Nazwa implementacji: Podstawy jQuery

Autor: michal.czyzewski

Opis implementacji:

Wykorzystując wiedzę z lekcji o formularzach HTML i JavaScript stworzymy napisany program tak, żeby komunikował się z użytkownikiem poprzez formularz i dokument HTML.

jQuery jest aktualnie najbardziej popularną biblioteką języka JavaScript ułatwiającą tworzenie aplikacji działających w przeglądarce internetowej. Korzystając z dostępnych w niej rozwiązań o wiele łatwiej stworzyć aplikację, która poprawnie działa pomiędzy różnymi przeglądarkami.

Aby dołączyć jQuery do zwykłego dokumentu HTML, dodaj w sekcji <head> linię:

```
<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.10.2/jquery.min.js"></script>
```

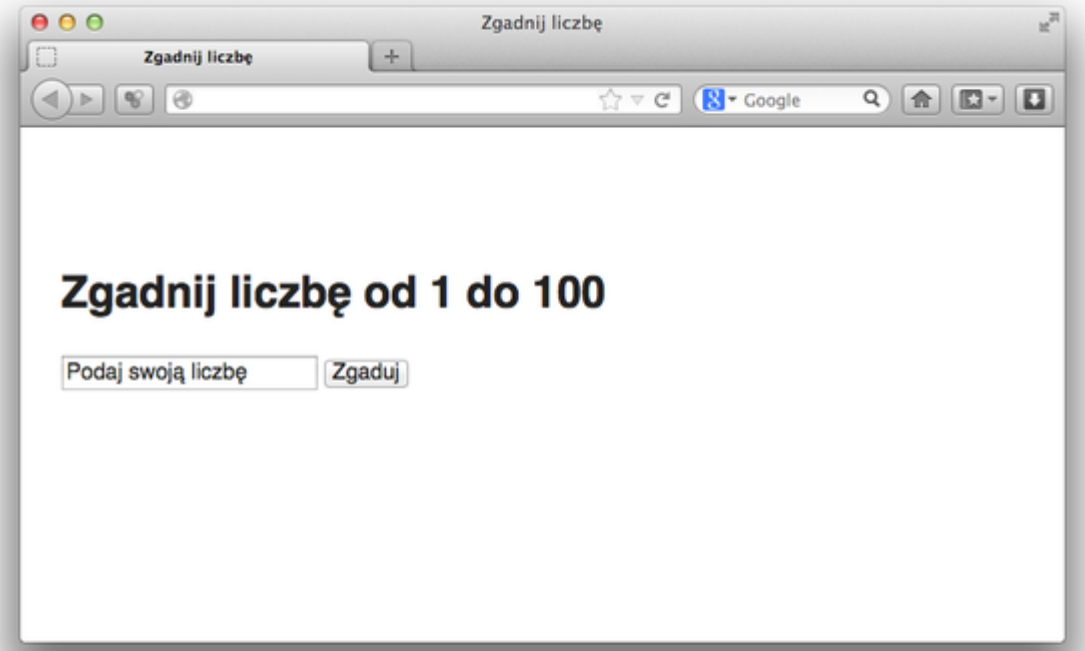
W przykładowym projekcie jest już dołączone jQuery (lokalna kopia pliku).

Zadanie: Przygotuj dokument HTML, który będzie służył do komunikacji pomiędzy programem, a użytkownikiem.

```
<body>
 <h1 class="naglowek">Zgadnij liczbę od 1 do 100</h1>
 <form>
   <input type="number" name="strzal" id="strzal"
     placeholder="Podaj swoją liczbę"
   />
   <input type="submit" value="Zgaduj" id="zgaduj" />
 </form>
 <ul id="odpowiedz"></ul>
 <script type="text/javascript" src="jquery.js"></script>
 <script type="text/javascript" src="script.js"></script>
</body>
```


KAPITAŁ LUDZKI NARODOWA STRATEGIA SPÓJNOŚCI
ORE
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.



Selectory CSS w jQuery

W celu wyszukania i dokonania operacji dowolnej operacji na elemencie HTML jQuery wykorzystuje takie same selektory, jak stosujemy w pliku CSS, żeby nadać pożądany wygląd. Selektor wprowadzamy jako łańcuch znaków do głównej funkcji jQuery oznaczonej znakiem $():

```
$('input') // wyszukuje wszystkie elementy <input>
	$('#zgaduj') // wyszukuje element z id=”zgaduj”
	$('.naglowek') // wyszukuje elementy z class=”naglowek”
```

Zadanie: W konsoli JavaScript wpisz powyższe przykłady i sprawdź czy jQuery zwraca poprawne elementy z dokumentu HTML. Następnie wprowadź ten sam kod do pliku script.js i wyświetl wyniki działania, używając console.log();. Czym się różnią?

