Санкт-Петербургский Национальный Исследовательский Университет Информационных Технологий, Механики и Оптики

Факультет инфокоммуникационных технологий

Отчет по дисциплине: Web-программирование

**Лабораторная работа 1.**

Выполнила:

Иванова Анна Сергеевна

Группа: K3322

Проверила:

Марченко Е.В.

Санкт-Петербург

2024

**Цель работы:**

**1. Структура HTML-кода**

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

В среде Notepad++ создадим файл с кодом и сохраним его как ex1.html (Рисунок 1).

Изображение выглядит как текст, снимок экрана, дисплей, программное обеспечение

Автоматически созданное описание

Рисунок 1 – HTML-код

Откроем файл ex1.html с помощью браузера Microsoft edge, чтобы посмотреть результат работы (Рисунок 2). Далее посмотрим код отображаемой в браузере страницы через браузер (Рисунок 3).

Изображение выглядит как текст, снимок экрана, программное обеспечение, Значок на компьютере

Автоматически созданное описание

Рисунок 2 – Результат работы

Изображение выглядит как текст, снимок экрана, программное обеспечение, Значок на компьютере

Автоматически созданное описание

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).

Изображение выглядит как текст, снимок экрана, программное обеспечение, дисплей

Автоматически созданное описание

Рисунок 4 –Добавление тэгов и комментариев

Сохраним внесенные изменения и посмотрим результат в браузере. Заметим, что видимых изменений не произошло и комментарии в браузере не отображаются. Просмотрим код HTML-страницы и убедимся, что комментарии в коде присутствуют (Рисунок 5).

Изображение выглядит как текст, снимок экрана, программное обеспечение, веб-страница

Автоматически созданное описание

Рисунок 5 – Код HTML-страницы

**Упражнение 3. Атрибуты HTML-тэгов**

Перед закрывающим тэгом </body> в файле ex1.html добавим строку, содержащую гиперссылку на страницу Консорциума Всемирной паутины http://www.w3.org. (Рисунок 5). Сохраним изменения и посмотрим результат в браузере. Перейдём по ссылке, чтобы убедиться, что она работает (Рисунок 6).

Изображение выглядит как текст, программное обеспечение, веб-страница, Значок на компьютере

Автоматически созданное описание

Рисунок 5 – Добавление гиперссылки

Изображение выглядит как текст, снимок экрана, программное обеспечение

Автоматически созданное описание

Рисунок 5 – Страница в браузере

В папку, где располагается файл ex1.html поместим изображение pic.jpg. Добавим строку, состоящую из тэга <img>, предназначенного для отображения изображений в формате GIF, JPEG или PNG, и его атрибутов alt, width и height. Добавим тэг <br/> сразу после гиперссылки, чтобы браузер отображал картинку под гиперссылкой, а не на одной с ней строке. Найдём абзац с текстом «Это текст моей первой страницы» и добавим к открывающему тэгу <p> информационный атрибут title (Рисунок 6). Сохраним изменения.

Изображение выглядит как текст, программное обеспечение, линия, число

Автоматически созданное описание

Рисунок 6 – Код HTML

Откроем страницу в браузере, подведём курсор к абзацу с фразой *«Это текст моей первой страницы»* и убедимся в появлении подсказки у курсора, содержащей значение атрибута title (рисунок 7).

Изображение выглядит как текст, снимок экрана, программное обеспечение, дисплей

Автоматически созданное описание

Рисунок 7 – Страница в браузере

### 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 8).

Изображение выглядит как текст, снимок экрана, программное обеспечение, линия

Автоматически созданное описание

Рисунок 8 – Код файла ex3.html

В файле ex2.html добавим атрибуты изменения цвета фона и текста к тэгу <body>. Добавим атрибут выравнивания абзаца <p>. К этому же элементу <p> добавим вложенный парный устаревший элемент <font> с изменяющими гарнитуру значениями атрибутов (рисунок 9). Посмотрим результат в браузере (рисунок 10).

Изображение выглядит как текст, снимок экрана, программное обеспечение, линия

Автоматически созданное описание

Рисунок 9 – Код файла ex2.html

Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Автоматически созданное описание

Рисунок 10 – Результат кода ex2.html

В файле ex3.html используем каскадную таблицу стилей CSS. Добавим в тэги <body> и <p> код стилевого оформления: <body **style = "background-color:SeaGreen; color:#800080"**> <p **style = "font:3em Arial; text-align:center"**> (рисунок 11). Сохраним изменения и откроем данный файл в браузере (рисунок 12).

Изображение выглядит как текст, снимок экрана, программное обеспечение, линия

Автоматически созданное описание

Рисунок 11 – Код файла ex3.html

Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Автоматически созданное описание

Рисунок 12 – Результат кода ex3.html

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

В редакторе откроем файл ex1.html. Добавим в любом новом абзаце перенос строк с помощью клавиши «Enter», а также произвольное количество идущих подряд пробельных символов (рисунок 13). Сохраним и просмотрим результат в браузере (рисунок 14).

Изображение выглядит как текст, снимок экрана, программное обеспечение, дисплей

Автоматически созданное описание

Рисунок 13 – Добавление абзацев и пробельных символов

Изображение выглядит как текст, снимок экрана, программное обеспечение, веб-страница

Автоматически созданное описание

Рисунок 14 – Текст в браузере

Заметим, что при отображении текста в браузере ничего не изменилось – HTML не поддерживает переход на другую строку с помощью клавиши «Enter», а любое количество идущих подряд пробелов в браузере отображается как один.

Вернёмся к открытому в Notepad++ исходному коду текста и заменим в абзаце с созданными ранее пробелами и переносами текстом тэги <p>…</p> на тэги <pre>…</pre> соответственно, не меняя текст абзаца. Просмотрим результат в браузере и убедимся, что количество отображаемых в браузере пробельных символов и переходов на новую строку теперь совпадает с количеством пробельных символов и переходов в исходном документе (рисунок 15).

Изображение выглядит как текст, снимок экрана, программное обеспечение, веб-страница

Автоматически созданное описание

Рисунок 15 – Измененный текст в браузере

В случае наличия длинных строк или написания всего текста в виде одной длинной строки в исходном HTML-коде, браузер разместит текст по ширине окна. Переносы текста будут добавлены автоматически в местах пробела. Если в тексте не будет пробелов, то в окне браузера появятся полосы прокрутки (рисунок 16).

Изображение выглядит как снимок экрана, текст, программное обеспечение, Значок на компьютере

Автоматически созданное описание

Рисунок 16 – Полосы прокрутки в браузере

### Упражнение 6. Использование спецсимволов

Откроем файл ex1.html и перед закрывающим тэгом </body> добавим следующий текст: <p> Цену товаров компании **&quot;**QWERTY **&trade;&quot;** необходимо пересчитать с учетом курса **&euro;** к **&pound;** **&plusmn;** наценка для региона</p></br> (рисунок 17).

Сохраним изменения, просмотрим результат в браузере (рисунок 18).

Изображение выглядит как текст, снимок экрана, линия, программное обеспечение

Автоматически созданное описание

Рисунок 17 – Полосы прокрутки в браузере

Изображение выглядит как текст, снимок экрана, программное обеспечение, веб-страница

Автоматически созданное описание

Рисунок 18 – Полосы прокрутки в браузере

**2. Основы логического форматирования контента**

### Упражнение 1. Использование заголовков

В редакторе Notepad++ создадим новую страницу с именем index.html, код представлен на рисунке 19.

Изображение выглядит как текст, электроника, снимок экрана, дисплей

Автоматически созданное описание

Рисунок 19 – Код файла index.html

Сохраним файл и увидим результат, отображаемый в браузере (рисунок 20).

Изображение выглядит как текст, снимок экрана, программное обеспечение, Значок на компьютере

Автоматически созданное описание

Рисунок 20 – Использование заголовков разного уровня

### Упражнение 2. Возможности логического форматирования текста

В файле index.html, добавим некоторые из тэгов, выделяющие текст с целью его специального обособления (рисунок 21).

Изображение выглядит как текст, линия, число, программное обеспечение

Автоматически созданное описание

Рисунок 21 – Выделяющие текст теги

Сохраним изменения и посмотрим на результат в браузере (рисунок 22).

Изображение выглядит как текст, снимок экрана, программное обеспечение, веб-страница

Автоматически созданное описание

Рисунок 22 – Результат в браузере

### Упражнение 3. Ссылки, цитаты, определения

Добавим в код теги <dfn> (используется для определения термина), <blockquote> (предназначен для выделения цитат), <address> (предназначен для представления контактной информации). Сохраним изменения (рисунок 23). Посмотрим изменения в браузере (рисунок 24).

Изображение выглядит как текст, снимок экрана, программное обеспечение, Шрифт

Автоматически созданное описание

Рисунок 23 – Добавление новых тегов

Изображение выглядит как текст, снимок экрана, программное обеспечение, веб-страница

Автоматически созданное описание

Рисунок 24 – Результат в браузере

### Упражнение 4. Элементы компьютерного кода

Для отформатированного представления элементов, связанных с компьютерным кодом, используем тэги <kbd>, <samp>, <code>, <var>. Тэг <kbd> используется для обозначения текста, который вводится с клавиатуры, <samp> - для представления примера вывода программы или системы, <code> - для выделения кода, <var> - для обозначения переменных. В файле code.html напишем код, используя теги выше (рисунок 25). Сохраним результат и посмотрим, как выглядит страница в браузере (рисунок 26).

Изображение выглядит как текст, снимок экрана, программное обеспечение, Значок на компьютере

Автоматически созданное описание

Рисунок 24 – Элементы компьютерного кода

Изображение выглядит как текст, снимок экрана, программное обеспечение, веб-страница

Автоматически созданное описание

Рисунок 25 – Элементы компьютерного кода в браузере

### Упражнение 5. Маркированные списки

Создадим новый файл classification.html***.*** В тело документа добавим фрагмент кода, создающий маркированный список с помощью средств HTML (рисунок 26). При таком подходе маркеры могут быть определены как *disc*, *circle* и *square*.

Изображение выглядит как текст, Шрифт, снимок экрана, линия

Автоматически созданное описание

Рисунок 26 – Отображение маркированного списка

Изменим маркеры списка с помощью CSS, для этого заменим созданный ранее код следующим кодом CSS (рисунок 27).

Изображение выглядит как текст, Шрифт, снимок экрана, линия

Автоматически созданное описание

Рисунок 27 – Отображение маркированного списка

Теперь изменим отдельные элементы списка с помощью CSS (рисунок 28).

