

Zaawansowane Programowanie Webowe

laboratorium nr 2 – Przypomnienie JavaScript + JQuery

Po nabyciu umiejętności projektowania responsywnych stron internetowych przyszedł czas na wprowadzenie do strony elementów reaktywności i dynamiki. Strona powinna zacząć reagować na zdarzenia generowane przez użytkownika. Jest to możliwe wykorzystując JavaScript ewentualnie najbardziej znaną bibliotekę do obsługi GUI – JQuery. W tym laboratorium przypomnimy sobie sposoby korzystania z natywnego JavaScript + JQuery.

Celem ćwiczenia jest zapoznanie studenta z możliwościami przetwarzania danych użytkownika wprowadzonych w formularzu po stronie klienta w oparciu o skrypt JavaScript i obiektowy model dokumentu DOM (ang. document object model). Realizując ćwiczenia student zapozna się z elementami języka JavaScript oraz możliwościami wykorzystania go do dynamicznej generacji zawartości strony internetowej oraz dostępu do elementów formularza.

Proszę starać się korzystać z JS w wersji standardu ES6.

Szczegółowe informacje na temat JavaScript można znaleźć w:

<https://javascript.info/>

lub ewentualnie tutaj (brak jednak opisu standardu ES6)

<http://www.w3schools.com/js/default.asp>

Edytor online do tworzenia kodu:

<https://es6console.com/>

Zadanie 1. Rozszerzony formularz walidacji

Twoim zadaniem jest przygotowanie i walidacja ankiety dla agencji modelek i modeli „Modelinki”. Strona ankiety musi umożliwić pobranie od każdego chętnego do pracy w charakterze modela następujących informacji:

- imię
- nazwisko
- adres zamieszkania
- telefon
- e-mail
- wiek
- wzrost
- waga
- kolor włosów

- kolor oczu
- rozmiar ubrania
- numer butów
- doświadczenie (wybierane od 1 do 5)
- płeć

Po wybraniu płci użytkownikowi zostaje wyświetlona dalsza część formularza w wersji dla kobiet albo dla mężczyzn, który umożliwia pobranie następujących informacji:

w przypadku kobiet:

- obwód biustu
- wielkość miseczki
- talia
- biodra
- długość nogi

w przypadku mężczyzn:

- klatka
- pas
- długość nogi

Ponieważ wszystkie pola są wymagane opracuje schemat sprawdzania pola w odpowiedniej funkcji. Sprawdzenie poprawności wpisanych wartości ma odbywać się dla wszystkich pól

Dla poniższych pól zastosuj podane reguły, dla reszty samodzielnie zdefiniuj reguły walidacji.

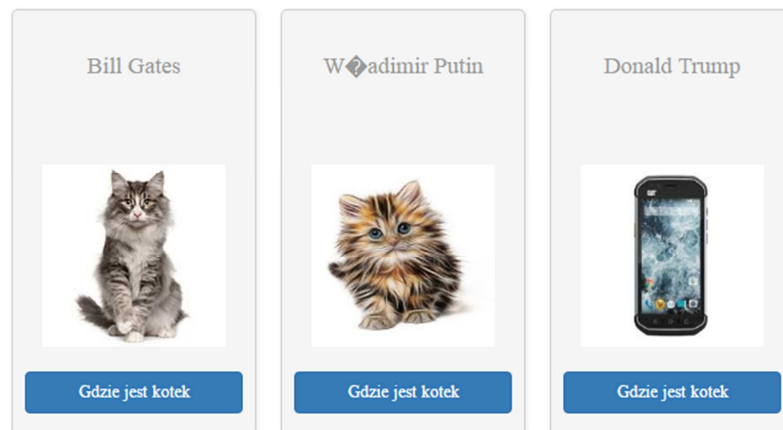
- E-mail (e-mail w formacie nazwa@serwer.domena)
- Kod pocztowy (kod odpowiedni dla naszego kraju)
- Wzrost – liczba w zakresie od 40 do 250 cm,
- Waga – liczba w zakresie od 20 do 80 kg

Wszystkie operacje sprawdzenia poprawności i wyświetlania elementów dla kobiet i mężczyzn mają być realizowane po stronie klienta w przeglądarce! Proszę zastosować odpowiednie wyrażenia regularne pozwalające na poprawną weryfikację danych. (W zadaniu należy zastosować zarówno elementy form z HTML5 + js)

Zadanie 2. Dynamiczne tworzenie zawartości strony.

