**РЕСПУБЛИКА БЕЛАРУСЬ**

**КОМИТЕТ ПО ОБРАЗОВАНИЮ МИНГОРИСПОЛКОМА**

учреждение образования

МИНСКИЙ ГОСУДАРСТВЕННЫЙ КОЛЛЕДЖ ЦИФРОВЫХ ТЕХНОЛОГИЙ

##### Группа 73 ТП

## Отчет

## по лабораторной работе № 8

**"Создание визуальных эффектов и анимации на HTML-странице"**

учебный предмет

“Программные средства создания Интернет приложений”

**Исполнитель: А.А. Артюшевский**

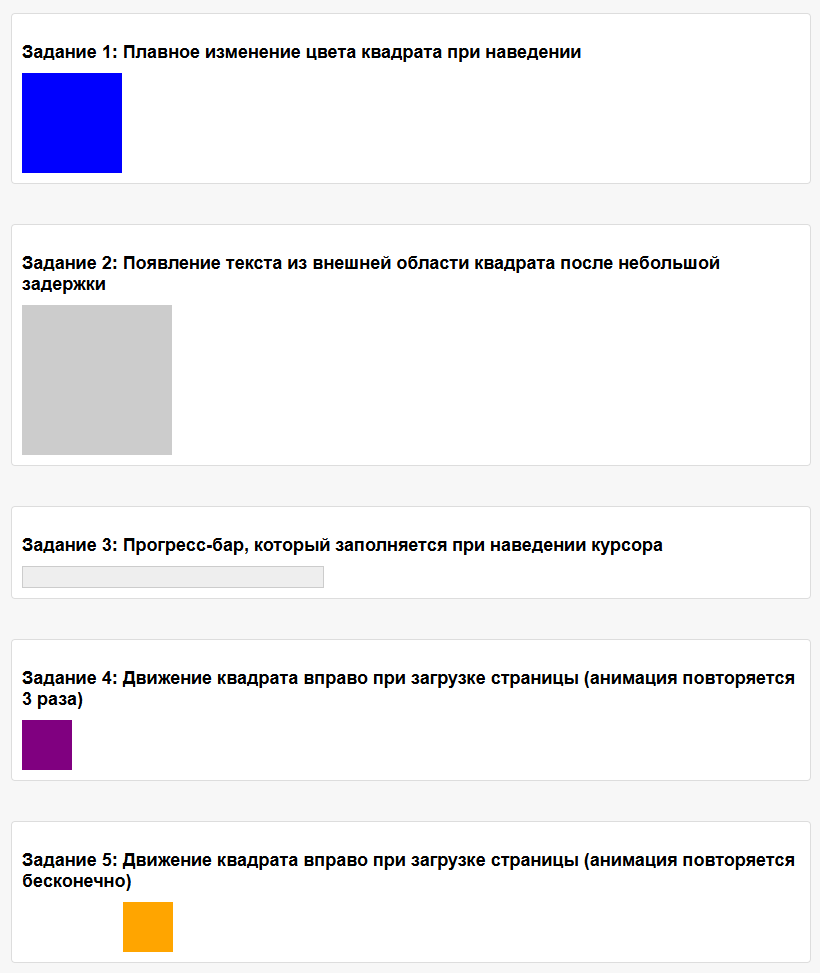
**Руководитель: Е.П. Сушкевич**

**Минск, 2025**

**Цель работы:**

Сформировать умение создавать визуальные эффекты и анимации на HTML-странице.

