

# JavaScript – zadanie 3 – strona z logowaniem i dodawaniem treści

Instrukcja krok po kroku – Adam Kwiatkowski

## Na czym polega zadanie

Zadanie 3 polega na przygotowaniu strony internetowej, wykorzystującej JavaScript do sprawdzenia danych logowania, wprowadzanych przez użytkownika. Nie jest to „prawdziwe” logowanie, a ledwie sprawdzenie, czy podawane przez użytkownika hasło jest zgodne z tym, które musimy w zadaniu samemu ustalić (do „prawdziwego” logowania potrzebować będziemy języka PHP, o czym kiedyś indziej na zajęciach). Kolejne kroki, które musimy zrobić w tym zadaniu:

- 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 będą trzy funkcje: logowanie, zastąpienie treści, dodanie zdjęć

## 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 3”
- <div id = „glownePudelko”> - zamiast <main> chcemy stworzyć <div>, w którym ma znaleźć się formularz logowania, czyli <form>, w którym potrzebujemy:
  - dwa pola <input>, jedno typu tekstowego, drugie typu „hasło”. Pierwszemu nadajemy id = „poleLogin”, drugiemu id = „poleHaslo”
  - przycisk do logowania, któremu przypiszemy funkcję logowanie() przy kliknięciu (onclick)
- <footer> - stopka strony, gdzie powinien znaleźć się symbol © oraz podpis

## 3. Skrypt JS

Skrypt JS ma składać się z trzech funkcji:

- logowanie() – funkcja sprawdzająca, czy użytkownik podał poprawne dane logowania. Kolejne kroki tej funkcji to:
  - utworzenie dwóch zmiennych, którym przypisane zostaną wartości pól logowania i hasła, np. let username = document.getElementById(„poleLogin”).value;
  - sprawdzenie, czy login składa się z minimum 8 znaków oraz czy hasło jest takie, jakie wymyślimy – potrzebujemy bloku if, w którym na raz sprawdzimy oba te warunki (zapis &&). Długość tekstu można sprawdzić metodą length (np. username.length).

- jeżeli i login, i hasło są OK, należy wywołać funkcję wstawiającą treść, a jako jej parametr ustalić zmienną, przechowującą login (np. `wstawTresc(username)`).
  - jeżeli albo login, albo hasło nie są OK, należy powiadomić użytkownika, że nie udało się zalogować (np. `alert()`)
- `wstawTresc(uzytkownik)` – funkcja z parametrem, która ma usunąć formularz logowania ze strony, a w jego miejsce wstawić inną zawartość. Kolejne kroki tej funkcji to:
- utworzenie nowej zmiennej, do której przypisany zostanie blok `<div>`, w którym jest formularz (np. `let tresc = document.getElementById(„glownyPojemnik”)`)
  - zamiana treści HTML tego bloku (metoda `tresc.innerHTML`) na blok `<main>`, w którym ma być:
    - nagłówek trzeciego stopnia w którym znajdzie się napis „Witaj” oraz pusty blok `<span>` o id = „nazwaUzyt”
    - blok `<section>` w którym znajdzie się napis „Oto twoje zdjęcia”
  - przypisanie wartości parametru `uzytkownik` jako tekst wewnętrzny elementu o id = „nazwaUzyt” (np. `document.getElementById(„nazwaUzyt”).innerText = uzytkownik`)
  - wywołanie funkcji dodającej obrazki (np. `dodajObrazki()`)
- `dodajObrazki()` – funkcja opierająca się na pętli tworzącej i dodającej nowe obrazki do strony. Kolejne kroki tej funkcji to:
- utworzenie nowej zmiennej, do której przypisany zostanie blok `<div>`, w którym teraz powinna być nowa treść, pochodząca z poprzedniej funkcji (np. `let tresc = document.getElementById(„glownyPojemnik”)`)
  - utworzenie nowej zmiennej, do której przypisany zostanie nowy blok `<div>`, w którym będziemy gromadzić zdjęcia, które chcemy dodać na stronę (np. `let pojemnikNaZdjecia = document.createElement(„div”)`)
  - nadanie temu pojemnikowi na zdjęcia reguł stylistycznych CSS takich, aby zdjęcia układały się obok siebie i były wycentrowane (korzystając z zapisu `pojemnikNaZdjecia.style.”regula”` trzeba nadać reguły `display: flex, flex-direction: row, justify-content: center`)
  - utworzenie zmiennej tablicowej, w której jako wartości mają być przechowane ścieżki do plików ze zdjęciami (np. `let zdjecia = [„img/zdjecie1.jpg”, „img/zdjecie2.jpg”...]`)
  - zbudowanie pętli, która ma wykonać się tyle razy, ile jest elementów w tablicy (`zdjecia.length`). Pętla powinna:
    - utworzyć nową zmienną, do której przypisany zostanie nowo utworzony element obrazkowy (np. `let noweZdjecie = document.createElement(„img”)`)
    - wybranie jako źródło tego nowego obrazka kolejnej wartości ze zmiennej tablicowej (np. `noweZdjecie.src = zdjecia[i]`)
    - podpięcie nowego zdjęcia jako element-dziecko dla pojemnika na zdjęcia (np. `pojemnikNaZdjecia.appendChild(noweZdjecie)`)
  - podłączenie pojemnika na zdjęcia (wypełnionego zdjęciami) jako element-dziecko głównego bloku z treścią strony (np. `tresc.appendChild(pojemnikNaZdjecia)`)

