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 № 5

«Drag and drop для галереи»

Выполнил студент

группы ИВТАПбд-12

Ступин А.О.

Проверил: Игонин А. Г.

Ульяновск, 2022

Оглавление

[Цель 3](#_Toc104349550)

[Описание метода выполнения 3](#_Toc104349551)

[Вывод 6](#_Toc104349552)

[Источники 7](#_Toc104349553)

[Код 7](#_Toc104349554)

## Цель

Целью работы являлась модификация персональной галереи, разработанной в Лабораторной работе №3. Необходимо было разбить фотографии на несколько блоков, чтобы их можно перелистывать удерживанием курсора/пальца, как свайпер. Для реализации можно было пользоваться PHP и JS.

## Описание метода выполнения

Лабораторная работа написана на языке Javascript.

Чтобы было, что листать, содержимое галереи было увеличено с 20 до 36 фотографий. Таким образом, она была разбита на 3 блока по 12 фотографий в каждом. Именно эти блоки и требуется перелистывать.

Далее – создаем файлы labfive.js и index.js, подключаем к gallery.php. Класс photos теперь не один – их три. Собственно они и будут листаться. Над ними создаем родительский класс gallery. Его мы делаем флекс-элементом.

Основная работа проводится в labfive.js. Там создаем класс Gallery и объявляем constructor. Элемент, передаваемый в него, сохраняем в containerNode, перед которым поставим ключевое слово this (которое еще много будем использовать).

В файле index.js создаем экземпляр галереи:

|  |
| --- |
| new Gallery(document.getElementById('gallery')); |

HTML прописываем также в JS файле, через manageHTML.

|  |
| --- |
| manageHTML() {  this.containerNode.classList.add(***GalleryClassName***);  this.containerNode.innerHTML = `  <div class="${***GalleryLineClassName***}">  ${this.containerNode.innerHTML}  </div>  <div class = "${***GalleryDotsClassName***}"></div>  `;  this.lineNode = this.containerNode.querySelector(`.${***GalleryLineClassName***}`);  this.dotsNode = this.containerNode.querySelector(`.${***GalleryDotsClassName***}`);   this.slideNodes = ***Array***.from(this.lineNode.children).map((childNode) =>  wrapElementByDiv({  element: childNode,  ***className***: ***GallerySlideClassName*** })  );  this.dotsNode.innerHTML = ***Array***.from(***Array***(this.size).keys()).map((key) => (  `<button class="${***GalleryDotClassName***} ${key === this.currentSlide ? ***GalleryDotActiveClassName*** : ' '}"></button>`  )).join('');   this.dotNodes = this.dotsNode.querySelectorAll(`${***GalleryDotClassName***}`);   } |

Перед этим в начале файла задав константы для имен классов. Наши “photos” выстраиваются в линию, которая имеет некоторую статичную обертку. Именно поэтому нам нужны дивы, которые мы добавляем окончательно с помощью функции:

|  |
| --- |
| function wrapElementByDiv({element, className}){  const wrapperNode = ***document***.createElement('div');  wrapperNode.classList.add(className);   element.parentNode.insertBefore(wrapperNode, element);  wrapperNode.appendChild(element);  return wrapperNode; } |

Создадим события для изменения окна браузера. Но чтобы пересчетов не было слишком много (особенно если пользователь меняет окно медленно), используем новую функцию, которая задает определенный период для пересчета размера окна (timeout):

|  |
| --- |
| function debounce(func, time = 100) {  let timer;  return function (event) {  clearTimeout(timer);  timer = setTimeout(func, time, event);  } } |

Далее делаем собственно движение. Для этого добавляем события pointerup, pointerdown и pointermove. Это события для отпускания, нажатия и перемещения мыши соответственно, передаем функции stopDrag, startDrag и dragging. Также стоит заметить, что перемещение у нас происходит только по горизонтали, поэтому вычисления перемещений производим только по оси Х.

