Zaawansowane Techniki WWW
(HTML, CSS i JavaScript)

Dr inż. Marcin Zieliński

Środa 15:30 - 17:00 sala: A-1-04

WYKŁAD 7

Wykład dla kierunku: **Informatyka Stosowana II rok**

Rok akademicki: **2015/2016 - semestr zimowy**

Przypomnienie z poprzedniego wykładu

UNIWERSYTET JAGIELLOŃSKI W KRAKOWIE

JavaScript jako język zdarzeniowy

Format JSON vs. XML

Asynchroniczność w JS

Dziedziczenie prototypowe w JS

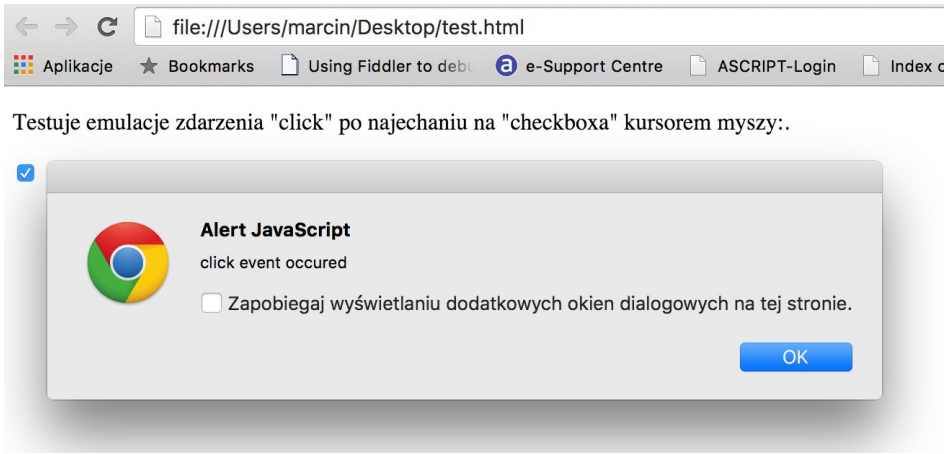
UNIWERSYTET JAGIELLOŃSKI W KRAKOWIE

Obiekty i zdarzenia na stronie

Przykład emulowania zdarzenia “click”:

```
<form>
  <input type="checkbox" id="myCheck" onmouseover="myFunction()" onclick="
alert('click event occured')">
</form>

<script>
function myFunction() {
    document.getElementById("myCheck").click();
}
</script>
```



Porównanie JSON vs. XML

```
{ "samochod": [
    {
        "Marka": "VW",
        "Model": "Golf",
        "Rocznik": 1999
    },
    {
        "Marka": "BMW",
        "Model": "S6",
        "Rocznik": 2007
    },
    {
        "Marka": "Audi",
        "Model": "A4",
        "Rocznik": 2009
    }
]}
```

```
<?xml version="1.0" encoding="utf-8"?>
<KOMIS>
  <Samochod>
    <Marka>VW</Marka>
    <Model>Golf</Model>
    <Rok>1999</Rok>
  </Samochod>
  <Samochod>
    <Marka>BMW</Marka>
    <Model>S6</Model>
    <Rok>2007</Rok>
  </Samochod>
  <Samochod>
    <Marka>Audi</Marka>
    <Model>A3</Model>
    <Rok>2009</Rok>
  </Samochod>
</KOMIS>
```

