



# Technologie internetowe

## Wprowadzenie do jQuery

Paweł Rajba

[pawel@cs.uni.wroc.pl](mailto:pawel@cs.uni.wroc.pl)

<http://pawel.ii.uni.wroc.pl/>

# Plan wykładu

- Wprowadzenie
- Składnia
- No conflict
- Selektory
- Zdarzenia
- Efekty
- Operowanie na HTML i CSS

# Wprowadzenie

- Co to jest i dlaczego jQuery?
  - Biblioteka JavaScript
  - Łatwa do nauczenia
  - Łatwa w użyciu
  - Znacznie upraszcza programowanie w JavaScript
  - Jest mnóstwo już napisanego kodu w jQuery, który
    - można dalej rozwijać
    - dzięki różnorodności pozwala zachować spójność i całość rozwiązania oprzeć tylko na jednej bibliotece (jQuery)
  - Strona domowa: <http://jquery.com/>

# Wprowadzenie

- Skrypt jQuery osadzamy wskazując
  - lokalizację lokalną
  - lokalizację na jednym z serwerów CDN
- Korzystamy przez zmienną: jQuery lub \$
- Niniejsze wprowadzenie na podstawie:
  - <http://learn.jquery.com/>
  - <http://api.jquery.com/>
  - <http://www.w3schools.com/jquery/>

# Wprowadzenie

- Gdzie jest void main()?
  - `$(document).ready(function() { ... });`
  - `$(window).load(function() { ... });`
    - `window.onload = function() { ... };`
  - `ready()` vs `window.onload`
    - `ready()` – tylko DOM (bez obrazków, ramek, itd.)
    - `window.onload` – wszystko
- Ważne, aby użyć jednego z powyższych
  - Pozwala uniknąć operowania na jeszcze nieistniejących obiektach

# Składnia

- Podstawowa składnia:  
`$(selektor).akcja1().akcja2().akcjaN()`
  
- **Przykład:** `hello-jquery.html`

# No conflict

- Do korzystania z funkcjonalności jQuery korzystamy z aliasu \$
  - jest krótki i wygodny
  - jednak czasami niepożądany, gdy korzystamy z innych bibliotek również korzystających z tego skrótu.
  - Rozwiązanie: możemy użyć mechanizmu nonConflict
- **Przykład:** noconflict.html
- Pod adresem:  
<http://api.jquery.com/jquery.noConflict/>  
mamy szereg jeszcze innych konstrukcji tego mechanizmu

# Selektory

- Ich rola jest analogiczna jak np. w CSS: służą do wyszukiwania odpowiednich węzłów w drzewie DOM
- Składnia jest podobna do tej używanej w CSS i XPath
- Kilka przykładów:
  - Selektory typu element:
    - \$("p")
    - \$("p.intro")
    - \$("#demo")



# Selektory

- Przykłady c.d.
  - Selektory typu atrybut
    - \$("[href]")
    - \$("[href='#']")
    - \$("[href!='#']")
    - \$("[href\$='.jpg']") [wartość kończy się na .jpg]
  - Selektor typu CSS  
(pozwala on na zmianę wartości właściwości)
    - \$("p").css("background-color","yellow");

# Traversing

- Mając wybrane elementy przez selektor, możemy je dalej przeglądać, filtrować, itp.
  - `each()` – iteracja po elementach
  - `filter()` – filtruje bieżącą listę
  - `find()` – wyszukuje podelementów w DOM
  - `end()` – wyłącza filtr i przywraca oryginalną listę elementów
  - `first()`, `next()`, etc.

# Selektory & Traversing

- **Przykład**

- selectors-traversing.html

- **Więcej**

- selektorów: [W3Schools](#), [jQuery Docs](#)
- o ich przeglądaniu (traversing): [jQuery Docs](#)

# Operowanie na HTML i CSS

- Mamy konstrukcje:
  - `$(selector).append(content)`,
  - `$(selector).prepend(content)`
  - `$(selector).after(content)`,
  - `$(selector).before(content)`
    - Append vs. After:  
<https://stackoverflow.com/questions/14846506/append-prepend-after-and-before>
  - `$(selector).css(name)`
  - `$(selector).css(name,value)`
  - `$(selector).css({properties})`
  - `$(selector).height(value)`
  - `$(selector).width(value)`

# Operowanie na HTML i CSS

- Przykłady:
  - `$("p").css("background-color","yellow");`
  - `$("p").css({  
    "background-color":"yellow",  
    "font-size":"200%"});`
  - `$("#div1").height("200px");`
- Więcej:
  - [http://www.w3schools.com/jquery/jquery\\_ref\\_html.asp](http://www.w3schools.com/jquery/jquery_ref_html.asp)

# Dane strukturalne

- Funkcja `.data()`

- [.data\( key, value \)](#)
  - [.data\( key, value \)](#)
  - [.data\( obj \)](#)
- [.data\( key\\_ \)](#)
  - [.data\( key\\_ \)](#)
  - [.data\(\)](#)

- Specyfikacja:

- <https://api.jquery.com/data/>

# Zdarzenia

- Dla wybranego elementu umożliwiają „podpięcie” funkcji pod pewną akcję
- Pojęcia bubbling i capturing/trickling
  - Do zapamiętania: bubble up, trickle down
- **Przykład**
  - **bubbling.html**
- Stackoverflow
  - <https://stackoverflow.com/questions/4616694/what-is-event-bubbling-and-capturing>

# Zdarzenia

- Najprościej skorzystać z następującej konstrukcji:
  - `$("button").click(function() { ... } )`
- Przykładowe zdarzenia:
  - `$(selector).click(function)`
  - `$(selector).blur(function)`
  - `$(selector).dblclick(function)`
  - `$(selector).focus(function)`
  - `$(selector).mouseover(function)`
  - `$(selector).keydown(function)`
- Więcej zdarzeń pod adresami:
  - [W3Schools](#), [jQuery Docs](#)
  - Przeglądamy kilka zdarzeń z dokumentacji



# Zdarzenia

- Bardziej ogólnie, podpinanie funkcji pod zdarzenia jest realizowane przez `on()` i `off()`
- W starszych wersjach `bind()` i `unbind()`
- Przykłady
  - `$(selektor).on('click',function() { ... });`
  - `$(selektor).off();`
  - `$(selektor).off('click');`
  - `$(selektor).on('mouseenter mouseleave',  
function() {  
 $(this).toggleClass('entered');  
});`

# Zdarzenia

- Jaki jest problem w następującym przypadku:
  - Mamy tablicę:

```
<table>  
<tr><td>Jan</td></tr>  
<tr><td>Albert</td></tr>  
</table>
```
  - Do każdego tr podpięte jest zdarzenie *click*
  - Za pomocą skryptu dodajemy:

```
<tr><td>Gerwazy</td></tr>
```

# Zdarzenia

- Typy zdarzeń: direct i delegated
  - <https://stackoverflow.com/questions/8110934/direct-vs-delegated-jquery-on>
- Obsługa delegated
  - Wykorzystywany jest mechanizm bubbling
  - Realizacja przez on()
  - Dawniejsze wersje
    - live() i die()
    - delegate() i undelegate()

# Zdarzenia

- Oba warianty można zrealizować za pomocą odpowiednich składni on()

```
1 | $( "#dataTable tbody tr" ).on( "click", function() {  
2 |     console.log( $( this ).text() );  
3 | });
```

```
1 | $( "#dataTable tbody" ).on( "click", "tr", function() {  
2 |     console.log( $( this ).text() );  
3 | });
```

- jQuery Maps

```
<script>  
$( "div.test" ).on({  
    click: function() {  
        $( this ).toggleClass( "active" );  
    }, mouseenter: function() {  
        $( this ).addClass( "inside" );  
    }, mouseleave: function() {  
        $( this ).removeClass( "inside" );  
    }  
});  
</script>
```

- Więcej rozwiązań [na stronach jQuery](#)

# Zdarzenia

- Ciekawa funkcja: hover

```
1 | $( "td" ).hover(  
2 |     function() {  
3 |         $( this ).addClass( "hover" );  
4 |     }, function() {  
5 |         $( this ).removeClass( "hover" );  
6 |     }  
7 | );
```

- Można też przekazać jedną funkcję, która będzie wywoływana przy obu zdarzeniach

# Zdarzenia

- Case study: podpinanie zdarzeń w pętli
- Przykład:
  - <https://jsfiddle.net/yg9vLw71/39/>
  - W plikach na stronie: data.html
- Do przeczytania:
  - <https://stackoverflow.com/questions/46027262/functions-declared-within-loops-referencing-an-outer-scoped-variable-may-lead-to>
  - <https://stackoverflow.com/questions/750486/javascript-closure-inside-loops-simple-practical-example>

# Efekty

- jQuery ma zaimplementowany szereg różnych efektów. Przykładowe:
  - Zwykłe pojawianie się i znikanie
    - `$(selector).hide(speed,callback)`
    - `$(selector).show(speed,callback)`
    - `$(selector).toggle(speed,callback)`
  - Zwijanie i rozwijanie
    - `$(selector).slideDown(speed,callback)`
    - `$(selector).slideUp(speed,callback)`
    - `$(selector).slideToggle(speed,callback)`

# Efekty

- Przykłady c.d.
  - Przenikanie
    - `$(selector).fadeIn(speed,callback)`
    - `$(selector).fadeOut(speed,callback)`
    - `$(selector).fadeToggle(speed,callback)`
    - `$(selector).fadeTo(speed,opacity,callback)`
  - Wartości w miejsce speed: "slow", "fast", "normal" lub liczba milisekund



# Efekty

- **Przykład:**
  - effects.html
- **Więcej efektów:**
  - <http://api.jquery.com/category/effects/>
  - [http://www.w3schools.com/jquery/jquery\\_ref\\_effects.asp](http://www.w3schools.com/jquery/jquery_ref_effects.asp)

# JQuery Source Viewer

- DEMO
  - Google → „jQuery Source Viewer”

# Biblioteka jQueryUI

- Strona domowa
  - [www.jqueryui.com](http://www.jqueryui.com)
- Przeglądamy możliwości
  - Styles & Themes, [ThemeRoller](#)
  - Interakcje
  - Widżety, każdy ma: Options, Events, Methods
  - Efekty