Также добавим функцию для перехода на следующий слайд:

|  |
| --- |
| if (dragShift > 50 && this.currentSlide > 0 && !this.currentSlideWasChanged) {  this.currentSlideWasChanged = true;  this.currentSlide = this.currentSlide - 1; }  if (dragShift < -50 && !this.currentSlideWasChanged && this.currentSlide < this.size - 1) {  this.currentSlideWasChanged = true;  this.currentSlide = this.currentSlide + 1; } |

Теперь слайды хорошо перелистываются. Более того, благодаря проделанному в третьей лабораторной работе, фотографии еще и открываются. Добавим точечки внизу, чтобы мы видели, на каком из трех слайдов находимся, и могли также переходить по ним с помощью точек. Задав константы для точек вверху страницы, объявим функции, отвечающие за смену цвета точки, соответствующей номеру слайда:

|  |
| --- |
| changeActiveDotClass() {  for (let i = 0; i < this.dotNodes.length; i++) {  this.dotNodes[i].classList.remove(***GalleryDotActiveClassName***);  }  this.dotNodes[this.currentSlide].classList.add(***GalleryDotActiveClassName***); // }  changeCurrentSlide() {  this.x = - this.currentSlide \* (this.width + this.settings.margin);  this.setStyleTransition();  this.setStylePosition();  this.changeActiveDotClass(); // } |

А также функцию клика по точке, с переданным в нее событием:

|  |
| --- |
| clickDots(evt) {  const dotNode = evt.target.closest('button') //closest ищет ближ. родителя по селектору button  if (!dotNode) {  return;  }   let dotNumber;  for(let i = 0; i < 2; i++) {  if(this.dotNodes [i] === dotNode) {  dotNumber = i;  break;  }   }  if (dotNumber === this.currentSlide) {  return;    }  this.currentSlide = dotNumber;  this.changeCurrentSlide(); // } |

Теперь страницы галереи действительно передвигаются мышкой:

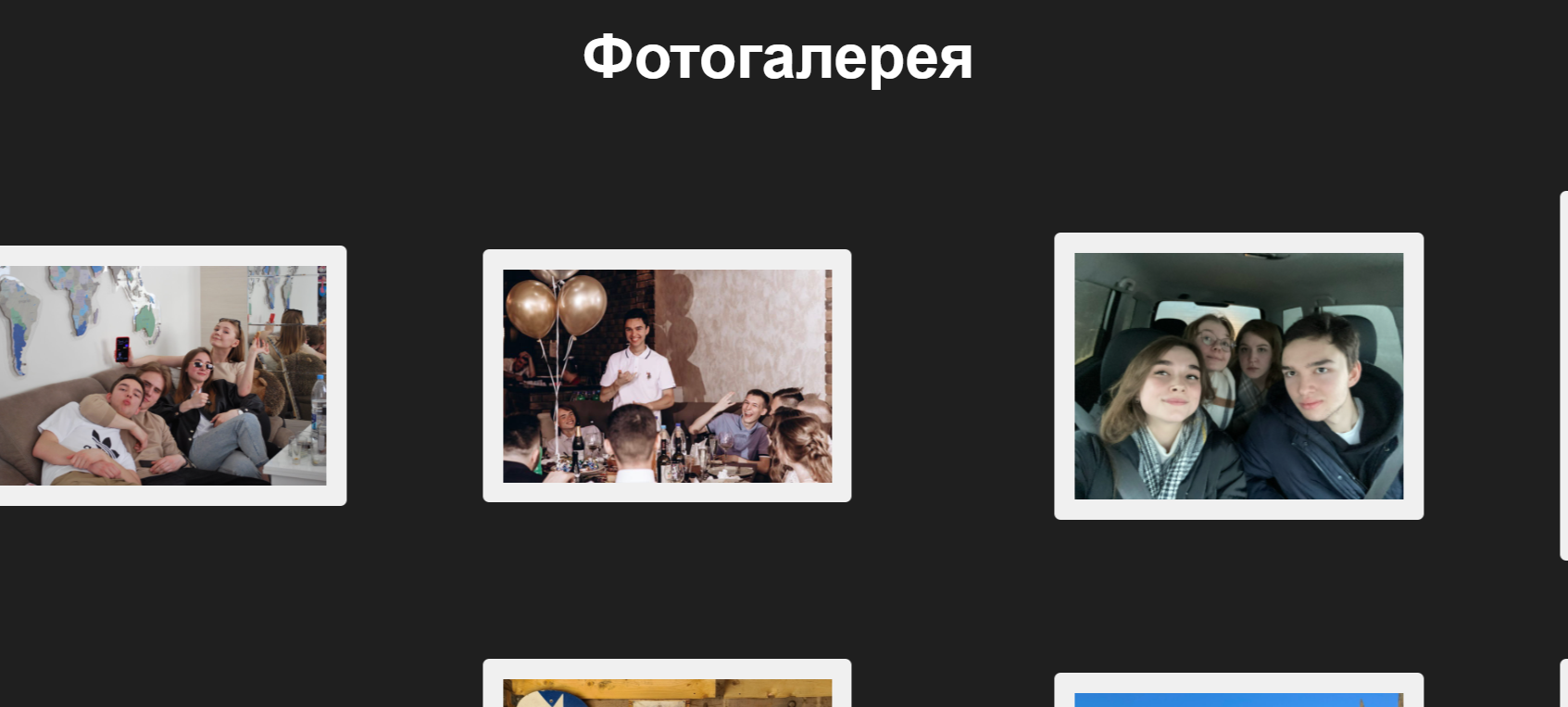


Рис. 1 – смена страниц

Пронаблюдаем это с телефона.