Изображение выглядит как текст, Шрифт, снимок экрана, линия

Автоматически созданное описание

Рисунок 28 – Изменение отдельного элемента

Сохраним изменения и убедимся, что элемент списка стал отображаться красным цветом (рисунок 29).

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Рисунок 29 – Отображение элементов в браузере

### Упражнение 6. Нумерованные списки

На странице classification.htmlпосле маркированного списка введём HTML-код для создания нумерованного списка (рисунок 30). Для изменения начального номера пунктов списка добавим к тэгу <ol> атрибут start, а для изменения типа нумерации пунктов списка к тэгу <ol> добавим еще один атрибут – type.

Изображение выглядит как текст, снимок экрана, Шрифт, число

Автоматически созданное описание

Рисунок 30 – Нумерованный список

Сохраним изменения и посмотрим результат в браузере (рисунок 31).

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Рисунок 31 – Отображение элементов в браузере

### Упражнение 7. Список определений

На странице classification.htmlпосле нумерованного списка введём HTML-код, представленный на рисунке 32.

Изображение выглядит как текст, Шрифт, снимок экрана, линия

Автоматически созданное описание

Рисунок 32 – Список определений

Обновим страницу и посмотрим на результат (рисунок 33).

Изображение выглядит как текст, Шрифт, снимок экрана, алгебра

Автоматически созданное описание

Рисунок 33 – Отображение списка определений в браузере

### Упражнение 8. Вложенные списки

Создадим новый файл findings.html. Организуем в нём многоуровневый список (рисунок 34), который бы отображался в браузере как представлено на рисунке 35.

Изображение выглядит как текст, снимок экрана, Шрифт, число

Автоматически созданное описание

Рисунок 34 – Создание вложенного списка

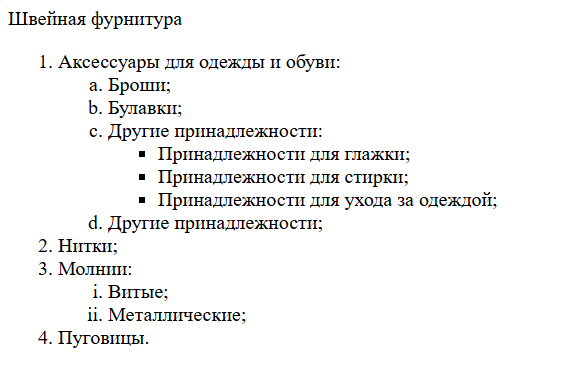


Рисунок 35 – Многоуровневый список

**3. Гиперссылки**

**Упражнение 1. Основы работы с гиперссылками**

Создадим две гиперссылки: одну со страницы index.html на страницу classification.html, вторую – ведущую обратно с classification.html на index.html. Для создания гиперссылок введём HTML текст, представленный на рисунке 36.



Рисунок 36 – Создание гиперссылки

На странице index.html создадим ссылку на страницу в википедии, а также на поисковую систему Яндекс. Дополним только что созданную гиперссылку атрибутом target (рисунок 37). Проверим работоспособность всех ссылок.

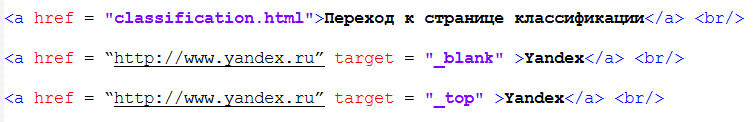


Рисунок 37 – Гиперссылки с атрибутом target

**Упражнение 2. Работа с цветом гиперссылок**

Создадим файл ex4.html, где разместим HTML текст со ссылкой на Yandex. Для изменения цветовых решений гиперссылки используем HTML. В этом случае цвета ссылок зададим в качестве атрибутов тега <body>. В код файла ex4.html добавим строку: <body alink="Fuchsia" vlink="Aqua" >. Обновим страницу и убедимся, что гиперссылка работает.

Рассмотрим более актуальный вариант с использованием CSS. В нем можно установить следующие основные визуальные решения: visited — стиль посещенной ссылки; active — стиль активной ссылки; hover — стиль для ссылки при наведении на нее мышью. Изменим код файла ex4.html (рисунок 38).

**Изображение выглядит как текст, снимок экрана, Шрифт, число

Автоматически созданное описание**

Рисунок 38 – Изменение цветовых решений гиперссылки

Сохраним изменения, просмотрим их результат в браузере. Пройдём по гиперссылке и убедимся в изменении цветовых решений (рисунок 39).

Изображение выглядит как текст, снимок экрана, Шрифт, дизайн

Автоматически созданное описание

Рисунок 39 – Цвет гиперссылки в браузере

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

В файле ex4.htmlв произвольном месте тела документа создадим ссылку вида: <a href="mailto:myAdress@domain.ru">Отправьте мне письмо</a>

Сохраним изменения, просмотрим результат работы в браузере. Убедимся, что при нажатии на гиперссылку вызывается почтовый клиент и открывается окно создания нового сообщения, при этом в поле получателя уже указан адрес, прописанный в гиперссылке (рисунок 40).

Изображение выглядит как текст, снимок экрана, Шрифт, диаграмма

Автоматически созданное описание

Рисунок 40 – Результат работы в браузере

**Упражнение 4. Внутренние ссылки документа**

Для организации навигации в рамках одного документа можно создать один или несколько якорей (т.е. внутренних ссылок документа). Создадим файл about\_buttons.html с большим количеством текста в редакторе. Найдём в тексте файла два парных заголовка <h2>…</h2>, заголовки содержат текст «История первая» и «История вторая» соответственно. В заголовки добавим метки вида: <h2 id = "history\_1">История первая</h2> и <h2 id = "history\_2">История вторая</h2>

Перейдём в верхнюю часть текущего документа и добавим простейшее меню для навигации по тексту страницы (рисунок 41).

Изображение выглядит как текст, Шрифт, снимок экрана, линия

Автоматически созданное описание

Рисунок 41 – Меню навигации по тексту страницы

Сохраним изменения. Откроем страницу в браузере и проверим работу созданных ссылок: при нажатии на ссылку браузер переходит к нужному разделу, перемещая соответствующий текст как можно выше к началу области просмотра экрана.

4. Таблицы

Упражнение 1. Создание таблицы

В рамках данного упражнения был создан файл ex5.html, содержащий код для создания таблицы. Результат выполнения задания можно увидеть на рисунке 42.



Рисунок 42 – Создание таблицы

Упражнение 2. Работа с границами таблицы

В этом упражнении таблица была дополнена границами, а её ширина была задана относительно ширины окна. Результат выполнения представлен на рисунке 43.

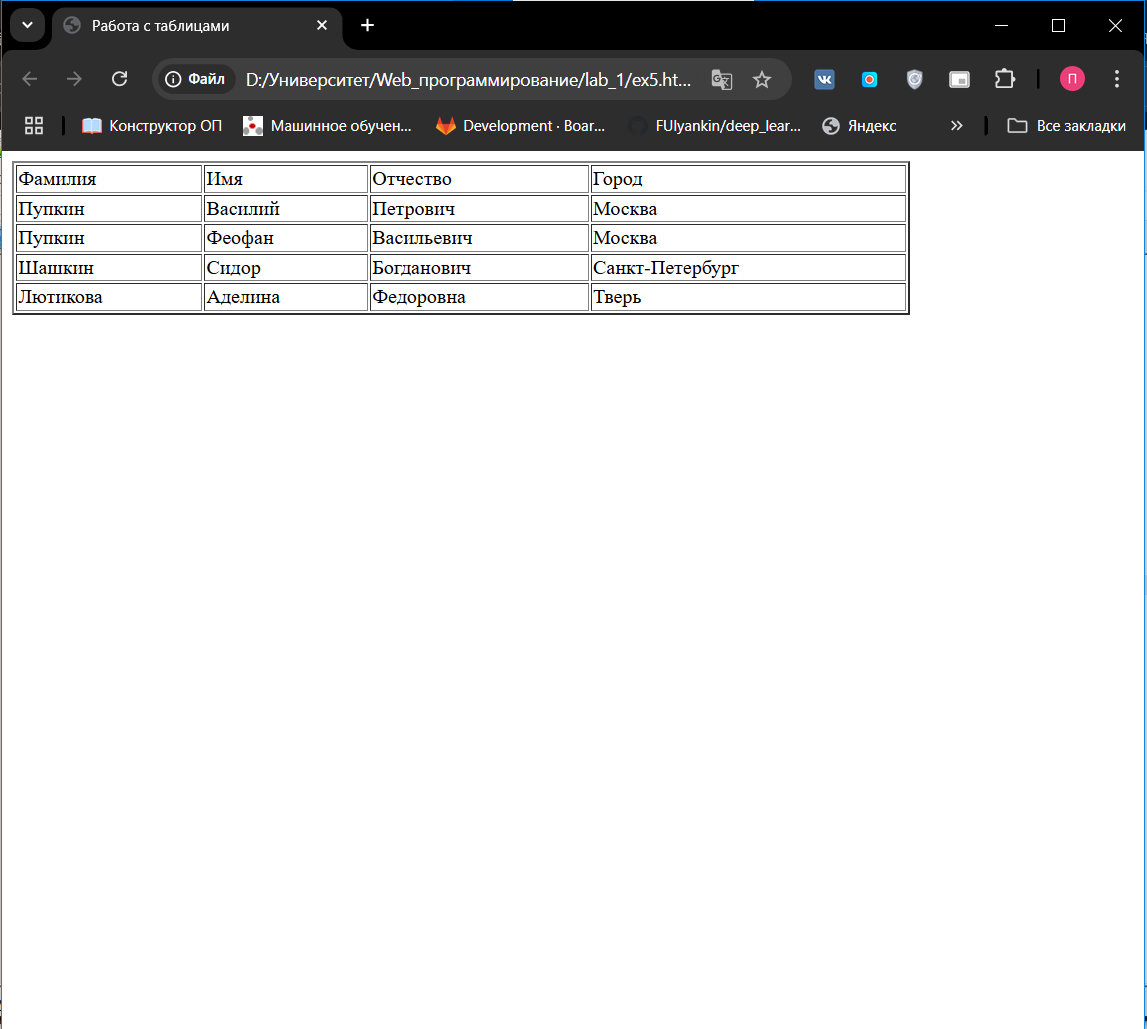


Рисунок 43 – Таблица с границами и установленной шириной

Упражнение 3. Создание заголовка и подписи таблицы

В ходе выполнения упражнения был изменён стиль оформления шапки таблицы, добавлен заголовок, размещённый под таблицей. Итоговый результат отображён на рисунке 44.