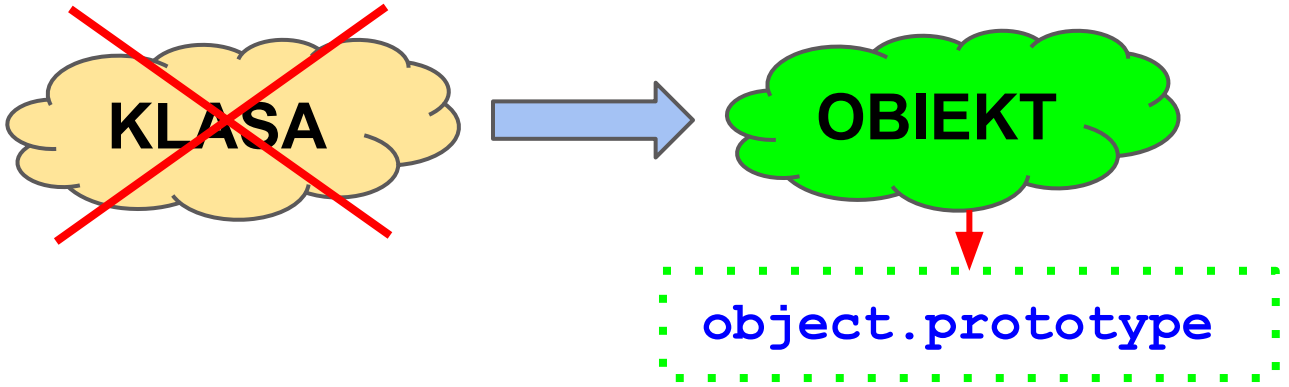
AJAX (asynchroniczność)

AJAX (Asynchronous JavaScript and XML) [Asynchroniczny JavaScript i XML] - model komunikacji sieciowej w której komunikacja pomiędzy klientem a serwerem odbywa się bez przeładowania dokumentu. Techniki służące do obsługi tej usługi są następujące:

1. **XMLHttpRequest - klasa która umożliwiająca asynchroniczne przesyłanie danych pomiędzy klientem a serwerem.**
2. **JavaScript - język skryptowy pośredniczący w komunikacji.**
3. **XML - język znaczników który opisuje dane (w ogólności może to być dowolny format np. JSON).**

Głównym zadaniem modelu AJAX jest otwarcie połączenia z pomiędzy klientem a serwerem.

Dziedziczenie prototypowe w JS

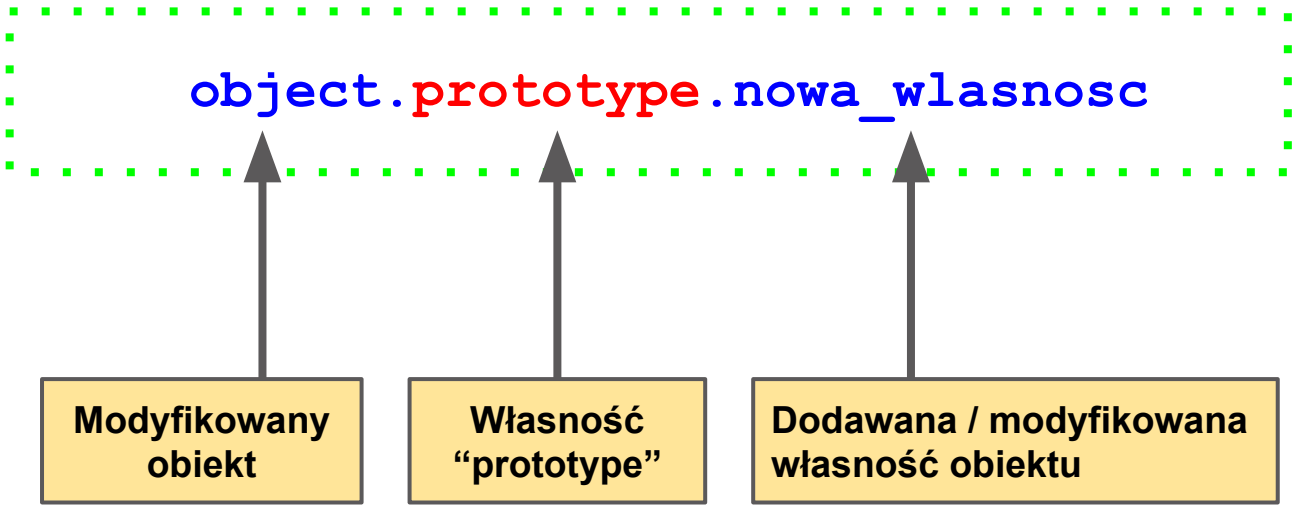


W języku JavaScript (a tym samym we wszystkich środowiskach programistycznych opartych o ten język) dziedziczenie odbywa się w sposób prototypowy. Każdy obiekt w JavaScript można traktować jako prototyp który w dowolnym momencie można rozszerzać, a jego własności przekazywać (cedować) na inne dziedziczące obiekty.