Решение:



index.html:

<!DOCTYPE html>

<html lang="ru">

<head>

    <meta charset="UTF-8">

    <title>Анимация квадратов</title>

    <link rel="stylesheet" href="/style.css">

</head>

<body>

    <div class="container">

        <!-- Задание 1 -->

        <div class="task" id="task1">

            <h3>Задание 1: Плавное изменение цвета квадрата при наведении</h3>

            <div class="square-hover"></div>

        </div>

        <!-- Задание 2 -->

        <div class="task" id="task2">

            <h3>Задание 2: Появление текста из внешней области квадрата после небольшой задержки</h3>

            <div class="hover-text-container">

                <div class="hover-text">Привет!</div>

            </div>

        </div>

        <!-- Задание 3 -->

        <div class="task" id="task3">

            <h3>Задание 3: Прогресс-бар, который заполняется при наведении курсора</h3>

            <div class="progress-bar">

                <div class="progress-fill"></div>

            </div>

        </div>

        <!-- Задание 4 -->

        <div class="task" id="task4">

            <h3>Задание 4: Движение квадрата вправо при загрузке страницы (анимация повторяется 3 раза)</h3>

            <div class="square-move-3"></div>

        </div>

        <!-- Задание 5 -->

        <div class="task" id="task5">

            <h3>Задание 5: Движение квадрата вправо при загрузке страницы (анимация повторяется бесконечно)</h3>

            <div class="square-move-infinite"></div>

        </div>

        <!-- Задание 6 -->

        <div class="task" id="task6">

            <h3>Задание 6: Непрерывное изменение фона квадрата при загрузке страницы</h3>

            <div class="square-bg-change"></div>

        </div>

        <!-- Задание 7 -->

        <div class="task" id="task7">

            <h3>Задание 7: Движение квадрата по контуру родительского элемента</h3>

            <div class="parent">

                <div class="square-border"></div>

            </div>

        </div>

        <!-- Задание 8 -->

        <div class="task" id="task8">

            <h3>Задание 8: Движение квадрата по контуру родителя с изменением фона в углах</h3>

            <div class="parent">

                <div class="square-border-color"></div>

            </div>

        </div>

        <!-- Задание 9 -->

        <div class="task" id="task9">

            <h3>Задание 9: Непрерывное движение квадрата с изменением фона на середине пути</h3>

            <div style="position: relative; height: 100px;">

                <div class="square-move-bg-change"></div>

            </div>

        </div>

    </div>

</body>

</html>

style.css:

body {

    font-family: Arial, sans-serif;

    padding: 20px;

    background: #f7f7f7;

}

.container {

    max-width: 800px;

    margin: 0 auto;

}

.task {

    margin-bottom: 40px;

    padding: 10px;

    background: #fff;

    border: 1px solid #ddd;

    border-radius: 4px;

}

.task h3 {

    margin-bottom: 10px;

    font-size: 18px;

}

.square-hover {

    width: 100px;

    height: 100px;

    background-color: blue;

    transition: background-color 0.5s ease;

}

.square-hover:hover {

    background-color: red;

}

.hover-text-container {

    width: 150px;

    height: 150px;

    background-color: #ccc;

    position: relative;

    overflow: hidden;

}

.hover-text {

    position: absolute;

    left: -100%;

    top: 50%;

    transform: translateY(-50%);

    white-space: nowrap;

    transition: left 0.5s ease 0.5s;

    font-size: 16px;

    color: #333;

}

.hover-text-container:hover .hover-text {

    left: 10px;

}

.progress-bar {

    width: 300px;

    height: 20px;

    background-color: #eee;

    position: relative;

    overflow: hidden;

    border: 1px solid #ccc;

}

.progress-fill {

    width: 0%;

    height: 100%;

    background-color: green;

    transition: width 1s ease;

}

.progress-bar:hover .progress-fill {

    width: 100%;

}

.square-move-3 {

    width: 50px;

    height: 50px;

    background-color: purple;

    position: relative;

    animation: moveRight3 2s linear 3;

}

@keyframes moveRight3 {

    from {

        left: 0;

    }

    to {

        left: 200px;

    }

}

.square-move-infinite {

    width: 50px;

    height: 50px;

    background-color: orange;

    position: relative;

    animation: moveRightInfinite 2s linear infinite;

}

@keyframes moveRightInfinite {

    from {

        left: 0;

    }

    to {

        left: 200px;

    }

}

.square-bg-change {

    width: 50px;

    height: 50px;

    animation: bgChange 3s linear infinite;

}

@keyframes bgChange {

    0% {

        background-color: red;

