

ДИСЦИПЛИНА	Фронтенд и бэкенд разработка
ИНСТИТУТ	ИПТИП
КАФЕДРА	Индустриального программирования
ВИД УЧЕБНОГО МАТЕРИАЛА	Методические указания к практическим занятиям
ПРЕПОДАВАТЕЛЬ	Астафьев Рустам Уралович
СЕМЕСТР	1 семестр, 2025/2026 уч. год

Ссылка на материал:

<https://github.com/astafiev-rustam/frontend-and-backend-development/tree/practice-1-11>

Практическое занятие 11: БЭМ, переменные и состояния в CSS

Теоретическая справка

Методология БЭМ (Блок-Элемент-Модификатор)

БЭМ - это методология именования CSS-классов, которая создает понятную и масштабируемую структуру стилей. Основная идея - разделение интерфейса на независимые компоненты.

Ключевые концепции БЭМ

Блок - самостоятельный компонент, который можно использовать повторно. Блоки независимы и не должны влиять на свое окружение.

```
/* Блоки - основные компоненты интерфейса */  
.header { }  
.menu { }  
.search-form { }
```

Элемент - составная часть блока, которая не имеет смысла вне своего блока. Элементы всегда принадлежат блоку.

```
/* Элементы - части блоков */  
.header__logo { } /* Логотип в хедере */  
.menu__item { } /* Пункт меню */  
.search-form__input { } /* Поле ввода в форме поиска */
```

Модификатор - определяет внешний вид, состояние или поведение блока/элемента.

```
/* Модификаторы - варианты блоков/элементов */
.button--large { } /* Большая версия кнопки */
.menu__item--active { } /* Активный пункт меню */
.search-form--collapsed { } /* Свернутое состояние формы */
```

Практический пример БЭМ

```
<!-- Блок навигации с модификатором -->
<nav class="nav nav--main">
  <!-- Элементы навигации -->
  <a class="nav__link nav__link--active" href="#">Главная</a>
  <a class="nav__link" href="#">О нас</a>
  <a class="nav__link nav__link--disabled" href="#">Контакты</a>
</nav>
```

```
/* Стилизация по БЭМ */
.nav { /* стили блока */ }
.nav--main { /* модификатор блока */ }
.nav__link { /* стили элемента */ }
.nav__link--active { /* модификатор элемента */ }
.nav__link--disabled { /* модификатор элемента */ }
```

CSS переменные (Custom Properties)

CSS переменные позволяют хранить значения для многократного использования в документе. Они обеспечивают централизованное управление стилями и простую темизацию.

Объявление и использование переменных

Переменные объявляются с двойным дефисом и обычно размещаются в селекторе `:root` для глобальной доступности.

```
/* Объявление переменных в корневом элементе */
:root {
  --primary-color: #3498db;
  --secondary-color: #2ecc71;
  --font-size-base: 16px;
  --spacing-unit: 1rem;
  --border-radius: 8px;
}

/* Использование переменных */
.button {
  background-color: var(--primary-color);
  font-size: var(--font-size-base);
}
```

```
padding: var(--spacing-unit);
border-radius: var(--border-radius);
}

/* Резервное значение если переменная не определена */
.element {
  color: var(--undefined-color, #000000);
}
```

Динамическое управление переменными

CSS переменные можно изменять через JavaScript и медиа-запросы, что делает их мощным инструментом для создания адаптивных интерфейсов.

```
// Изменение переменной через JavaScript
document.documentElement.style.setProperty('--primary-color', '#e74c3c');
```

```
/* Адаптивные переменные для разных экранов */
@media (max-width: 768px) {
  :root {
    --font-size-base: 14px;
    --spacing-unit: 0.5rem;
  }
}
```

Состояния в CSS

Состояния определяют как элементы выглядят и ведут себя в различных условиях. CSS предоставляет несколько механизмов для работы с состояниями.

