 МІНІСТЕРСТВО ОСВІТИ I НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ   
ІМЕНІ ІГОРЯ СІКОРСЬКОГО»

ФАКУЛЬТЕТ БІОМЕДИЧНОЇ ІНЖЕНЕРІЇ

КАФЕДРА БІОМЕДИЧНОЇ КІБЕРНЕТИКИ

**Комп’ютерний практикум № 6**

з дисципліни «Веб-технології та веб-дизайн»

**Виконав:**

студент гр. БС-82

Грініх К.А.

**Перевірив:**

ас. каф. БМК Давидько О. Б.

Зараховано від \_\_\_.\_\_\_.\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(підпис викладача)

Київ-2020

**Завдання:**

1) Створити користувацький HTML елемент - галерею зображень. Галерея має бути організована у вигляді таблиці, всі зображення мають бути приведені до однакового розміру.   
2) Список картинок має передаватись через атрибути елементу, наприклад: imageList=”resources/image1;resources/image2;resources/image3”  
3) При натисненні на зображення, воно має відкритись на весь екран. Перехід до збільшеного режиму має бути з анімацією.  
4) Галерея має бути адаптивною, тобто при зміні розміру вікна браузера має змінюватись кількість колонок, при мінімальній ширині екрану має залишитись лише одна колонка.  
5) При відсутності або неможливості завантажити зображення, замість галереї має бути виведене зображення placeholder.  
6) Зробити preloader з анімацією для галереї.  
7) Вивести галерею на окрему сторінку існуючого сайту

**Лістинг:**

index.html

Начало формы

Конец формы

|  |
| --- |
| <!DOCTYPE html> |
|  | <html> |
|  |  |
|  | <head> |
|  | <meta charset="utf-8"> |
|  | <title>Gallery</title> |
|  | <link rel="stylesheet" href="style.css"> |
|  | </head> |
|  |  |
|  | <body> |
|  | <my-gallery preloader="resources/preloader.gif" imageList=" |
|  | resources/1.jpg; resources/2.jpg; resources/3.jpg; |
|  | resources/4.jpg; resources/5.jpg; resources/6.jpg; |
|  | resources/7.jpg; resources/8.jpg"> |
|  | </my-gallery> |
|  | <script src="script.js"></script> |
|  | </body> |
|  |  |
|  | </html> |

Начало формы

Конец формы

## **style.css**

|  |
| --- |
| \* { margin: 0; padding: 0; } |
|  |  |
|  | img { |
|  | object-fit: contain; |
|  | } |
|  |  |
|  | .gallery-container { |
|  | display: grid; |
|  | gap: 1rem; |
|  | max-width: 1200px; |
|  | margin: 0 auto; |
|  | } |
|  |  |
|  | @media (min-width: 600px) { |
|  | .gallery-container { |
|  | grid-template-columns: repeat(2, 1fr); |
|  | } |
|  | } |
|  |  |
|  | @media (min-width: 900px) { |
|  | .gallery-container { |
|  | grid-template-columns: repeat(3, 1fr); |
|  | } |
|  | } |
|  |  |
|  | .image-container { |
|  | width : 250px; |
|  | height: 250px; |
|  | margin: 0 auto; |
|  | } |
|  |  |
|  | .image-container img { |
|  | position: absolute; |
|  | width : inherit; |
|  | height: inherit; |
|  | } |
|  |  |
|  | .preloader, .placeholder { |
|  | position: absolute; |
|  | width : 400px; |
|  | height: auto; |
|  | top : 50%; |
|  | left : 50%; |
|  | transform: translate(-50%, -50%); |
|  | } |

## **script.js**

|  |
| --- |
| class MyGallery extends HTMLElement { |
|  | constructor() { |
|  | super(); |
|  |  |
|  | const imageList = this.hasAttribute('imageList') ? |
|  | this.getAttribute('imageList').split(';') : |
|  | []; |
|  |  |
|  | this.totalImages = imageList.length; |
|  | this.imagesLoaded = 0; |
|  |  |
|  | this.galleryContainer = document.createElement('div'); |
|  | this.galleryContainer.className = 'gallery-container'; |
|  |  |
|  | for (const resource of imageList) { |
|  | const imageContainer = document.createElement('div'); |
|  | imageContainer.className = 'image-container'; |
|  |  |
|  | const img = document.createElement('img'); |
|  | img.src = resource; |
|  | img.onerror = this.onerror.bind(this); |
|  | img.onload = this.onload.bind(this); |
|  | img.onclick = fullscreen; |
|  |  |
|  | imageContainer.append(img); |
|  | this.galleryContainer.append(imageContainer); |
|  | } |
|  |  |
|  | this.preloader = document.createElement('img'); |
|  | this.preloader.className = 'preloader'; |
|  | this.preloader.src = this.getAttribute('preloader'); |
|  | this.append(this.preloader); |
|  | } |
|  |  |
|  | onerror() { |
|  | if (this.error) { |
|  | return; |
|  | } |
|  |  |
|  | this.error = true; |
|  |  |
|  | setTimeout(() => { |
|  | const placeholder = document.createElement('img'); |
|  | placeholder.className = 'placeholder'; |
|  | placeholder.src = 'https://via.placeholder.com/600x400?text=Gallery+Here'; |
|  | placeholder.onload = (() => { |
|  | this.preloader.remove(); |
|  | this.append(placeholder); |
|  | }); |
|  | }, 1500); |
|  | } |
|  |  |
|  | onload() { |
|  | if (++this.imagesLoaded == this.totalImages) { |
|  | setTimeout(() => { |
|  | this.preloader.remove(); |
|  | this.append(this.galleryContainer); |
|  | }, 1500); |
|  | } |
|  | } |
|  | } |
|  |  |
|  | customElements.define('my-gallery', MyGallery); |
|  |  |
|  | function fullscreen({ target: el }) { |
|  | if (el.animation) { |
|  | el.animation.cancel(); |
|  | el.animation = null; |
|  | el.style['z-index'] = 0; |
|  | return; |
|  | } |
|  |  |
|  | const { width, height, top, left } = el.getBoundingClientRect(); |
|  | const keyframes = [ |
|  | { top: `${window.scrollY + top}px`, left: `${left}px`, width: `${width}px`, height: `${height}px` }, |
|  | { top: `${window.scrollY}px`, left: 0, width: '100%', height: '100%' } |
|  | ]; |
|  |  |
|  | el.style['z-index'] = 1; |
|  | el.animation = el.animate(keyframes, { fill: 'forwards', duration: 200, easing: 'ease' }); |
|  | } |

**Результат виконання програми:**



