Liczba godzin: 21

1. Frameworki JavaScript
2. jQuery vs. JavaScript
3. Przegląd funkcjonalności jQuery
4. Selektory
5. Filtry podstawowe
6. Filtry atrybutów
7. Filtry treści, widoczności
8. Filtrowanie elementów formularza
9. Przemieszczanie się po dokumencie
10. Deklaracje łańcuchowe
11. Manipulacja treścią strony
12. Manipulacja atrybutami
13. Opakowywanie, zastępowanie, usuwanie treści
14. Praca z CSS
15. Obsługa zdarzeń
16. Efekty
17. AJAX
18. Implementacja pluginów dostępnych w sieci
19. Pisanie własnych pluginów
20. Co to jest biblioteka jQuery?

# Spis treści

1. Co to jest jQuery i do czego służy
2. Osadzanie biblioteki jQuery w pliku HTML
3. Podział dokumentu HTML na warstwy
4. Selektory CSS
5. Zdarzenia HTML
6. Modyfikacja wyglądu elementów
7. Odczyt i modyfikacja treści elementów
8. Odczyt i modyfikacja atrybutów
9. Dodawanie i usuwanie węzłów drzewa DOM

## 1.Co to jest jQuery i do czego służy

jQuery jest biblioteką dla języka JavaScript, której pierwsze wydanie pojawiło się 26 sierpnia 2006 roku. Kosztem niewielkiego spadku wydajności(praktycznie niezauważalnego) ułatwia korzystanie i tworzenie efektów animacji, dodawanie dynamicznych zmian strony, wczytywanie elementów strony bez potrzeby przeładowania(AJAX) itd.

Od 18 kwietnia 2013(wersja 2.0) porzucono obsługę przestarzałych przeglądarek IE 6-8.  
Najnowsza stabilna wersja jQuery oznaczona jest numerem 3.5.1(<https://jquery.com/>).

jQuery umożliwia korzystanie z takich funkcjonalności jak:

* selektorów
* atrybutów
* modyfikowaniem drzewa DOM
* zmieniania stylów elementów
* obsługa zdarzeń
* efektów
* AJAX

## 2. Osadzanie biblioteki jQuery w pliku HTML

Plik z biblioteką jQuery załączamy do strony za pomocą elementu <script>.

*<script type=’text/javascript’ src=’jquery.js’></script>*  
  
Bibliotekę możemy dołączyć bezpośrednio do plików projektu lub jako CDN ze źródła zewnętrznego.

**Przykład**

* **$(document).ready()**
* **$(window).load()**

|  |
| --- |
| **Zadanie 1 (5 minut)**  Napisz skrypt jQuery, który po załadowaniu **kompletnej** strony WWW wyświetli komunikat powitalny “Witaj”. Do wyświetlenia komunikatu użyj funkcji alert(). Pamiętaj o zastosowaniu funkcji o nazwie **$** z parametrem **document.** Wynikiem będzie obiekt, na którym należy zastosować metodę **ready().**  **$(document).ready(function(){** *komunikat* **})** |

|  |
| --- |
| **Zadanie 2 (5 minut)**  Wykonaj zadanie 1 za pomocą skróconej notacji do obsługi zdarzeń.  **$(function(){** *komunikat* **});** |

## 3. Podział dokumentu HTML na warstwy

HTML + CSS + JS

**Przykład**

|  |
| --- |
| **Zadanie 3 (5 minut)**  Napisz skrypt jQuery(plik zewnętrzny, dołączony do pliku HTML), który spowoduje, że po kliknięciu akapitu *Lorem ipsum* zostanie wyświetlone okno informacyjne z komunikatem *Witaj*.  **$(‘p’).click(function(){** *komunikat* **});** |

|  |
| --- |
| **Zadanie 4 (5 minut)**  Wykonaj powyższe zadanie bez użycia jQuery. |

|  |
| --- |
| **Zadanie 5 (10 minut)**  Przygotuj dokument HTML zawierający listę ul/li. Oprogramuj zdarzenie onclick elementów listy. Kliknięcie elementu ma powodować wyświetlenie komunikatu. Zadanie rozwiąż, przygotowywując trzy pliki: index.html, styles.css, scripts.js. |

|  |
| --- |
| **Zadanie 6 (5 minut)**  Wykonaj powyższe zadanie bez zastosowania jQuery. |

## 4. Selektory CSS

W celu pracy z biblioteką będziemy wskazywać elementy strony, które będziemy chcieli poddać jakiejś interakcji/modyfikacji.   
Najczęściej będziemy stosować:

* selektor elementu,
* selektor identyfikatora,
* selektor klasy.

