

Moduł_6\task-js-events\assets\js\script.js

```
1  const init = function () {
2      const imagesList = document.querySelectorAll('.gallery__item')
3      imagesList.forEach(img => {
4          img.dataset.sliderGroupName = Math.random() > 0.5 ? 'nice' : 'good'
5      }) // za każdym przeładowaniem strony przydzielaj inną nazwę grupy dla zdjęcia
6
7      runJSSlider()
8  }
9
10 document.addEventListener('DOMContentLoaded', init)
11
12 const runJSSlider = function () {
13     const imagesSelector = '.gallery__item'
14     const sliderRootSelector = '.js-slider'
15
16     const imagesList = document.querySelectorAll(imagesSelector)
17     const sliderRootElement = document.querySelector(sliderRootSelector)
18
19     initEvents(imagesList, sliderRootElement)
20     initCustomEvents(imagesList, sliderRootElement, imagesSelector)
21 }
22
23 const initEvents = function (imagesList, sliderRootElement) {
24     imagesList.forEach(function (item) {
25         item.addEventListener('click', function (e) {
26             fireCustomEvent(e.currentTarget, 'js-slider-img-click')
27         })
28     })
29
30     // todo:
31     // utwórz nasłuchiwanie eventu o nazwie [click], który ma uruchomić event [js-slider-
img-next]
32     // na elemencie [.js-slider__nav--next]
33     const navNext = sliderRootElement.querySelector('.js-slider__nav--next')
34     navNext.addEventListener('click', e => {
35         fireCustomEvent(navNext, 'js-slider-img-next')
36     })
37     // todo:
38     // utwórz nasłuchiwanie eventu o nazwie [click], który ma uruchomić event [js-slider-
img-prev]
39     // na elemencie [.js-slider__nav--prev]
40     const navPrev = sliderRootElement.querySelector('.js-slider__nav--prev')
41     navPrev.addEventListener('click', e => {
42         fireCustomEvent(navPrev, 'js-slider-img-prev')
43     })
44
45     // todo:
46     // utwórz nasłuchiwanie eventu o nazwie [click], który ma uruchomić event [js-slider-
close]
47     // tylko wtedy, gdy użytkownik kliknie w [.js-slider__zoom]
48     const zoom = sliderRootElement.querySelector('.js-slider__zoom')
49     zoom.addEventListener('click', e => {
50         e.stopPropagation()
```

```
51         if (e.target === zoom) {
52             fireCustomEvent(zoom, 'js-slider-close')
53         }
54     })
55 }
56
57 const fireCustomEvent = function (element, name) {
58     console.log(element.className, '=>', name)
59
60     const event = new CustomEvent(name, {
61         bubbles: true,
62     })
63
64     element.dispatchEvent(event)
65 }
66
67 const initCustomEvents = function (imagesList, sliderRootElement, imagesSelector) {
68     imagesList.forEach(function (img) {
69         img.addEventListener('js-slider-img-click', function (event) {
70             onImageClick(event, sliderRootElement, imagesSelector)
71         })
72     })
73
74     sliderRootElement.addEventListener('js-slider-img-next', onImageNext)
75     sliderRootElement.addEventListener('js-slider-img-prev', onImagePrev)
76     sliderRootElement.addEventListener('js-slider-close', onClose)
77 }
78
79 const onImageClick = function (event, sliderRootElement, imagesSelector) {
80     // todo:
81     // 1. dodać klasę [.js-slider--active], aby pokazać całą sekcję
82     sliderRootElement.classList.add('js-slider--active')
83
84     // 2. wyszukać ścieżkę (atrybut [src]) do klikniętego elementu i wstawić do [.js-
85     slider__image]
86     const clickedImage = event.target.querySelector('img')
87     const sliderImage = document.querySelector('.js-slider__image')
88     sliderImage.src = clickedImage.src
89
90     // 3. pobrać nazwę grupy zapisaną w dataset klikniętego elementu
91     const groupName = event.target.dataset.sliderGroupName
92
93     // 4. wyszukać wszystkie zdjęcia należące do danej grupy, które wykorzystasz do
94     osadzenia w dolnym pasku
95     const imagesList = []
96     const allImages = document.querySelectorAll(imagesSelector)
97
98     allImages.forEach(image => {
99         if (image.dataset.sliderGroupName === groupName) {
100             imagesList.push(image)
101             // console.log(image.firstElementChild.src)
102         }
103     })
104
105     // 5. utworzyć na podstawie elementu [.js-slider__thumbs-item--prototype] zawartość
106     dla [.js-slider__thumbs]
```

```
104
105     const thumbsContainer = document.querySelector('.js-slider__thumbs')
106
107     thumbsContainer.innerHTML = ''
108
109     imagesList.forEach(img => {
110         const thumb = document.createElement('figure')
