

## TECHNIKI WWW (WFAIS.IF-C125)

(zajęcia 30.11.2016 r.)

### 1. Biblioteka jQuery ([www.jquery.com](http://www.jquery.com)) - wprowadzenie

Biblioteka napisana w języku JavaScript o "lekkim" charakterze, obsługująca przestrzenie nazw z mechanizmem łatwej rozszerzalności umożliwiającą manipulowanie elementami struktury DOM (Document Object Model). Najnowsze wydanie stabilne:

jQuery 2.2.4 (20 maj 2016)

jQuery 3.1.1 (22 wrzesień 2016)

Biblioteka występuje:

#### V. 2.2.4

- normalnej - <http://code.jquery.com/jquery-2.2.4.js>
- skompresowanej - <http://code.jquery.com/jquery-2.2.4.min.js>

#### V.3.1.1

- normalnej - <http://code.jquery.com/jquery-3.1.1.js>
- skompresowanej - <http://code.jquery.com/jquery-3.1.1.min.js>

Dla zwiększenia wydajności aplikacji lepiej używać wersji (b). Opcjonalnie można używać wersji udostępnianej na zewnętrznym serwerze np.

<https://ajax.googleapis.com/ajax/libs/jquery/2.2.4/jquery.min.js>

Aby dodać funkcjonalności biblioteki jQuery do naszej aplikacji, musimy ją tą bibliotekę jako skrypt używając znaczników <script>:

```
<!DOCTYPE html>
<html>
<head>
<title>Pierwsza strona z jQuery</title>
<script type="text/javascript" src="jquery-2.2.4.min.js"></script>
</head>
<body>
  <h1>Testujemy bibliotekę jQuery</h1>
</body>
</html>
```

Pierwszą funkcją ją możemy napisać jest wyświetlenie komunikatu:

```
<head>
<title>Pierwsza strona z jQuery</title>
<script type="text/javascript" src="jquery.2.4.1.min.js"></script>
<script type="text/javascript">
  $(document).ready(function() {
    alert('Pierwsza operacja jQuery');
  });
</script>
</head>
```

Zaprezentowana konstrukcja:

```
$(document).ready( );
```

tworzy nowy obiekt na podstawie argumentu "document" i wykonuje na nim metodę "ready()". Jest to skrótowy zapis składni jQuery gdzie znak "\$" wskazuje na przestrzeń nazw "jQuery". Konstrukcja ta w wersji pełnej wygląda następująco:

Dodatkowo metoda "ready()" jest tak naprawdę funkcją obsługi zdarzenia "ready", oznaczająca załadowanie całego dokumentu hipertekstowego. Natomiast funkcja która jest przekazywana jako argument funkcji "ready()" jest tzw. funkcją anonimową, (bez nazwy) i w tym konkretnym wypadku pełni rolę wywołania zwrotnego tzw. "callbacka". Jako że większość, metod jQuery chcemy wykonywać po załadowaniu dokumenty hipertekstowego można tą konstrukcję skrócić jeszcze bardziej do:

```
$(function() {  
    alert('Pierwsza operacja jQuery');  
});
```

Funkcja ta realizuje proste wyświetlanie komunikaty "alert()".

## 2. Selektory i zdarzenia w jQuery

Biblioteka jQuery wspiera operacje pozwalające na manipulowanie strukturą drzewa DOM. Aby operacje te mogły być wykonywane musimy określić które elementy drzewa DOM mają zostać zmodyfikowane. W jQuery robimy to stosując te same selektory co w przypadku arkuszy stylu CSS. Przykłady:

```
$( 'div' )      wybiera <div> </div>  
$( 'h1' )      wybiera <h1> </h1>  
  
$( '#tresc' )  wybiera <div id="tresc"> </div>  
$( '.tresc' ) wybiera <div class="tresc"> </div>
```

Zatem można poruszać się po drzewie wykorzystując te same własności co w CSS. Dodatkowo obowiązują te same zależności dziedziczenia pomiędzy selektorami co w przypadku CSS.

Poza selektorami dla w jQuery możemy używać tych samych funkcji obsługi zdarzeń co w normalnym języku JavaScript umożliwiając aplikacji reagowanie na działania podejmowane przez użytkownika. Najpopularniejszymi zdarzeniami są: "click()", "mouseover()", "mouseout()".