Obiekty dziedziczą po obiektach!

Dziedziczenie prototypowe w JS

Do zmiany lub rozszerzenia utworzonego obiektu możemy użyć własności prototype (który też jest obiektem):

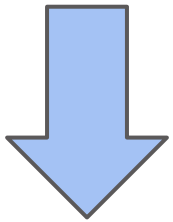


UNIWERSYTET
JAGIELLOŃSKI
W KRAKOWIE

Wprowadzenie do jQuery

UNIWERSYTET JAGIELLOŃSKI W KRAKOWIE

Wprowadzenie do jQuery



http://www.jquery.com/

UNIWERSYTET JAGIELLOŃSKI W KRAKOWIE

Wprowadzenie do jQuery

Biblioteka napisana w języku JavaScript o “lekkim” charakterze, obsługująca przestrzenie nazw z mechanizmem łatwej rozszerzalności umożliwiająca manipulowanie elementami struktury DOM (Document Object Model).

jQuery
write less, do more.

Biblioteka jest dotępna na licencji: GPL i MIT.

Pierwsze wydanie biblioteki w wersji miało miejsce 26 sierpnia 2006 roku

John Resig

UNIWERSYTET JAGIELLOŃSKI W KRAKOWIE

Wprowadzenie do jQuery

Biblioteka napisana w języku JavaScript o “lekkim” charakterze, obsługująca przestrzenie nazw z mechanizmem łatwej rozszerzalności umożliwiająca manipulowanie elementami struktury DOM (Document Object Model).

jQuery
write less, do more.

Biblioteka jest dotępna na licencji: GPL i MIT.

Pierwsze wydanie biblioteki w wersji miało miejsce 26 sierpnia 2006 roku

Najnowsze wydanie stabilne:

jQuery 2.1.4 (z 28 kwietnia 2015 r.)

jQuery 1.11.3 (z 28 kwietnia 2015 r.)

John Resig

UNIWERSYTET JAGIELLOŃSKI W KRAKOWIE

Wprowadzenie do jQuery

Biblioteka napisana w języku JavaScript o “lekkim” charakterze, obsługująca przestrzenie nazw z mechanizmem łatwej rozszerzalności umożliwiająca manipulowanie elementami struktury DOM (Document Object Model).

jQuery
write less, do more.

Biblioteka jest dotępna na licencji: GPL i MIT.

Pierwsze wydanie biblioteki w wersji miało miejsce 26 sierpnia 2006 roku

Najnowsze wydanie stabilne:

jQuery 2.1.4 (z 28 kwietnia 2015 r.)

jQuery 1.11.3 (z 28 kwietnia 2015 r.)

Uwaga od wersji 2.X.X brak wsparcia dla IE 6 - 8 (redukcja rozmiaru pliku biblioteki)

John Resig

UNIWERSYTET JAGIELLOŃSKI W KRAKOWIE

Wprowadzenie do jQuery

Najnowsze wydanie stabilne:

jQuery 2.1.4 (z 28 kwietnia 2015 r.)

Biblioteka jQuery występuje w dwóch wersjach:

UNIWERSYTET JAGIELLOŃSKI W KRAKOWIE

Wprowadzenie do jQuery

Najnowsze wydanie stabilne:

jQuery write less, do more.

jQuery 2.1.4 (z 28 kwietnia 2015 r.)

Biblioteka jQuery występuje w dwóch wersjach:

Zwykłej:
jquery-2.1.4.js

rozmiar: 82.4 KB

Czytelny kod, struktura hierarchiczna, komentarze

Spakowanej:
jquery-2.1.4.min.js

rozmiar: 27.7 KB

Brak czytelności, wszystko w jednej linii, brak komentarzy

UNIWERSYTET JAGIELLOŃSKI W KRAKOWIE

Wprowadzenie do jQuery

Najnowsze wydanie stabilne:

jQuery
write less, do more.

jQuery 2.1.4 (z 28 kwietnia 2015 r.)

