**Министерство науки и высшего образования Российской Федерации**

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

**НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО**

**ITMO University**

**ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ 1**

**По дисциплине** Web-программирование

**Тема работы** Основы работы с HTML

**Обучающийся** Колтунова Полина Владимировна

**Факультет** Факультет инфокоммуникационных технологий

**Группа** К3320

**Направление подготовки** 11.03.02 Инфокоммуникационные технологии и системы связи

**Образовательная программа** Программирование в инфокоммуникационных системах

**Обучающийся**  Колтунова П.В.

(дата) (подпись) (Ф.И.О.)

**Руководитель**  Марченко Е.В.

(дата) (подпись) (Ф.И.О.)

Санкт-Петербург

2024

ВВЕДЕНИЕ

Цель работы: изучить основы работы с HTML и CSS.

ГЛАВА 1

# Часть 1. Основы работы с HTML

## 1. Структура HTML-кода

### Упражнение 1. Создание первой HTML-страницы

Был написан код HTML, содержащий теги <html>, <head>, <body>, <title> и <p>.

В результате выполнения упражнения была получена web-страница, представленная на рис.1.

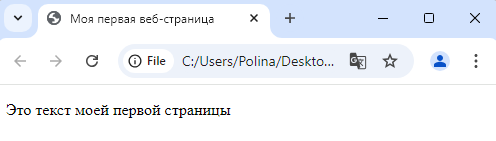


Рисунок 1 - Первая страница

Просмотр кода страницы в браузере, представлен на рис.2.

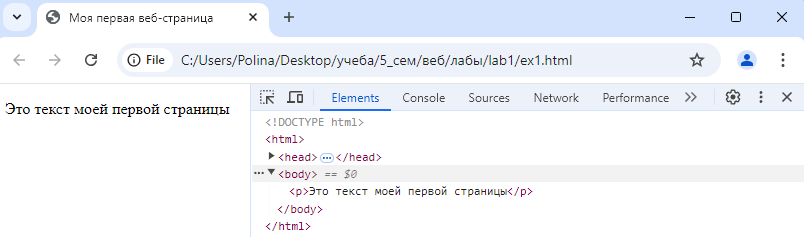


Рисунок 2 - Просмотр кода страницы из браузера

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

После добавления тега верхнего уровня <meta charset=”utf-8">, использующегося для задания кодировки символов, используемой на странице, видимых изменений со страницей не произошло, что можно увидеть на рис.3.

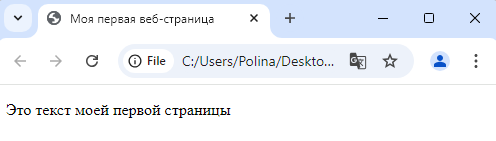


Рисунок 3 - Страница после добавления тега верхнего уровня

Просмотр кода страницы в браузере, представлен на рис.4.

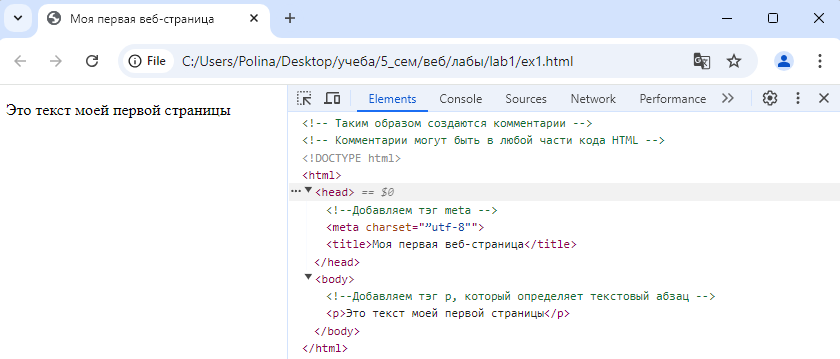


Рисунок 4 - Просмотр кода страницы из браузера

### Упражнение 3. Атрибуты HTML-тэгов

Был добавлен тег <a> гиперссылки с атрибутом href, содержащего в качестве значения адрес ссылки.

Также был добавлен тег <img>, отображающий изображение в формате .jpg с атрибутами alt, width и height, которые выводит альтернативный текст для изображения, и задают ширину и высоту изображения соответственно.

