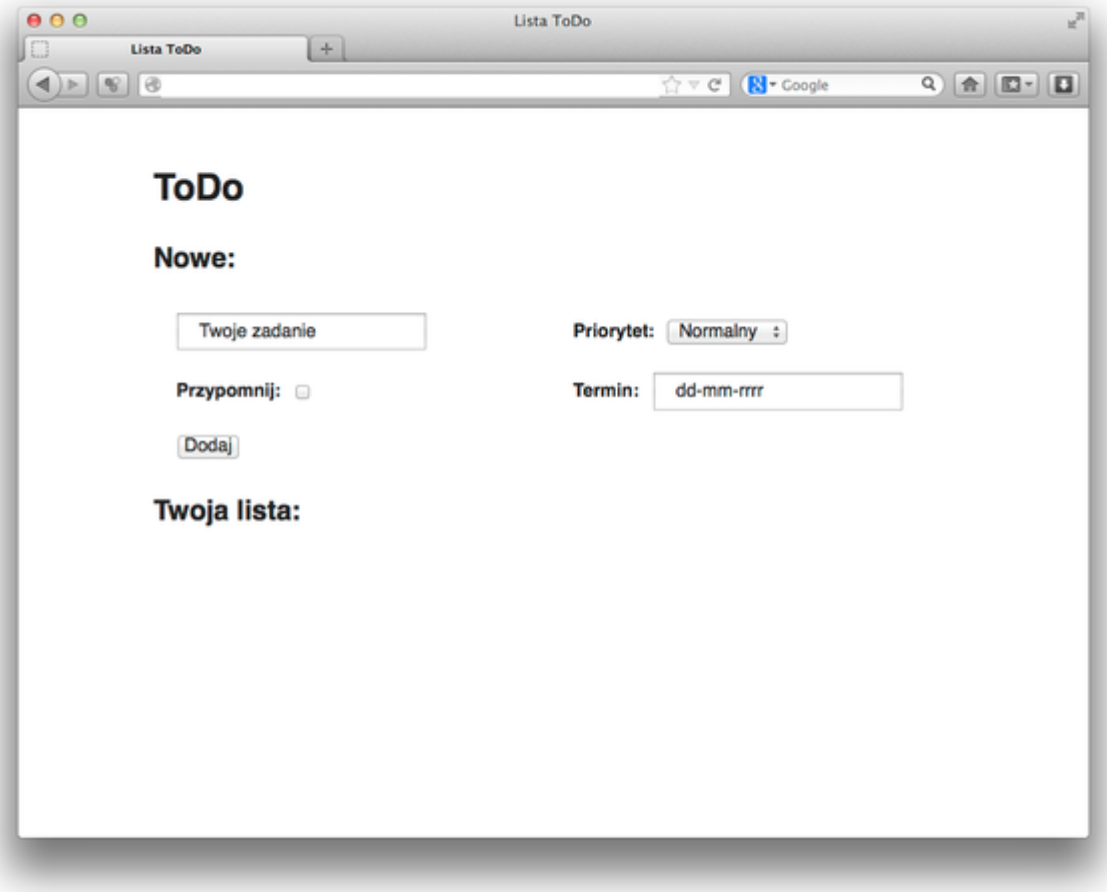
Nazwa implementacji: jQuery - zdarzenia i efekty **Autor:** michal.czyzewski **Opis implementacji:** Do już znanych podstaw jQuery wprowadzimy dodatkowe informacje o tym jak reagować na różne operacje dokonywane przez użytkownika oraz prezentować wynik działań poprzez zmiany w dokumencie HTML używając odpowiednich efektów. Na koniec całość pozwoli nam stworzyć zalążek aplikacji ToDo, która będzie rozwijana przy kolejnych lekcjach.

Poznaliśmy już podstawowe sposoby na wykrywanie tego, co użytkownik robi w aplikacji, oraz jak dodawać kolejne elementy do dokumentu HTML. Wykorzystamy tą wiedzę i rozszerzymy ją o bardziej zaawansowane przykłady, tworząc prostą aplikację umożliwiającą nam tworzenie list zadań, które mamy do wykonania, oznaczania ich priorytetów i statusu.

Zadanie: Przygotuj prosty dokument HTML zawierający pustą listę elementów i formularz HTML.