Biblioteka jQuery występuje w dwóch wersjach:

jquery-2.1.4.js

```
(function( global, factory ) {

	if ( typeof module === "object" && typeof module.exports === "object" ) {
		// For CommonJS and CommonJS-like environments where a proper `wi
		// is present, execute the factory and get jQuery.
		// For environments that do not have a `window` with a `document`
		// (such as Node.js), expose a factory as module.exports.
		// This accentuates the need for the creation of a real `window`.
		// e.g. var jQuery = require("jquery")(window);
		// See ticket #14549 for more info.
		module.exports = global.document ?
			factory( global, true ) :
			function( w ) {
				if ( !w.document ) {
					throw new Error( "jQuery requires a windo
				}
				return factory( w );
			};
	} else {
		factory( global );
	}

// Pass this if window is not defined yet
}(typeof window !== "undefined" ? window : this, function( window, noGlobal ) {

// Support: Firefox 18+
// Can't be in strict mode, several libs including ASP.NET trace
// the stack via arguments.caller.callee and Firefox dies if
```

jquery-2.1.4.min.js

```
/*! jQuery v2.1.4 | (c) 2005, 2015 jQuery Foundation, Inc. | jquer
!function(a,b){"object"==typeof module&&"object"==typeof module.ex
requires a window with a document");return b(a)}:b(a)}("undefined"
[],d=c.slice,e=c.concat,f=c.push,g=c.indexOf,h={},i=h.toString,j=h
[\s\uFEFF\xA0]+|[\s\uFEFF\xA0]+$/g,p=/^-ms-/,q=/-([\da-z])/gi,r=fu
{jquery:m,constructor:n,selector:"",length:0,toArray:function(){re
(this)},pushStack:function(a){var b=n.merge(this.constructor(),a);
(this,a,b)},map:function(a){return this.pushStack(n.map(this,funct
(this,arguments))},first:function(){return this.eq(0)},last:functi
this.pushStack(c>=0&&b>c?[this[c]]:[])},end:function(){return this
(null)},push:f,sort:c.sort,splice:c.splice},n.extend=n.fn.extend=f
("boolean"==typeof g&&(j=g,g=arguments[h]||{},h++),"object"==typeo
a)c=g[b],d=a[b],g!==d&&(j&&d&&(n.isPlainObject(d)||(e=n.isArray(d)
(j,f,d)):void 0!==d&&(g[b]=d));return g},n.extend({expando:"jQuery
(a)},noop:function(){},isFunction:function(a){return"function"===n
=a&&a===a.window},isNumeric:function(a){return!n.isArray(a)&&a-par
n.isWindow(a)?1:a.constructor&&!j.call(a.constructor.prototype,"i
0},type:function(a){return null==a?a+"":"object"==typeof a||"funct
b,c=eval;a=n.trim(a),a&&(1===a.indexOf("use strict")?(b=l.createEl
(a))},camelCase:function(a){return a.replace(p,"ms-").replace(q,r)
```

UNIWERSYTET JAGIELLOŃSKI W KRAKOWIE

Wprowadzenie do jQuery

Najnowsze wydanie stabilne:

jQuery write less, do more.

jQuery 2.1.4 (z 28 kwietnia 2015 r.)

Biblioteka jQuery występuje w dwóch wersjach:

jquery-2.1.4.js

```
(function( global, factory ) {

	if ( typeof module === "object" && typeof module.exports === "object" ) {
		// For CommonJS and CommonJS-like environments where a proper `wi
		// is present, execute the factory and get jQuery.
		// For environments that do not have a `window` with a `document`
		// (such as Node.js), expose a factory as module.exports.
		// This accentuates the need for the creation of a real `window`.
		// S
		modu

	} else {
		fact
	}

// Pass this if window is not defined yet
}(typeof window !== "undefined" ? window : this, function( window, noGlobal ) {

// Support: Firefox 18+
// Can't be in strict mode, several libs including ASP.NET trace
// the stack via arguments.caller.callee and Firefox dies if
```