И был добавлен тег </br>, устанавливающий перевод строки в том месте, где находится.

После добавления гиперссылки, изображения, тега перевода строки и информационного атрибута страница приняла вид, представленный на рис.5.

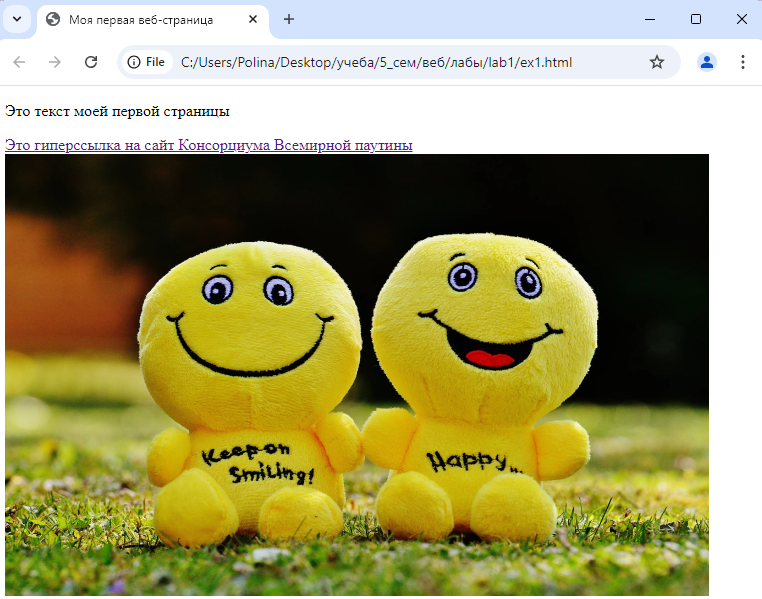


Рисунок 5 - Страница с рисунком и гиперссылкой

### Упражнение 4. Устаревшие HTML-атрибуты

Было произведено сравнение устаревших атрибутов bgcolor, align и вложенного парного атрибута <font face>, что применяет инлайновое форматирование, затрудняющее поддержку и изменение стилей, с актуальным стилевым оформлением, использующим CSS для стилизации через атрибут style, что позволяет более гибко и удобно управлять стилями и разделяет структуру и оформление.

На рис.6 страница с устаревшим решением, на рис.7 страница с актуальным решением. Отличие заключается в размере отступа перед и после текста из-за сочетания стандартных стилей браузеров, каскадности и наследования стилей, а также конфликтов между устаревшими атрибутами и современными CSS.

Сравнение результатов. На рис.6 страница с устаревшим решением, на рис.7 страница с актуальным решением. Отличие в размере отступа перед и после текста.