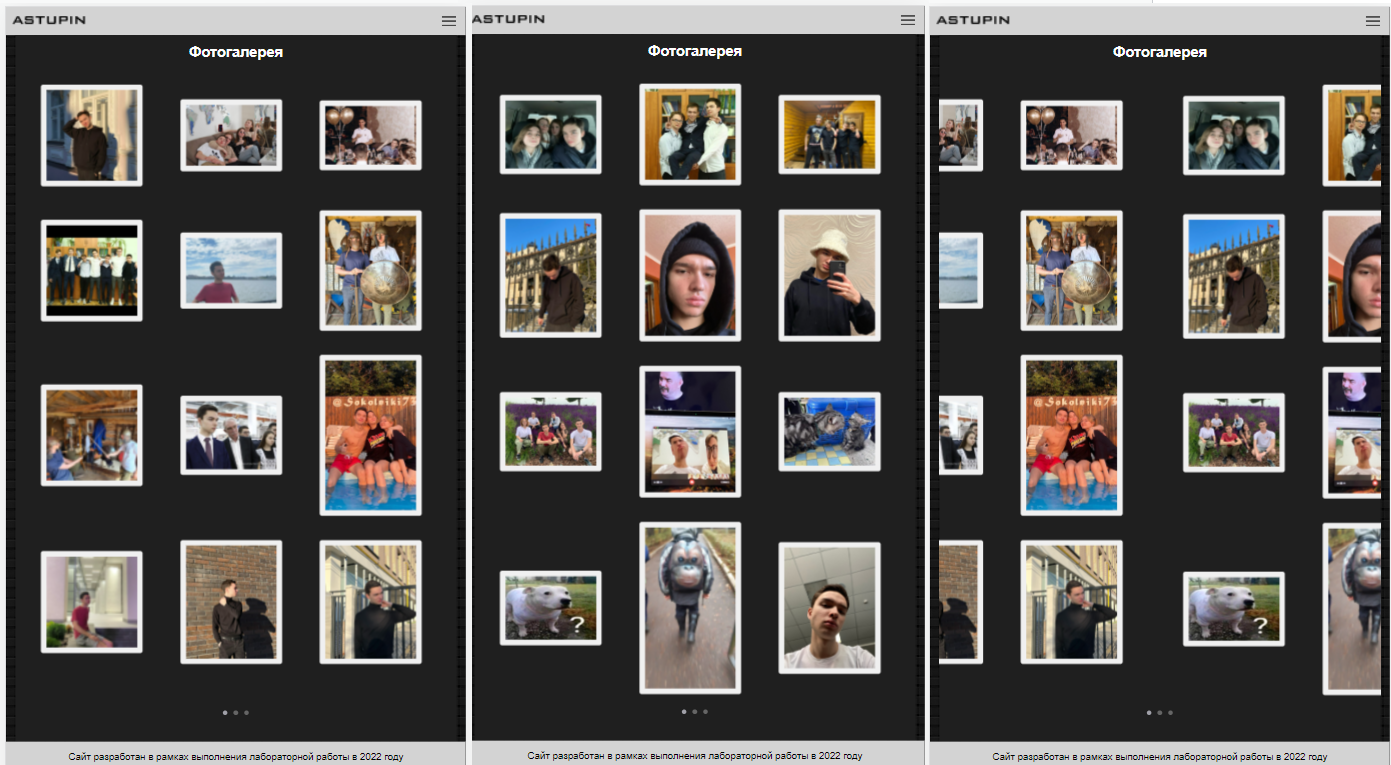


Рис. 2 – смена страниц на телефоне

## Вывод

Нам удалось модифицировать галерею, добавив в нее возможность пролистывания удержанием курсора или пальца. Такая галерея особенно удобна для пользователей мобильных устройств.

## Источники

* Ключевое слово this в JavaScript для начинающих [электронный ресурс] URL: <https://habr.com/ru/company/ruvds/blog/419371/>
* Создаём drag & touch галерею на чистом JS [электронный ресурс] URL: <https://www.youtube.com/watch?v=MxzenEvt7X>
* Drag'n'Drop с событиями мыши [электронный ресурс] URL: <https://learn.javascript.ru/mouse-drag-and-drop>
* Объект события [электронный ресурс] URL: <https://learn.javascript.ru/obtaining-event-object>

## Код

Labfive.js

const ***GalleryClassName*** = 'gallery';  
const ***GalleryDraggableClassName*** = 'gallery-draggable';  
const ***GalleryLineClassName*** = 'photos-items';  
const ***GallerySlideClassName*** = 'photosline';  
const ***GalleryDotsClassName*** = 'gallerydots';  
const ***GalleryDotClassName*** = 'gallerydot';  
const ***GalleryDotActiveClassName*** = 'gallerydotactive';  
  
class Gallery {  
 constructor(element, options = {}) {  
 this.containerNode = element;  
 this.size = element.childElementCount; //слайды  
 this.currentSlide = 0;  
 this.currentSlideWasChanged = false;  
 this.settings = {  
 margin: options.margin || 0  
 }  
  
 this.manageHTML = this.manageHTML.bind(this);  
 this.setParameters = this.setParameters.bind(this);  
 this.setEvents = this.setEvents.bind(this);  
 this.resizeGallery = this.resizeGallery.bind(this);  
 this.startDrag = this.startDrag.bind(this);  
 this.stopDrag = this.stopDrag.bind(this);  
 this.dragging = this.dragging.bind(this);  
 this.setStylePosition = this.setStylePosition.bind(this);  
 this.clickDots = this.clickDots.bind(this);  
 this.changeActiveDotClass = this.changeActiveDotClass.bind(this);  
 this.changeCurrentSlide = this.changeCurrentSlide.bind(this);  
  
 this.manageHTML();  
 this.setParameters();  
 this.setEvents();  
 }  
  
 manageHTML() {  
 this.containerNode.classList.add(***GalleryClassName***);  
 this.containerNode.innerHTML = `  
 <div class="${***GalleryLineClassName***}">  
 ${this.containerNode.innerHTML}  
 </div>  
 <div class = "${***GalleryDotsClassName***}"></div>  
 `;  
 this.lineNode = this.containerNode.querySelector(`.${***GalleryLineClassName***}`);  
 this.dotsNode = this.containerNode.querySelector(`.${***GalleryDotsClassName***}`);  
  
 this.slideNodes = ***Array***.from(this.lineNode.children).map((childNode) =>  
 wrapElementByDiv({  
 element: childNode,  
 ***className***: ***GallerySlideClassName*** })  
 );  
 this.dotsNode.innerHTML = ***Array***.from(***Array***(this.size).keys()).map((key) => (  
 `<button class="${***GalleryDotClassName***} ${key === this.currentSlide ? ***GalleryDotActiveClassName*** : ' '}"></button>`  
 )).join('');  
  
 this.dotNodes = this.dotsNode.querySelectorAll(`${***GalleryDotClassName***}`); //задаем события, вытаскиваем  
  