jquery-2.1.4.min.js

```
/*! jQuery v2.1.4 | (c) 2005, 2015 jQuery Foundation, Inc. | jquer
!function(a,b){"object"==typeof module&&"object"==typeof module.ex
requires a window with a document");return b(a)}:b(a)}("undefined"
[],d=c.slice,e=c.concat,f=c.push,g=c.indexOf,h={},i=h.toString,j=h
[\s\uFEFF\xA0]+|[\s\uFEFF\xA0]+$/g,p=/^-ms-/,q=/-([\da-z])/gi,r=fu
{jquery:m,constructor:n,selector:"",length:0,toArray:function(){re
ructor(),a);
ap(this,funct
,last:functi
){return this
n.fn.extend=f
bject"==typeo
=n.isArray(d)
bando:"jQuery
function"===n
=a&&a===a.window},isNumeric:function(a){return!n.isArray(a)&&a-par
n.isWindow(a)?!1:a.constructor&&!j.call(a.constructor.prototype,"i
0},type:function(a){return null==a?a+"":"object"==typeof a||"funct
b,c=eval;a=n.trim(a),a&&(1===a.indexOf("use strict")?(b=l.createEl
(a))},camelCase:function(a){return a.replace(p,"ms-").replace(q,r)
```

Dla optymalizacji działania aplikacji zaleca się używanie wersji “min”.

UNIWERSYTET JAGIELLOŃSKI W KRAKOWIE

Pobranie i instalacja jQuery

Pobieramy plik w najnowszej wersji ze strony:

http://jquery.com/download/

UNIWERSYTET JAGIELLOŃSKI W KRAKOWIE

Pobranie i instalacja jQuery

Pobieramy plik w najnowszej wersji ze strony:

http://jquery.com/download/

jQuery write less, do more.

jquery-2.1.4.js

jquery-2.1.4.min.js

```
<!DOCTYPE html>
<html>
<head>
<title>Pierwsza strona z jQuery</title>
<script type="text/javascript" src="jquery-2.1.4.min.js"></script>
</head>
<body>
 <h1>Testujemy bibliotekę jQuery</h1>
</body>
</html>
```

UNIWERSYTET JAGIELLOŃSKI W KRAKOWIE

Pobranie i instalacja jQuery

Pobieramy plik w najnowszej wersji ze strony:

http://jquery.com/download/

jQuery write less, do more.

jquery-2.1.4.js

jquery-2.1.4.min.js

```
<!DOCTYPE html>
<html>
<head>
<title>Pierwsza strona z jQuery</title>
<script type="text/javascript" src="jquery-2.1.4.min.js"></script>
</head>
<body>
 <h1>Testujemy bibliotekę jQuery</h1>
</body>
</html>
```

Od tego momentu mamy dostęp do wszystkich funkcjonalności biblioteki jQuery

Pobranie i instalacja jQuery

Pobieramy plik w najnowszej wersji ze strony:

http://jquery.com/download/

jquery-2.1.4.js

jquery-2.1.4.min.js

```
<!DOCTYPE html>
<html>
<head>
<title>Pierwsza strona z jQuery</title>
<script type="text/javascript" src="https://ajax.googleapis.
com/ajax/libs/jquery/2.1.4/jquery.min.js"></script>
</head>
<body>
 <h1>Testujemy bibliotekę jQuery</h1>
</body>
</html>
```

W ten sposób odwołujemy się do zewnętrznego źródła biblioteki jQuery.

Jakie to ma zalety ??

Od tego momentu mamy dostęp do wszystkich funkcjonalności biblioteki jQuery

UNIWERSYTET JAGIELLOŃSKI W KRAKOWIE

Jak używać jQuery

jQuery write less, do more.