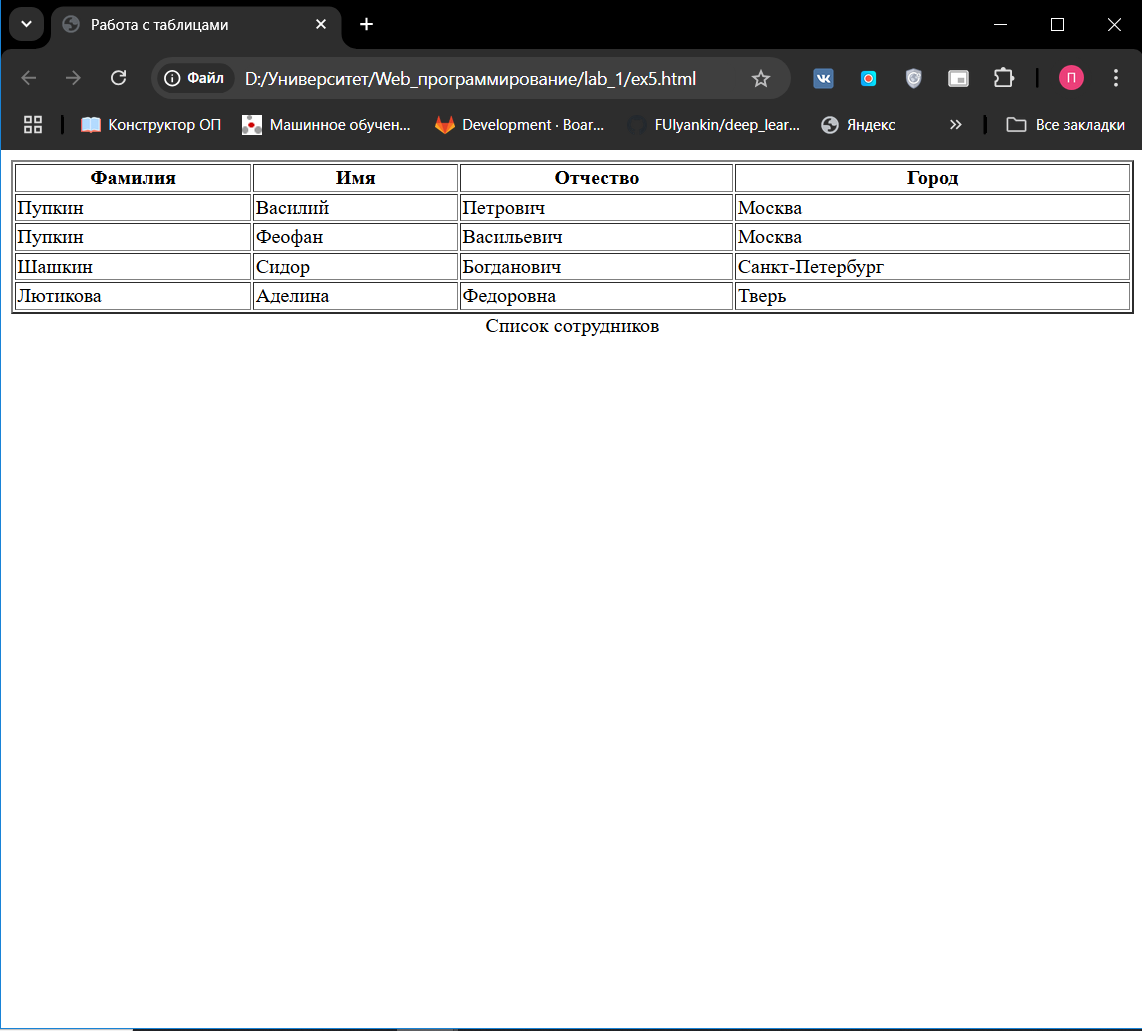


Рисунок 44 – Таблица с измененной шапкой и заголовком

Упражнение 4. Объединение ячеек

В процессе работы ячейки таблицы были объединены по горизонтали и вертикали. Итоговая структура представлена на рисунке 45.

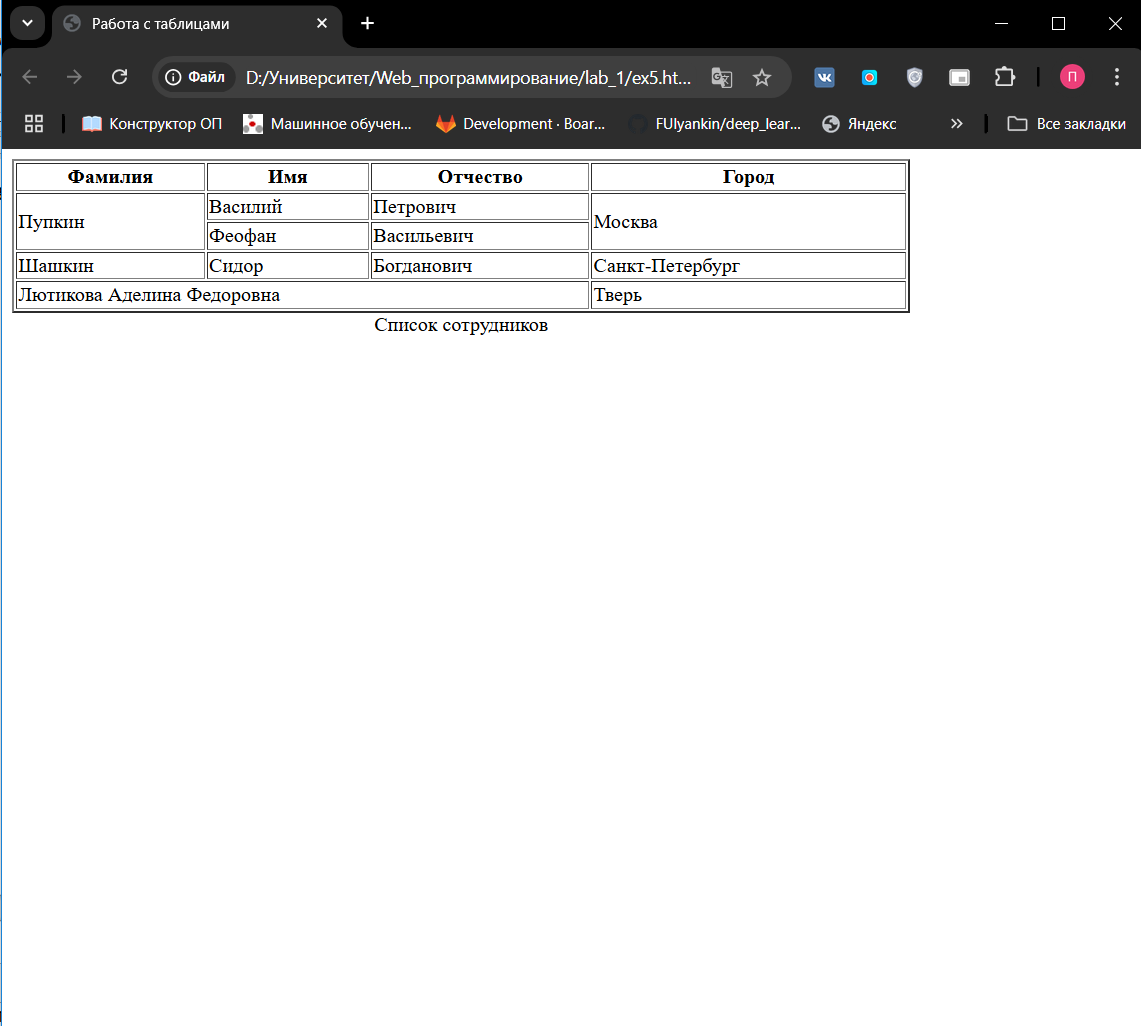


Рисунок 45 – Объединение ячеек

Упражнение 5. Структурные блоки таблиц

В этом упражнении структура таблицы была изменена с помощью тегов <tbody>, <thead> и <tfoot>. Кроме того, для некоторых полей таблицы была настроена ширина и стиль с использованием <colgroup>. Результат проделанной работы представлен на рисунке 46.