Псевдоклассы состояний

Псевдоклассы позволяют стилизовать элементы на основе их состояния или положения в документе.

```
/* Состояния взаимодействия */
.button:hover { } /* При наведении курсора */
.button:active { } /* В момент активации */
.button:focus { } /* При получении фокуса */
.button:disabled { } /* В отключенном состоянии */

/* Структурные состояния */
.item:first-child { } /* Первый элемент */
.item:last-child { } /* Последний элемент */
.item:nth-child(odd) { } /* Нечетные элементы */

/* Состояния форм */
```

```
.input:valid { }      /* Валидное значение */  
.input:invalid { }    /* Невалидное значение */  
.input:required { }   /* Обязательное поле */
```

Кастомные состояния через data-атрибуты

Data-атрибуты позволяют создавать собственные состояния для сложных компонентов.

```
<div class="tab" data-state="active">Вкладка 1</div>  
<div class="tab" data-state="inactive">Вкладка 2</div>
```

```
/* Стилизация на основе data-атрибутов */  
.tab[data-state="active"] {  
  background-color: var(--primary-color);  
  color: white;  
}  
  
.tab[data-state="inactive"] {  
  background-color: #f0f0f0;  
  color: #666;  
}
```

Состояния через модификаторы БЭМ

В методологии БЭМ состояния часто выражаются через модификаторы.

```
/* Состояния через БЭМ модификаторы */  
.menu__item--active { }      /* Активный пункт меню */  
.modal--open { }             /* Открытое модальное окно */  
.accordion--expanded { }     /* Развернутый аккордеон */  
.form--valid { }             /* Валидная форма */
```

Комбинированный пример

Вот как эти три концепции работают вместе для создания поддерживаемого и гибкого кода:

```
/* CSS переменные для темизации */  
:root {  
  --primary-color: #667eea;  
  --hover-color: #764ba2;  
  --transition: all 0.3s ease;  
}  
  
/* БЭМ компонент с состояниями */
```

```
.button {
  padding: 12px 24px;
  border: none;
  border-radius: 6px;
  transition: var(--transition);
  cursor: pointer;
}

.button--primary {
  background-color: var(--primary-color);
  color: white;
}

/* Состояния кнопки */
.button--primary:hover {
  background-color: var(--hover-color);
  transform: translateY(-2px);
}

.button--disabled {
  opacity: 0.6;
  cursor: not-allowed;
}

.button--disabled:hover {
  transform: none;
}
```

```
<button class="button button--primary">Обычная кнопка</button>
<button class="button button--primary button--disabled">Отключенная
кнопка</button>
```

Этот подход объединяет преимущества всех трех технологий: четкую структуру БЭМ, гибкость CSS переменных и интерактивность состояний.

ЭТАП 1. Практический пример

Отредактируем файл `index.html` так, чтобы реализовать некоторые из концепций теоретического блока в примере, так же отредактируем `styles.css` в соответствии с новой системой классов.

Таким образом, результат реализации следующий:

```
<!DOCTYPE html>
<html lang="ru">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Главная - Портфолио</title>
  <link rel="stylesheet" href="style/styles.css">
```

```
</head>
<body>
  <!-- Основной контейнер -->
  <div class="page">

    <!-- Шапка -->
    <header class="header">
      <h1 class="header__title">Моё портфолио</h1>
    </header>

    <!-- Навигация -->
    <nav class="nav">
      <ul class="nav__list">
        <li class="nav__item nav__item--active">
          <a class="nav__link" href="index.html">Главная</a>
        </li>
        <li class="nav__item">
          <a class="nav__link" href="pages/news.html">Новости</a>
        </li>
        <li class="nav__item">
          <a class="nav__link" href="pages/contacts.html">Контакты</a>
        </li>
      </ul>
    </nav>

    <!-- Основной контент -->
    <main class="main">

      <!-- Профиль -->
      <section class="profile">
        <div class="profile__card">
          
          <div class="profile__info">
            <h2 class="profile__name">Иванов Иван</h2>
            <p class="profile__group">ЭФБО-01-22</p>
          </div>
        </div>
      </section>

      <!-- Обо мне -->
      <section class="about">
        <h2 class="about__title">Обо мне</h2>
        <div class="about__content">
          <p class="about__text">
            Студент, увлекаюсь веб-разработкой. Изучаю HTML, CSS и
            JavaScript.
            Люблю создавать красивые и функциональные сайты.
          </p>
        </div>
      </section>

      <!-- Навыки -->
      <section class="skills">
        <h2 class="skills__title">Мои навыки</h2>
```

```

        <div class="skills__list">
            <div class="skill skill--html">
                <span class="skill__name">HTML</span>
            </div>
            <div class="skill skill--css">
                <span class="skill__name">CSS</span>
            </div>
            <div class="skill skill--js">
                <span class="skill__name">JavaScript</span>
            </div>
        </div>
    </section>

    <!-- Проекты -->
    <section class="projects">
        <h2 class="projects__title">Проекты</h2>
        <div class="projects__grid">
            <div class="project-card">
                <h3 class="project-card__title">Личный сайт</h3>
                <p class="project-card__desc">Сайт-портфолио на HTML и
CSS</p>
            </div>
            <div class="project-card">
                <h3 class="project-card__title">Todo приложение</h3>
                <p class="project-card__desc">Простое приложение для
задач</p>
            </div>
        </div>
    </section>

</main>

<!-- Подвал -->
<footer class="footer">
    <p class="footer__text">&copy; 2024 Мой сайт</p>
</footer>
</div>
</body>
</html>

