Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Ульяновский государственный технический университет»

Кафедра «Вычислительная техника»

Дисциплина «Интернет-программирование»

Лабораторная работа № 3

«Создание анимированной галереи»

Выполнил студент

группы ИВТАПбд-12

Ступин А.О.

Проверил: Игонин А. Г.

Ульяновск, 2022

Оглавление

[Цель работы 3](#_Toc104304523)

[Описание метода выполнения 3](#_Toc104304524)

[Вывод 9](#_Toc104304525)

[Источники 9](#_Toc104304526)

[Код 9](#_Toc104304527)

## Цель работы

Целью работы является создание персональной галереи на личном сайте. Фотографии должны листаться, и переход к следующей/предыдущей фотографии должен быть анимирован.

## Описание метода выполнения

В лабораторной работе использовались HTML, CSS и JavaScript.

Первым делом сохраняем некоторое количество фотографий (пусть будет 20) в специальную папку gallery, созданную в папке images. Для удобства назовем их одинаково, лишь присваивая разные номера по возрастанию. Также сохраним туда изображения для кнопок будущего модального окна.

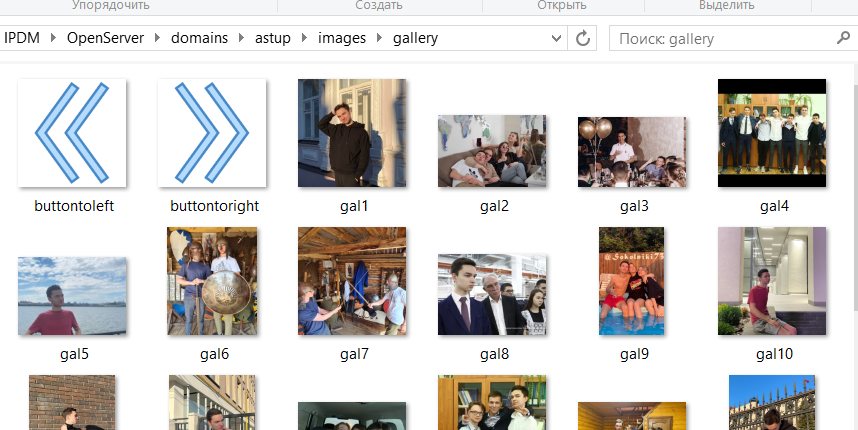


Рис. 1 – Фотографии, которые мы будем использовать

Файл gallery.php имеет следующую html-структуру:

|  |
| --- |
| <div class = "photos">  <ul>  <li><img src = "images/gallery/gal1.jpg" onclick="Show(1)" title = "Я" alt="Фото не прогрузилось"></li>  <li><img src = "images/gallery/gal2.jpg" onclick="Show(2)" title = "В гостях" alt="Фото не прогрузилось"></li>  <li><img src = "images/gallery/gal3.jpg" onclick="Show(3)" title = "Выпускной" alt="Фото не прогрузилось"></li>  </ul>  </div> |

Класс photos занимает всю страницу галереи. Он разбит на ненумерованные списки, в каждом из которых по 3 фотографии.

Дабы расположить фотографии в 3 колонны, классу photos ul и родительскому photos были заданы css-свойства:

|  |
| --- |
| .photos {  justify-content: center;  display: flex;  flex-direction: column;  margin: 0 10% 0 10%;  width: 80%;  position: relative; }  .photos ul {  list-style: none;  display: flex;  flex-direction: row;  align-items: center;  justify-content: center; } |

Также были даны свойства ячейкам, и самим изображениям (li и li img соответственно), чтобы они занимали треть каждой «строки» и имели примерно одинаковую ширину на странице:

|  |
| --- |
| .photos ul li {  margin-top: 0.5%;  width: 33%;  }  .photos ul li img{  cursor: pointer;  width: 80%;  margin: 0 0 0 3.5%;  background: #f0f0f0;  border-radius: 5px;  padding: 4%; }  .photos ul li img:hover {  transform: scale(1.1); } |

Псевдо-класс hover был добавлен, чтобы при наведении курсора на изображение оно увеличивалось на 10%.

Фотографии добавлены, теперь нужно, чтобы они открывались. Для этого, после класса photos, в gallery.php был добавлен div c классом window, внутри него – класс content с содержимым окна, а в содержимом: кнопка выхода и класс main-gallery, в котором находятся собственно изображение и кнопки перехода вперед/назад.

Всю последующую работу выполняет Javascript. В файле gallery.js объявляем переменную numImg для номера каждой картинки (изначальное её значение задаем 1), а также константы cntImg (кол-во изображений, равное 20) и pathImg (путь до изображения, «/images/gallery/»).

