

# Małpia lista zadań 3

1. Za pomocą metody `Window#prompt` z API przeglądarkowego zaimplementuj program sprawdzający wiek użytkownika przed pokazaniem mu zdjęcia smakowitego napoju. Strona ma prezentować użytkownikowi wiadomość, że wymagana jest weryfikacja wieku oraz przycisk, który ma zapoczątkować operację weryfikacji pełnoletniości. W przypadku wpisania wartości większej lub równej 18, treść strony ma zostać zastąpiona zdjęciem ww. napoju wraz ze zmianą tła na wybrany przez Ciebie kolor. Obsłuż zdarzenie `click` dodając odpowiedni *listener* i użyj metody `parseInt` ze standardowej biblioteki JavaScript w celu zamiany wpisanej przez użytkownika wartości na liczbę. Postaraj się wykorzystać funkcję w celu abstrakcji operacji które wykonujesz.
2. Stwórz stronę zawierającą dwa przyciski `+` oraz `-` wraz z etykietą reprezentującą *obrót w stopniach* załączonego obrazka. Domyślnie strona rozpoczyna ze stanem obrotu  $0^\circ$  kliknięcie przycisku `+` ma obrócić obrazek o kolejne  $90^\circ$  zgodnie z ruchem wskazówek zegara, zaś `-` ma go obrócić o  $90^\circ$  przeciwne do ruchu wskazówek zegara. W celu obrotu obrazka zdefiniuj trzy klasy CSS które obracają obrazek za pomocą właściwości `transform` dla kątów  $90$ ,  $180$  oraz  $270$  stopni. Za pomocą API `HTMLImageElement` odpowiednio dodawaj i usuwaj klasy CSS po kliknięciu przycisku. Przycisk `+` ma być wyłączony (atrybut `disabled`) dla wartości  $270$ . Przycisk `-` ma być wyłączony przy wartości  $0$ . Dodaj i usuwaj odpowiedni atrybut `disabled` za pomocą JavaScript. Za pomocą właściwości `transition` zanimuj tę rotację w CSS. Postaraj się wykorzystać funkcje w celu abstrakcji operacji które wykonujesz.
3. Zmodyfikuj działanie przycisków `+` i `-` z poprzedniego zadania - tym razem mają one dodawać bądź odejmować 10 stopni do obrotu w odpowiednią stronę. W tym celu nie wystarczą (no dobra, wystarczą jak chcesz wprowadzić 35 klas) już klasy CSS i należy to zrobić modyfikując atrybut `style` z poziomu JavaScript.
4. Dodaj przycisk `SSIJ PAŁĘ` obok przycisków `+` i `-`. Po kliknięciu tego przycisku należy wyłączyć (atrybut `disabled`) przyciski `+` i `-`. Po kliknięciu tego przycisku jego etykieta ma się zmienić na `JADĘ DO BIAŁEGOSTOKU`. Obrazek ma się zacząć samoczynnie kręcić dopóki użytkownik nie kliknie w przycisk o zmienionej etykiecie ponownie. W tym celu wykorzystaj API przeglądarki `setInterval` które wykonuje

działanie w kółko co zadaną w drugim parametrze liczbę milisekund (`10^-4` sekundy). W celu wyłączenia tego działania możemy skorzystać z metody `clearInterval` która przyjmuje zwrócony przez `setInterval` identyfikator jako argument. Po kliknięciu ponownie przycisku stan przycisków powinien być poprawny, tj. przyciski `+` i `-` mają się odblokować (chyba że trafimy na `0` lub `270` przy wyłączeniu oczywiście), etykieta przycisku ma się zmienić. Etykieta reprezentująca obrót również ma się aktualizować przy tym samoczynnym obrocie. Jeżeli nie postarałeś się stworzyć dobrych funkcji przy okazji zadań 2 i 3 spodziewaj się, że ugryzie Cię to bardzo mocno w dupę robiąc to zadanie :).

5. Stwórz cztery kwadraty o różnych kolorach za pomocą elementów `div`. Za pomocą zdarzeń `mouseenter` i `mouseleave` spraw, by pierwszy kwadrat rozszerzał się dwukrotnie i zwężał przy najechaniu/odjechaniu kursorem. Drugi ma zmieniać kolor. Trzeci ma zmieniać swoją pozycję o 100 pikseli w dół (tak, masz podążać za nim kursorem żeby ten efekt się nie “zwinął”). Czwarty ma się obracać o 45 stopni zgodnie z ruchem wskazówek zegara. Wykorzystaj dodawanie/usuwanie klas CSS oraz dodaj animacje tych zachowań za pomocą właściwości `transition`.
6. Dodaj zliczanie zakończonych animacji używając zdarzeń `animationstart` oraz `animationend`. Stwórz cztery osobne liczniki zliczające zakończone animacje.
7. Zamień statyczne kwadraty na tagi `IMG` stanowiących twarze żółtego rodzaju małp. Zmieniaj kolor za pomocą właściwości `filter` w CSS zamiast ustawiania `background-color`. Reszta transformacji powinna zostać bez zmian.
8. **(Dłuższe)** Pobierz odpowiednie zasoby dla stworzenia swojej wersji aplikacji `TodoMVC`. Aplikacja ma działać jak ta tutaj: <https://todomvc.com/examples/vanillajs/>. Możesz korzystać z jej źródeł w celu inspiracji, ale staraj się zrozumieć zawsze to czym się inspirujesz. Spróbuj wydać tą aplikację publicznie za pomocą Netlify.