```

Для файла `styles.css`:

```

/* CSS переменные */
:root {
    --color-main: #2c3e50;
    --color-accent: #3498db;
    --color-light: #ecf0f1;
    --color-dark: #34495e;
    --color-text: #2c3e50;
    --color-white: #ffffff;

    --shadow: 0 2px 5px rgba(0,0,0,0.1);

```

```
--radius: 8px;

--space-sm: 10px;
--space-md: 20px;
--space-lg: 30px;
}

/* Базовые стили */
* {
  margin: 0;
  padding: 0;
  box-sizing: border-box;
}

body {
  font-family: Arial, sans-serif;
  line-height: 1.6;
  color: var(--color-text);
  background-color: var(--color-light);
}

/* Блок page */
.page {
  max-width: 1200px;
  margin: 0 auto;
  background: var(--color-white);
  min-height: 100vh;
}

/* Блок header */
.header {
  background: var(--color-main);
  color: var(--color-white);
  padding: var(--space-lg);
  text-align: center;
}

.header__title {
  font-size: 2rem;
}

/* Блок nav */
.nav {
  background: var(--color-dark);
}

.nav__list {
  list-style: none;
  display: flex;
  justify-content: center;
}

.nav__item {
  margin: 0;
```



```
}

.nav__link {
  display: block;
  padding: var(--space-md);
  color: var(--color-white);
  text-decoration: none;
  transition: background-color 0.3s;
}

.nav__link:hover {
  background: var(--color-accent);
}

/* Модификатор --active */
.nav__item--active .nav__link {
  background: var(--color-accent);
  font-weight: bold;
}

/* Блок main */
.main {
  padding: var(--space-lg);
}

/* Блок profile */
.profile {
  margin-bottom: var(--space-lg);
}

.profile__card {
  display: flex;
  align-items: center;
  gap: var(--space-lg);
  background: var(--color-white);
  padding: var(--space-lg);
  border-radius: var(--radius);
  box-shadow: var(--shadow);
}

.profile__photo {
  width: 150px;
  height: 150px;
  border-radius: 50%;
  border: 3px solid var(--color-accent);
}

.profile__name {
  font-size: 1.5rem;
  color: var(--color-main);
  margin-bottom: var(--space-sm);
}

.profile__group {
```

```
        color: var(--color-accent);
        font-size: 1.1rem;
    }

    /* Блок about */
    .about {
        background: var(--color-white);
        padding: var(--space-lg);
        margin-bottom: var(--space-lg);
        border-radius: var(--radius);
        box-shadow: var(--shadow);
    }

    .about__title {
        color: var(--color-main);
        margin-bottom: var(--space-md);
        font-size: 1.5rem;
    }

    .about__text {
        line-height: 1.6;
    }

    /* Блок skills */
    .skills {
        margin-bottom: var(--space-lg);
    }

    .skills__title {
        color: var(--color-main);
        margin-bottom: var(--space-md);
        font-size: 1.5rem;
    }

    .skills__list {
        display: flex;
        gap: var(--space-md);
        flex-wrap: wrap;
    }

    /* Блок skill с модификаторами */
    .skill {
        padding: var(--space-md);
        border-radius: var(--radius);
        color: var(--color-white);
        font-weight: bold;
        transition: transform 0.3s;
    }

    .skill:hover {
        transform: translateY(-3px);
    }

    .skill--html {
```

```
        background: #e34c26;
    }

    .skill--css {