W pierwszym przykładzie będziemy wskazywać znacznik elementu. W miejsce w cudzysłowiu możemy wstawić każdy znacznik, który będziemy chcieli wskazać do dalszej pracy z nim: $(‘div’)

W celu odwołania się do elementu po jego ID: $(‘#nazwaid)

Natomiast do modyfikacji elementu oznaczonego klasą: $(‘.nazwaklasy’).

Inne selektory, które możesz spotkać to:

* [data-info] - selektor atrybutu
* [data-info=’true’] - selektor atrybutu o podanej dokładnej wartości
* [lang|=’en’] - selektor języka
* \* - selektor uniwersalny
* selektor1, selektor2, selektor3 - selektor grupowy

**Przykład**

* (target=’\_blank’)
* metoda find()

## 5. Zdarzenia HTML

* onclick,
* ondblclick,
* onmousedown,
* onmouseup,
* onmouseover,
* onmousemove,
* onmouseout,
* onkeypress,
* onkeydown,
* onkeyup.

Dla każdego z tych zdarzeń w jQuery zaimplementowano metody o takiej samej nazwie, ale pozbawione przedrostka ‘on-’:

* click(),
* dblclick(),
* mousedown(),
* mouseup(),
* mouseover(),
* mousemove(),
* mouseout(),
* keypress(),
* keydown(),
* keyup().

**Przykład**

|  |
| --- |
| **Zadanie 7 (15 minut)**  Na bazie wcześniejszych szablonów stwórz dokument z kilkoma akapitami(<p>), przyciskami(<button>), oraz polami polami formularza (<input>). Każdemu z elementów dodaj klasy lub identyfikatory i przetestuj działanie każdego z powyższych zdarzeń. |

|  |
| --- |
| **Zadanie 8 (10 minut)**  Stwórz dokument HTML, który zawiera trzy elementy **div**, jeden z nich jest oznaczony identyfikatorem *#tresc*, drugi jest oznaczony klasą .*klik* a trzeci posiada atrybut *data-info.*  Każdemu z selektorów przypisz inne zdarzenie wywołujące różne komunikaty. |

## 6. Modyfikacja wyglądu elementów

Podstawową funkcją umożliwiającą modyfikowanie wyglądu elementów jest funkcja css().

*$(‘.nazwaKlasy’).css(‘background’, ‘green’);  
$(‘p’).css(‘background’, $(‘.red’).css(‘color’));*

Kolejnym sposobem modyfikacji jest dodawanie klasy za pomocą metody addClass().  
  
*$(‘.nazwaKlasy’).addClass(‘alert’);  
  
.alert {*

*color: red;*

*}*

Odwrotnym w działaniu do poprzedniego sposobu jest metoda removeClass().

$(‘.nazwaKlasy’).removeClass(‘alert’);

**Przykład**

* metoda css()
* metoda addClass()
* metoda removeClass()
* metoda toggleClass()

|  |
| --- |
| **Zadanie 9 (15 minut)**  Dokument HTML zawiera akapit tekstu oraz trzy kwadraty w kolorach: czerwony, zielony oraz niebieski. Kolorowe kwadraty zostały wykonane jako elementy span ozdobione odpowiednimi stylami CSS. Napisz skrypt jQuery, który oprogramuje zdarzenie onclick w taki sposób, by po naciśnięciu kolorowego kwadratu jego kolor został przypisany jako kolor tła akapitu. |

|  |
| --- |
| **Zadanie 10 (10 minut)**  Rozwiąż powyższe zadanie jeszcze raz, z tą różnicą aby zastosować zdarzenie mouseover() dla zmiany koloru akapitu oraz mouseout() aby na powrót ustawić domyślny kolor tła(np. biały). |

|  |
| --- |
| **Zadanie 11 (10 minut)**  Stwórz dokument HTML z przyciskiem rozwijającym menu w wersji mobilnej.  Ps: Pamiętaj o rozdzieleniu HTML, CSS i JS na oddzielne pliki. |