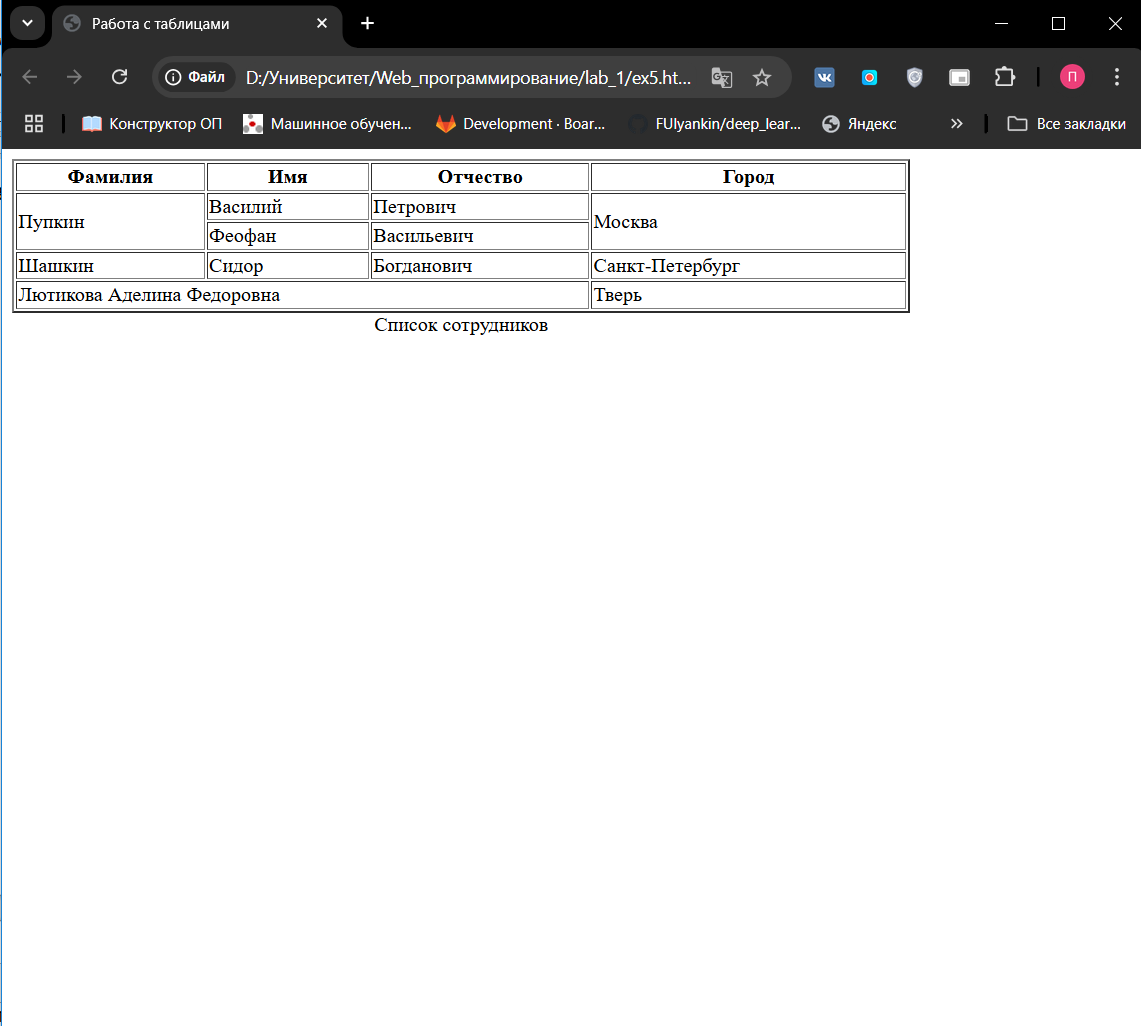


Рисунок 46 – Стилевое оформление таблицы

5. Изображения и медиаконтент

Упражнение 1. Основы работы с изображениями

В файле index.html перед разделом с контактной информацией был добавлен код для размещения изображения, включающий атрибуты альтернативного текста и подсказки. Итог работы отражён на рисунке 47.

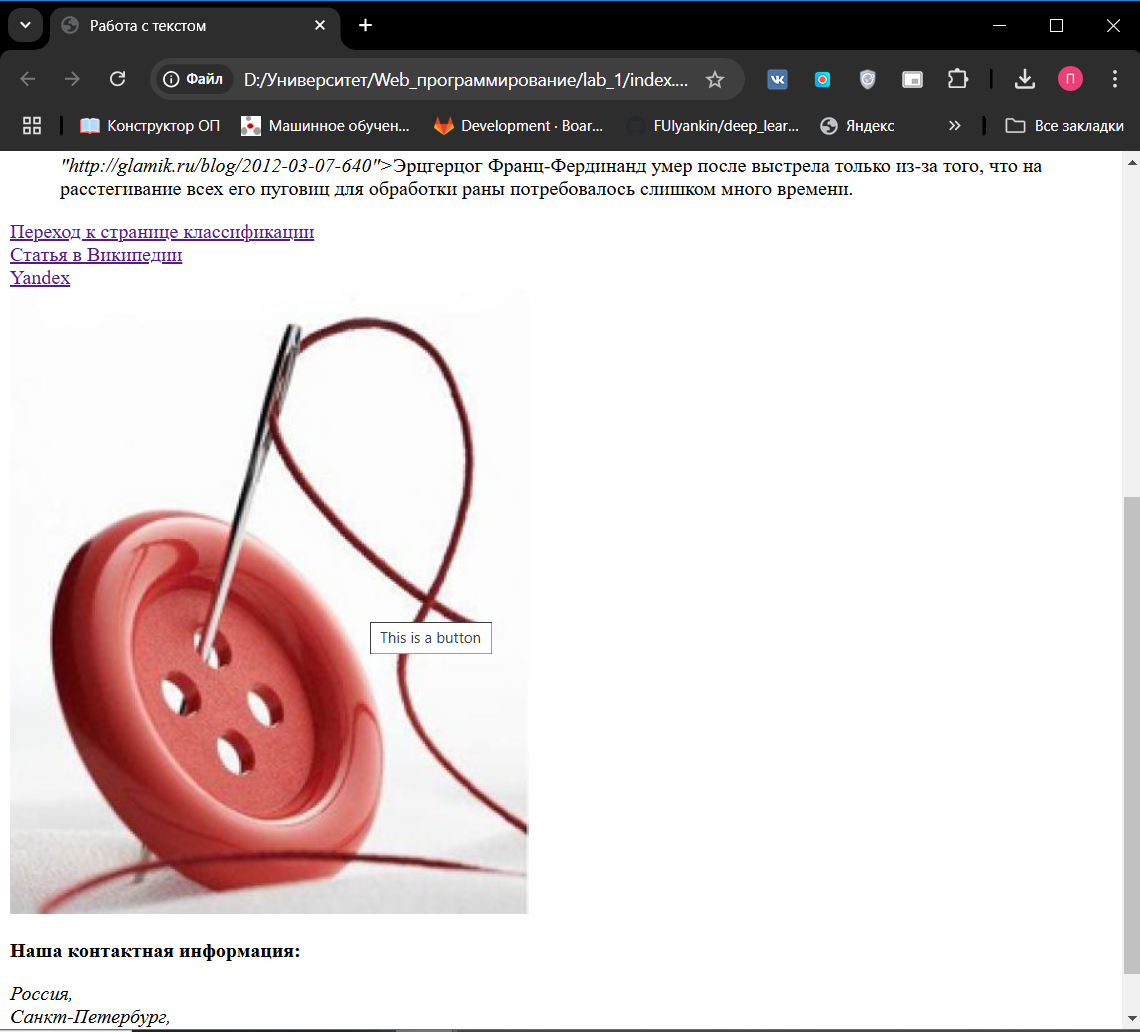


Рисунок 47 – Размещение изображения на странице

Упражнение 2. Изменение размеров изображения

Размер изображения из предыдущего задания был изменён, что показано на рисунке 48.

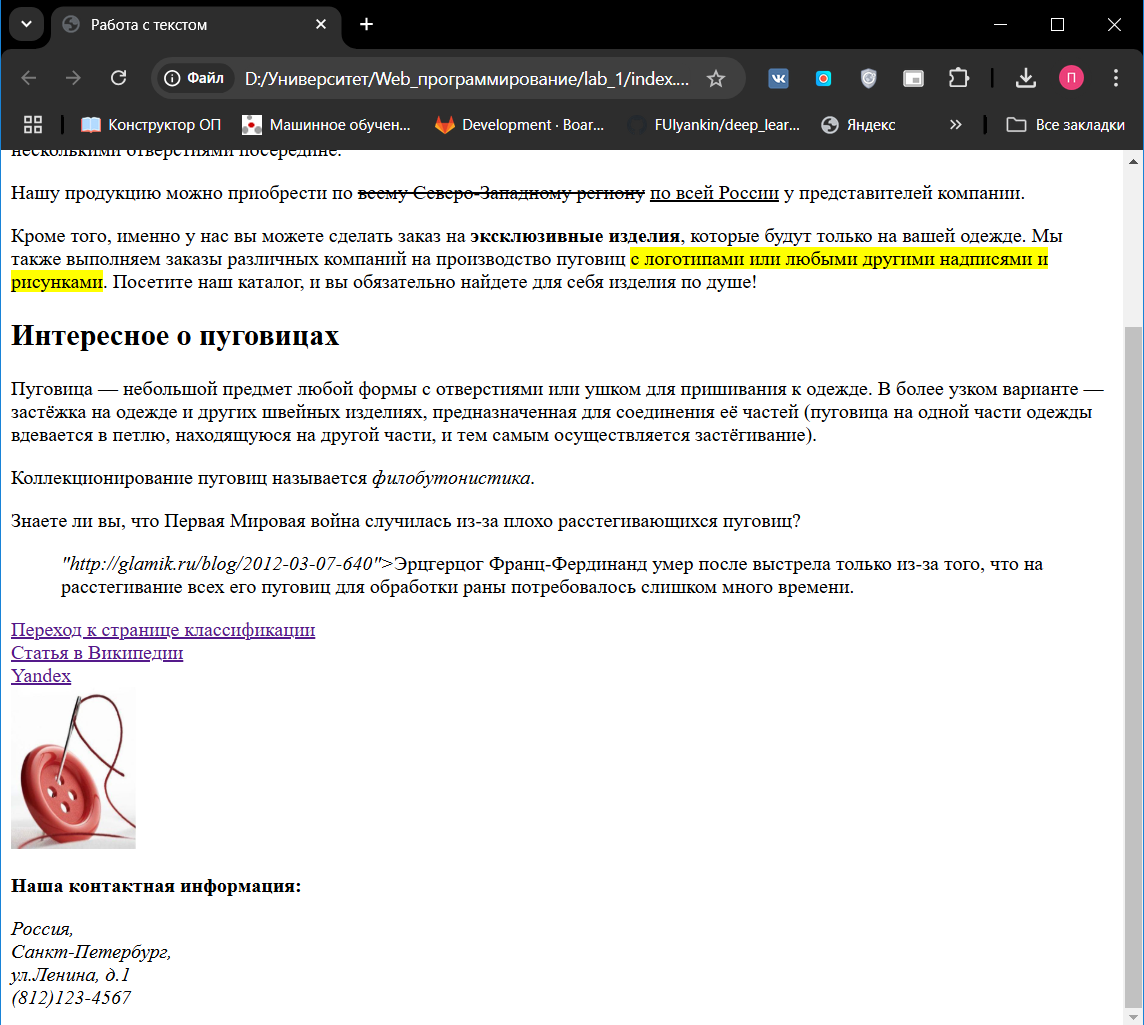


Рисунок 48 – Изменение размера изображения

Упражнение 3. Создание изображения-гиперссылки

Гиперссылка со страницы index.html на страницу classification.html была преобразована в изображение-гиперссылку. Результат показан на рисунке 49.