        background: #264de4;
    }

    .skill--js {
        background: #f0db4f;
        color: var(--color-text);
    }

    /* Блок projects */
    .projects__title {
        color: var(--color-main);
        margin-bottom: var(--space-md);
        font-size: 1.5rem;
    }

    .projects__grid {
        display: grid;
        grid-template-columns: repeat(auto-fit, minmax(300px, 1fr));
        gap: var(--space-md);
    }

    /* Блок project-card */
    .project-card {
        background: var(--color-white);
        padding: var(--space-lg);
        border-radius: var(--radius);
        box-shadow: var(--shadow);
        border-left: 4px solid var(--color-accent);
        transition: transform 0.3s;
    }

    .project-card:hover {
        transform: translateY(-5px);
    }

    .project-card__title {
        color: var(--color-main);
        margin-bottom: var(--space-sm);
    }

    .project-card__desc {
        color: #666;
    }

    /* Блок footer */
    .footer {
        background: var(--color-main);
        color: var(--color-white);
        text-align: center;
        padding: var(--space-md);
    }
```

```
    margin-top: var(--space-lg);
}

/* Адаптивность */
@media (max-width: 768px) {
    .profile__card {
        flex-direction: column;
        text-align: center;
    }

    .nav__list {
        flex-direction: column;
    }

    .skills__list {
        justify-content: center;
    }

    .projects__grid {
        grid-template-columns: 1fr;
    }
}

/* Добавляем в конец файла styles.css */

/* Блок news */
.news {
    background: var(--color-white);
    padding: var(--space-lg);
    border-radius: var(--radius);
    box-shadow: var(--shadow);
}

.news__title {
    color: var(--color-main);
    margin-bottom: var(--space-md);
    font-size: 1.5rem;
}

.news__list {
    display: flex;
    flex-direction: column;
    gap: var(--space-md);
}

/* Блок news-item */
.news-item {
    padding: var(--space-md);
    border-left: 4px solid var(--color-accent);
    background: var(--color-light);
    border-radius: var(--radius);
}

.news-item__title {
```

```
    color: var(--color-main);
    margin-bottom: var(--space-sm);
}

.news-item__date {
    color: var(--color-accent);
    font-size: 0.9rem;
    margin-bottom: var(--space-sm);
}

.news-item__text {
    line-height: 1.5;
}

/* Блок contacts */
.contacts {
    background: var(--color-white);
    padding: var(--space-lg);
    border-radius: var(--radius);
    box-shadow: var(--shadow);
}

.contacts__title {
    color: var(--color-main);
    margin-bottom: var(--space-md);
    font-size: 1.5rem;
}

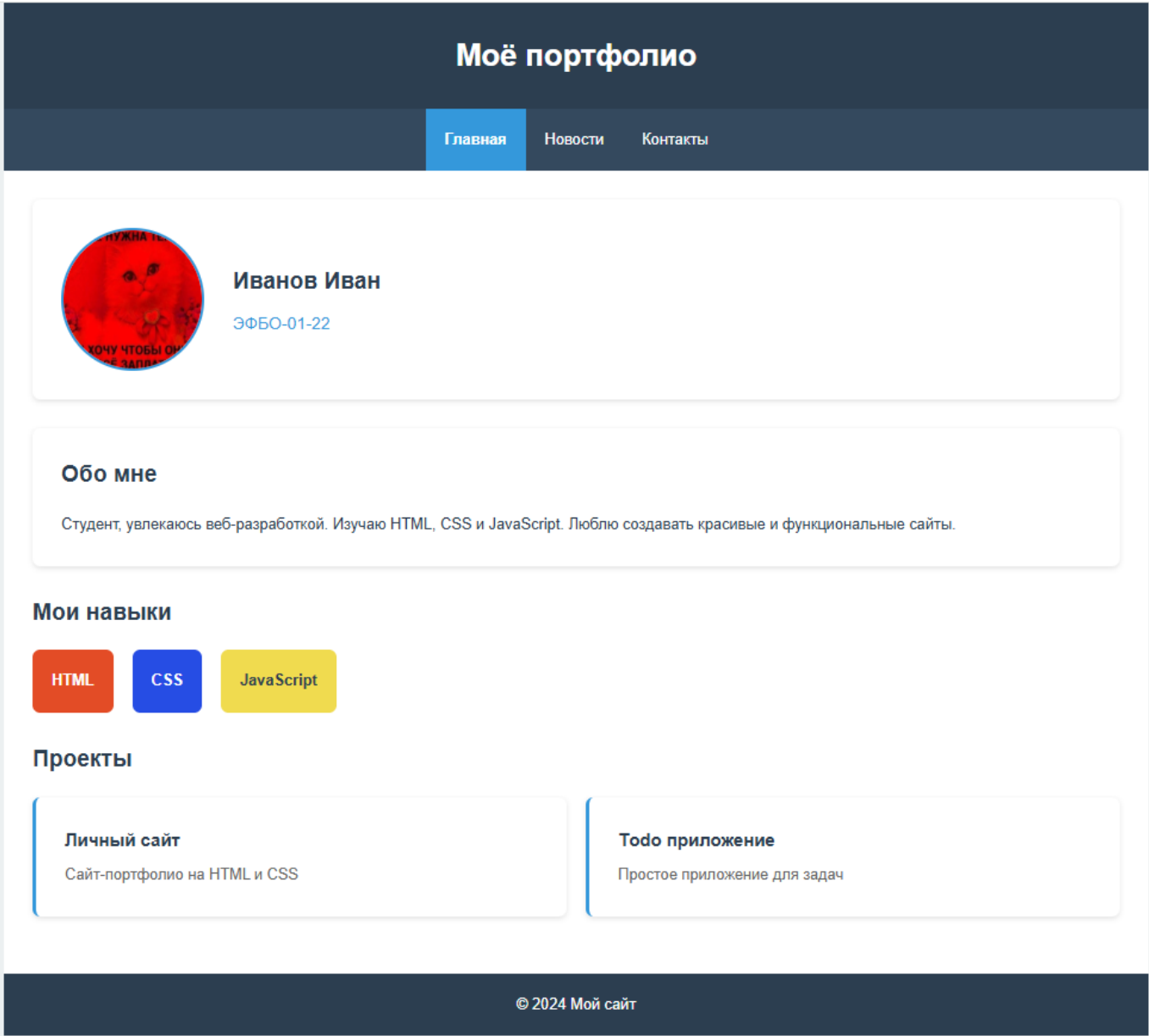
.contacts__list {
    display: grid;
    grid-template-columns: repeat(auto-fit, minmax(250px, 1fr));
    gap: var(--space-md);
}

/* Блок contact-item */
.contact-item {
    padding: var(--space-md);
    background: var(--color-light);
    border-radius: var(--radius);
    text-align: center;
}

.contact-item__title {
    color: var(--color-main);
    margin-bottom: var(--space-sm);
    font-size: 1.1rem;
}

.contact-item__value {
    color: var(--color-accent);
    font-weight: bold;
}
```

