

## Практическое задание №2: Галерея с фильтрацией и анимацией (Можно группой из 2-х человек)

**Цель:** Разработать интерактивную галерею изображений с возможностью фильтрации по категориям и плавной анимацией, применяя знания HTML, CSS, JavaScript и принципы командной работы.

**Описание:** Вам предстоит создать веб-страницу, которая отображает галерею изображений в виде карточек. Каждая карточка должна содержать изображение, название, описание и теги (категории). Пользователь должен иметь возможность фильтровать изображения по тегам, а процесс фильтрации должен сопровождаться плавной анимацией.

### Функциональные требования:

1. **Отображение карточек:** Изображения должны быть представлены в виде карточек с названием, описанием и тегами (категориями). Расположение карточек должно быть адаптивным и выглядеть гармонично на разных разрешениях экрана.
2. **Фильтрация:** Пользователь должен иметь возможность фильтровать изображения по тегам. Фильтрация должна быть реализована с помощью кнопок или другого удобного интерфейса. При выборе тега должны отображаться только карточки, соответствующие этому тегу.
3. **Анимация при фильтрации:** При применении фильтра карточки должны плавно появляться или исчезать. Избегайте резких переключений.
4. **Добавление новых изображений (опционально, + к оценке):** Реализуйте возможность добавления новых изображений в галерею. Это можно сделать двумя способами:
  - **Вручную:** Через форму на странице, где пользователь может загрузить изображение, указать название, описание и теги.
  - **Из JSON-файла:** Предоставьте возможность загрузить данные о новых изображениях из JSON-файла.
5. **Адаптивный дизайн:** Галерея должна корректно отображаться на разных устройствах (десктоп, планшет, мобильный).

### Темы для повторения:

- **HTML:** Структура HTML документа, семантические теги, работа с изображениями.
- **CSS:** Селекторы, стилизация элементов, Flexbox/Grid (для расположения карточек), анимации и переходы, медиа-запросы (для адаптивности).
- **JavaScript:** Работа с DOM, обработка событий, работа с массивами данных, динамическое создание и удаление элементов на странице, асинхронные запросы (для загрузки данных из JSON), (опционально: работа с формами для загрузки изображений).

### Инструменты (рекомендации):

- **Контроль версий:** Git (обязательно для групповой работы)

### Критерии оценки:

- **Функциональность (+1):** Все функции работают корректно.

- **Дизайн (+1):** Визуальное оформление галереи, гармоничность, удобство использования.
- **Качество кода (+1):** Чистота и структурированность кода, использование комментариев, отсутствие ошибок.
- **Анимации (+1):** Качество и плавность анимаций.
- **Адаптивность (+1):** Корректное отображение на разных устройствах.
- **Дополнительно (+5):** Возможность добавления новых изображений.

#### Что будет проверяться:

- Работоспособность всех функций (отображение, фильтрация, анимация).
- Соответствие дизайна требованиям.
- Качество и чистота кода.
- История коммитов в Git (для оценки командной работы).
- Адаптивность верстки.

#### Полезные ресурсы и примеры кода:

- Работа с массивами в JavaScript: [MDN Array](#)
- Fetch API (для загрузки данных из JSON): [MDN Fetch API](#)
- CSS анимации: [CSS Transitions](#), [CSS Animations](#)
- Примеры галерей с фильтрацией: Поищите на CodePen, GitHub, Dribbble примеры галерей изображений с фильтрацией. Используйте их для вдохновения, но **не копируйте код напрямую**.

#### Сдача работы:

Работу необходимо сдать двумя способами:

1. **Ссылка на репозиторий Git:** Создайте репозиторий на GitHub и загрузите туда весь код вашего проекта. Убедитесь, что репозиторий публичный (или доступен преподавателю). Пришлите ссылку на репозиторий.
2. **Анимация работы приложения (GIF или видео):** Запишите короткий GIF или видео (например, в формате MP4), демонстрирующий работу вашего приложения. Покажите на видео все основные функции: отображение карточек, фильтрацию по тега, анимации, а также (опционально) добавление новых изображений. Загрузите GIF/видео на любой файлообменник и пришлите ссылку.