Zdarzenia formularza

W momencie, gdy użytkownik wykonuje operacje na elementach należących do formularza HTML, mogą zostać wywołane następujące wydarzenia:

submit – wywoływane tylko i wyłącznie przez element <form> w momencie, kiedy użytkownik zatwierdza formularz. focus – wywoływane na elemencie, który staje się aktywny w celu wprowadzenia danych przez użytkownika. blure – wywoływane na elemencie, który przestaje być aktywny, gdyż użytkownik przeszedł do kolejnego elementu lub kliknął w dowolnym miejscu poza formularzem. change – wywoływane na elemencie, którego zawartość została zmieniona przez użytkownika. Uwaga: To zdarzenie najlepiej sprawdza się przy elementach takich jak <input type=”checkbox”>, <input type=”radio”> i <select>. W przypadku elementu <input type=”text”> lub <textarea> to zdarzenie jest wywoływane razem z blure. Żeby na bieżąco

KAPITAŁ LUDZKI NARODOWA STRATEGIA SPÓJNOŚCI
ORE
FWIOO
UNIA EUROPEJSKA EUROPEJSKI FUNDUSZ SPOŁECZNY

Projekt “Strategia Wolnych i Otwartych Implementacji jako innowacyjny model zainteresowania kierunkami informatyczno-technicznymi oraz wspierania uczniów i uczennic w kształtowaniu kompetencji kluczowych” współfinansowany ze środków Unii Europejskiej w ramach Europejskiego Funduszu Społecznego.

sprawdzać wprowadzane informacje, lepiej skorzystać z keyup lub keydown omówione w punkcie Zdarzenia klawiatury.

Zadanie: Dodaj następującą obsługę zdarzeń:

- Gdy priorytet zadania zostanie zmieniony na wysoki lub niski, zmień kolor tekstu w <label> na odpowiednio czerwony lub zielony.
- Ukryj pole do wprowadzania terminu wykonania zadania i wyświetlaj je dopiero po zaznaczeniu pola Przypomnij.
- Gdy użytkownik aktywuje pole do wprowadzania terminu wykonania zadania, automatycznie wprowadź aktualną datę jako podpowiedź. Uwaga: Żeby wygenerować aktualną datę, skorzystaj z funkcji: new Date().toLocaleFormat('%d-%m-%Y').
- Gdy użytkownik zatwierdzi formularz, pobierz z niego wszystkie informacje i dodaj nowe zadanie na listę. Dodatkowo: Zamiast wyszukiwać każde pole formularza z osobna, skorzystaj z funkcji: $('form').serializeArray(); zwracającej w tablicy wartości wprowadzone we wszystkie pola dodane do formularza.

Uwaga: W momencie, kiedy piszemy funkcje obsługującą zdarzenie na konkretnym elemencie i potrzebujemy odwołać się do jednego z jego atrybutów, na przykład przy wywołaniu zdarzenia change chcemy pobrać wartość atrybutu checked zamiast tworzyć nowy selektor, korzystamy ze specjalnej zmiennej $(this), wskazującej na obsługiwany element:

```
$('#przypomnij').on('change', function(event) {

var $this = $(this);

var zaznaczony = $this.attr('checked');

});
```

Zdarzenia klawiatury

W momencie, kiedy użytkownik wprowadza wartości do pola tekstowego przy pomocy klawiatury, wywoływane są następujące zdarzenia:

- keydown – została wciśnięty przycisk. Wartość odpowiadająca wciśniętemu przycisku jest przechowywana w zmiennej event.which.
- keypress – tak samo jak przy keydown, ale w zmiennej event.which przechowywana jest wartość wprowadzanego znaku, a nie wciśniętego przycisk (uwzględnione są informacje o wciśniętych przyciskach alt lub shfit). Natomiast w przypadku przycisków specjalnych (na przykład strzałki), wartość wynosi 0.
- keyup – użytkownik przestał wciskać przycisk. To wydarzenie najlepiej się nadaje do sprawdzenia, jaka jest zawartość pola <input> po wprowadzonych zmianach.

Zadanie: Korzystając z console.log() napisz obsługę wszystkich powyższych zdarzeń, wyświetlając w konsoli zawartość zmiennej event.which. Porównaj, w jakiej kolejności wywoływane są zdarzenia. Następnie wybierając jedno ze zdarzeń, weryfikuj i podpowiadaj użytkownikowi, czy wprowadzana data jest w poprawnym formacie: dd-mm-yyyy.

Uwaga: Dla sprawdzenia czy wprowadzana data jest poprawna, wykorzystaj funkcję używającą wyrażenia regularne:

```
if ( $this.val().match(/^[0-9]{1,2}\-[0-9]{1,2}\-[0-9]{4}$/) )
```


KAPITAŁ LUDZKI NARODOWA STRATEGIA SPÓJNOŚCI
ORE
FWIOO
UNIA EUROPEJSKA EUROPEJSKI FUNDUSZ SPOŁECZNY

Projekt “Strategia Wolnych i Otwartych Implementacji jako innowacyjny model zainteresowania kierunkami informatyczno-technicznymi oraz wspierania uczniów i uczennic w kształtowaniu kompetencji kluczowych” współfinansowany ze środków Unii Europejskiej w ramach Europejskiego Funduszu Społecznego.

