

JavaScript – zadanie 4 – galeria

Instrukcja krok po kroku – Adam Kwiatkowski

Na czym polega zadanie

Zadanie 4 polega na przygotowaniu strony internetowej, wykorzystującej JavaScript do zbudowania, wyświetlenia i obsługi galerii zdjęć, która ma działać jak karuzela. Jeżeli w galerii umieścimy pięć zdjęć, to po dotarciu do ostatniego powinniśmy móc zapętlić się na początek galerii (w obie strony). Problem, który musimy rozwiązać polega na zrozumieniu, jak określić, który obrazek jest obecnie wyświetlany. Kolejne rzeczy do wykonania:

- przygotować katalog roboczy (osobne foldery na CSS, JS i obrazy)
- zbudować dokument HTML, w którym będzie formularz logowania
- przygotować arkusz CSS, aby nadać stronie wygląd (w zewnętrznym pliku)
- pobrać/przygotować zdjęcia, które będziemy chcieli dodać na stronę
- napisać plik skryptowy, w którym będzie skrypt inicjujący galerię oraz funkcję obsługującą przewijanie

1. Wskazówki co do Visual Studio Code

Przydatna wskazówka: Visual Studio Code pozwala na „wygenerowanie” początkowych linijek kodu związanych z dokumentem HTML – wystarczy utworzyć nowy plik i zapisać go jako plik .html, a następnie napisać w nim sam znak wykrzyknik (!) i nacisnąć Enter. Trzeba tylko pamiętać o zmianie zadeklarowanego języka i tytułu dokumentu, a także o podłączeniu arkusza stylów

Druga przydatna wskazówka – w <head> dokumentu można napisać „link:css” i skorzystać z kolejnego skrótu, który generuje linijkę kodu odpowiedzialną za dodanie pliku CSS

Trzecia przydatna wskazówka – możemy automatycznie utworzyć potrzebne pliki CSS i JS poprzez najpierw napisanie linijek, które mają podłączyć pliki o odpowiedniej nazwie (umieszczone w odpowiednich folderach), a następnie trzymając Ctrl kliknąć na nazwę pliku (którego jeszcze nie mamy) – Visual Studio Code poinformuje nas, że nie można utworzyć takiego pliku (bo go nie ma), ale zapyta, czy chcemy go utworzyć.

2. Dokument HTML

Dokument HTML ma składać się z trzech głównych bloków:

- <header> - tu ma znaleźć się treść nagłówka strony: nagłówek drugiego stopnia z napisem „JavaScript – zadanie 4”
- <main> - blok centralny, który podzielimy na trzy bloki:
 - dwa blok <div> z klasą „pojemnikNaPrzycisk” – mają być po bokach. W każdym z nich ma znaleźć się przycisk <button>, do którego przypisana ma być funkcja przewijająca z odpowiednim parametrem (np. przycisk z lewej strony ma mieć funkcję przewin(„lewo”), a przycisk z prawej strony funkcję przewin(„prawo”)
 - blok <div> o id = „pojemnikNaGalerie”
- <footer> - stopka strony, gdzie powinien znaleźć się symbol © oraz podpis

3. Skrypt JS

Skrypt JS ma składać się ze skryptu, który załaduje do galerii pierwsze zdjęcie oraz funkcji przewin() z parametrem określającym kierunek przewijania. Kolejne kroki do wykonania:

- stworzenie zmiennej tablicowej, w której będą przechowywane ścieżki do zdjęć (np. let obrazy = [„img/zdj1.jpg”, „img/zdj2.jpg”...])
- stworzenie zmiennej, która ma być licznikiem, wskazującym na numer zdjęcia. Ta zmienna ma przyjąć wartość równą indeksowi pierwszego elementu z tablicy ze zdjęciami, czyli 0 (np. let nrZdjecia = 0)
- stworzenie zmiennej, do której przypiszemy nowo utworzony element typu obrazkowego. Będzie to pierwsze wyświetlane zdjęcie w galerii (np. let pierwszeZdjecie = document.createElement(„img”)
- przypisanie pierwszemu zdjęciu źródła – wybranie pierwszego elementu z tablicy (np. pierwszeZdjecie.src = obrazy[nrZdjecia])

- przypisanie do zmiennej pojemnika na galerię (np. let galeria = document.getElementById(„pojemnikNaGalerie”))
- podpięcie pierwszego zdjęcia do galerii (np. galeria.appendChild(pierwszeZdjecie))
- zbudowanie funkcji przewin(kierunek):
 - użycie switch(case) jako przełącznika, określającego kierunek przewijania (np. switch(kierunek))
 - dla case „lewo”:
 - sprawdzenie, czy obecny numer zdjęcia jest równy pierwszemu zdjęciu z tablicy (np. if (nrZdjecia == 0)):
 - jeżeli tak, to ustalenie numeru zdjęcia na numer ostatniego zdjęcia, wyliczone na podstawie długości tablicy (np. nrZdjecia = obrazy.length – 1)
 - jeżeli nie, to ustalenie numeru zdjęcia na o jeden mniejszy (np. nrZdjecia--)
 - dla case „prawo”:
 - sprawdzenie, czy obecny numer zdjęcia jest równy ostatniemu zdjęciu z tablicy (np. if (nrZdjecia == obrazy.length – 1)):
 - jeżeli tak, to ustalenie numeru zdjęcia na numer pierwszego zdjęcia (np. nrZdjecia = 0)
 - jeżeli nie, to ustalenie numeru zdjęcia na o jeden większy (np. nrZdjecia++)
 - stworzenie nowej zmiennej do której zostanie przypisany nowy element obrazkowy (np. let nowyObr = document.createElement(„img”);
 - przypisanie nowemu obrazkowi źródła na podstawie numeru zdjęcia (np. nowyObr.src = obrazy[nrZdjecia])
 - zastąpienie starego zdjęcia nowym poprzez usunięcie starego (np. galeria.innerHTML = „”), a następnie wstawienie nowego (np. galeria.appendChild(nowyObr))

4. Arkusz CSS

Stylizyka strony dowolna, ale jakaśkolwiek wymagana. Warto zadbać o ustalenie z góry wielkości zdjęć. Możemy napisać reguły CSS dla elementów, które jeszcze nie są umieszczone na stronie (a pojawią się tam później, np. w wyniku działania skryptu). Przydatne reguły:

- margin: auto;
- display: flex;
- flex-direction: row;
- justify-content: center;
- align-items: center;

Błokowi <main> warto przydzielić wszystkie reguły dotyczące wyświetlania elastycznego.

JavaScript - zadanie 4.