111         thumb.classList.add('js-slider__thumbs-item')
112         img.classList = ''
113         img.classList.add('js-slider__thumbs-image')
114         img.setAttribute('src', img.firstElementChild.getAttribute('src'))
115         thumb.append(img)
116         thumbsContainer.append(thumb)
117     })
118
119     // 6. zaznaczyć przy pomocy klasy [.js-slider__thumbs-image--current], który element
    jest aktualnie wyświetlany)
120     const thumbImages = document.querySelectorAll('.js-slider__thumbs-image')
121
122     const currentThumb = Array.from(thumbImages).find(item => {
123         return item.firstElementChild.src === clickedImage.src
124     })
125
126     currentThumb.classList.add('js-slider__thumbs-image--current')
127     console.log(currentThumb)
128     // thumbImages.forEach(img => {
129     //     if (img.firstElementChild.src === clickedImage.src) {
130     //         img.classList.add('js-slider__thumbs-image--current')
131     //         console.log(img)
132     //     }
133     // })
134
135     autoChange(onImageNext)
136 }
137
138 const autoChange = function (changeElement) {
139     setInterval(changeElement, 2000)
140 }
141
142 const onImageNext = function (event) {
143     // console.log(this, 'onImageNext')
144     // [this] wskazuje na element [.js-slider]
145     // todo:
146     // 1. wyszukać aktualny wyświetlany element przy pomocy [.js-slider__thumbs-image--
current]
147     // // 2. znaleźć element następny do wyświetlenia względem drzewa DOM dla [.js-
slider__thumbs]
148
149     const currentImg = document.querySelector('.js-slider__thumbs-image--current')
150     const thumbImages = document.querySelectorAll('.js-slider__thumbs-image')
151     const currentImgIndex = Array.from(thumbImages).indexOf(currentImg)
152     let nextImg = thumbImages[currentImgIndex + 1]
153
154     // 3. sprawdzić czy ten element istnieje - jeśli nie to [.nextElementSibling] zwróci
[null]
155     // 4. przełączyć klasę [.js-slider__thumbs-image--current] do odpowiedniego elementu
```

```
156 // 5. podmienić atrybut o nazwie [src] dla [.js-slider__image]
157 if (nextImg === undefined) {
158     nextImg = thumbImages[0]
159 }
160
161 if (nextImg !== null) {
162     currentImg.classList.remove('js-slider__thumbs-image--current')
163     nextImg.classList.add('js-slider__thumbs-image--current')
164     const sliderImage = document.querySelector('.js-slider__image')
165     const currentImgSrc = document.querySelector('.js-slider__thumbs-image--current')
166     .getAttribute('src')
167     sliderImage.src = currentImgSrc
168 }
169
170 const onImagePrev = function (event) {
171     console.log(this, 'onImagePrev')
172     // [this] wskazuje na element [.js-slider]
173
174     // todo:
175     // 1. wyszukać aktualny wyświetlany element przy pomocy [.js-slider__thumbs-image--current]
176     // 2. znaleźć element poprzedni do wyświetlenia względem drzewa DOM dla [.js-slider__thumbs]
177
178     const currentImg = document.querySelector('.js-slider__thumbs-image--current')
179     const thumbImages = document.querySelectorAll('.js-slider__thumbs-image')
180     const currentImgIndex = Array.from(thumbImages).indexOf(currentImg)
181     let prevImg = thumbImages[currentImgIndex - 1]
182
183     // 3. sprawdzić czy ten element istnieje i czy nie posiada klasy [.js-slider__thumbs-item--prototype]
184     // 4. przełączyć klasę [.js-slider__thumbs-image--current] do odpowiedniego elementu
185     // 5. podmienić atrybut [src] dla [.js-slider__image]
186     if (prevImg === undefined) {
187         prevImg = thumbImages[thumbImages.length - 1]
188     }
189
190     if (prevImg !== null && !prevImg.classList.contains('js-slider__thumbs-item--prototype')) {
191         currentImg.classList.remove('js-slider__thumbs-image--current')
192         prevImg.classList.add('js-slider__thumbs-image--current')
193         const sliderImage = document.querySelector('.js-slider__image')
194         const currentImgSrc = document.querySelector('.js-slider__thumbs-image--current')
195         .getAttribute('src')
196         sliderImage.src = currentImgSrc
197     }
198 }
199
200 const onClose = function (event) {
201     // todo:
202     // 1. należy usunąć klasę [js-slider--active] dla [.js-slider]
203     const slider = document.querySelector('.js-slider')
204     slider.classList.remove('js-slider--active')
205
206     // 2. należy usunąć wszystkie dzieci dla [.js-slider__thumbs] pomijając [.js-slider__thumbs-item--prototype]
```