Функция смены изображения, где также реализована анимация, названа imgChange. Содержимое функции выглядит так:

|  |
| --- |
| $("#mainimg").fadeOut("slow", function () {  $("#mainimg").attr('src', ***pathImg*** + "gal" + ***numImg*** + ".jpg");  $("#mainimg").animate({  height:'toggle',  }, 1000);  $("#mainimg").fadeIn("slow"); |

Здесь первая строка – анимация «затухания» картинки, от которой мы уходим. Вторая строка – ищет расположение файла с картинкой, к которой мы переходим. Третья строка – анимация «увеличения» новой картинки с заданным временем этого увеличения, а последняя строка – её медленное появление. Все это применяется к mainimg, который мы задали в модальном окне в классе photo.

Для реализации функций анимации используется библиотека jquery.

Далее – функции самих переходов к предыдущему и следующему изображению. Для первой мы проверяем номер изображения. Если он больше 1, то мы его просто уменьшаем, если же он равен 1, то мы приравниваем номер изображения к количеству, переходя, таким образом, с первой на последнюю фотографию.

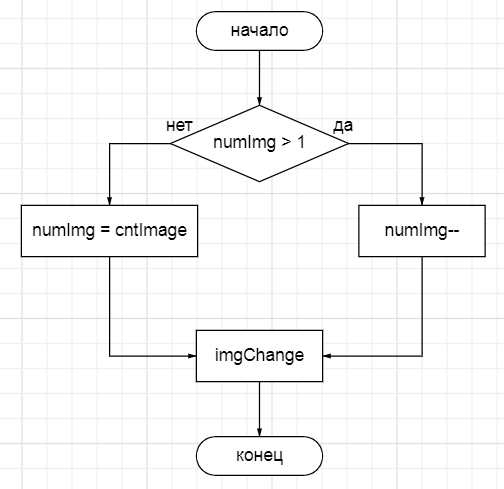


Рис. 2 – блок-схема функции перехода к предыдущему изображению

Функция перехода к следующему изображению аналогично, только сравниваем мы номер не с 1, а с количеством. Если номер меньше количества, то мы увеличиваем номер, если же нет, то приравниваем его к единице, перехода с последней фотографии на первую.

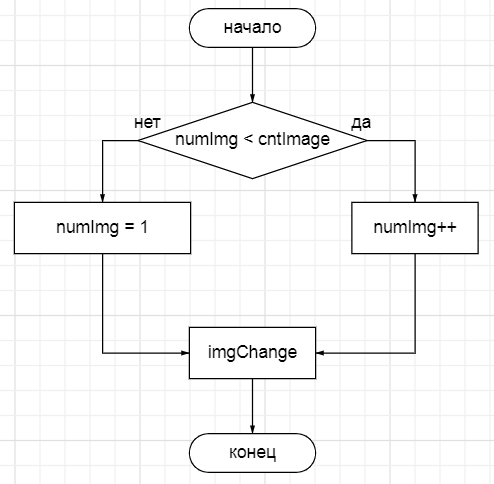


Рис. 3 – блок-схема функции перехода к следующему изображению

Последняя функция, Show, предназначена для того, чтобы модальное окно открывалось по клику. Для этого к каждому изображению в галерее было добавлено события Show(), где в скобках был указан номер картинки. В js-файле в скобки был помещен x, переменная numImg также была приравнена к х.

В последней функции класс window (для которого в css был прописан display:none) меняется на класс window\_active, а body меняется на hidden, для лучшего вида модального окна. Наконец, для закрывающей иконке (icon\_close) прописана работающая по клику функция, которая наоборот меняет window\_active на window и hidden на body.

