

Министерство образования и науки Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования

Нижегородский государственный технический
университет им. Р.Е.Алексеева

Институт радиоэлектроники и информационных технологий
Кафедра прикладной математики и информатики

Отчёт № 3

«Селекторы и свойства CSS»

по дисциплине

web-разработка

Выполнила:

студент группы 23-ПМ-1

Блинов А. С.

Проверил:

Курушин Е. А.

Нижний Новгород

2025 год

Оглавление

Цели и задачи.....	3
Теоретическая справка	4
Реализация	8
Вывод	12
Приложение	13

Цели и задачи

Основной целью данной лабораторной работы является освоение методов подключения таблиц стилей CSS к веб-странице, изучение различных способов стилизации элементов с использованием разных типов селекторов, а также практическое применение CSS для оформления блоков с учётом современных возможностей языка.

Для достижения поставленной цели необходимо выполнить следующие задачи:

1. Подключить к созданной ранее странице таблицы стилей.
2. Добавить к каждому элементу на странице несколько стилей. Не менее 5 шт. Для каждого элемента выполнить обращение по разному типу селекторов.
3. Оформить имеющийся блок на странице, либо добавить новый блок, в качестве самостоятельного элемента и добавить к нему стили CSS. Не менее 3 свойств.

Теоретическая справка

Современная веб-разработка требует не только грамотного построения HTML-структуры, но и создания привлекательного интерфейса, который обеспечивает удобство использования сайта. Одним из ключевых инструментов для стилизации веб-страниц является CSS (Cascading Style Sheets) – технология, позволяющая управлять внешним видом элементов, их расположением и адаптацией под различные устройства.

В данной работе рассматриваются основные принципы использования CSS, включая его роль в веб-дизайне, разновидности селекторов, новые свойства, добавленные в CSS3, а также основные концепции позиционирования и настройки отступов. Разбор этих аспектов поможет глубже понять принципы стилизации и гибкой верстки, что является важным навыком для создания профессиональных и адаптивных веб-сайтов.

1. Что такое CSS?

CSS (Cascading Style Sheets) — это язык стилей, используемый для описания внешнего вида документа, написанного на HTML или XML. CSS позволяет управлять цветом, шрифтами, отступами, размерами и другими аспектами визуального представления веб-страниц.

2. Для чего применяются стили CSS?

CSS применяется для:

- Управления внешним видом веб-страниц.
- Разделения содержимого (HTML) и его оформления (CSS).
- Создания адаптивных и кросс-браузерных дизайнов.
- Упрощения поддержки и изменения стилей на сайте.

3. Какие бывают селекторы?

Селекторы используются для выбора элементов, к которым применяются стили. Основные типы:

- Селектор по тегу (например, `div`, `p`).
- Селектор по классу (например, `.class-name`).
- Селектор по идентификатору (например, `#id-name`).
- Универсальный селектор (`*`).
- Селектор по атрибуту (например, `[type="text"]`).
- Комбинаторы (например, `div > p`, `div + p`, `div ~ p`).
- Псевдоклассы (например, `:hover`, `:nth-child(n)`).
- Псевдоэлементы (например, `::before`, `::after`).

4. Назовите не менее 5 свойств, появившихся в CSS3.

- `border-radius` (скругление углов).
- `box-shadow` (тень для элементов).
- `gradient` (градиентные фоны).
- `transition` (анимация переходов).
- `flexbox` (гибкая раскладка элементов).
- `@media` (медиа-запросы для адаптивного дизайна).

5. В чем отличие `margin` от `padding`?

• `Margin` — это внешний отступ элемента, который создает пространство между этим элементом и другими элементами.

• `Padding` — это внутренний отступ элемента, который создает пространство между содержимым элемента и его границей.

6. Что такое и какие бывают типы позиционирования? Позиционирование определяет, как элемент располагается на странице. Основные типы:

- static (по умолчанию) — элемент располагается в нормальном потоке документа.

- relative — элемент смещается относительно своего обычного положения.

- absolute — элемент позиционируется относительно ближайшего позиционированного родителя (или окна браузера).

- fixed — элемент фиксируется относительно окна браузера.

