



# Technologie internetowe

## Wprowadzenie do jQuery

Paweł Rajba

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

<http://itcourses.eu/>

# Plan wykładu

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

# Wprowadzenie

- JSON: format wymiany danych

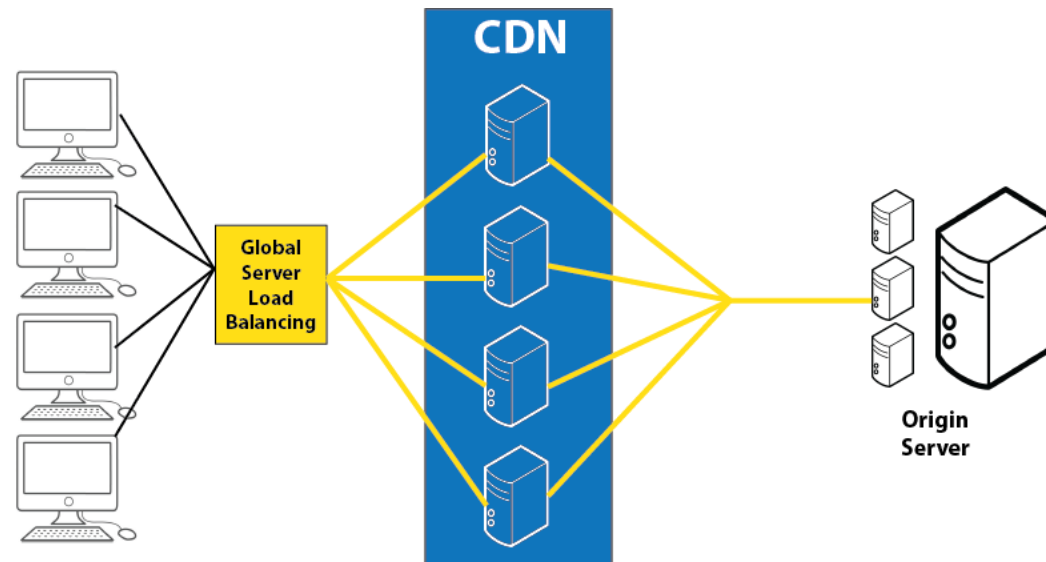
```
    {"menu": {  
      "id": "file",  
      "value": "File",  
      "popup": {  
        "menuitem": [  
          {"value": "New", "onclick": "CreateNewDoc()"},  
          {"value": "Open", "onclick": "OpenDoc()"},  
          {"value": "Close", "onclick": "CloseDoc()"}  
        ]  
      }  
    }  
  }  
}
```

- WWW:

- <http://www.json.org/>
- <http://pl.wikipedia.org/wiki/JSON>
- <http://tools.ietf.org/html/rfc4627>

# Wprowadzenie

- CDN, czyli Content Delivery Network



- Ciekawe [wprowadzenie](#)
- Biblioteki na serwerach: [Microsoft](#), [Google](#)

# 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)

# Wprowadzenie

- Skrypt jQuery osadzamy wskazując
  - lokalizację lokalną
  - lokalizację na jednym z serwerów CDN
- Korzystamy przez zmienną: jQuery lub \$
- Niniejsze wprowadzenie na podstawie:
  - [http://docs.jquery.com/Tutorials:Getting\\_Started\\_with\\_jQuery](http://docs.jquery.com/Tutorials:Getting_Started_with_jQuery)
  - <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
- Ciekawe [podsumowanie](#)

# 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 odo 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)`
    - dopisuje za/przed każdym znalezionym elementem
  - `$(selector).after(content)`,
  - `$(selector).before(content)`
    - dopisuje za/przed wszystkimi znalezionymi elementami
  - `$(selector).css(name)`
  - `$(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)
- [http://www.w3schools.com/jquery/jquery\\_ref\\_css.asp](http://www.w3schools.com/jquery/jquery_ref_css.asp)

# Zdarzenia

- Dla wybranego elementu umożliwiają „podpięcie” funkcji pod pewną akcję
- Pojęcia bubbling i capturing/trickling
  - Do zapamiętania: bubble up, trickle down
- **Przykład**
  - **bubbling.html**



# 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

- Funkcje `live()` i `delegate()`
- `live()` i `die()`
  - Zdarzenie podpięte pod Document Object
  - Wykorzystywany jest mechanizm bubbling
  - Deprecated – usunięte w wersji 1.9
  - `$('#div').live('click', func);`
- `delegate()` i `undelegate()`
  - Nowsza wersja `live()`, dodana w 1.4
  - Nowym elementem jest określenie kontekstu
  - `$('#OsobyTable').delegate('tr', 'click', func);`

# Zdarzenia

- Wszystkie 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

- Na koniec 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

# 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