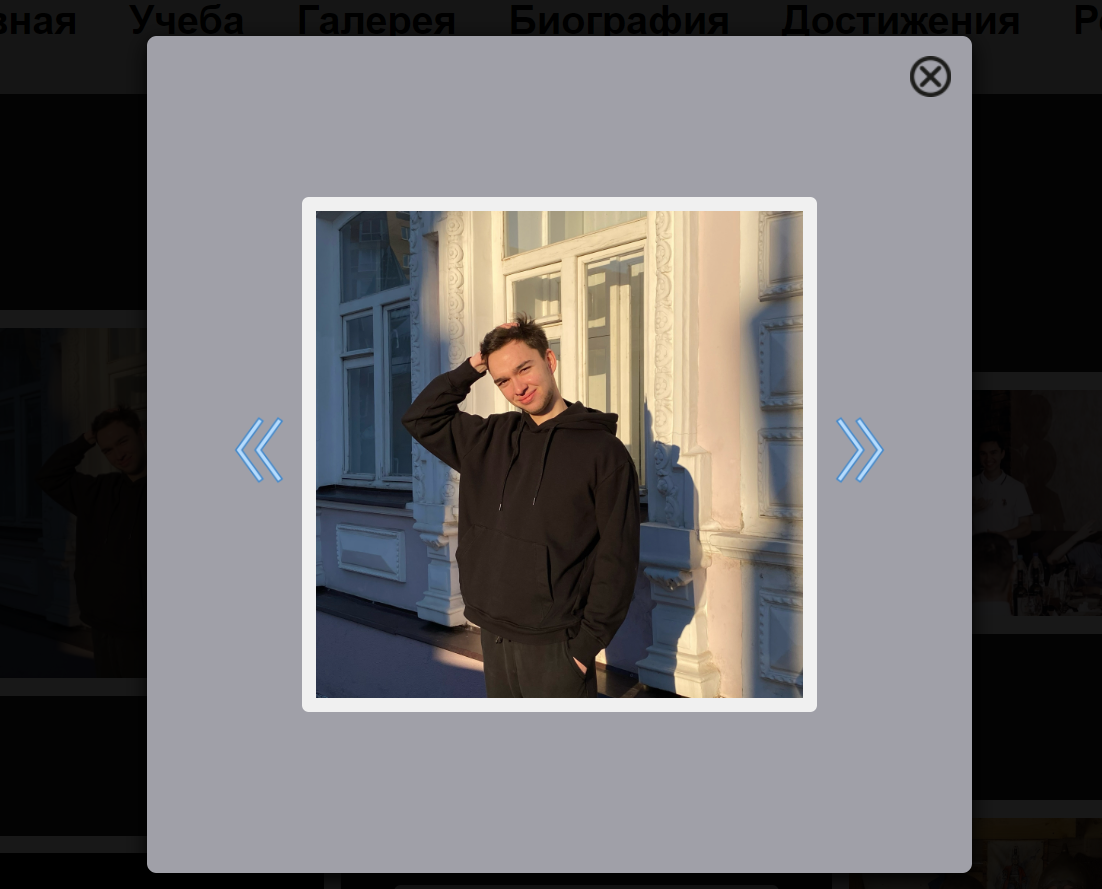


Рис. 4 – вид модального окна



Рис. 5 – затухание предыдущего изображения

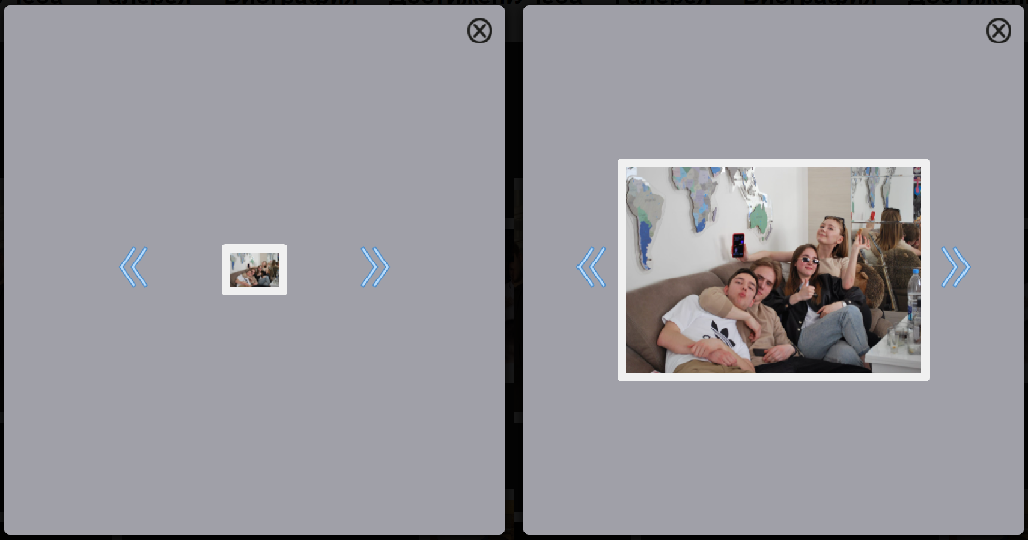


Рис. 6 – увеличение следующего изображения

## Вывод

В ходе выполнения лабораторной работы была сделана галерея, с возможностью просмотра картинок в модальном окне и добавлением анимации перехода между ними.