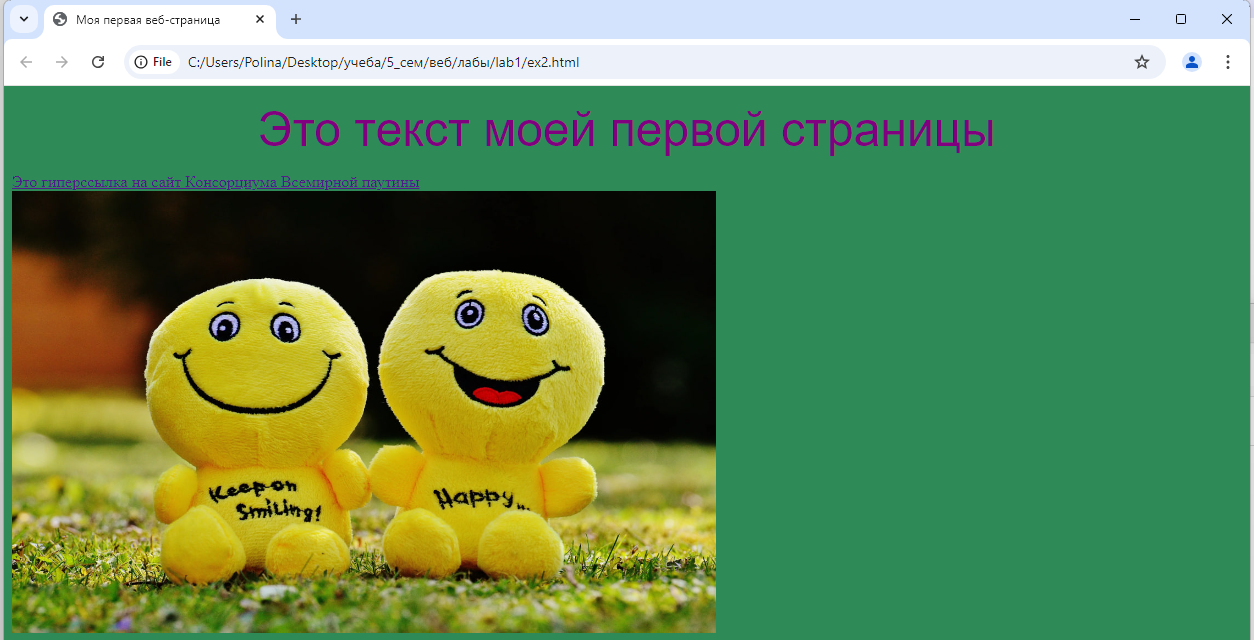


Рисунок 6 - Устаревшее решение

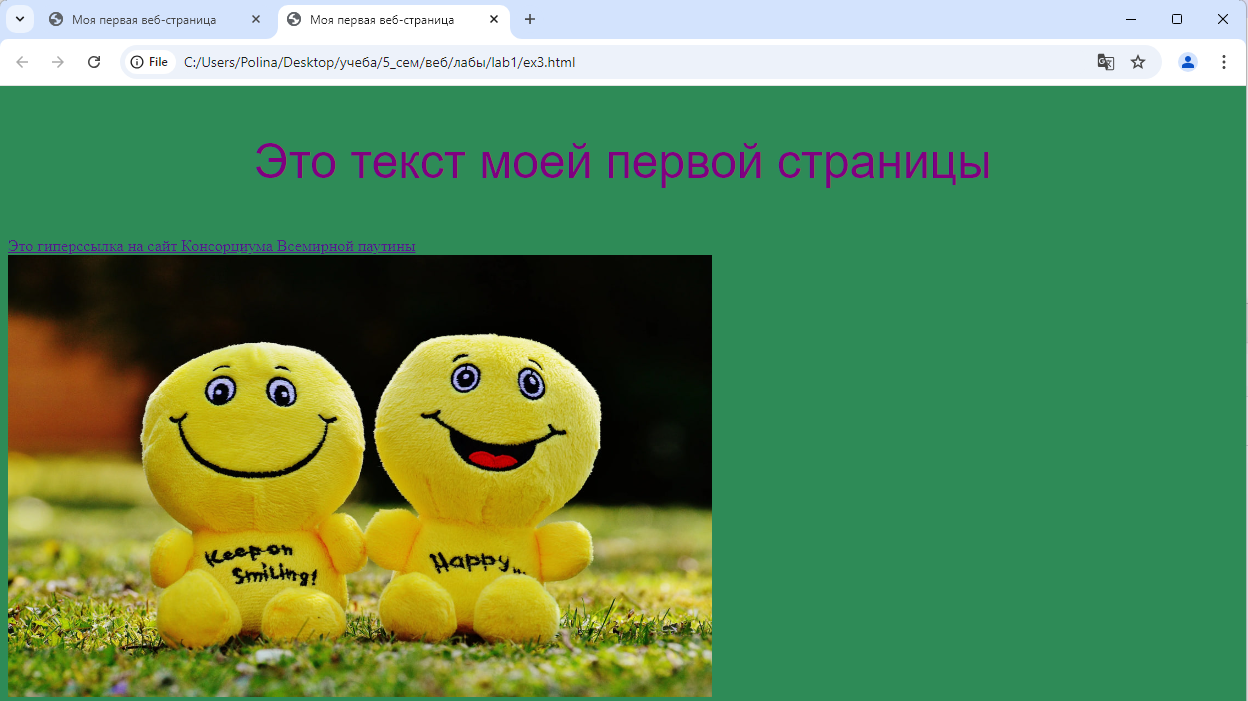


Рисунок 7 - Актуальное решение

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

Был добавлен абзац текста с тегом <pre>, позволяющий отображать количество пробельных символов и переходов на новую строку в браузере, как в исходном документе (рис.8-9).

Также была добавлена длинная строка текста без пробелов, после чего в браузере появились полосы прокрутки (рис.8). С добавлением пробелов в коде документа, появился автоматический перенос текста на следующую строку в браузере (рис.9).