W katalogu zad2 mamy stronę index.html wraz z plikiem do stylizacji oraz 3 zdjęcia. Proszę stworzyć plik main.js za pomocą którego wygenerujemy następującą zawartość strony. Dane należy zdefiniować bezpośrednio w pliku js w postaci tablicy obiektów.

Kto jest najważniejszy



Format tablicy jest następujący:

```
var vip = [  
  {  
    title: "Bill Gates",  
    where: "http://www.onet.pl",  
    url: "cat1.jpg"  
  },  
  {  
    title: "Władimir Putin",  
    where: "http://www.interia.pl",  
    url: "cat2.jpg"  
  },  
  {  
    title: "Donald Trump",  
    where: "http://www.wp.pl",  
    url: "cat3.jpg"  
  }  
];
```

Zadanie 3. Slajder zdjęć.

Napisz slajder zdjęć. Zdjęcia przesuwają się horyzontalnie po naciśnięciu na zdjęciu. (Zachownie podobne do obiektu typu caruzela w Bootstrap.). Materiały znajdziesz w katalogu zad3.

Przypomnienie informacji - bezpośrednio u źródła
www.jquery.com

lub w niezawodnym
<http://www.w3schools.com> w sekcji JQuery

Przypomnienie najważniejszych informacji:

Filozofia jQuery polega na znajdowaniu elementów w dokumencie a następnie odpowiednie działanie na tych elementach. Aby odnaleźć interesujące elementy na stronie używane są selektory.

Składnia jQuery jest bardzo prosta.

`$(selektor).action()`

gdzie:

\$ - znak dostępu do jQuery (można użyć także słowa "jQuery" w miejsce znaku "\$"),
selektor - służą do przeszukiwania dokumentu i znajdowania określonych części kodu,
action() - metoda dostępna w jQuery, operująca na elementach odpowiadających selektorowi.

Przykładowy kod jQuery:

```
$("#idElementu").toggle();
```

Selektory - służą do odnajdywania elementów strony.

Dostępne selektory:

selektory ogólne:

- wyszukuje elementy po id (element o podanym id powinien być tylko jeden w całym dokumencie, id powinno być unikalne),
. - wyszukuje elementy po klasie, (elementów jednej klasy może być więcej niż jeden),
* - wszystkie elementy,
this - obecny element,
div - wszystkie divy, (analogicznie inne znaczniki html).

filtry:

:first - pierwszy element,
:last - ostatni element,
:eq(indeks) - konkretny element (indeksowanie od 0),
:odd - nieparzyste elementy,
:even - parzyste elementy,
:contains(tekst) - element, zawierający podany tekst,
:hidden - elementy ukryte oraz pola typu `<input type="hidden"/>`,
:visible - elementy widoczne.

selektory formularzy:

:text - element input z atrybutem type, ustawionym na „text”:

<input type=„text”>

:submit - element input z atrybutem type, ustawionym na „submit”:

<input type=„submit”>

:checkbox - element input z atrybutem type, ustawionym na „checkbox”:

<input type=„checkbox”>

:radio - element input z atrybutem type, ustawionym na „radio”:

<input type=„radio”>

:input - wszystkie elementy typu input,

:checked - zaznaczone pola wyboru,

:selected - wybrane elementy opcji wyboru.

I wiele innych - więcej szczegółów -> <http://api.jquery.com/category/selectors/>

Akcje w JQuery umożliwiają pełne skanowanie oraz modyfikacje zawartości modelu DOM.