Uruchomienie programu

W poprzednim punkcie w zadaniu ten sam kod zwraca inne wartości w przypadku uruchomienia z konsoli, a inny w przypadku umieszczenia go w pliku. Dzieje się tak dlatego, że kod umieszczony w pliku uruchamia się natychmiast po załadowaniu, zanim jeszcze przeglądarka zdąży przygotować dokument HTML, na którym ma zostać dokonana operacja. Żeby zapobiec takiej sytuacji, jQuery udostępnia mechanizm, który powoduje, że kod uruchomi się dopiero po tym, jak strona jest gotowa:

```
$(document).ready(function(){
	/* Kod programu */
});
```

lub w jeszcze krótszej formie:

```
$(function(){
	/* Kod programu */
});
```

Zadanie: Skopiuj kod z poprzedniego zadania i uruchom w funkcji $(document).ready();. Porównaj wyniki działania.

KAPITAŁ LUDZKI NARODOWA STRATEGIA SPÓJNOŚCI
ORE
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.

Zdarzenia w jQuery

Żeby program mógł poprawnie działać i reagować na to, co wykonuje użytkownik w przeglądarce, potrzebujemy zdefiniować funkcję, która będzie uruchamiana w momencie wykrycia danej operacji na dokumencie HTML. Najprostszym i najczęściej używanym zdarzeniem jest click:

```
$('input').on('click', function(event){
       /* Kod wykonywany po wykryciu kliknięcia */
});
```

Zadanie: Napisz kod, który po kliknięciu przycisku #zgaduj będzie wyświetlał prosty komunikat przy pomocy alert();.

Pobieranie informacji z formularza

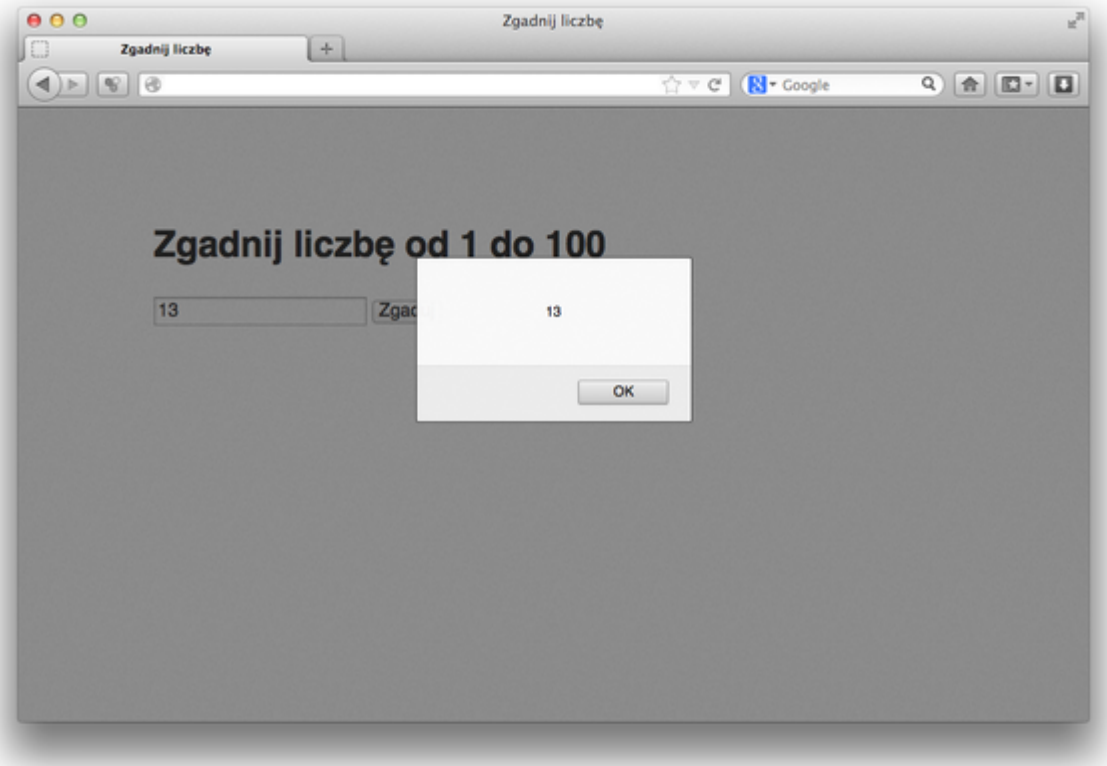
jQuery udostępnia szereg metod, które umożliwiają pobieranie i modyfikowanie elementów HTML:

- $('input').attr('id') – zwraca wartość umieszczoną pod danym atrybutem. W momencie, kiedy podamy dodatkowy 2 parametr, zamiast pobierać wartość, będziemy mogli ją zmodyfikować. Na przykład: $('#strzal').attr('value', 100);.
- $('input').val() – jest to skrócona forma od $('input').attr('value');
- $('p').text() – umożliwia pobieranie i ustawianie zawartości tagu w postaci tekstu.
- $('div').html() – jak wyżej tylko pobierany lub modyfikowany jest kod HTML znajdujący się wewnątrz danego elementu.
- $('div').addClass('selected'); – dla atrybutu class stworzony jest specjalna metoda, która pozwala dodawać nową klasę bez modyfikowania poprzednich.
- $('div').removeClass('selected'); – jak wyżej, tylko zamiast dodawać klasę do elementu, jest ona usuwana.

Zadanie: Zmodyfikuj kod tak, żeby po kliknięciu przycisku #zgaduj w komunikacie uruchamianym poprzez alert(); była wyświetlana zawartość z pola tekstowego #strzal.

KAPITAŁ LUDZKI NARODOWA STRATEGIA SPÓJNOŚCI
ORE
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.



Wstrzymanie wydarzenia

Nasz program potrafi już pobrać wartość wprowadzoną przez użytkownika do pola tekstowego, ale w momencie zamknięcia okna z komunikatem strona jest przeładowana i wprowadzona wartość znika. Spowodowane jest to standardowym zachowaniem przeglądarki, która po zatwierdzeniu formularza wysyła wprowadzone informacje do serwera i przeładowuje stronę. Żeby zapobiec takiemu zachowaniu, możemy zatrzymać jej działanie poprzez wywołanie:

event.preventDefault();

lub zwróceniu wartości false po skończeniu obsługi zdarzenia:

$('#zgaduj').on('click', function(event) {

alert($('#strzal').val());

return false;

});

Obie metody wykonują tą samą operacje, różnią się jedynie tym, że w momencie, gdy w trakcie obsługi zdarzenia nastąpi błąd w programie w przypadku event.preventDefault(); akcja przeglądarki zostanie zatrzymana, a przy zwracaniu wartości false; zachowa się, jakby obsługa zdarzenia nie istniała i wykona domyślną operację: wysłanie formularza / przejście do kolejnej strony.

Zadanie: Uzupełnij kod o dowolną z powyższych metod tak, żeby po zamknięciu komunikatu wartość wprowadzona przez użytkownika była dalej wyświetlona w polu tekstowym.

KAPITAŁ LUDZKI NARODOWA STRATEGIA SPÓJNOŚCI
ORE
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.

Modyfikowanie dokumentu HTML

Ciągłe wyświetlanie komunikatów alert(); nie jest wygodnym sposobem na interakcje. Zamiast tego, możemy na bieżąco modyfikować nasz dokument HTML, dodając i modyfikując elementy HTML. Do takich operacji mamy cały zestaw metod w jQuery:

- $(' ') – tworzy nam element HTML zgodnie z tagami podanymi jako łańcuch znaków. Uwaga: Należy odróżniać tworzenie elementu $(' ') od wyszukiwania w dokumencie: $('div').
- $('#odpowiedz').prepend($elem); – dodaje element podany jako argument wewnątrz elementu #odpowiedz przed innymi już istniejącymi elementami.
- $('#odpowiedz').append($elem); – jak wyżej, tylko dodanie elementu następuje po wszystkich pozostałych elementach znajdujących się wewnątrz #odpowiedz.
- $('#odpowiedz').replaceWith($elem); – zastępuje cały tag #odpowiedz nowym elementem przekazanym w parametrze.

Więcej metod i ich dokładny opis modyfikujących dokument HTML można znaleźć pod adresem:

http://api.jquery.com/category/Manipulation/

Zadanie: Na początku załadowania strony dodaj do listy <ul id=”odpowiedz”> pierwszy element <li> zawierający komunikat tłumaczący zasady i zachęcający użytkownika do gry. Następnie po każdym kliknięciu przycisku zgadnij, dodaj kolejny element z informacją o dokonanej próbie i wyzeruj zawartość elementu input, korzystając z metody .val('') (ważne jest podanie pustego ciągu znaków: '', w przeciwnym wypadku wartość w polu tekstowym nie zostanie zmieniona).