Получим следующий результат:



ЭТАП2. Самостоятельная работа

Вам необходимо модернизировать существующий проект, внедрив методологию БЭМ для именования классов, создав систему CSS переменных для управления дизайном и улучшив интерактивность через состояния элементов.

Задача 1: Создание системы CSS переменных

Создайте централизованную систему CSS переменных для управления дизайном проекта. Определите переменные для цветовой палитры, типографики, размеров и отступов. Замените прямые значения в существующих стилях на созданные переменные.

Задача 2: Рефакторинг по методологии БЭМ

Переименуйте CSS классы в соответствии с методологией БЭМ. Выделите независимые блоки, их элементы и создайте модификаторы для различных состояний и вариантов отображения. Убедитесь, что структура классов становится более понятной и масштабируемой.

Задача 3: Реализация интерактивных состояний

Добавьте визуальную обратную связь для интерактивных элементов. Реализуйте состояния при наведении, фокусе и активном состоянии для кнопок, карточек и элементов навигации. Используйте плавные переходы для анимации изменений.

Задача 4: Создание тематического переключателя

Реализуйте функциональность переключения между светлой и темной темой оформления. Создайте соответствующие наборы CSS переменных для каждой темы и механизм переключения между ними.

Задача 5: Компонент карточки товара с состояниями

Разработайте универсальный компонент карточки с использованием БЭМ методологии. Реализуйте различные состояния карточки: базовое, при наведении, выбранное, со скидкой. Добавьте соответствующие модификаторы для каждого состояния.

Требования к результату

- Все CSS классы именуются по методологии БЭМ
- Дизайн управляется через CSS переменные
- Интерактивные элементы имеют визуальную обратную связь
- Реализована возможность смены темы оформления
- Компоненты переиспользуемы и масштабируемы
- Код хорошо структурирован и комментирован

Базовое задание

В качестве результата по итогам занятия необходимо прикрепить ссылку в соответствующий раздел рабочей области в СДО.