

Министерство образования Республики Беларусь

Учреждение образования «Белорусский государственный университет  
информатики и радиоэлектроники»

Инженерно-экономический факультет

Кафедра экономической информатики

Дисциплина: Распределенные системы обработки информации

Индивидуальная практическая работа №2

на тему

«Создание веб-страниц с использованием с использованием динамических  
элементов дизайна (DHTML)»

Выполнил студент 5-го курса  
группы 694051 специальности  
«Электронный маркетинг»

Кузьмич Павел  
Валерьевич

Проверил старший преподаватель  
кафедры экономической  
информатики

Атрощенко Натэлла  
Александровна

Минск, 2020

## **Содержание**

1. Описание HTML кода страницы.....	3
2. Описание CSS кода каскадных таблиц стилей.....	4
3. Описание функциональной части.....	5
Выводы.....	8

## 1. Описание HTML кода страницы

Страница проекта содержит динамическую форму, с полями для введения исходных данных — текстовое поле, заданное тегом `<input type = «text»>`, и выпадающий селектор, заданных тегом `<select>` — а также несколько кнопок — радио (`<input type=«radio»>`) для выбора варианта обновления текста (до или после) и две кнопки `<input type="button">` для исполнения функциональной логики. Также с помощью тега `<div>` задано отдельное поле для вывода результатов. После тела страницы с использованием тега `<script src=«script.js»>` задан путь к JavaScript файлу с кодом для динамического изменения страницы. Листинг кода приведён ниже.

Листинг 1.1 Структура HTML страницы.

```
<html>
<head>
    <meta charset="utf-8">
    <link rel="stylesheet" href="content.css">
    <title>ИПП #2</title>
</head>
<body>
<div class="form-box">
    <form>
        <label for="input-text" class="input-field">
            Enter some text:
        </label><br>
        <input type="text" id="input-text" name="input">
        <select id="prefix-selector" class="select-box">
            <option>prefix#1</option>
            <option>prefix#2</option>
            <option>prefix#3</option>
        </select><br>
        <input type="radio" id="before-radio-button" name="type"
            value="before">
        <label for="before-radio-button">
            before
        </label><br>
        <input type="radio" id="after-radio-button" name="type"
            value="after">
        <label for="after-radio-button">
            after
        </label><br>
```

Листинг 1.1 Продолжение.

```
<input type="button" id="style-magic-button" class="button"
      value="*style magic*">
<input type="button" id="update-button" class="button"
      value="Update text"><br><br>
<div class="result-box" id="result-box"></div>
</form>
</div>
</body>
<script src="script.js"></script>
</html>
```

## 2. Описание CSS кода каскадных таблиц стилей

Для оформления страницы и формы ввода используются следующие CSS элементы. Для формирования фона страницы, а также определения положения и стиля формы ввода используется код приведённый в листинге 2.1. Градиент фона задан с помощью атрибута «background-image: linear-gradient(135deg, #f5f7fa 0%, #c3cfe2 100%);», а для формы ввода и вывода результатов стиль задан в классе «form-box».

Листинг 2.1 Описание стиля страницы и формы ввода.

```
body {
  display: flex;
  justify-content: center;
  align-items: center;
  background-image: linear-gradient(135deg, #f5f7fa 0%, #c3cfe2
                                100%);
}
.input-field {
  width: 100%;
}
.form-box {
  display: flex;
  flex-direction: column;
  align-items: center;
  padding: 50px 40px;
  color: #fff;
  background: rgba(0, 0, 0, 0.8);
  border-radius: 10px;
  box-shadow: 0 0.4px 0.4px rgba(128, 128, 128, 0.109), 0 12px 12px
              rgba(128, 128, 128, 0.35);
}
```

Для оформления кнопок создан класс «button», в котором заданы размеры, цвет, шрифт, переход при нажатии и другие атрибуты. Для придания кнопкам дополнительной анимации при наведении на кнопку используется атрибут «hover». Формат выпадающего селектора и окна вывода результатов заданы с помощью классов «select-box» и «result-box» соответственно. Код приведён в листинге 2.2.