Metody do przechodzenia po elementach DOM

przodkowie (ancestors):

parent() - zwraca element rodzica,

parents() - zwraca wszystkich rodziców, aż do znacznika <html>,

parentsUntil() - zwraca wszystkich przodków aż do wybranego elementu.

potomkowie (descendants):

children() - zwraca tylko dzieci (tylko jeden poziom),

find() - zwraca wszystkich potomków.

rodzeństwo (siblings):

siblings() - zwraca całe rodzeństwo,

next() - zwraca kolejny element z rodzeństwa,

nextAll() - zwraca wszystkie kolejne elementy z rodzeństwa,

nextUntil() - zwraca kolejnych członków rodzeństwa aż do określonego elementu,

prev() - zwraca poprzedni element z rodzeństwa,

prevAll() - zwraca wszystkie poprzednie elementy z rodzeństwa,

prevUntil() - zwraca poprzednich członków rodzeństwa aż do określonego elementu.

Metody w JQuery do zarządzania zawartością strony.

text() - zwraca lub ustawia treść z/do określonego elementu,

html() - zwraca lub ustawia treść razem z kodem html z/do określonego elementu,

val() - zwraca lub ustawia treść z/do pola wejściowego input,

attr() - ustawia/zmienia wartość atrybutu,

append() - wstawia zawartość na końcu wybranego elementu,

prepend() - wstawia zawartość na początku wybranego elementu,

before() - wstawia zawartość przed wybranym elementem,

after() - wstawia zawartość po wybranym elemencie,
remove() - usuwa element razem z podelementami (child elements),
empty() - usuwa elementy dzieci z wybranego elementu,
addClass() - dodaje klasy do wybranych elementów,
removeClass() - usuwa klasy z wybranych elementów,
toggleClass() - usuwa lub dodaje elementy do wybranych elementów,
css() - ustawia lub zwraca style wybranych elementów.

Zadania startowe.

1. W katalogu zad4 znajduje się przykładowy kod. Proszę wykorzystując JQuery wykonać następujące modyfikacje:
 - a. Dla wszystkich do wszystkich elementów listy dodaj dodatkowy akapit <p> zawierający jakikolwiek tekst np. fragment „Lorum ipsum”
 - b. Zmień tło pierwszego elementu w każdej liście na yellow.
 - c. Zmień czcionkę w tekstach nie znajdujących się w liście
 - d. Wymień grafiki odnoszące się do krajów leżących w Ameryce na dowolną inną.
2. Zaprojektuj stronę z możliwością dynamicznej modyfikacji wyglądu - za pomocą przycisku przełączania stylów - tzw. Mechanizm skórki. Przełączanie powinno odbywać się z wykorzystaniem efektów wizualnych – dotyczy zdjęć, formularza, tekstu. Realizacja w oparciu o dynamiczną wymianę klas CSS przygotowanych wcześniej w pliku style.css
3. Dodawanie komentarzy na stronie. Mamy pole tekstowe, pole na nick oraz przycisk. Po naciśnięciu przycisku pojawia się nowy komentarz na stronie w sekcji komentarzy. Drugi przycisk usuwa komentarz – usuwany jest pierwszy komentarz na stronie
4. Stwórz interaktywną tabelę – z naprzemiennym kolorowaniem wierszy. Tabela powinna mieć możliwość dodawania, usuwania i modyfikacji wierszy (obok każdego wiersza umieszczony przycisk). Nagłówki mają być dodatkowo wyróżnione. W zależności od zawartości poszczególnych komórek (tekst czy wartość liczbową) inna wizualizacja komórki.
Podsumowanie:
 - o Dodawanie naprzemiennego kolorowania wierszy
 - o Użycie dla wierszy prostego efektu hover
 - o Wyróżnienie pierwszego i ostatniego wiersza tabeli
 - o Usuwanie wiersza przy użyciu selektora filtra
 - o Dodawanie wiersza po wierszu na podstawie jego wartości indeksu
 - o Usuwanie wiersza na podstawie jego wartości indeksu
 - o Dodawanie komunikatu po wierszach z określoną treścią
 - o Usuwanie wiersza na podstawie jego treści
 - o Sortowanie wierszy przy użyciu dodatku tablesorter
 - o zmiana domyślnej kolejności sortowania