Przykład zastosowania selektora i zdarzenia click:

```
<script type="text/javascript">  
    $(function() {  
        $( '#button' ).mouseover(function() {  
            alert("Klikam w link do strony UJ");  
        });  
    });  
</script>
```

To pokazuje jak w łatwy sposób można pisać sterowniki modyfikujące zawartość DOM.

### 3. Modyfikacja wyglądu strony za pomocą CSS

W jQuery możemy wpływać na wygląd styli CSS używając metody “css()” na dowolnym elemencie. Metoda ta przyjmuje jako dwa argumenty: cechę i wartość cechy którą chcemy nadać elementowi DOM (cecha i wartość muszą być podane jako string). Przykład:

```
<script type="text/javascript">
    $(function() {
        $('span').css('background', '#CCC000');
    });
</script>
```

Zatem nadaliśmy cechy pojedynczemu elementowi. Możliwe jest również odczytywanie wartości cech CSS elementów i nadawanie ich innym elementom:

```
<script type="text/javascript">
    $(function() {
        $('#div#tresc').mouseover(function() {
            $('p').css('background', $(this).css('background-color'));
        }).mouseout(function() {
            $('p').css('background', 'white');
        });
    });
</script>
```

Ważne aby odczytywana cechy była cechą JEDNOWARTOŚCIOWĄ. Pojawiające się w tym kontekście słowo kluczowe “this” wskazuje na element drzewa który wywołał funkcję obsługi zdarzenia “mouseover()”.

Możemy poszczególnym elementom nadawać cechy CSS dodając ich do odpowiednich klas lub usuwając je z nich:

```
<style type="text/css">
    .wazny { color: red; }
</style>
```

```
<script type="text/javascript">
    $(function() {
        $('li').mouseover(function() {
            $(this).addClass('wazny');
        }).mouseout(function() {
            $(this).removeClass('wazny');
        });
    });
</script>
```

Zatem możemy przygotować w CSS odpowiednie definicje reguł i w zależności od sytuacji nadawać je elementom.

W powyższym przykładzie widać również ważną cechę jQuery, tzw. “łańcuch wywołań” (omówiliśmy go na wykładzie)>

## 4. Ukrywanie elementów DOM

W jQuery możemy wykorzystać trzy metody do ukrywania i pokazywania elementów: “hide()”, “show()”, “toggle()”. Przykłady użycia:

```
$(function() {
  $('#button#hide').click(function() {
    $('p').hide();
  });
  $('#button#show').click(function() {
    $('p').show();
  });
});
```

To samo na jednym elemencie można uzyskać za pomocą metody toggle:

```
$(function() {
  $('#button').click(function() {
    $('#span').toggle();
  });
});
```

## 4. Modyfikacja elementów DOM

jQuery został stworzony do modyfikacji elementów DOM. Dwom najprostszymi operacjami jakie chcemy wykonywać jest dodawanie tekstu i nowych znaczników.

```
$(function() {
  $('#div#tresc1').text('Nowa treść wpisana dynamicznie w
jQuery');
  $('#div#tresc2').html('Nowa treść wpisana dynamicznie w
jQuery');
});
```

Różnica polega na tym że metoda “text()” sparsuje wszystkie znaki specjalne i zamieni je na encje HTML.

```
$(function() {
  $('#div#tresc1').text('<span> To jest treść</span>');
  $('#div#tresc2').html('<span> To jest treść</span>');
});
```

Proszę przetestować działanie obu metod.

## 5. Modyfikacja atrybutów elementów DOM

jQuery możemy również manipulować atrybutami konkretnych elementów DOM, służy do tego metoda “attr()”. Przykład zastosowania:

```
$(function() {
  $('img').attr('src', 'obrazek.png');
});
```

Zadanie:

Proszę przygotować dokument HTML w którym będzie lista nienumerowana `<ul>` z kilkoma pozycjami. Następnie proszę przygotować skrypt który będzie reagował na zdarzenie `click()` na elemencie listy i w wyniku wywołania zwrótnego usuwał daną pozycję listy.