## 7. Odczyt i modyfikacja treści elementów

Dostęp do treści elementów HTML można uzyskać po zastosowaniu metody text() oraz html(). Pierwsza z nich operuje tylko tekstem, natomiast druga kodem HTML. Mogą one być wywoływane bezparametrowo lub z jednym parametrem. Wywołanie bezparametrowo zwraca obecną zawartość elementu, zaś wywołanie z parametrem ustawia nową zawartość.

*$(‘selektor’).text()* - zwraca tekst danego elementu.

*$(‘selektor’).text(‘Lorem ipsum...’)* - dodaje tekst do wskazanego elementu.

**Przykład**

* **tablica**
* **pętla for**

|  |
| --- |
| **Zadanie 12 (10 minut)**  Przygotuj dokument HTML, który będzie zawierać pusty akapit(<p>) o id=”tresc”. Stwórz skrypt jQuery, który element #tresc wypełni dowolną treścią. |

|  |
| --- |
| **Zadanie 13 (5 min)**  Zmodyfikuj powyższy skrypt tak aby wykorzystać element <button> oraz metodę click() do wypełnienia elementu #tresc. |

|  |
| --- |
| **Zadanie 14 (20 minut)**  Dokument HTML zawiera element div o identyfikatorze #names. W kodzie JS stwórz zmienną names będącą tablicą zawierającą imiona. Stwórz skrypt, który doda do div#names tabelę z listą imion. |

## 8. Odczyt i modyfikacja atrybutów

Elementy HTML mogą mieć różne atrybuty:

* <div id=’tresc’></div>
* <p class=’opis’></p>
* <a href=’lorem.html’>lorem</a>
* <img src=’obrazek.jpg’ alt=’opis obrazka’ />

Dostęp do atrybutów zapewnia metoda attr(). Wywołana z jednym parametrem zwraca wartość, zaś z dwoma parametrami ustala podaną wartość atrybutu.

Wykorzystując metodę attr(), możemy:

* wymienić obrazy wyświetlane przez znacznik img,
* opracować wskaźnik wybranej opcji zależny od adresu URL dokumentu,
* na bieżąco odczytywać treści wpisywane do formularzy,
* modyfikować działanie linków tak, by powodowały otwieranie nowego okna.

|  |
| --- |
| **Zadanie 15 (5 minut)**  Stwórz dokument ze znacznikiem img, po kliknięciu którego będzie podmieniane zdjęcie na inne. |

|  |
| --- |
| **Zadanie 16 (5 minut)**  Stwórz w tekście kilka linków z atrybutem target=’blank’. Stwórz skrypt, który doda atrybut title z informacją o otwarciu takich linków w nowej karcie. |

|  |
| --- |
| **Zadanie 17 (20 minut)**  Stwórz dokument z 3 zakładkami(tabami). Ostyluj je a następnie stwórz skrypt do przełączania się pomiędzy nimi. |

## 9. Dodawanie i usuwanie węzłów drzewa DOM

Poza wspomnianymi wcześniej metodami text() oraz html() istnieje wiele innych metod do tworzenia nowych węzłów w drzewie DOM.

* append()
* appendTo()
* prepend()
* prependTo()
* after()
* insertAfter()
* before()
* wrap()
* wrapInner()

Metody append() oraz appendTo(), a także prepend() i prependTo() dodają do drzewa DOM nowo utworzony element. Metoda append() oraz appendTo() umieszczają nowy element drzewa jako ostatnie dziecko podanego elementu, zaś metody prepend() i prependTo() dodają element jako pierwsze dziecko elementu.

|  |
| --- |
| **Zadanie 18 (20 minut)**  Sprawdź w praktyce użycie każdej z wyżej wymienionych metod. |