XX

Efekt Parallaxy oraz zastosowanie animacji

Efekty parallax to obecnie jedne z częściej wykorzystywanych efektów na stronach internetowych. Najpopularniejsze odnoszą się do dwóch zastosowań:

1. Efekt tła dającego wrażenie trójwymiarowości
2. Efekt płynnego przewijania strony typu OnePage.

Ćwiczenie 1. Parallax background

1. Na początek dodajmy efekt tła. Przygotuj 3-4 zdjęcia o wymiarach 1600x600px, nadaj im nazwy *bg1*, *bg2*, *itd.* i zapisz je w katalogu ze stroną w folderze *img*.
2. W edytorze html otwórz plik *index.html*. Nad pierwszą sekcją `<section>` wstaw nową sekcję i nadaj jej klasę „*parallax parallax-1*”. To będzie nasz nasze zdjęcie parallax.

index.html

```
<section class="parallax parallax-1"></section>
```

3. Okienko parallax wstawione, trzeba je ostylewać. W *style.css* dodamy więc style umiejscawiające obraz w odpowiednim miejscu na stronie.

style.css

```
section.parallax {  
  height: 600px;  
  background-position: 50% 50%;  
  background-repeat: no-repeat;  
  background-attachment: fixed;  
  background-size: cover;  
}  
.parallax-1 {  
  background-image: url("../img/bg1.jpg") ;  
}
```

Zapisz i sprawdź jak to wygląda.

Zmodyfikuj parametry tła w *section.parallax*. Ustaw tło, które się:

- nie przewija: `background-attachment`;
- znajduje na środku sekcji: `background-position`;
- wypełnia całą sekcję: `background-size`
- nie powtarza.

4. W sekcji z parallaxem dodaj teraz nagłówek H1 z tytułem i odpowiednio go ostyluj.

5. Zmodyfikuj stronę w taki sposób, żeby po kliknięciu na odnośnik w menu strona przewijała się do obrazka, a nie sekcji poniżej – przeniesienie ID.

6. Analogicznie dodaj kolejne sekcje ze zdjęciami do poszczególnych działów strony.

Zadanie samodzielne

1. Wykorzystując *media_queries*

http://www.w3schools.com/css/css3_mediaqueries.asp i punkty przełomu Bootstrapa

<http://getbootstrap.com/css/#responsive-utilities> zmień rozmiary obrazków tła dla poszczególnych parallaxów w zależności od rozdzielczości ekranu.

Ćwiczenie 2. Parallax scrolling

W kolejnym zadaniu wykorzystamy bibliotekę *jQuery* do osiągnięcia efektu płynnego przewijania do poszczególnych sekcji strony. Wykorzystamy do tego metody:

- *scroll()*: http://www.w3schools.com/jquery/event_scroll.asp

- *scrollTop()* - odpowiedzialną za ustawienie dokumentu na określonej pozycji względem okna.

Można powiedzieć, że funkcja ta przewija dokument do zadanej w pikselach pozycji:

<https://api.jquery.com/scrollTop/>

Czyli, żeby móc płynnie przewinąć dokument np. do sekcji o *id* = „*contact*”, musimy wiedzieć, o ile pikseli od góry strony znajduje się ta sekcja.

- *offset()* - odległość sekcji względem góry strony <https://api.jquery.com/offset/>

`$ ('#contact').offset().top`

- *animate()* – animacje http://www.w3schools.com/jquery/jquery_animate.asp

Funkcja ta ma postać:

`$(selector).animate({params}, speed, callback);`

1. W katalogu *js* utwórz plik *parallax-scroll.js* i dołącz go do strony *index.html* pod załączoną biblioteką *jQuery*.