Kiedy mamy już dołączoną bibliotekę jQuery do naszej aplikacji możemy wykorzystywać wbudowane funkcjonalności.

```
<!DOCTYPE html>
<html>
<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>
<body>
 <h1>Testujemy bibliotekę jQuery</h1>
</body>
</html>
```

Jak używać jQuery

Kiedy mamy już dołączoną bibliotekę jQuery do naszej aplikacji możemy wykorzystywać wbudowane funkcjonalności.

```
<!DOCTYPE html>
<html>
<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>
<body>
 <h1>Testujemy bibliotekę jQuery</h1>
</body>
</html>
```

```
jQuery(document).ready(function(){
  alert('Pierwsza operacja jQuery');
});
```

Jak używać jQuery

Kiedy mamy już dołączoną bibliotekę jQuery do naszej aplikacji możemy wykorzystywać wbudowane funkcjonalności.

```
<!DOCTYPE html>
<html>
<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>
<body>
 <h1>Testujemy bibliotekę jQuery</h1>
</body>
</html>
```

```
jQuery(document).ready(function(){
  alert('Pierwsza operacja jQuery');
});
```

Konstrukcja ta oznacza dostęp do zdefiniowanej przestrzeni nazw biblioteki:

$( )

(wersja skrócona)

Jak używać jQuery

Kiedy mamy już dołączoną bibliotekę jQuery do naszej aplikacji możemy wykorzystywać wbudowane funkcjonalności.

```
<!DOCTYPE html>
<html>
<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>
<body>
 <h1>Testujemy bibliotekę jQuery</h1>
</body>
</html>
```

```
jQuery(document).ready(function(){
  alert('Pierwsza operacja jQuery');
});
```

Konstrukcja ta oznacza dostęp do zdefiniowanej przestrzeni nazw biblioteki:

$( )	=	jQuery( )
(wersja skrócona)		(wersja pełna)

Jak używać jQuery

$( ) = jQuery( )

```
<script type="text/javascript">
     $(document).ready(function(){
       alert('Pierwsza operacja jQuery');
     });
</script>
```

UNIWERSYTET JAGIELLOŃSKI W KRAKOWIE

Jak używać jQuery

jQuery write less, do more.

$( ) = jQuery( )

```
<script type="text/javascript">
      $(document).ready(function(){
        alert('Pierwsza operacja jQuery');
      });
</script>
```

Co robi ta funkcja ?

UNIWERSYTET JAGIELLOŃSKI W KRAKOWIE

Jak używać jQuery

jQuery write less, do more.

$( ) = jQuery( )

```
<script type="text/javascript">
      $(document).ready(function(){
        alert('Pierwsza operacja jQuery');
      });
</script>
```

Co robi ta funkcja ?

W wyniku zadziałania funkcji $() tworzony jest obiekt, który posiada metodę ready(), a argumentem tej metody jest wywołanie zwrotne (funkcja nienazwana), która wykonuje określone działenie.

Jak używać jQuery

$( ) = jQuery( )

```
<script type="text/javascript">
      $(document).ready(function(){
        alert('Pierwsza operacja jQuery');
      });
</script>
```

Co robi ta funkcja ?

W wyniku zadziałania funkcji $() tworzony jest obiekt, który posiada metodę ready(), a argumentem tej metody jest wywołanie zwrotne (funkcja nienazwana), która wykonuje określone działenie.

Co robi metoda ready() oraz czym jest argument document ?

UNIWERSYTET JAGIELLOŃSKI W KRAKOWIE

Jak używać jQuery

jQuery write less, do more.

Co robi metoda ready() oraz czym jest argument document ?

Metoda ready(), zachodzi wtedy kiedy zajdzie zdarzenie “ready”, czyli cały (kompletny) dokument hipertekstowy zostanie załadowany, natomiast argument “document” symbolizuje ten dokument.

```
<script type=”text/javascript”>
      $(document).ready(function(){
        alert(‘Pierwsza operacja jQuery’);
      });
</script>
```

UNIWERSYTET JAGIELLOŃSKI W KRAKOWIE

Jak używać jQuery

jQuery write less, do more.

Co robi metoda ready() oraz czym jest argument document ?

