**Практическая работа Создание фотогалерии на CSS**

#### Пошаговый план создания этой галереи следующий:

1. Разработка адаптивной сетки.
2. Оформление миниатюр.
3. Стилизация подписей.
4. Финальные штрихи.
5. Дополнительно: подключение плагина для всплывающих окон.

### Загрузка файлов

Скачайте архив с файлами и откройте в удобном для вас редакторе кода файлы gallery.html и style.css (из папки css). В теге <head> подключен файл сброса стилей и основную таблицу стилей , а также шрифт Google Fonts. Дополнительно добавлен еще одна таблица стилей lightbox.min.css, а в конце документа — скрипт lightbox-plus-jquery.min.js.

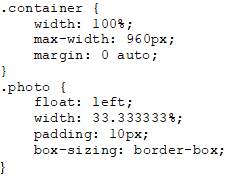
## Создание фотогалереи

Изучите структуру HTML-страницы. В теле документа расположен блок-контейнер, в котором есть заголовок <h1> и основной блок <div> с идентификатором #gallery. Внутри блока галереи находится девять HTML5-тегов <figure> с классом .photo, каждый из которых содержит тег <img> с миниатюрой изображения и тег <figcaption> с подписью к фото. Кроме этого, каждое изображение заключено в тег <a>, который содержит ссылку на соответствующий полноразмерный графический файл.

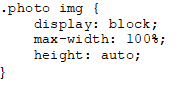
## 1. Разработка адаптивной сетки

Сетка является своего рода каркасом, определяющим расположение элементов на веб-странице. Ширина блока-контейнера будет иметь максимальную ширину 960 пикселей, а миниатюры будут выстраиваться в три столбца одинаковой ширины. Каждая сторона миниатюры будет иметь внутренний отступ в размере 10 пикселей.

Основываясь на описании выше, запишите первый стиль в файл style.css:

.

Проверьте адаптивность сетки. Измените ширину окна браузера несколько раз и понаблюдайте за поведением элементов. При ширине менее 977 пикселей элементы начинают некорректно себя вести. Это происходит потому, что изображения миниатюр не подстраиваются под ширину контейнера. Добавьте следующий стиль для обеспечения адаптивности картинок:



## 2. Оформление миниатюр

Согласно макету, каждая миниатюра должна быть обрамлена в рамку светло-серого цвета. Ширина рамки с каждой стороны составляет 10 пикселей. Реализовать эту рамку можно двумя способами, и визуально они ничем не будут отличаться.

##### Способ I: свойство border

Первый способ — создать рамку нужного цвета и ширины для каждого тега <img>.  
Добавьте этот код к селектору .photo img:

border: 10px solid #eee;

box-sizing: border-box;

Обновите страницу в браузере и запомните результат.

##### Способ II: свойства background-color и padding

Второй способ заключается в том, чтобы имитировать вид рамки посредством создания 10-пиксельных внутренних отступов и закрашивания фона в светло-серый цвет. Замените код из первого способа на следующий:

padding: 10px;

background-color: #eee;

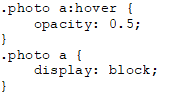
box-sizing: border-box;

Сохраните изменения и обновите веб-страницу.

### Стиль при наведении

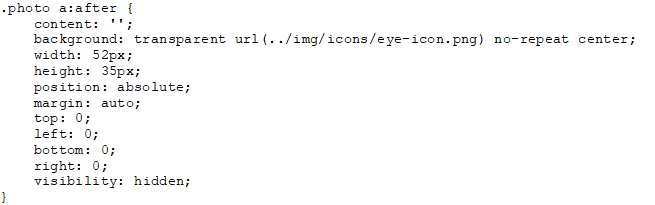
Миниатюра должна выглядеть в состоянии наведения на нее курсора следующим образом: фотография становится полупрозрачной, а в центре элемента появляется иконка глаза, намекающая на то, что клик по миниатюре означает просмотр полного изображения.

Запишите стиль для псевдокласса .photo a:hover:



Теперь, когда вы сохраните таблицу стилей, обновите страницу в браузере и наведете курсор на любую из фотографий, то увидите, что она стала полупрозрачной.

Добавление иконки глаза. Для этого понадобится задействовать псевдоэлемент :after. Добавьте следующие стили в вашу таблицу:

Скрыли псевдоэлемент с иконкой для того чтобы делать его видимым только при наведении курсора на ссылку-миниатюру. И для этого мы снова обратимся к псевдоклассу :hover. Запишем следующий код:

.photo:hover > a:after {

visibility: visible;

}

Обновите страницу и посмотрите, как теперь реагируют элементы на наведение курсора.

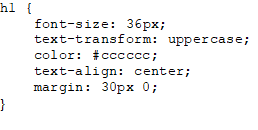
## 3. Стилизация подписей

В идеале, подпись к фотографии должна выглядеть лаконично и не слишком бросаться в глаза. Поэтому стиль для нее будет очень простым:

## 

## 4. Финальные штрихи

Стилизация заголовка над галереей.



Чтобы не дописывать свойство font-family к каждому элементу, необходимо задать его для всего тега <body>, после чего стереть эту строку из стиля для .photo figcaption — она там теперь лишняя:

body {

font-family: 'Open Sans', sans-serif;

}

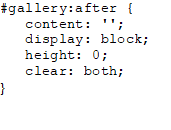
А чтобы сам блок галереи не упирался в низ веб-страницы, добавьте ему небольшой отступ снизу:

#gallery {

margin-bottom: 50px;

}

Отключите игнорирование float-элементов родителем с помощью следующего стиля:



Добавленный нижний отступ в 60 пикселей теперь находится там, где и требовалось.

## 5. Подключение плагина для всплывающих окон

Плагин Lightbox обеспечит красивое открытие полноразмерных фотографий во всплывающем окне, без покидания страницы, а также добавит возможность перелистывать фото прямо на месте.

Плагин Lightbox, работающий на базе JavaScript-библиотеки jQuery, используется для наложения изображений поверх текущей страницы. Это удобный инструмент, который работает в любом современном браузере.

В начале урока подключили необходимый скрипт и таблицу стилей к HTML-документу. Вам останется лишь инициализировать его, добавив атрибут data-lightbox="roadtrip" к каждому тегу <a>, который ссылается на изображение. Этот код говорит плагину, что ссылку нужно открыть во всплывающем окне, а также добавить возможность переключаться между фотографиями, используя боковые стрелки.

Продублируйте атрибут для каждой из девяти ссылок галереи:

<a href="img/originals/img-01.jpg" data-lightbox="roadtrip">

<img src="img/img-01-min.jpg" alt="Eagle" />

</a>

Затем сохраните изменения в документе и просмотрите его в браузере.

Чтобы плагин отображал в открытом окне и подпись к фотографии. Для этого добавьте к ссылке еще один атрибут — data-title="", а внутрь его кавычек поместите текст подписи, скопировав из тега <figcaption>:

<a href="img/originals/img-01.jpg" data-lightbox="roadtrip" data-title="Eagle">

<img src="img/img-01-min.jpg" alt="Eagle" />

</a>

Повторите эти действия для остальных ссылок и проверьте результат.