## Источники

1. HTML5 book «3.10. Введение в jQuery» [электронный ресурс] URL: <https://html5book.ru/vvedenie-v-jquery/>
2. HTML5 book «2.21. CSS3-трансформации» [электронный ресурс] URL: <https://html5book.ru/css3-transform/>
3. Справочник по JavaScript[2022]. [Электронный ресурс] URL: <https://developer.mozilla.org/ru/docs/Web/JavaScript/Reference>

Код

Gallery.js:

let ***numImg*** = 1;  
const ***cntImage*** = 36;  
const ***pathImg*** = "/images/gallery/";  
  
function imgChange() {  
 $("#mainimg").fadeOut("slow", function () {  
 $("#mainimg").attr('src', ***pathImg*** + "gal" + ***numImg*** + ".jpg");  
 $("#mainimg").animate({  
 height:'toggle',  
 }, 1000);  
 $("#mainimg").fadeIn("slow");  
 });  
  
  
}  
  
function leftClick() {  
 if(***numImg*** > 1) {  
 ***numImg***--;}  
 else {  
 ***numImg*** = ***cntImage***;  
 }  
 imgChange();  
}  
  
function rightClick() {  
 if(***numImg*** < ***cntImage***) {  
 ***numImg***++;}  
 else {  
 ***numImg*** = 1;  
 }  
 imgChange();  
}  
  
function Show(x){  
 ***numImg***=x;  
 imgChange();  
 $('.window').addClass('window\_active');  
 $('body').addClass('hidden');  
 $('.icon\_close').click(function () {  
 $('.window').removeClass('window\_active');  
 $('body').removeClass('hidden');  
 });  
  
}

css для галереи:

.photos {  
 justify-content: center;  
 display: flex;  
 flex-direction: column;  
 margin: 0 10% 0 10%;  
 width: 80%;  
 position: relative;  
}  
  
.photos ul {  
 list-style: none;  
 display: flex;  
 flex-direction: row;  
 align-items: center;  
 justify-content: center;  
}  
  
.photos ul li {  
 margin-top: 0.5%;  
 width: 33%;  
  
}  
  
.photos ul li img{  
 cursor: pointer;  
 width: 80%;  
 margin: 0 0 0 3.5%;  
 background: #f0f0f0;  
 border-radius: 5px;  
 padding: 4%;  
}  
  
.photos ul li img:hover {  
 transform: scale(1.1);  
}  
  
#mainimg{  
 width: 100%;  
 align-items: center;  
 display: flex;  
}  
.main-gallery{  
 display: flex;  
 flex-direction: row;  
 justify-content: center;  
 align-items: center;  
}  
.buttonGallery{  
 margin: 225px auto;  
}  
  
.but{  
 width: 51px;  
 height: 51px;  
 border: none;  
}  
button{  
 border: none;  
 background-color:#a0a0a8;  
}  
button:hover{  
 cursor: pointer;  
}  
.window {  
 position: absolute;  
 display: none;  
 overflow-y: scroll;  
}  
.hidden{  
 overflow: hidden;  
}  
.window\_active {  
 position: fixed;  
 transition: all 0.3s ease 0s;  
 top: 0;  
 right: 0;  
 bottom: 0;  
 left: 0;  
 background: rgba(0, 0, 0, 0.9);  
 display: flex;  
 justify-content: center;  
 align-items: flex-start;  
 padding: 5% 10% 5% 10%;  
}  
  
.content {  
 width: 100%;  
 max-width: 500px;  
 padding: 50px;  
 background: #a0a0a8;  
 border-radius: 7px;  
 position: relative;  
 box-shadow: 0 5px 15px black;  
 margin: 0.5% 0 0.5% 0;  
}  
  
.photo {  
 align-items: center;  
 display: flex;  
 background: #f0f0f0;  
 border-radius: 5px;  
 padding: 10px;  
}  
  
.icon\_close {  
 height: 50px;  
 width: 50px;  
 display: flex;  
 justify-content: center;  
 align-items: center;  
 position: absolute;  
 right: 5px;  
 top: 5px;  
 cursor: pointer;  
 transition: all 0.3s ease 0s;  
}

Html для модального окна:

<div class="window">  
 <div class="content">  
 <button class="icon\_close"><img src="images/gallery/iclose.png" width="30" alt=""></button>  
 <div class="main-gallery">  
 <div class="buttonGallery" onclick="leftClick()">  
 <button><img class="but" src="images/gallery/buttontoleft.png"></button>  
 </div>  
 <div class="photo">  
 <img id="mainimg">  
 </div>  
 <div class="buttonGallery" onclick="rightClick()">  
 <button><img class="but" src="images/gallery/buttontoright.png"></button>  
 </div>  
 </div>  
 </div>  
</div>