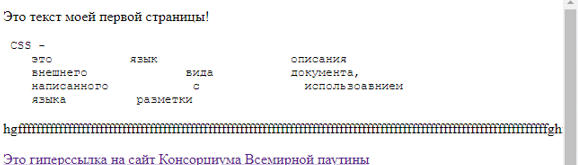


Рисунок 8 - Прокрутка браузера

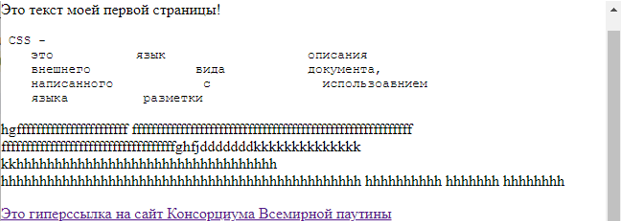


Рисунок 9 - Переносы текста на новую строку

### Упражнение 6. Использование спецсимволов

Добавлены спецсимволы для отображения кавычек &quot, символа торговой марки &trade, значков евро &euro и фунтов стерлингов &pound, и символа «плюс минус» &plusmn. Результат на рис.10.



Рисунок 10 - Строка со спецсимволами

## 2. Основы логического форматирования контента

### Упражнение 1. Использование заголовков

Использованы заголовки разного уровня при помощи тегов <h1>…<h6>

Результат представлен на рис.11.

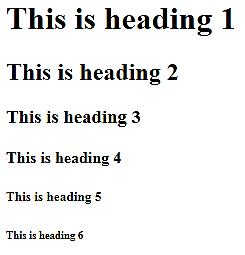


Рисунок 11 - Использование заголовков разного уровня

### Упражнение 2. Возможности логического форматирования текста

Использованы теги, выделяющие текст со специальным обособлением (рис.12).

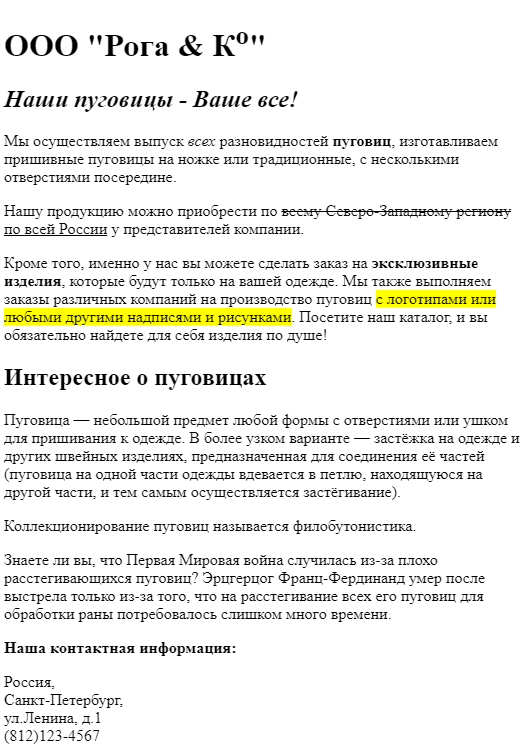


Рисунок 12 - Текст с обособлением

Где <b> - bold - жирное начертание;

<em> - emphasized – смысловое ударение;

<i> - italic – курсив;

<small> - smaller – шрифт, уменьшенный на единицу относительно текущего;

<strong> - important – акцентирование текста;

<sub> - subscripted – подстрочный текст;

<sup> - superscripted – надстрочный текст;

<ins> - inserted – выделение добавленного в новую версию документа текст;

<del> - deleted – удаленный (вычеркнутый) текст;

<mark> - marked/highlighted – выделенный текст.

### Упражнение 3. Ссылки, цитаты, определения

Добавлены: определение термина с помощью парного тега <dfn>; ссылка на источник (парный тег <cite>); указание контактной информации (парный тег <address>). Результат представлен на рис.13.