2. Na początek wpisujemy kod odpowiadający za uruchomienie jQuery oraz wyłączamy domyślne zachowanie odnośników z menu:

`parallax-scroll.js`

```
$(document).ready(function() {  
  $(".navbar-nav a").click(function(event) {  
    event.preventDefault();  
  });  
})
```

3. Pora na dodanie opcji płynnego przewijania do przycisków. Najprościej byłoby zdefiniować *id* dla każdego przycisku w menu i połączyć go z odpowiednią sekcją:

`index.html`

```
<a href="#contact" id="contactScroll">Kontakt</a>  
<section id="contact"></section>
```

a następnie dopisać do niego obsługę przewijania:

`parallax-scroll.js`

```
$(document).ready(function() {  
  $(".navbar-nav a").click(function(event) {  
    event.preventDefault();  
  });  
  $('#contactScroll').click(function() {  
    $('html, body').animate({  
      scrollTop: $('#contact').offset().top  
    }, 1000, 'swing');  
    return false;  
  });  
});
```

Jak to działa?

Na początek wybraliśmy odnośnik menu o *id* = „*contactScroll*” i zdefiniowaliśmy, co się ma dzieć po kliknięciu. Żeby uzyskać efekt płynnego przewijania użyliśmy na dokumencie funkcji do animacji *animate()* z opcją *swing*.

W naszym przypadku parametrem funkcji będzie właściwość *scrollTop*, której przypiszemy odległość elementu od góry dokumentu. Pobierzemy ją za pomocą metody *.offset().top*

Prędkość animacji ustawiliśmy na 1000ms. Natomiast jako funkcję *callback* wywołujemy parametr ‘*swing*’ dzięki czemu sekcje płynnie pływają.

4. Analogicznie dodaj obsługę dla pozostałych przycisków menu.

Zadanie samodzielne

Wykorzystane rozwiązanie ma pewne wady. Zauważ, że dla każdego przycisku musisz zdefiniować oddzielną obsługę zdarzenia, mimo że zmienia się tylko ID przycisku oraz ID warstwy. Lepiej byłoby, gdyby można było to zapisać w jeden uniwersalny sposób.

1. Zmodyfikuj skrypt funkcji tak, żeby był uniwersalny dla wszystkich elementów nawigacji.

Podpowiedź: wykorzystaj selector z atrybutem href.

Ćwiczenie 3. Animate.css

Wykorzystując bibliotekę animate.css <https://github.com/daneden/animate.css/> oraz nabytą wiedzę o JavaScript oraz jQuery dodaj do strony efekty animacji po najechaniu myszką na wybrane elementy: obrazki, logo, kolumny tekstu, nagłówki.

Zadanie dodatkowe

Korzystając z biblioteki animate.css dodaj własny skrypt js lub jQuery uruchamiający animacje w momencie, gdy znajdują się w obrębie ekranu

Podpowiedź: wykorzystanie metody scroll() oraz porównanie położenia dokumentu oraz elementu względem okna przeglądarki:

```
window.pageYOffset;  
$( 'element' ).offset().top
```

XX

Zadanie do dodania - domowe?

Napisz aplikację typu Quiz. Wyświetlane jest pytanie i 4 odpowiedzi. Na wybór użytkownik ma 20 sekund. Po upływie czasu lub wybraniu odpowiedzi pojawia się nowe pytanie. Na górze są dwa paski: pasek postępów (pokazujący przy którym pytaniu jesteś i jakie były wcześniejsze odpowiedzi: czerwone: złe, szare brak odpowiedzi, zielone – OK. Drugi pasek to pasek upływającego czasu odpowiedzi.

Po zakończeniu kwizu otrzymuje podsumowanie pokazując je w kontekście wcześniejszych odpowiedzi.