Листинг 2.2. Описание стиля кнопок и окна вывода.

```
.form-box .button {
    width: 100%;
    margin: 10px 0 0px 0;
}
.button:hover {
    color: rgba(255, 255, 255, 1);
    text-transform: uppercase;
    box-shadow: 0px 3px 10px lightskyblue;
}
.select-box {
    position: relative;
    display: block;
    width: 100%;
    margin: 10px 0 0px 0;
    font-size: 18px;
    color: #60666d;
}
.result-box {
    display: flex;
    flex-direction: column;
    align-items: center;
    background: rgba(128, 128, 128, 0.109);
    border-radius: 10px;
}
```

### 3. Описание функциональной части

Основная динамическая логика проекта прописана в файле «script.js» с использованием языка JavaScript. Для получения информации об элементах страницы используется метод `getElementById` объекта `document`, где получения объектов кнопок или полей предназначен их уникальный идентификатор (листинг 3.1). Здесь же заданы вспомогательные функции для получения текущего значения выпадающего окна префикса/суффикса, а также обновления цвета и типа шрифта элемента страницы, например, значения по умолчанию из словаря `defaultStyle` (чёрный цвет и тип Verdana).

### Листинг 3.1.

```
var defaultStyle = {
    "color": "black",
    "fontFamily": "Verdana",
}
var resultBox = document.getElementById("result-box")
var inputField = document.getElementById("input-text")
var styleMagicButton = document.getElementById('style-magic-button');
var updateButton = document.getElementById('update-button');
var beforeButton = document.getElementById('before-radio-button');
var afterButton = document.getElementById('after-radio-button');

function getPrefixSelector() {
    var prefixSelectorButton = document.getElementById('prefix-selector');
    return prefixSelectorButton.value;
}

function updateStyle(input, expectedStyle) {
    input.style.fontFamily = expectedStyle.fontFamily;
    input.style.color = expectedStyle.color;
}
```

Для обеспечения функциональной работы кнопок к ним добавлены «обработчики событий» (event listeners), которые при нажатии (событие типа «click») вызывают код, прописанный в теле соответствующих функций. Так для кнопки «update» прописана логика: создаётся результат вывода, параграф заполненный текстом из поля ввода, далее к этому тексту с помощью метода insertAdjacentText добавляется суффикс или префикс, в зависимости от того, какая нажата радио-кнопка - «до» или «после»; в завершении работы функции обновляется стиль вывода и он добавляется в окно вывода «result-box». Для кнопки «style magic» в свою очередь задана функция, которая случайным образом формирует объект, содержащий тип и цвет шрифта, которым после обновляется значение в поле ввода. Код функций приведён в листинге 3.2.

### Листинг 3.2 Функции работы кнопок.

```
updateButton.addEventListener('click', function() {
    var resultOutput = document.createElement("P");
    resultOutput.innerText = inputField.value;
    if (beforeButton.checked) {
        resultOutput.insertAdjacentText('afterbegin', getPrefixSelector());
    }
    if (afterButton.checked) {
        resultOutput.insertAdjacentText('beforeend', getPrefixSelector());
    }
});
```

Листинг 3.2 Продолжение.

```
        console.log(resultOutput);
        updateStyle(resultOutput, inputField.style);
        resultBox.appendChild(resultOutput)
    });

    inputField.addEventListener('click', function() {
        updateStyle(this, defaultStyle)
    });

    styleMagicButton.addEventListener('click', function() {
        var fontFamily = [
            "Arial",
            "Helvetica",
            "Roboto",
            "Trebushet MS",
            "Times New Roman"
        ];
        var color = [
            "red",
            "bisque",
            "cyan",
            "orange",
            "listcyan",
            "mistyrose",
            "gray",
            "darkred"
        ];

        function getRandomStyle(styleArray) {
            var index = Math.floor(Math.random() * styleArray.length);
            return styleArray[index];
        }

        var randomStyle = {
            "color": getRandomStyle(color),
            "fontFamily": getRandomStyle(fontFamily),
        }
        updateStyle(inputField, randomStyle);
    });
```

Итоговые скриншоты проекта приведены на рисунках 1 и 2, протестированные в браузерах Mozilla Firefox и Chromium Browser.