- sticky — элемент ведет себя как relative, пока не достигнет определенной точки, после чего становится fixed.

7. Какие бывают и в чем особенность типов отображения? (Свойство display) Свойство display определяет, как элемент отображается на странице. Основные значения:

- block — элемент занимает всю ширину родителя и начинается с новой строки (например, div, p).

- inline — элемент занимает только необходимое пространство и не начинает новую строку (например, span, a).

- inline-block — элемент ведет себя как inline, но может иметь ширину, высоту и отступы как block.

- none — элемент полностью скрыт и не занимает места на странице.

- flex — элемент становится flex-контейнером, позволяя управлять расположением дочерних элементов с помощью flexbox.

- grid — элемент становится grid-контейнером, позволяя создавать сложные сетки для расположения дочерних элементов.

Реализация

В рамках выполнения лабораторной работы была разработана веб-страница с современным дизайном и интуитивно понятным интерфейсом. Она включает несколько ключевых блоков: заголовок страницы, описание проекта, секцию с преимуществами, блок с автоматизацией задач и контактную информацию. Работа велась в среде разработки Visual Studio Code. Ниже рассмотрены основные этапы реализации.

1. Создание структуры HTML-документа

Была разработана базовая HTML-структура, включающая:

- Заголовок страницы – определён с использованием тега `<title>`.
- Метаданные – добавлены теги `<meta>` для указания кодировки (UTF-8) и адаптивности страницы (viewport).
- Основные секции – страница разделена на логические блоки с помощью семантических тегов:
 - `<header>` – для шапки сайта с логотипом и навигацией.
 - `<section>` – для контента, включая описание проекта и функциональные блоки.
 - `<footer>` – для контактных данных и формы обратной связи.

2. Разработка навигационного меню

В верхней части страницы создано навигационное меню, включающее ссылки на основные разделы. Используются теги `<a>`, при этом каждая ссылка дополнительно содержит:

- `href="#"` – временную ссылку,

- title – всплывающую подсказку,
- aria-label – описание для улучшения доступности.

3. Оформление веб-страницы с помощью CSS

Для стилизации страницы использованы каскадные таблицы стилей (CSS).

3.1. Общие стили

- Определены шрифты (Arial, sans-serif), отступы и цветовая палитра (#f8f8f8 – фон, #333 – текст).
- Установлены глобальные настройки margin и padding для корректного отображения контента.

3.2. Стилизация заголовка (header)

- Определены параметры размещения (position: absolute; height: 75px;).
- Логотип оформлен с использованием background-image.
- Ссылки в навигации стилизованы: изменён цвет (color: #000), размер шрифта (font-size: 15px;), добавлены отступы.

3.3. Блок "Описание проекта"

- Центрирование контента (text-align: center;).
- Использован display: flex; для расположения карточек с преимуществами.
- Карточки (.feature) оформлены с закруглёнными углами (border-radius: 10px;) и фоновым цветом (#f4f4f4).

3.4. Блок "Set, forget, and then track"

- Элементы размещены в строку (display: flex;).

- Фон блока задан с использованием `background-color: #2b7db7;`
`color: white;`.
- Добавлены изображения (`max-width: 500px;`).
- Оформлен список (`.custom-list`), в котором элементы выровнены по левому краю (`align-items: center;`).

3.5. Блок автоматизации задач

- Использована сетка (`display: flex;`) для равномерного распределения карточек.
- Карточки (`.task`) получили мягкие тени и плавные анимации при наведении.

4. Добавление формы обратной связи

- Создана форма с полями `<input>` и `<textarea>`.
- Для кнопки отправки (`button`) определены стили (`background: #2b7db7; color: white; padding: 10px; border-radius: 5px;`).
- Добавлен эффект при наведении (`hover`).

5. Финальное тестирование

После завершения работы был проведён тест рендеринга страницы в браузере, проверена корректность отображения всех элементов и их адаптивность. Веб-страница полностью готова к использованию. Вид страницы представлен на рисунке 1.