Delegowanie obsługi zdarzeń

Napisane przez nas aktualne obsługi zdarzeń działają tylko na elementach, które znajdują się w dokumencie HTML tuż po załadowaniu strony, ale gdy w efekcie dokonanych operacji dodajemy kolejne elementy, nie uda nam się w ten sposób zareagować przy akcji użytkownika na tych elementach: $('#all-tasks li').on('click', ...); – nie zadziała poprawnie.

Żeby coś takiego było możliwe, przypinamy obsługę zdarzenia do elementu, który będzie zawierał nowo dodawane elementy i podajemy dodatkowy parametr w postaci selektora wskazującego, jakich nowych elementów ma nasza obsługa zdarzenia wyszukiwać:

$('#all-tasks').on('click', 'li', function(event){ ...

Na dzieciach elementu #all-tasks wyszukujemy wszystkie dynamicznie tworzone li i oczekujemy na akcje użytkownika.

Zadanie: Napisz kod, który będzie wyszukiwał dynamicznie dodawane zadania do listy i gdy użytkownik kliknie w zadanie zostanie oznaczone jako wykonane.

Efekty i animacje

jQuery umożliwia dodawanie, ukrywanie i pokazywanie elementów z użyciem płynnych animacji, które sprawiają, że aplikacja jest bardziej estetyczna i wygodniej się z niej korzysta. Podstawowe efekty dostępne w bibliotece to:

animate – pozwala na płynną zmianę w dowolnym parametrze CSS od aktualnej do nowo podanej wartości. fadeIn / fadeOut – powodują płynne pojawienie się i zniknięcie elementu poprzez zmianę jego przezroczystości. slideDown / slideUp – powodują zwinięcie i rozwinięcie elementu modyfikując jego wysokość.

dodatkowo do jQuery można dołączyć rozszerzenie w postaci jQuery UI, które posiada kilkanaście gotowych animacji wywoływanych poprzez metode effect(). Zobacz więcej na stronie: http://jqueryui.com/effect/

Każdy z efektów przyjmuje między innymi 2 parametry:

czas, jaki ma trwać animacja w milisekundach; funkcje, która zostanie wykonana po zakończeniu animacji, co pozwala nam na przykład na łączenie animacji w łańcuchy.

Na przykład:

$('li').fadeIn(1000, function() {

$(this).animate({'background-color': '#999'}, 500);

});

Element <li> najpierw będzie powoli pojawiał się na stronie przez 1 sekundę, a następnie, jego kolor zmieni się do szarego w 0.5 sekundy. Całość animacji będzie trwało 1.5 sekundy.

Natomiast:

$('li').fadeIn(1000);

$('li').animate({'background-color': '#999'}, 500);

Spowoduje, że obie animacje zostaną odpalone praktycznie równocześnie i element <li> będzie miał kolor szary, zanim w pełni pojawi się na stronie, a całość animacji będzie trwałą 1 sekundy.

Zadanie: Dodaj animację fadeIn i slideDown zastępując zwykłe wyświetlanie nowych elementów.

KAPITAŁ LUDZKI NARODOWA STRATEGIA SPÓJNOŚCI
ORE
FWIOO
UNIA EUROPEJSKA EUROPEJSKI FUNDUSZ SPOŁECZNY

Projekt “Strategia Wolnych i Otwartych Implementacji jako innowacyjny model zainteresowania kierunkami informatyczno-technicznymi oraz wspierania uczniów i uczennic w kształtowaniu kompetencji kluczowych” współfinansowany ze środków Unii Europejskiej w ramach Europejskiego Funduszu Społecznego.

Uwaga: W momencie, gdy dodajemy zupełnie nowy element, należy go najpierw ukryć poprzez hide(), a dopiero potem wywołać animację stopniowo wyświetlającą go na stronie.

Zadanie dodatkowe: Dołącz do aplikacji rozszerzenie jQuery UI i wykorzystaj w aplikacji rozbudowane animacje dostępne z listy: http://jqueryui.com/effect/.

KAPITAŁ LUDZKI NARODOWA STRATEGIA SPÓJNOŚCI
ORE
FWIOO
UNIA EUROPEJSKA EUROPEJSKI FUNDUSZ SPOŁECZNY

Projekt “Strategia Wolnych i Otwartych Implementacji jako innowacyjny model zainteresowania kierunkami informatyczno-technicznymi oraz wspierania uczniów i uczennic w kształtowaniu kompetencji kluczowych” współfinansowany ze środków Unii Europejskiej w ramach Europejskiego Funduszu Społecznego.