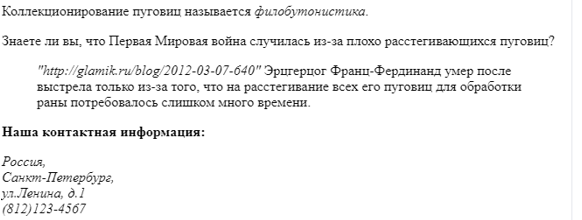


Рисунок 13 - Добавление определения, ссылки, контактной информации

### Упражнение 4. Элементы компьютерного кода

Для отформатированного представления элементов, связанных с компьютерным кодом, были использованы тэги <kbd>, <samp>, <code>, <var>. Результат представлен на рис.14.

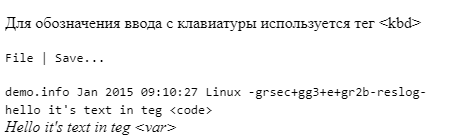


Рисунок 14 - Элементы компьютерного кода

### Упражнение 5. Маркированные списки

Маркированный список был создан с помощью тега <ul>.

Пункты списка были созданы с помощью тега <li>.

Вид маркера был изменен средствами CSS: <ul style=”list-style-type:circle”>.

Цвет пункта был изменен средствами CSS: <li style=”color:red”>.

Изучение маркированных списков представлено на рис.15.

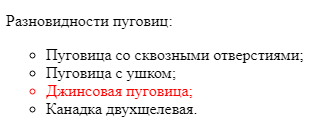


Рисунок 15 - Маркированный список

### Упражнение 6. Нумерованные списки

Нумерованный список был создан с помощью тега <ol>.

Пункты списка были созданы с помощью тега <li>.

Начальный номер пунктов списка был изменен с помощью атрибута start: <ol start=”5”>.

Тип нумерации пунктов списка был изменен с помощью атрибута type: <ol start=”5” type=”i”>.

Изучение нумерованных списков представлено на рис.16.

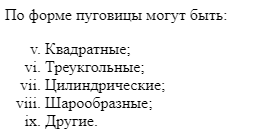


Рисунок 16 - Нумерованный список

### Упражнение 7. Список определений

Список определений был создан с помощью тега <dl>.

Пункты списка для терминов были созданы с помощью тега <dt>.

Пункты списка для определений были созданы с помощью тега <dd>.

Изучение списка определений представлено на рис.17.

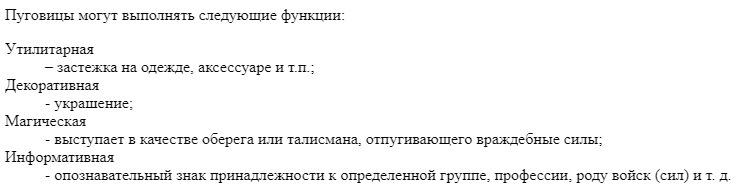


Рисунок 17 - Список определений

### Упражнение 8. Вложенные списки

Были изучены вложенные списки. Результат представлен на рис.18.

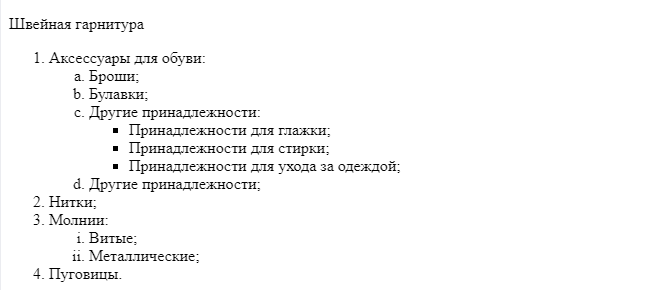


Рисунок 18 - Вложенные списки

## 3. Гиперссылки

### Упражнение 1. Основы работы с гиперссылками

Были созданы гиперссылки:

* С переходом на созданную страницу и обратно. Для этого в значении атрибута href прописано название файла, содержащего HTML-код страницы, на которую необходимо перейти;
* С переходом на статью. Для этого в значении атрибута href прописана ссылка на статью;
* Открывающие в том же окне браузера страницы Яндекса дзен. Для этого был добавлен атрибут target со значением \_top.

Результат представлен на рис.19.

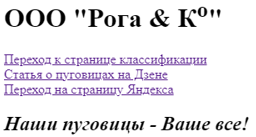


Рисунок 19 - Гиперссылки

### Упражнение 2. Работа с цветом гиперссылок

Были установлены стили CSS для изменения цвета ссылок:

<style>

a:visited{color:blue;} <!-- Стиль посещенной ссылки -->

a:active{color:Fuchsia;} <!-- Стиль активной ссылки -->

a:hover{color:yellow;} <!-- Стиль для ссылки при наведении на нее мышью -->

</style>

Изменение цвета ссылки продемонстрированно на рис.20.