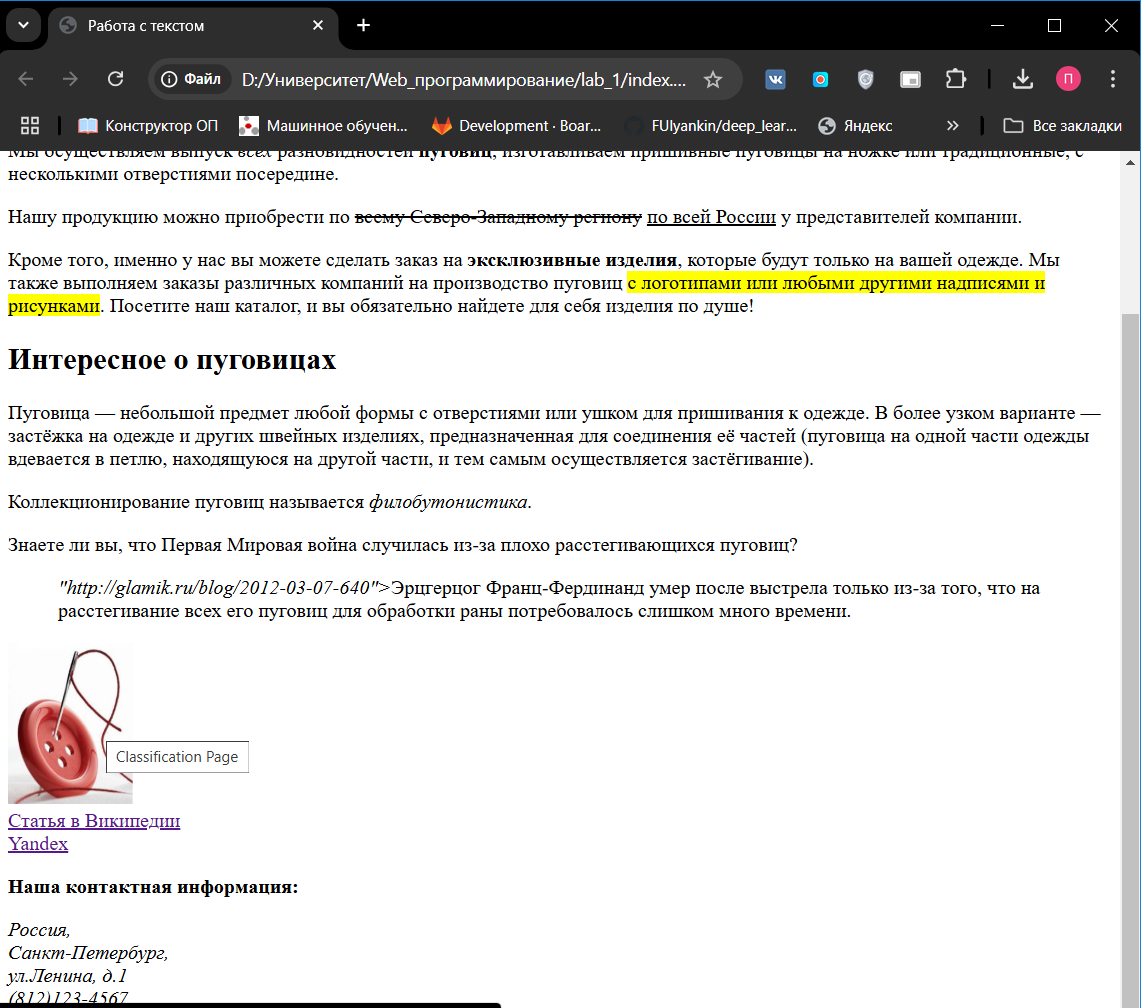


Рисунок 49 – Изображение-гиперссылка

Упражнение 4. Карты изображений

В файле map.html был добавлен код изображения с наложенной картой, определяющей активные области. Этапы добавления областей отображены на рисунке 50.

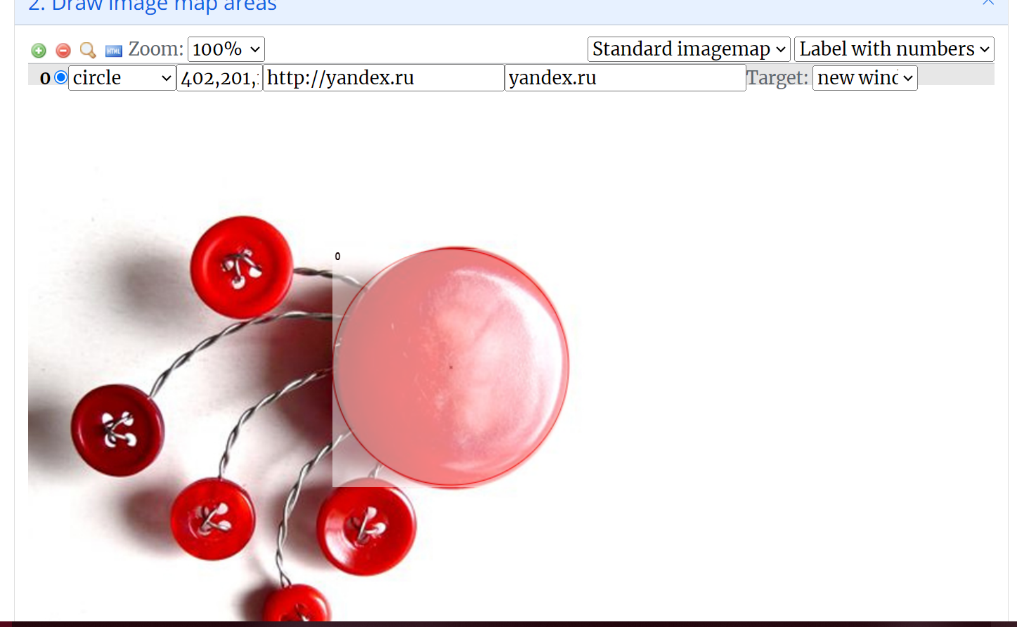


Рисунок 50 – Добавление областей на изображение

Упражнение 5. Добавление медиаконтента

Для добавления видеоконтента на страницу был создан файл ex6.html. Итоговое решение показано на рисунке 51.

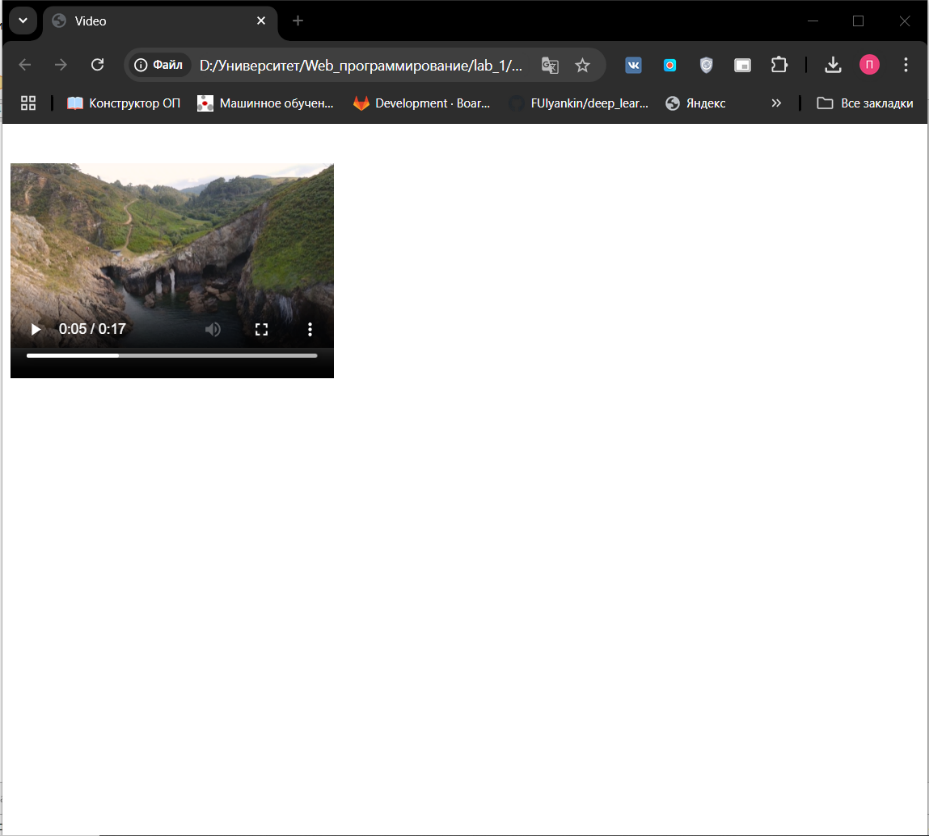


Рисунок 51 – Добавление видеоконтента на страницу

6. Формы

Упражнение 1. Текстовые поля формы

В рамках этого упражнения был создан HTML-файл ex7.html, в который добавлена пустая форма. В форму были включены элементы:

* Однострочные текстовые поля с разными настройками, включая поле с ограничением на ввод не более 5 символов.
* Многострочное текстовое поле (textarea) для ввода объемного текста.
* Поле ввода пароля, скрывающее вводимые символы.
* Скрытое поле, используемое для передачи данных без отображения на странице.

Результат работы представлен на рисунке 52.