## 4. Arkusz CSS

Stylizacja 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;`

### JavaScript - zadanie 3

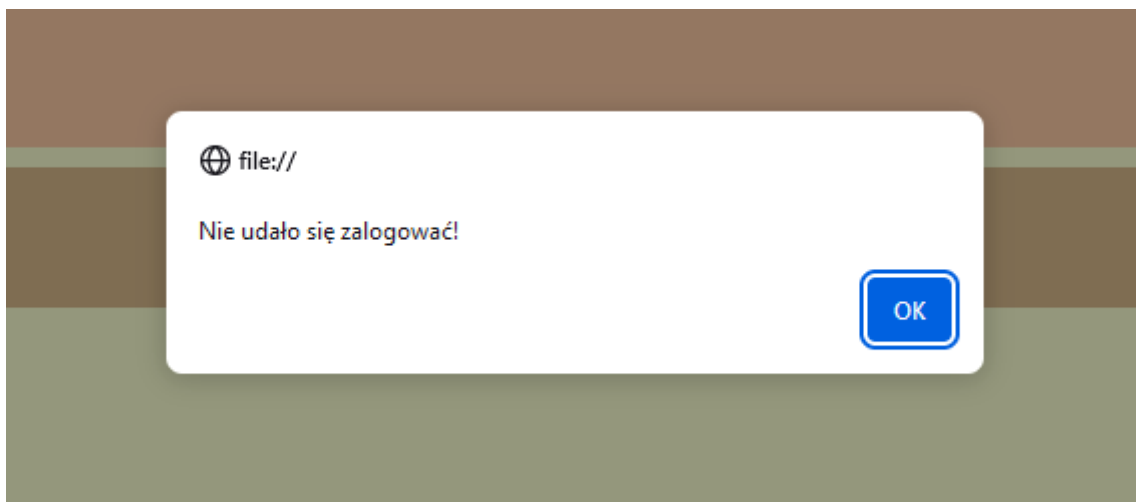
Interaktywna strona z logowaniem

Zaloguj się:

Login:   
Hasło:

© Adam Kwiatkowski

Rysunek 1. Strona z zadania 3. przed zamianą treści



Rysunek 2. Problem z logowaniem powinien być zwrócony użytkownikowi jako błąd

### JavaScript - zadanie 3

Interaktywna strona z logowaniem

Witaj AdamKwiatkowski!

Oto twoje zdjęcia kotków.



Rysunek 3. Po zalogowaniu się powinny zniknąć paski formularza. W ich miejscu ma znaleźć się napis powitalny oraz zdjęcia. W nagłówku powinien znaleźć się także login użytkownika