Рисунок 20 - Изменение цвета гиперссылки

### Упражнение 3. Ссылка на адрес электронной почты

Была добавлена ссылка на адрес электронной почты при помощи указания значения mailto:example@mail.ru в атрибуте href.

Добавление ссылки на адрес электронной почты , открытие почтового клиента и открытие окна создания нового сообщения с указанием адреса получателя продемонстрированно на рис.20.

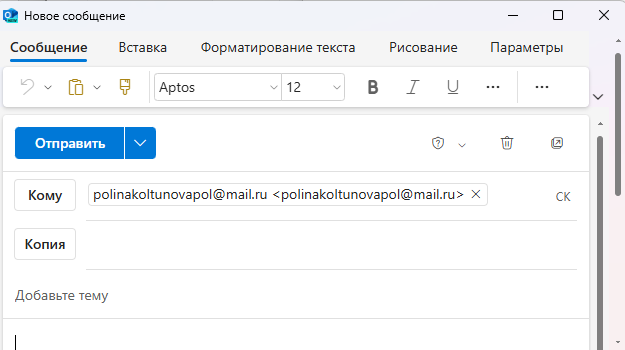


Рисунок 21 - Добавление ссылки на адрес электронной почты

### Упражнение 4. Внутренние ссылки документа

Якоря в документе были созданы с помощью меток:

<h2 **id = "history\_1"**>История первая</h2>

<h2 **id = "history\_2"**>История вторая</h2>

Меню для навигации по тексту страницы создано с помощью ссылок:

<a href = "**#history\_1**">История первая</a><br/>

<a href = "**#history\_2**">История вторая</a><br/>

Результат представлен на рис.22.

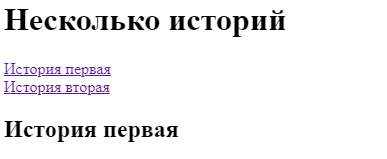


Рисунок 22 - Меню навигации

## 4. Таблицы

### Упражнение 1. Создание таблицы

Была создана таблица при помощи следующих тегов:

<table>…</table> - создание таблицы

<tr> …</tr>- создание строк

<td>… </td> - создание ячеек

Результат представлен на рис.23.

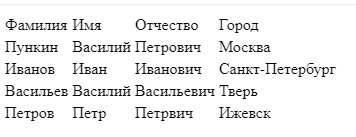


Рисунок 23 - Отображение табличной структуры

### Упражнение 2. Работа с границам таблицы

Для работы с границами таблицы был использован атрибут border, значение которого равно ширине границы. Также был добавлен CSS стиль для изменения ширины таблицы: <table border = "2” **style="width:80%"**>

Результат представлен на рис.24 в упражнении 4.

### Упражнение 3. Создание заголовка и подписи таблицы

Были изменены теги первой строки таблицы. Теги <td> заменены на <th>, при этом текст ячейки поменян на жирный шрифт и отцентрован.

Создан заголовок таблицы при помощи парного тэга <caption>.

Добавлен CSS стиль для изменения местоположения заголовка:

<caption **style = "caption-side:bottom"**>Список сотрудников</caption>.

Результат представлен на рис.24 в упражнении 4.

### Упражнение 4. Объединение ячеек

Были объединены ячейки таблицы по горизонтали при помощи атрибута colspan со значением равным 3 у тега td.

Для объединения ячеек по вертикали был использован атрибут rowspan со значением 2 того же тега td, при этом были удалены две ячейки из другого ряда.

Результат упражнений 2-4 представлен на рис.24.



Рисунок 24 -Итоговый вариант таблицы с сотрудниками

### Упражнение 5. Структурные блоки таблиц

Была изменена структура таблицы, добавлены тэги <tbody>,<thead> и <tfoot> и применены к ним стили CSS. Добавлен тег <colgroup>, задающий ширину и стиль поля(ей) таблицы.

Результат представлен на рис.25.