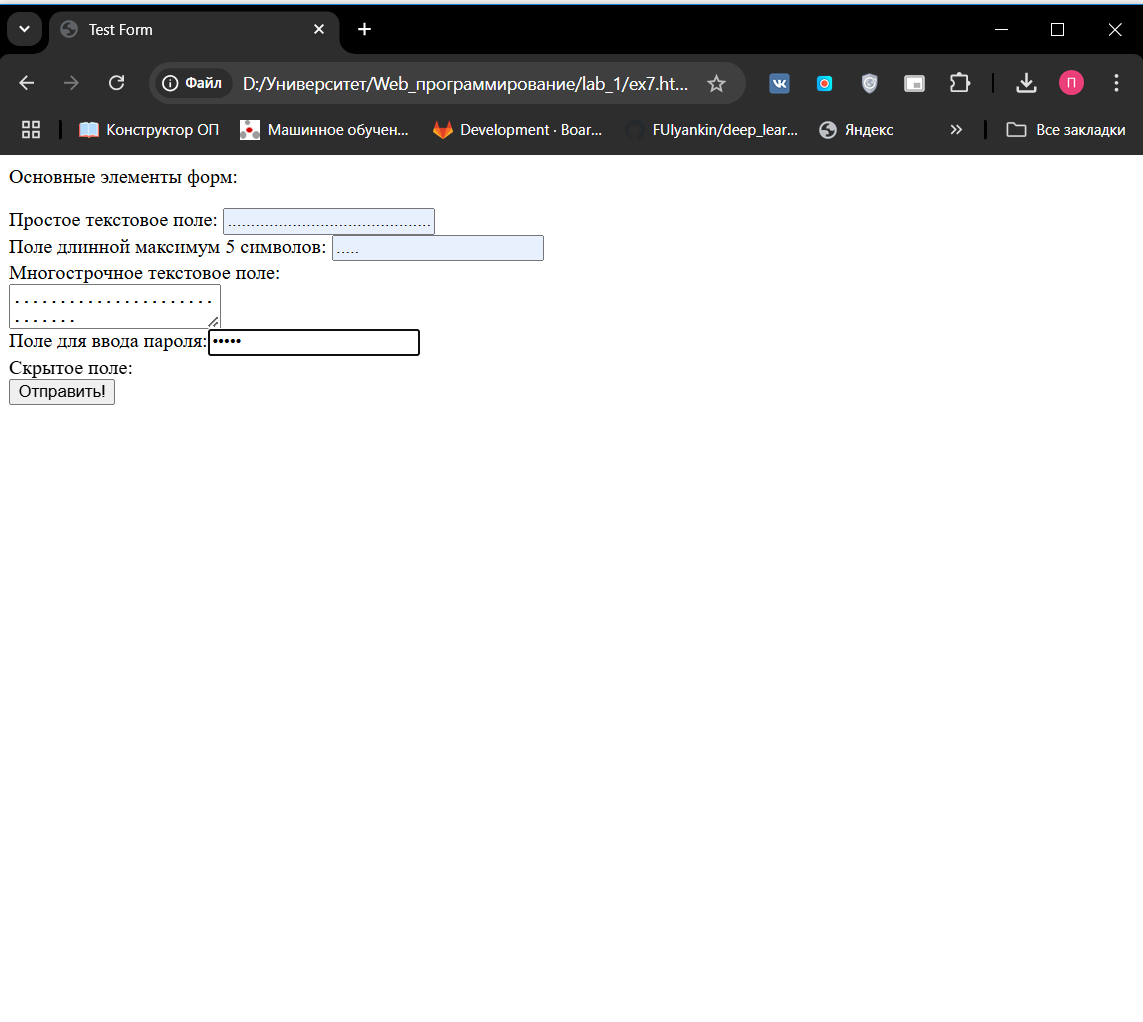


Рисунок 52 – Создание формы

Упражнение 2. Типы полей HTML5

Для исследования новых типов полей HTML5 в файл ex7.html добавлена дополнительная форма с кнопкой отправки данных. Последовательно в нее были включены следующие типы полей:

* Выбор цвета (color).
* Поле ввода даты (date).
* Проверка e-mail (email).
* Ввод числовых данных (number).
* Выбор дня недели (week).
* Проверка URL-адреса (url).

Результат проделанной работы представлен на рисунке 53.

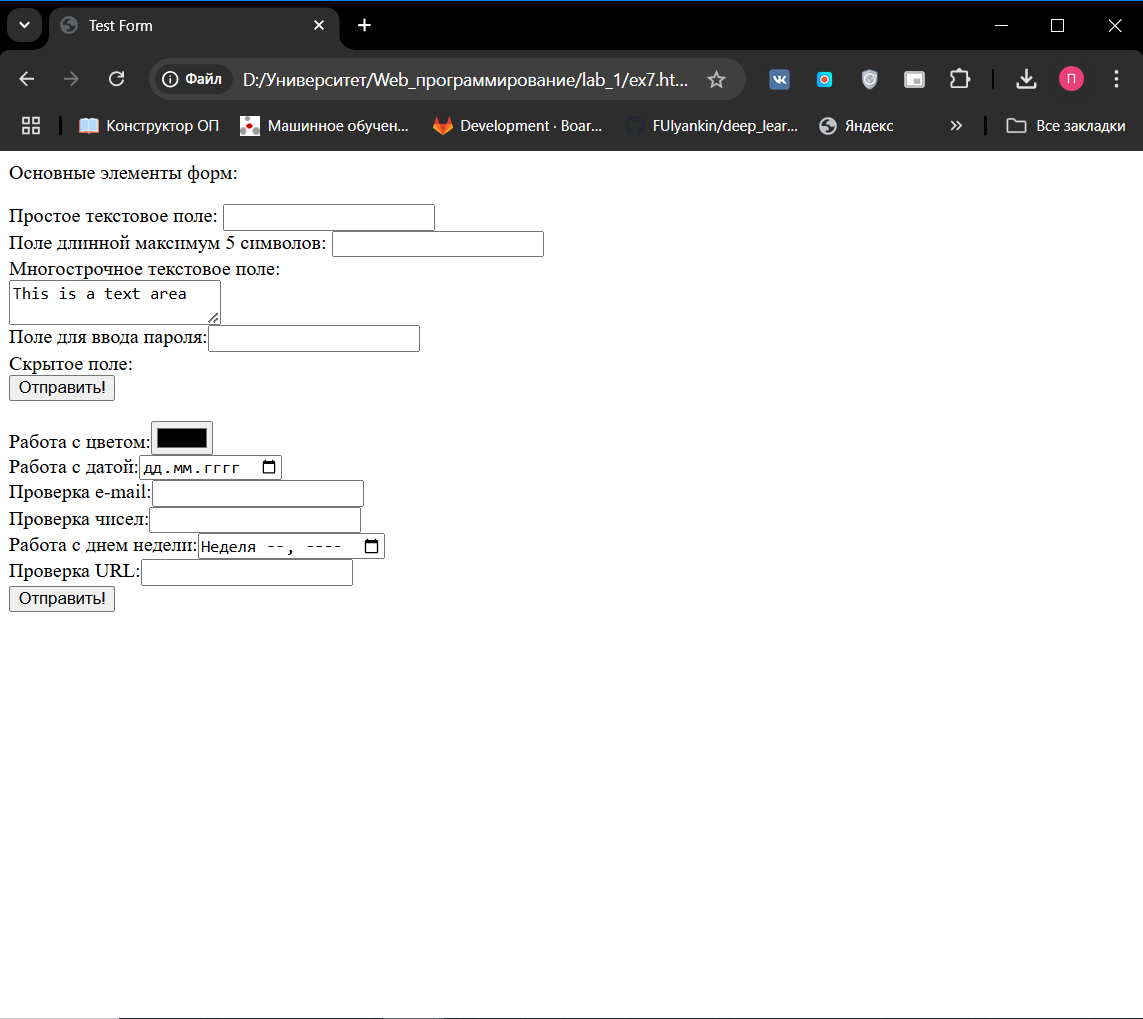


Рисунок 53 – Добавление новых типов полей

Упражнение 3. Кнопки

В этом упражнении была создана новая форма, содержащая различные типы кнопок:

* Кнопка очистки для сброса данных.
* Кнопка отправки данных (submit).
* Простая кнопка (button).
* Кнопка с изображением и всплывающей подсказкой, добавленная с использованием тега <button> и элемента <img>.

Результат работы отображен на рисунке 54.

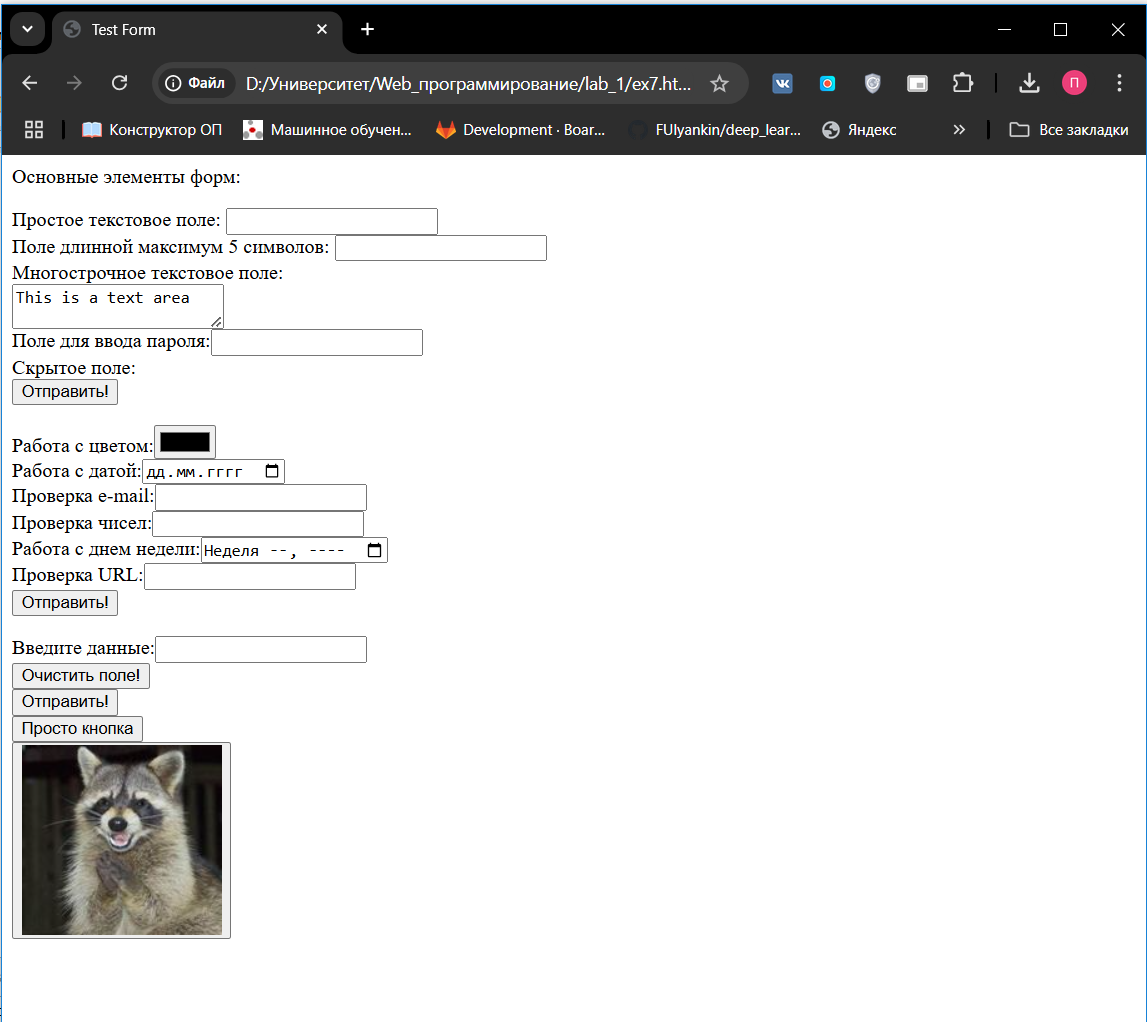


Рисунок 54 – Различные виды кнопок

Упражнение 4. Флажки и переключатели

В файле ex7.html были созданы две формы:

* Форма с переключателями (radio buttons), где элементы одной группы имели одинаковое значение атрибута name и позволяли выбрать только один вариант.
* Форма с флажками (checkboxes), которые позволяют выбрать несколько значений из группы.  
  Обе формы корректно отображались в браузере. Результат работы представлен на рисунке 55.