    }

    33% {

        background-color: green;

    }

    66% {

        background-color: blue;

    }

    100% {

        background-color: red;

    }

}

.parent {

    width: 220px;

    height: 220px;

    position: relative;

    border: 2px solid #333;

    margin: 10px 0;

}

.square-border {

    width: 30px;

    height: 30px;

    background-color: teal;

    position: absolute;

    animation: borderMove 4s linear infinite;

}

@keyframes borderMove {

    0% {

        top: 0;

        left: 0;

    }

    25% {

        top: 0;

        left: calc(100% - 30px);

    }

    50% {

        top: calc(100% - 30px);

        left: calc(100% - 30px);

    }

    75% {

        top: calc(100% - 30px);

        left: 0;

    }

    100% {

        top: 0;

        left: 0;

    }

}

.square-border-color {

    width: 30px;

    height: 30px;

    position: absolute;

    animation: borderColorMove 4s linear infinite;

}

@keyframes borderColorMove {

    0% {

        top: 0;

        left: 0;

        background-color: red;

    }

    25% {

        top: 0;

        left: calc(100% - 30px);

        background-color: green;

    }

    50% {

        top: calc(100% - 30px);

        left: calc(100% - 30px);

        background-color: blue;

    }

    75% {

        top: calc(100% - 30px);

        left: 0;

        background-color: yellow;

    }

    100% {

        top: 0;

        left: 0;

        background-color: red;

    }

}

.square-move-bg-change {

    width: 50px;

    height: 50px;

    position: absolute;

    animation: moveAndChange 3s linear infinite alternate;

}

@keyframes moveAndChange {

    0% {

        left: 0;

        background-color: purple;

    }

    50% {

        left: 100px;

        background-color: pink;

    }

    100% {

        left: 200px;

        background-color: purple;

    }

}

Контрольные вопросы:

1. Перечислите основные CSS-свойства, которые используются для создания анимаций и визуальных эффектов на веб-странице.

Ответ:

CSS предлагает множество свойств для создания анимаций и эффектов:

* transition – позволяет плавно изменять значения свойств (например, цвет, размер, прозрачность).
* transform – используется для трансформации элементов (например, rotate(), scale(), translate()).
* animation – управляет анимациями с использованием ключевых кадров.
* keyframes – задает последовательность изменений стилей во времени.
* opacity – позволяет создавать эффекты появления и исчезновения.
* box-shadow – добавляет тени для создания глубины.
* filter – добавляет визуальные эффекты, такие как размытие или изменение цвета.

2. Объясните разницу между CSS-анимациями и анимациями, реализованными с помощью JavaScript. В каких случаях предпочтительнее использовать каждый из подходов?

Ответ:

CSS-анимации:

* Преимущество: работают плавно и оптимизированы браузером (аппаратное ускорение).
* Недостаток: ограниченные возможности взаимодействия (например, нельзя динамически менять анимацию в процессе ее выполнения).
* Используются для простых эффектов: затухания, движения, трансформаций.

JavaScript-анимации:

* Преимущество: обеспечивают полный контроль и динамическое изменение параметров анимации.
* Недостаток: требуют больше ресурсов (особенно если не оптимизированы).
* Используются для сложных сценариев: интерактивных эффектов, анимации графиков, параллакс-скролла.

3. Опишите процесс создания анимации с помощью ключевых кадров (keyframes) в CSS.

Ответ:

Ключевые кадры (@keyframes) позволяют задавать изменение стиля элемента в разные моменты времени:

@keyframes example-animation {

0% { opacity: 0; transform: translateY(-20px); }

50% { opacity: 0.5; transform: translateY(10px); }

100% { opacity: 1; transform: translateY(0); }

}

.element {

animation: example-animation 1s ease-in-out infinite;

}

Здесь:

* Анимация изменяет прозрачность (opacity) и положение (transform).
* 0%, 50%, 100% задают этапы анимации.
* 1s – длительность, ease-in-out – вид ускорения, infinite – бесконечное повторение.