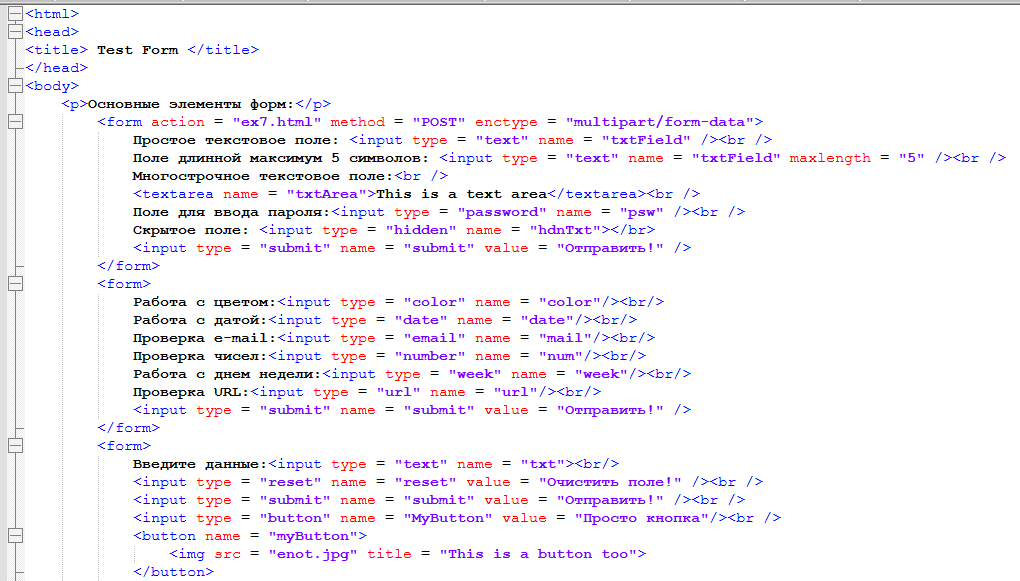
Данная часть кода представлена на рисунке 52:

Рисунок 52 - Файл ex7 1 часть

Посмотрим результат работы в браузере (Рисунок 53)

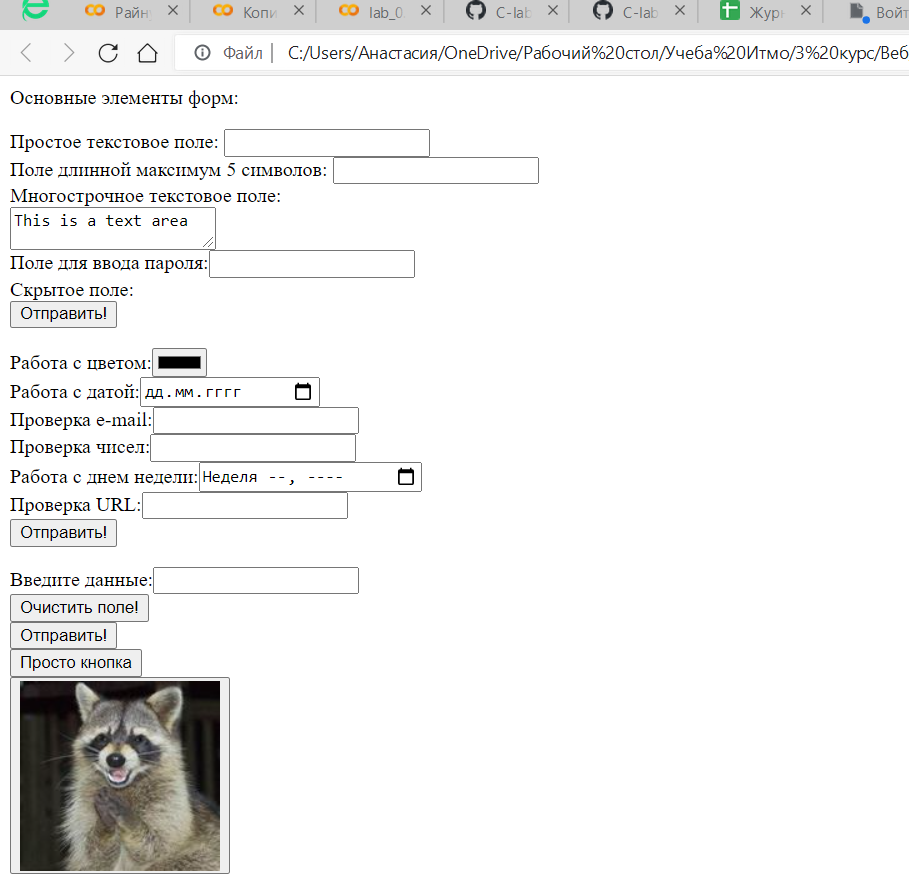
В следующем упражнении в файл были добавлены переключатель (type=”radio”), где можно выбрать только 1 вариант и флажки(type=”checkbox”), где можно выбрать несколько вариантов. Далее с помощью тега <select> было создано поле со списком для выбора элемента (модели машины). Также с помощью тега <optgroup> внутрь списка были добавлены названия марок машин. В конце с помощью атрибута type = "file" тега <input> была добавлена форма с полем для загрузки файлов.