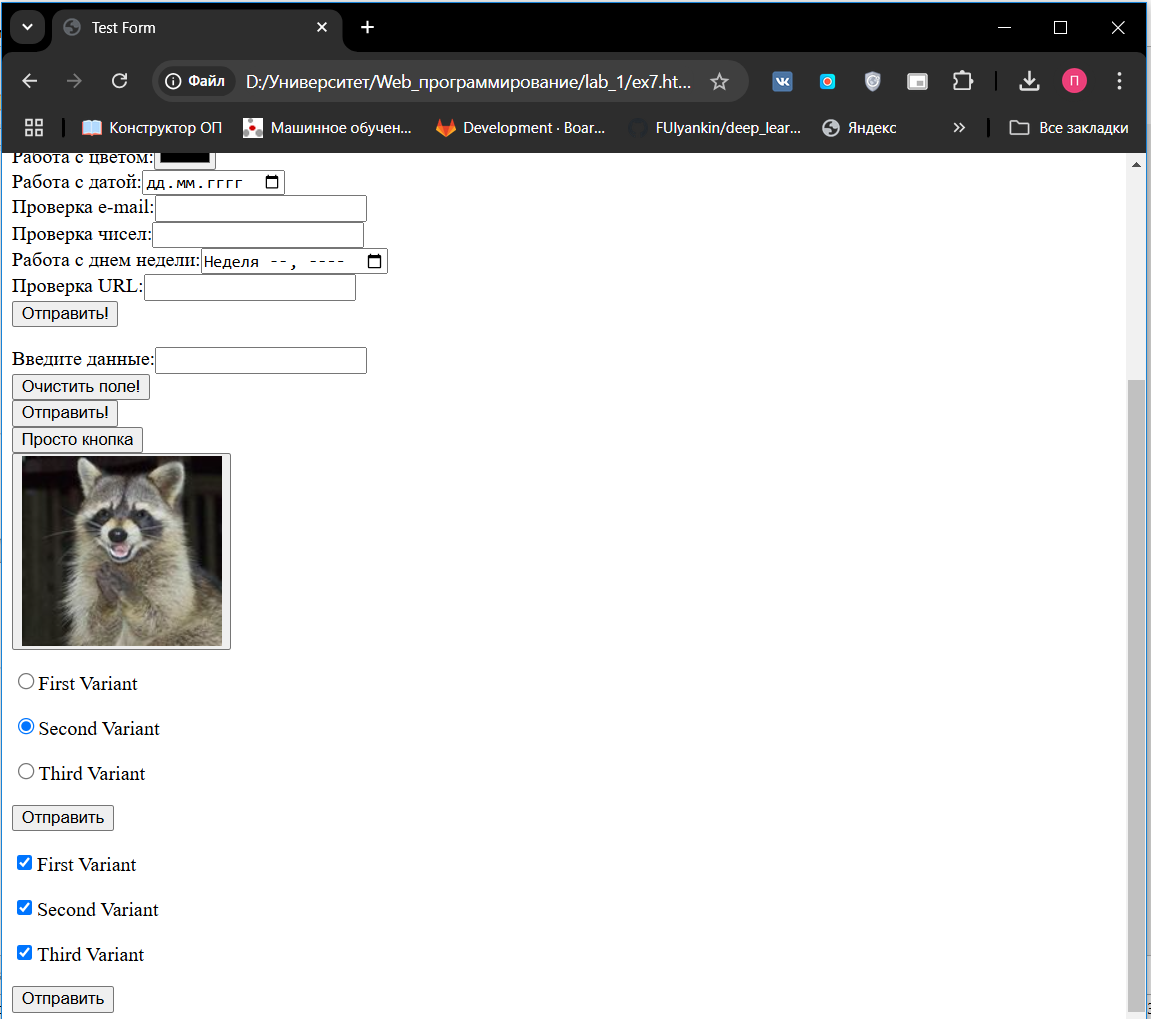


Рисунок 55 – Создание флажков и переключателей

Упражнение 5. Поле со списком

В этом упражнении был создан выпадающий список.

* На первом этапе добавлен простой список с элементами.
* Затем список был разделен на группы (optgroup), что улучшило структуру и удобство использования.

Группировка помогла организовать элементы по категориям, например, "Ford", "Volkswagen", "Chevrolet". Результат работы представлен на рисунке 56.

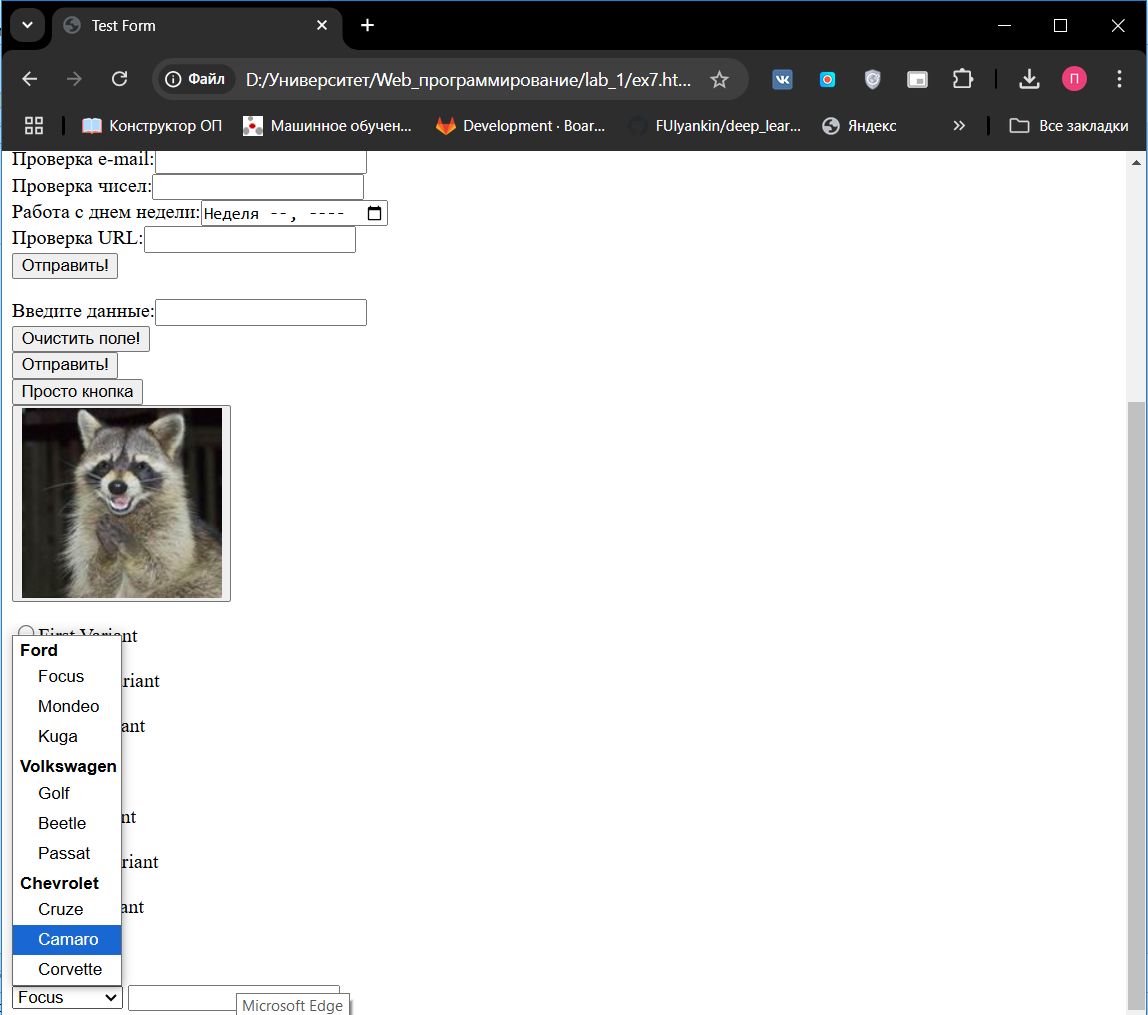


Рисунок 56 – Выпадающий список

Упражнение 6. Поле для загрузки файлов

Добавлено поле для загрузки файлов. Пользователь мог выбрать несколько файлов с помощью файлового менеджера (например, проводника Windows).   
 Результат работы представлен на рисунке 57.