 }  
  
 setParameters() {  
 const coordsContainer = this.containerNode.getBoundingClientRect();  
 this.width = coordsContainer.width;  
 this.maximumX = - (this.size - 1) \* (this.width + this.settings.margin);  
 this.x = -this.currentSlide \* (this.width + this.settings.margin);  
 this.resetStyleTransition();  
 this.lineNode.style.width = `${this.size \* (this.width + this.settings.margin)}px`;  
 this.setStylePosition();  
 ***Array***.from(this.slideNodes).forEach((slideNode) => {  
 slideNode.style.width = `${this.width}px`;  
 slideNode.style.marginRight = `${this.settings.margin}px`;  
 })  
  
 }  
  
 setEvents() {  
 this.debouncedResizeGallery = debounce(this.resizeGallery);  
 ***window***.addEventListener('resize', this.debouncedResizeGallery);  
 this.lineNode.addEventListener('pointerdown', this.startDrag);  
 ***window***.addEventListener('pointerup', this.stopDrag);  
 ***window***.addEventListener('pointercancel', this.stopDrag);  
  
 this.dotsNode.addEventListener('click', this.clickDots);  
 }  
  
 destroyEvents() {  
 ***window***.removeEventListener('resize', this.debouncedResizeGallery);  
 this.lineNode.removeEventListener('pointerdown', this.startDrag);  
 ***window***.removeEventListener('pointerup', this.stopDrag);  
 ***window***.removeEventListener('pointercancel', this.stopDrag);  
 this.dotsNode.removeEventListener('click', this.clickDots);  
 }  
  
 resizeGallery() {  
 this.setParameters();  
 }  
  
 startDrag(evt) {  
 this.currentSlideWasChanged = false;  
 this.clickX = evt.pageX;  
 this.startX = this.x;  
 this.resetStyleTransition();  
 this.containerNode.classList.add(***GalleryDraggableClassName***);  
 ***window***.addEventListener('pointermove', this.dragging);  
 }  
  
 stopDrag() {  
 ***window***.removeEventListener('pointermove', this.dragging);  
 this.x = - this.currentSlide \* (this.width + this.settings.margin);  
 this.setStylePosition();  
 this.setStyleTransition();  
  
 this.containerNode.classList.remove(***GalleryDraggableClassName***);  
 }  
  
 dragging(evt) {  
 this.dragX = evt.pageX;  
 const dragShift = this.dragX - this.clickX;  
 const easing = dragShift / 5;  
 this.setStylePosition();  
 this.x = ***Math***.max(***Math***.min(this.startX + dragShift, easing), this.maximumX + easing);  
  
 //смена активного слайда  
 if (dragShift > 50 && this.currentSlide > 0 && !this.currentSlideWasChanged) {  
 this.currentSlideWasChanged = true;  
 this.currentSlide = this.currentSlide - 1;  
 }  
  
 if (dragShift < -50 && !this.currentSlideWasChanged && this.currentSlide < this.size - 1) {  
 this.currentSlideWasChanged = true;  
 this.currentSlide = this.currentSlide + 1;  
 }  
  
 }  
  
 changeActiveDotClass() {  
 for (let i = 0; i < this.dotNodes.length; i++) {  
 this.dotNodes[i].classList.remove(***GalleryDotActiveClassName***);  
 }  
 this.dotNodes[this.currentSlide].classList.add(***GalleryDotActiveClassName***); //  
 }  
  
 changeCurrentSlide() {  
 this.x = - this.currentSlide \* (this.width + this.settings.margin);  
 this.setStyleTransition();  
 this.setStylePosition();  
 this.changeActiveDotClass(); //  
 }  
  
 clickDots(evt) {  
 const dotNode = evt.target.closest('button') //closest ищет ближ. родителя по селектору button  
 if (!dotNode) {  
 return;  
 }  
  
 let dotNumber;  
 for(let i = 0; i < 2; i++) {  
 if(this.dotNodes [i] === dotNode) {  
 dotNumber = i;  
 break;  
 }  
  