Рисунок 25 - Изменение стиля таблицы

## 5. Изображения и медиаконтент

### Упражнение 1. Основы работы с изображениями

Добавлено изображение на страницу при помощи тега <img> с атрибутом src, задающим адрес файла с изображением, атрибутом alt, размещающим альтернативный текст и атрибутом title, задающим подсказку к изображению.

В результате при наведении курсора на изображение появилась всплывающая подсказка.

### Упражнение 2. Изменение размеров изображения

Был добавлен стиль CSS для изменения размера изображения.

<img src= “button\_1.jpg" alt="Button" title = “Button” **style="width:100px; height:130px"**>

### Упражнение 3. Создание изображения-гиперссылки

Была создана изображение-гиперссылка с помощью тега <a>, который оборачивает тег <img>, указывая, что при нажатии на картинку будет открываться страница "classification.html".

### Упражнение 4. Карты изображений

Была создана карта изображения, позволяющая привязывать ссылки к разным областям одного изображения. Для этого был использован тег <img> с атрибутом usemap, который связывает его с тегом <map>, где с помощью тега <area> задаются конкретные области, к которым привязываются ссылки.

<img src="button\_4.jpg" usemap="#MyMap">  
<map name="MyMap">  
 <area shape="circle" alt="yandex.ru" title="" coords="406,202,103" href="http://yandex.ru" target="" />  
</map>

Парный тег <map> определяет конфигурацию накладываемой на изображение карты.

Атрибут name задает имя конфигурации.

Атрибут usemap указывает браузеру, что изображение является картой.

Тег <area/> сформировывает активные области.

На сайте сервиса Online Image Map Editor была добавлена область, найдены ее координаты, и указана внешняя ссылка, которые были добавлены в код.

### Упражнение 5. Добавление медиаконтента

Был добавлен видеоконтент на страницу при помощи парного тега <video> с атрибутами width и height, задающими размеры видео, оборачивающего тег <source> с атрибутом src, указывающим путь к видеофайлу, и атрибутом type, указывающим формат видеофайла.

Результат представлен на рис.26.

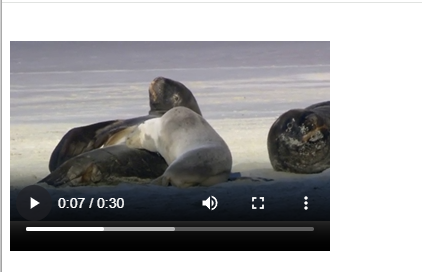


Рисунок 26 – Медиаконтент на странице

## 6. Формы

### Упражнение 1. Текстовые поля формы

Были созданы текстовые поля при помощи тега <form>, имеющего следующие атрибуты:

* action указывает веб-адрес программы, которая обрабатывает данные формы.
* method указывает http метод, который используется браузером для отправки формы на сервер.

Также был использован парный тег <input> для создания текстовых полей.

Для создания поля с заданной длиной использовался атрибут maxlength.

Для создания многострочных текстовых полей использовался парный тег <textarea>.

Для создания полей ввода пароля использовался тип password атрибута type.

Для создания скрытого поля использовался тип hidden атрибута type.

Результат представлен на рис.27 в упражнении №6.

### Упражнение 2. Типы полей HTML5

Были добавлены несколько типов полей тега <input>:

* color - работа с цветом;
* date - работа с датой;
* email - проверка e-mail;
* number - проверка чисел;
* week - работа с днем недели;
* url - проверка URL.

Результат представлен на рис.27 в упражнении №6.

### Упражнение 3. Кнопки

Были добавлены различные виды кнопок:

* reset – кнопка очистки поля;
* submit – кнопка отправки формы;
* button – кнопка.

Затем был создан функционал для кнопки myButton, которая содержит в себе картинку.

Результат представлен на рис.27 в упражнении №6.

### Упражнение 4. Флажки и переключатели

Были созданы переключатели при помощи типа поля radio парного тега <input>.

Были созданы флажки при помощи типа поля checkbox парного тега <input>.

Результат представлен на рис.27 в упражнении №6.

### Упражнение 5. Поле со списком

Было создано поле с раскрывающимся списком при помощи тега парного <select>.

Парный тег <optgroup> представляет из себя контейнер, внутри которого находятся элементы option. Он выделяется жирным шрифтом, и смещает вправо пункты элемента option.