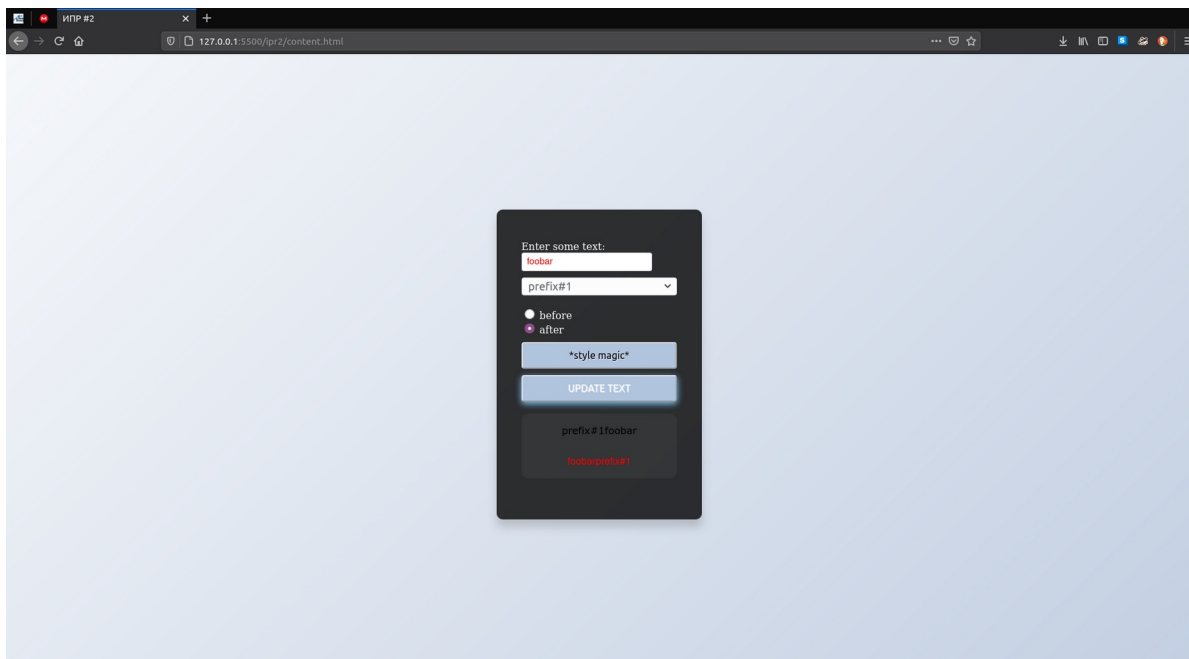


Рисунок 1— Скриншот проекта в Mozilla Firefox.

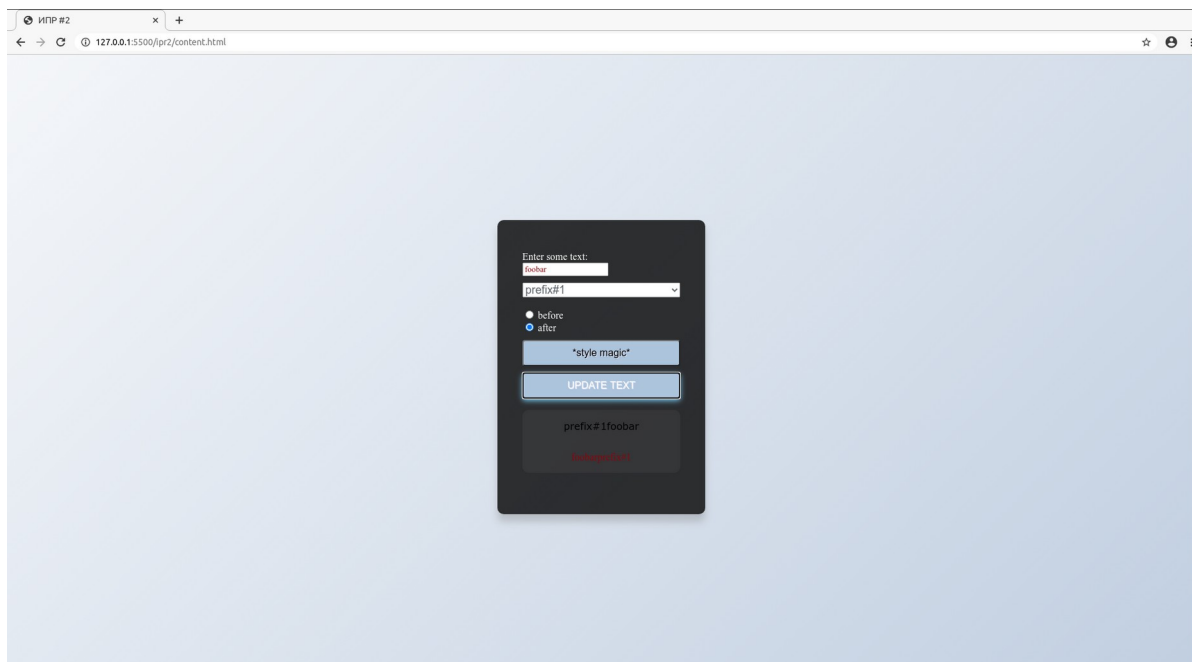


Рисунок 2 — Скриншот проекта в Chromium Browser.

## Выводы

В результате выполнения практического задания был создан проект, состоящий из 3 файлов (content.html, content.css и script.js), обеспечивающих структуру, стиль и логику функционирования с использованием возможностей динамического формирования веб-страниц в JavaScript. Использованы и изучены основные сущности языка JavaScript такие как переменные, типы, функции, условные операторы и методы DOM-модели.