Рисунок 1 – Стилизация главной страницы

Вывод

В ходе выполнения лабораторной работы была разработана веб-страница с современным дизайном и удобной структурой. В процессе работы были применены технологии HTML и CSS для создания адаптивного интерфейса, а также реализована навигация, блоки с информацией и форма обратной связи.

Использование семантических тегов позволило улучшить читаемость кода и повысить доступность сайта. Применение CSS помогло стилизовать элементы, а flexbox обеспечил удобное расположение контента. В результате получилась функциональная веб-страница, соответствующая современным требованиям веб-разработки.

Приложение

```
/* Общие стили */
```

```
body {  
    font-family: Arial, sans-serif;  
    margin: 0;  
    padding: 0;  
    background-color: #f8f8f8;  
    color: #333;  
}
```

```
/* Хедер */
```

```
.site-header {  
    position: absolute;  
    height: 75px;  
    left: 0%;  
    text-align:center;  
    right: 0%;  
    top: 0px;  
}
```

```
.site-header nav a {  
    margin-right: 15px;  
    text-decoration: none;  
    color: #000;  
    font-family: 'Inter';  
    font-style: normal;  
    font-weight: 400;  
    font-size: 15px;
```

```
line-height: 24px;
```

```
}
```

```
.image-up {
```

```
position: absolute;
```

```
width: 222px; /* Фиксированная ширина */
```

```
height: 186px; /* Фиксированная высота */
```

```
background-image: url('../Screenshot_1.png'); /* Путь к изображению */
```

```
background-size: contain; /* Масштабирование внутри блока */
```

```
background-repeat: no-repeat; /* Отключение повторов */
```

```
background-position: center; /* Центрирование изображения */
```

```
z-index: 2;
```

```
}
```

```
/* Описание проекта */
```

```
.about-section {
```

```
position: relative;
```

```
text-align: center;
```

```
padding: 60px 20px;
```

```
}
```

```
.features {
```

```
display: flex;
```

```
justify-content: center;
```

```
gap: 20px;
```

```
margin-top: 30px;
```

```
}
```

```
.feature {  
    width: 30%;  
    padding: 20px;  
    background: #f4f4f4;  
    border-radius: 10px;  
    z-index: 5;  
}
```

```
/* БЛОК "Set, forget, and then track" */
```

```
.track-section {  
    display: flex;  
    align-items: center;  
    background-color: #2b7db7;  
    padding: 60px 20px;  
    color: white;  
}
```

```
.track-section img {  
    width: 50%;  
    max-width: 500px;  
}
```

```
.track-section-text {  
    padding-left: 40px;  
}
```

```
.track-section h2 {  
    font-size: 28px;  
}
```

```
.custom-list {  
    list-style: none;  
    padding: 0;  
}
```

```
.custom-list li {  
    display: flex;  
    align-items: center;  
    margin: 10px 0;  
}
```

```
.check-icon {  
    width: 20px;  
    margin-right: 10px;  
}
```

```
/* Автоматизация задач */
```

```
.automation-section {  
    text-align: center;  
    padding: 60px 20px;  
    background-color: #fff;  
}
```

```
.task-grid {
```



```
display: flex;
justify-content: center;
gap: 30px;
margin-top: 30px;
}
```

```
.task {
width: 30%;
padding: 20px;
background: #f4f4f4;
border-radius: 10px;
}
```

```
.task img {
width: 50px;
margin-bottom: 10px;
}
```

```
/* Футер */
```

```
.footer {
display: flex;
justify-content: space-between;
padding: 40px;
background: #f8f8f8;
}
```

```
.footer__section {
width: 25%;
```

```
}
```

```
.footer__logo {  
  width: 50px;  
  margin-top: 10px;  
}
```

```
.feedback-form {  
  display: flex;  
  flex-direction: column;  
  gap: 10px;  
}
```

```
.feedback-form input,  
.feedback-form textarea {  
  padding: 8px;  
  border: 1px solid #ccc;  
  border-radius: 5px;  
}
```

```
.feedback-form button {  
  background: #2b7db7;  
  color: white;  
  border: none;  
  padding: 10px;  
  cursor: pointer;  
}
```

```
.feedback-form button:hover {  
    background: #1a5a85;  
}
```