Uwaga: Tag <li> możemy stworzyć od razu, podając treść, jaką ma zawierać:

$('
Komunikat do użytkownika
');

lub wykorzystać metody omówione w punkcie 4: text() lub html():

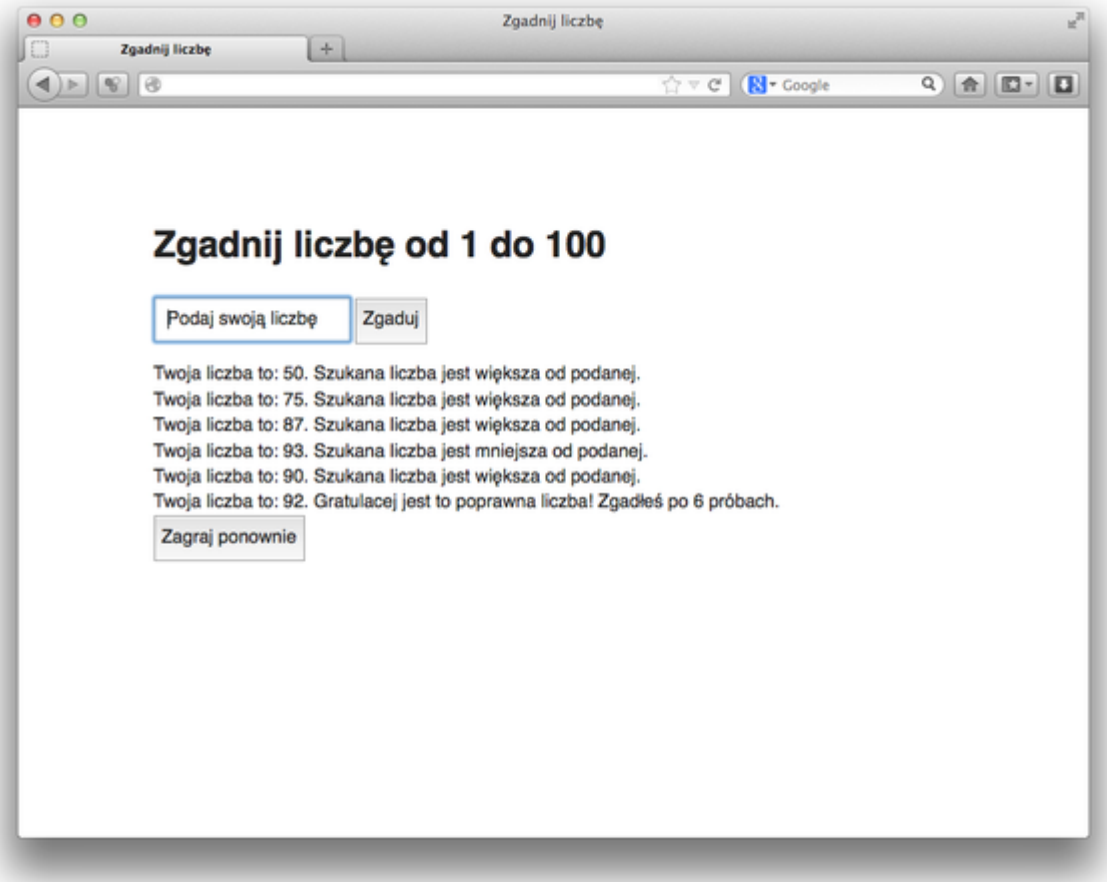
$('<li>').text('Komunikat do użytkownika');

Wypróbuj obie metody.

Zadnie końcowe: Uzupełnij do końca aplikację, korzystając z kodu z poprzednich zajęć, informując użytkownika o dokonanych próbach i tego czy poszukiwana liczba jest większa lub mniejsza od podanej. Gdy użytkownik skończy grę, wyświetl przycisk umożliwiający rozpoczęcie grę od nowa. Po jego kliknięciu wszystkie utworzone komunikaty powinny zostać usunięte z dokumentu HTML oraz wylosowana nowa liczba do zgadywania.

KAPITAŁ LUDZKI NARODOWA STRATEGIA SPÓJNOŚCI
ORE
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.



KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI
ORE
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.