 }  
 if (dotNumber === this.currentSlide) {  
 return;  
  
  
 }  
 this.currentSlide = dotNumber;  
 this.changeCurrentSlide(); //  
 }  
  
  
 setStylePosition() {  
 this.lineNode.style.transform = `translate3d(${this.x}px, 0, 0)`  
  
 }  
  
 setStyleTransition() {  
 this.lineNode.style.transition = `all 0.25s ease 0s`;  
 }  
  
 resetStyleTransition() {  
 this.lineNode.style.transition = `all 0s ease 0s`;  
 }  
  
}  
 function wrapElementByDiv({element, className}){  
 const wrapperNode = ***document***.createElement('div');  
 wrapperNode.classList.add(className);  
  
 element.parentNode.insertBefore(wrapperNode, element);  
 wrapperNode.appendChild(element);  
 return wrapperNode;  
 }  
  
 function debounce(func, time = 100) {  
 let timer;  
 return function (event) {  
 clearTimeout(timer);  
 timer = setTimeout(func, time, event);  
 }  
 }

Gallery.php

<?php include\_once "template/header.php" ?>  
<!DOCTYPE html>  
<html lang="en">  
<head>  
 <meta charset="UTF-8">  
 <title>Галерея</title>  
 <link rel="icon" type="image/x-icon" href="images/favicon.png">  
 <link rel="stylesheet" href="/styles/styles.css">  
 <script type="text/javascript" src="/scripts/jquery-3.6.0.js"></script>  
 <script type="text/javascript" src="/scripts/gallery.js"></script>  
 <script type="text/javascript" src="/scripts/labfive.js"></script>  
</head>  
<body>  
<div class = "main\_content">  
 <h2 align="center">Фотогалерея</h2>  
 <div id="gallery">  
 <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>  
 <ul>  
 <li><img src = "images/gallery/gal4.jpg" onclick="Show(4)" title = "Лицей, группа 1-19" alt="Фото не прогрузилось"></li>  
 <li><img src = "images/gallery/gal5.jpg" onclick="Show(5)" title = "В Казани" alt="Фото не прогрузилось"></li>  
 <li><img src = "images/gallery/gal6.jpg" onclick="Show(6)" title = "В Свияжске" alt="Фото не прогрузилось"></li>  
 </ul>  
 <ul>  
 <li><img src = "images/gallery/gal7.jpg" onclick="Show(7)" title = "В Свияжске" alt="Фото не прогрузилось"></li>  
 <li><img src = "images/gallery/gal8.jpg" onclick="Show(8)" title = "На фабрике Boston" alt="Фото не прогрузилось"></li>  
 <li><img src = "images/gallery/gal9.jpg" onclick="Show(9)" title = "На дне рождения" alt="Фото не прогрузилось"></li>  
 </ul>  
 <ul>  
 <li><img src = "images/gallery/gal10.jpg" onclick="Show(10)" title = "Гуляю" alt="Фото не прогрузилось"></li>  
 <li><img src = "images/gallery/gal11.jpg" onclick="Show(11)" title = "Гуляю" alt="Фото не прогрузилось"></li>  
 <li><img src = "images/gallery/gal12.jpg" onclick="Show(12)" title = "Гуляю" alt="Фото не прогрузилось"></li>  
 </ul>  
 </div>  
 <div class = "photos">  
 <ul>  
 <li><img src = "images/gallery/gal13.jpg" onclick="Show(13)" title = "Гуляю" alt="Фото не прогрузилось"></li>  
 <li><img src = "images/gallery/gal14.jpg" onclick="Show(14)" title = "Илья, Андрей и Я" alt="Фото не прогрузилось"></li>  
 <li><img src = "images/gallery/gal15.jpg" onclick="Show(15)" title = "Новый год 2022" alt="Фото не прогрузилось"></li>  
 </ul>  
 <ul>  
 <li><img src = "images/gallery/gal16.jpg" onclick="Show(16)" title = "Гуляю" alt="Фото не прогрузилось"></li>  