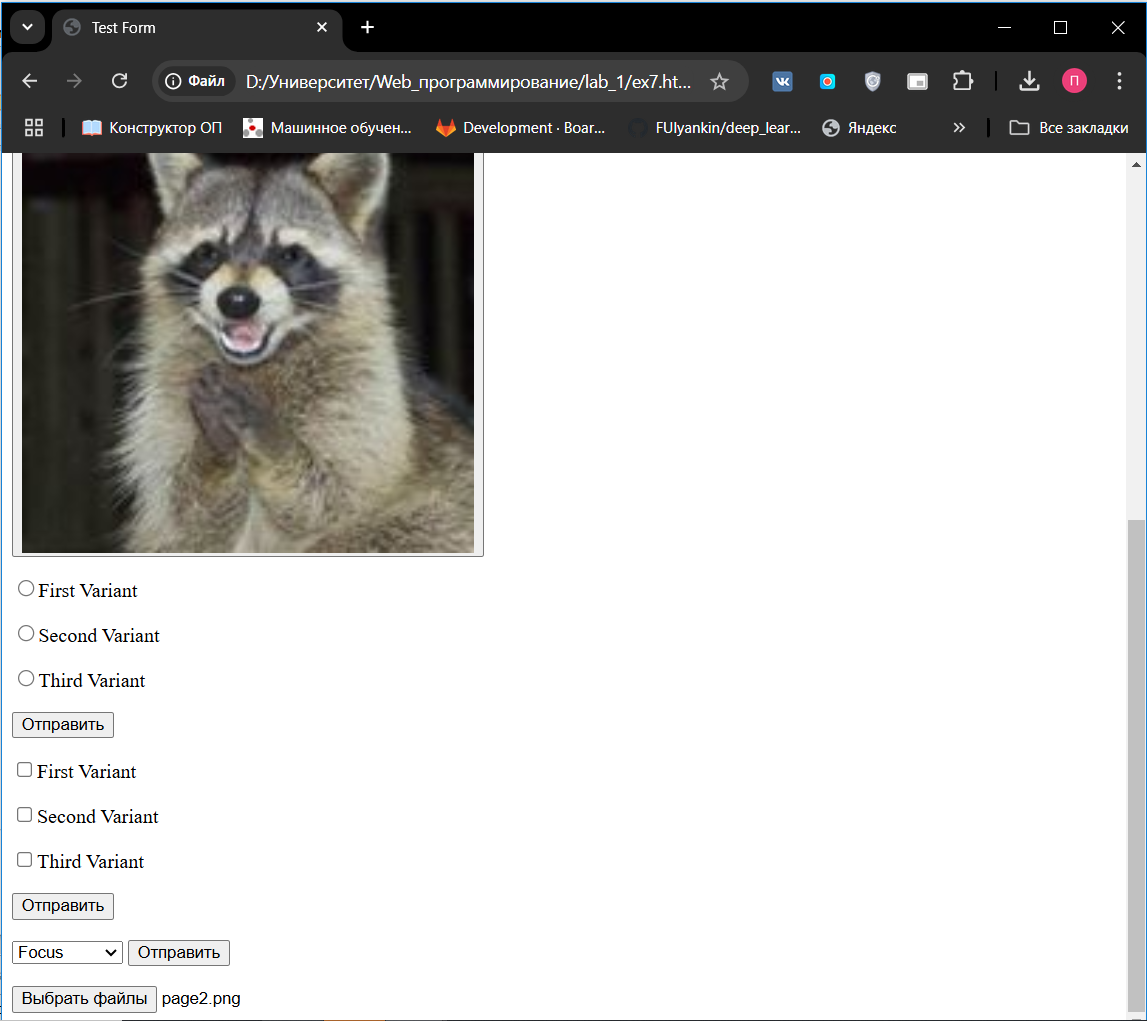


Рисунок 57 – Загрузка файлов

7. Основы работы с блочной структурой

Упражнение 1. Основы организации структуры документа

В файл block\_1.html добавлена ссылка на файл стилей block\_1.css. Стили использовались для задания визуального оформления блочных элементов, таких как <div>. Стили применялись к элементам по идентификаторам (id).

Результат представлен на рисунке 58.

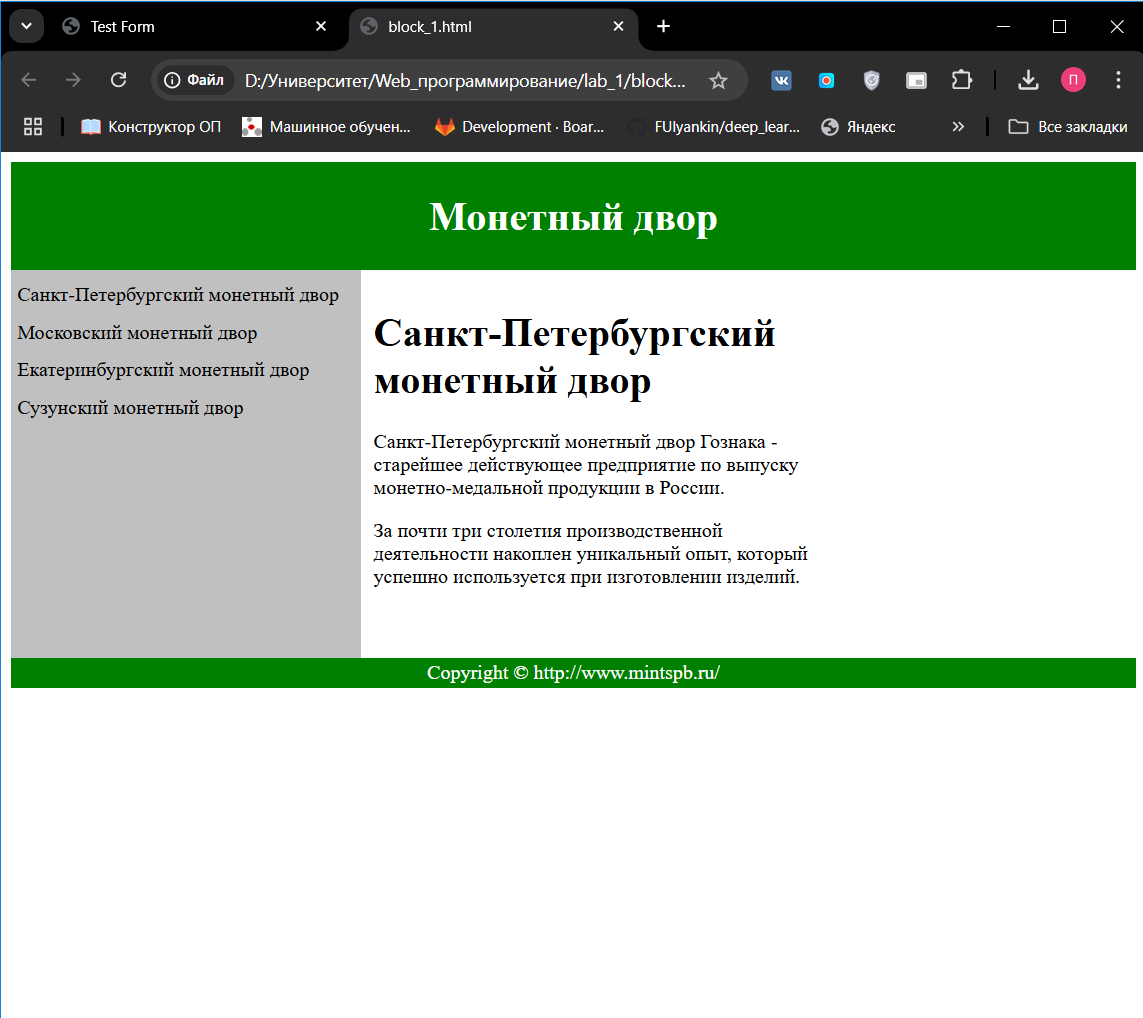


Рисунок 58 – Подключение файла CSS к block\_1.html

Упражнение 2. Работа с разделами документа

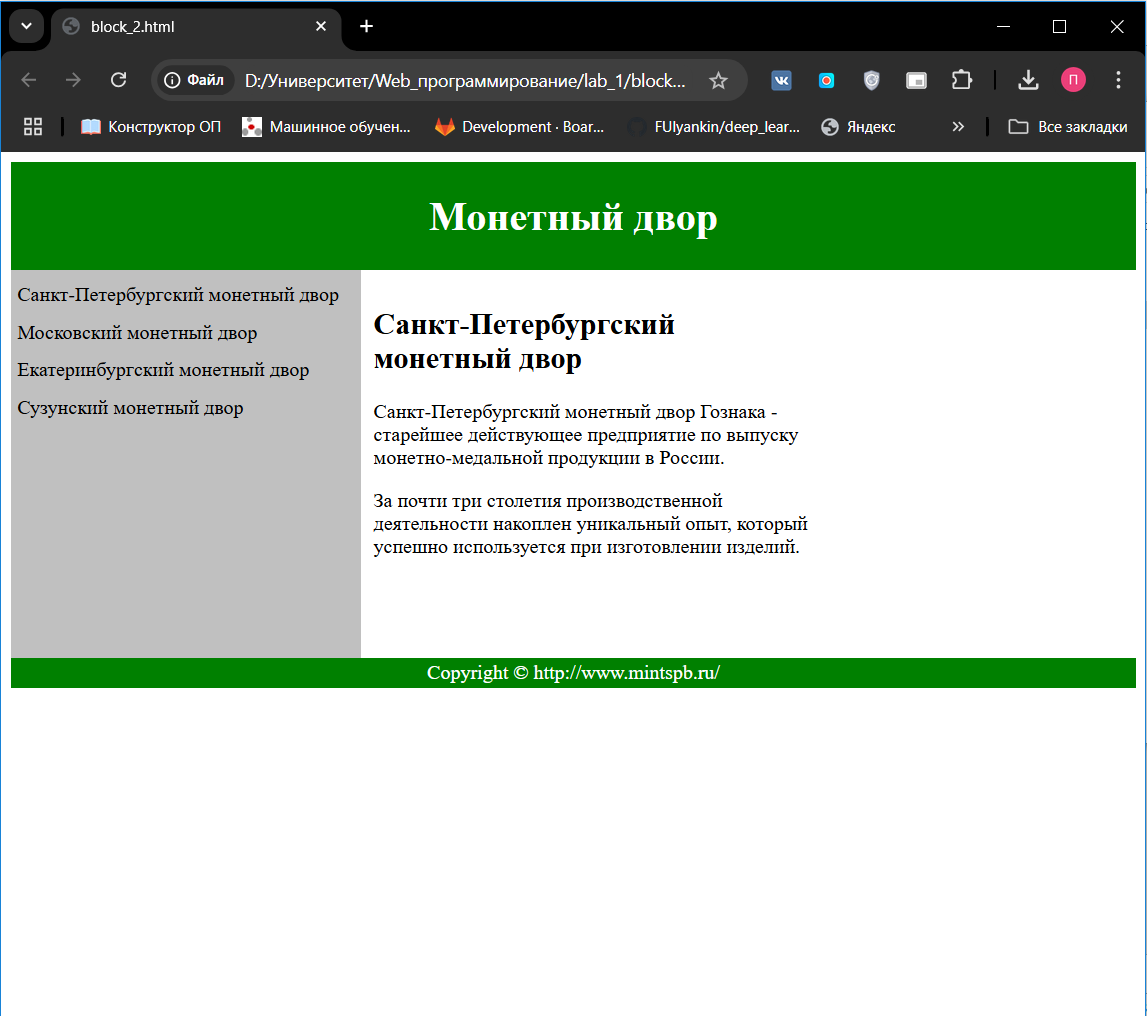
В файл block\_2.html добавлена ссылка на файл стилей block\_2.css. В отличие от предыдущего упражнения, стили применялись ко всем элементам определенного типа, что обеспечило единообразие оформления.  
Результат работы представлен на рисунке 59.

Рисунок 59 – Подключение файла CSS к block\_2.html

Заключение

В результате выполнения лабораторной работы получилось выполнить поставленную цель, так как были изучены базовые возможности, структура и основные элементы HTML.