Рисунок 53 - Результат работы

Вторая часть кода представлена на рисунке 54:

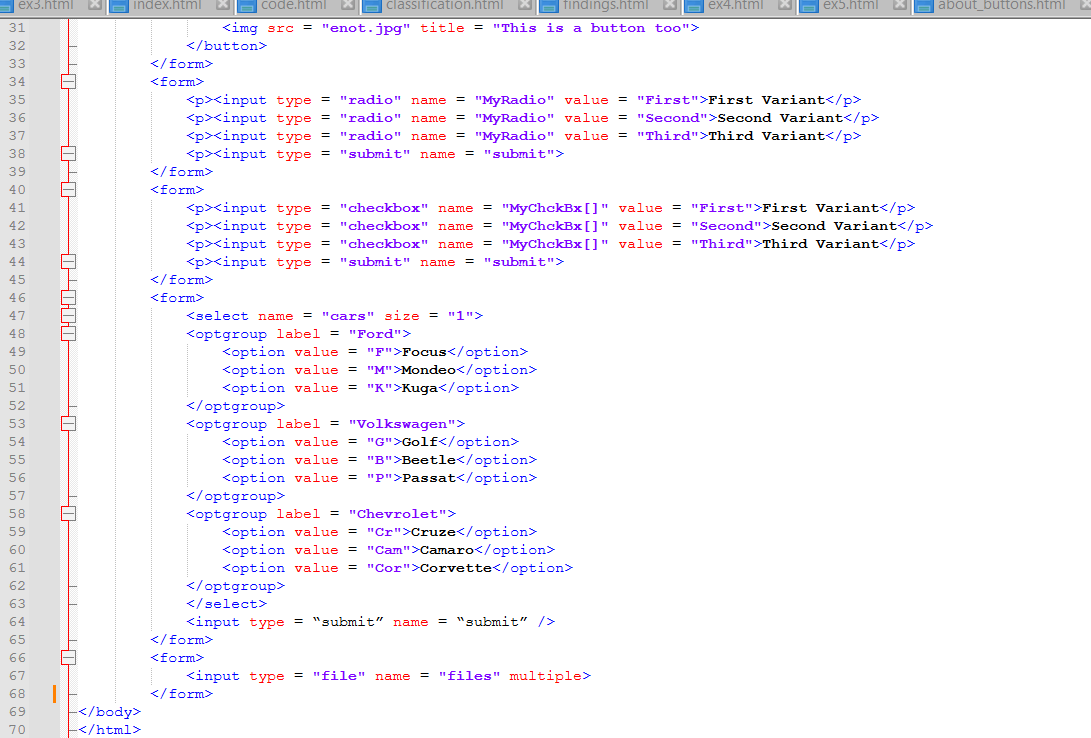
Посмотрим результат работы в браузере (Рисунок 55)

Рисунок 54 - Файл ex7 2 часть

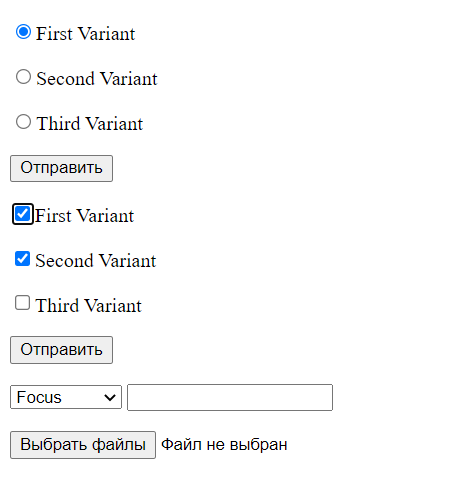


Рисунок 55 - Результат работы

## **7. Основы работы с блочной структурой**

В данном разделе рассматривалась стилевая работа с блочной архитектурой. Были предоставлены две пары html+css файлов. Сайты давали одинаковый результат, но структурно отличались. В первой паре использовались теги div и в атрибутах были указаны их классы, которые в css стилистически оформлялись. Во второй паре не использовались классы, а css обращался к ним с помощью изменения свойств html-тега

## **Вывод**

В ходе данной работы были изучены основы языков HTML (основные теги, структура HTML файла, специальные теги для работы с разными типами элементов, блочная структура HTML страницы) и CSS (стилевые атрибуты) для создания веб-страниц. Цель достигнута.