 <li><img src = "images/gallery/gal17.jpg" onclick="Show(17)" title = "Я" alt="Фото не прогрузилось"></li>  
 <li><img src = "images/gallery/gal18.jpg" onclick="Show(18)" title = "Я" alt="Фото не прогрузилось"></li>  
 </ul>  
 <ul>  
 <li><img src = "images/gallery/gal19.jpg" onclick="Show(19)" title = "В Казани" alt="Фото не прогрузилось"></li>  
 <li><img src = "images/gallery/gal20.jpg" onclick="Show(20)" title = "В гостях" alt="Фото не прогрузилось"></li>  
 <li><img src = "images/gallery/gal21.jpg" onclick="Show(21)" title = "" alt=""></li>  
 </ul>  
 <ul>  
 <li><img src = "images/gallery/gal22.jpg" onclick="Show(22)" title = "" alt=""></li>  
 <li><img src = "images/gallery/gal23.jpg" onclick="Show(23)" title = "" alt=""></li>  
 <li><img src = "images/gallery/gal24.jpg" onclick="Show(24)" title = "" alt=""></li>  
 </ul>  
 </div>  
 <div class="photos">  
 <ul>  
 <li><img src = "images/gallery/gal25.jpg" onclick="Show(25)" title = "" alt=""></li>  
 <li><img src = "images/gallery/gal26.jpg" onclick="Show(26)" title = "" alt=""></li>  
 <li><img src = "images/gallery/gal27.jpg" onclick="Show(27)" title = "" alt=""></li>  
 </ul>  
 <ul>  
 <li><img src = "images/gallery/gal28.jpg" onclick="Show(28)" title = "" alt=""></li>  
 <li><img src = "images/gallery/gal29.jpg" onclick="Show(29)" title = "" alt=""></li>  
 <li><img src = "images/gallery/gal30.jpg" onclick="Show(30)" title = "" alt=""></li>  
 </ul>  
 <ul>  
 <li><img src = "images/gallery/gal31.jpg" onclick="Show(31)" title = "" alt=""></li>  
 <li><img src = "images/gallery/gal32.jpg" onclick="Show(32)" title = "" alt=""></li>  
 <li><img src = "images/gallery/gal33.jpg" onclick="Show(33)" title = "" alt=""></li>  
 </ul>  
 <ul>  
 <li><img src = "images/gallery/gal34.jpg" onclick="Show(34)" title = "" alt=""></li>  
 <li><img src = "images/gallery/gal35.jpg" onclick="Show(35)" title = "" alt=""></li>  
 <li><img src = "images/gallery/gal36.jpg" onclick="Show(36)" title = "" alt=""></li>  
 </ul>  
 </div>  
 </div>  
 <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>  
</div>  
<script type="text/javascript" src="/scripts/index.js"></script>  
</body>  
  
<?php include\_once "template/footer.php" ?>

CSS:

.gallery {  
 width: 100%;  
 overflow: hidden;  
 user-select: none;  
 align-items: center;  
 justify-content: center;  
}  
  
.photosline {  
 float: left;  
 min-height: 1px;  
 touch-action: none;  
}  
  
.gallery-draggable {  
 cursor: grabbing;  
}  
  
  
.photos\_items {  
 display: flex;  
 justify-content: center;  
}  
  
.photos {  
 justify-content: center;  
 display: flex;  
 flex-direction: column;  
 width: 100%;  
 position: relative;  
 float: left;  
 transform: translateZ(0);  
  
}  
  
.gallerydots {  
 display: flex;  
 justify-content: center;  
 width: 300px;  
 margin: auto;  
 margin-bottom: 50px;  
}  
  
.gallerydot {  
 width: 10px;  
 height: 10px;  
 border-radius: 50%;  
 background-color: #656565;  
 margin: 0.4rem;  
 cursor: pointer;  
 transition: all 0.3s;  
 border: none;  
 outline: none;  
 padding: 0;  
}  
  
.gallerydot:hover {  
 background-color: #a0a0a8;  
}  
  
.gallerydotactive {  
 background-color: #a0a0a8;  
}