Metoda ready(), zachodzi wtedy kiedy zajdzie zdarzenie “ready”, czyli cały (kompletny) dokument hipertekstowy zostanie załadowany, natomiast argument “document” symbolizuje ten dokument.

```
<script type=”text/javascript”>
      $(document).ready(function(){
        alert(‘Pierwsza operacja jQuery’);
      });
</script>
```

Czy da się w zawiązku z tym jeszcze bardziej uprościć ten zapis ?

Jak używać jQuery

Co robi metoda ready() oraz czym jest argument document ?

Metoda ready(), zachodzi wtedy kiedy zajdzie zdarzenie “ready”, czyli cały (kompletny) dokument hipertekstowy zostanie załadowany, natomiast argument “document” symbolizuje ten dokument.

```
<script type=”text/javascript”>
      $(document).ready(function(){
        alert(‘Pierwsza operacja jQuery’);
      });
</script>
```

Czy da się w zawiązku z tym jeszcze bardziej uprościć ten zapis ?

TAK

```
<script type=”text/javascript”>
      $(function(){
        alert(‘Pierwsza operacja jQuery’);
      });
</script>
```

Jak używać jQuery

Co robi metoda ready() oraz czym jest argument document ?

Metoda ready(), zachodzi wtedy kiedy zajdzie zdarzenie “ready”, czyli cały (kompletny) dokument hipertekstowy zostanie załadowany, natomiast argument “document” symbolizuje ten dokument.

```
<script type=”text/javascript”>
      $(document).ready(function(){
        alert(‘Pierwsza operacja jQuery’);
      });
</script>
```

Czy da się w zawiązku z tym jeszcze bardziej uprościć ten zapis ?

TAK

```
<script type=”text/javascript”>
      $(function(){
        alert(‘Pierwsza operacja jQuery’);
      });
</script>
```

Dlaczego ?

UNIWERSYTET JAGIELLOŃSKI W KRAKOWIE

jQuery i selektory

Jedną z podstawowych zalet biblioteki jQuery jest to, że umożliwia manipulowanie wybranymi elementami DOM, wykorzystując do tego celu selektory - tak jak jest robione w przypadku arkuszy CSS.

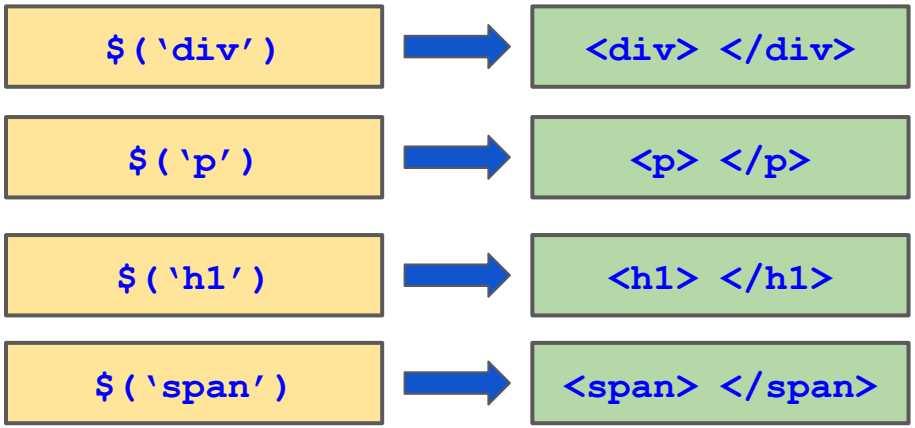
UNIWERSYTET JAGIELLOŃSKI W KRAKOWIE

jQuery i selektory

jQuery write less, do more.

Jedną z podstawowych zalet biblioteki jQuery jest to, że umożliwia manipulowanie wybranymi elementami DOM, wykorzystując do tego celu selektory - tak jak jest robione w przypadku arkuszy CSS.