Результат представлен на рис.27 в упражнении №6.

### Упражнение 6. Поле для загрузки файлов

Добавлено поле для отправки файлов при помощи типа file парного тега <input> с атрибутом multiple, позволяющим пользователю выбрать несколько файлов одновременно для загрузки.

Результат упражнений 1-6 представлен на рис.27.

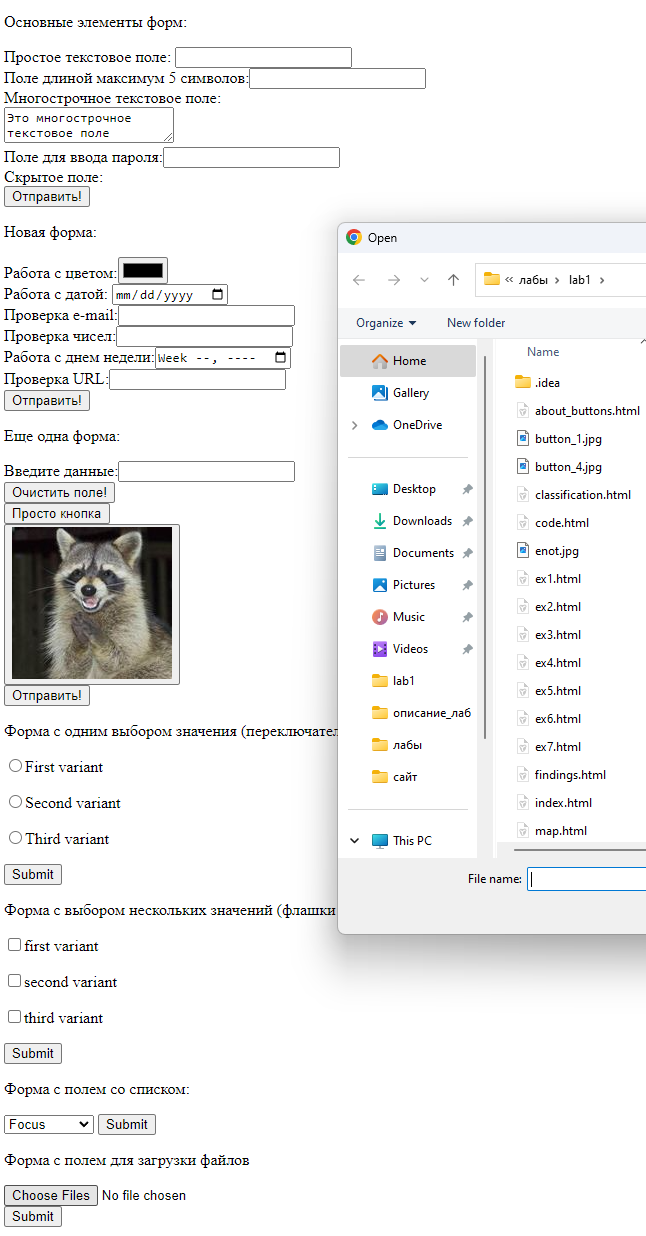


Рисунок 27 – Все формы

## 7. Основы работы с блочной структурой

### Упражнение 1. Основы организации структуры документа

Была добавлена разметка из файла с расширением.css при помощи связывающего тега <link>, имеющего атрибуты:

* rel – указывает, что подключаемый файл является таблицей стилей;
* type – определяет тип содержимого;
* href – указывает путь к файлу.

Результат представлен на рис.28.



Рисунок 28 – Страница с блочными тегами

### Упражнение 2. Работа с разделами документа

Было произведено сравнение структур файлов. В файле block\_1.html используются блочные теги с идентификаторами id <div id="header">, <div id="nav">, <div id="section">, и <div id="footer">. Эти теги создают блоки на странице, которые занимают всю ширину родительского элемента и начинаются с новой строки.

В файле block\_2.html используются семантические HTML5 теги <header>, <nav>, <section>, и <footer>, которые заменяют соответствующие div с идентификаторами в block\_1.html, что позволяет описывать смысл содержимого, делает структуру и содержание страницы более понятными. Результат работы кода с семантическими тегами представлен на рис.29.

Внешне страницы не отличаются друг от друга, но семантические теги явно описывают логическую структуру документа.



Рисунок 29 – Страница с семантическими тегами