Przykład selektora:

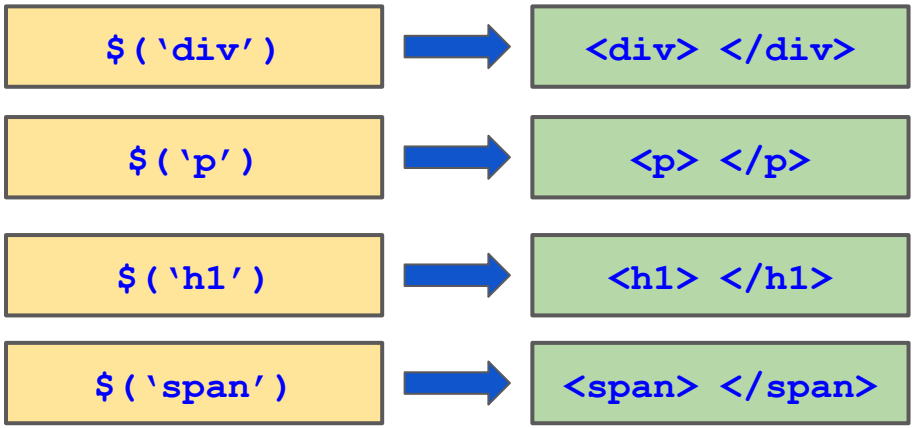


jQuery i selektory

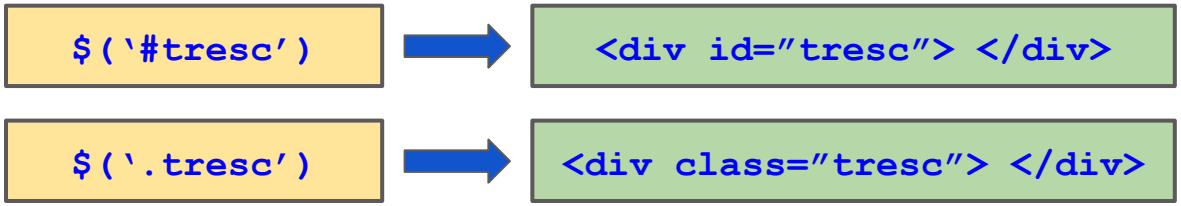
jQuery write less, do more.

Jedną z podstawowych zalet biblioteki jQuery jest to, że umożliwia manipulowanie wybranymi elementami DOM, wykorzystując do tego celu selektory - tak jak jest robione w przypadku arkuszy CSS.

Przykład selektora:

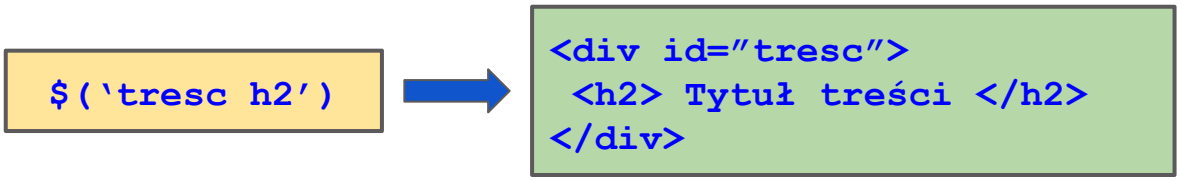


Selektorami tak jak w CSS mogą być klasy i identyfikatory:



jQuery i selektory

W jQuery obowiązują również, wszystkie reguły dziedziczenia, łączenia selektorów oraz używania selektorów atrybutów:



jQuery i selektory

W jQuery obowiązują również, wszystkie reguły dziedziczenia, łączenia selektorów oraz używania selektorów atrybutów:

```
$('tresc h2')
```

```
<div id="tresc">
 <h2> Tytuł treści </h2>
</div>
```

```
<script type="text/javascript">
     $(function(){
       $('tresc h2').click(function(){
          alert("Klikam w h2");
       });
     });
</script>
```

Zdarzenie zostanie obsłużone tylko wtedy gdy znacznik <h2> jest potomkiem znacznika o identyfikatorze “tresc”.

jQuery i selektory

W jQuery obowiązują również, wszystkie reguły dziedziczenia, łączenia selektorów oraz używania selektorów atrybutów:

```
$('tresc h2')
```

```
<div id="tresc">
 <h2> Tytuł treści </h2>
</div>
```

```
<script type="text/javascript">
     $(function(){
       $('tresc h2').click(function(){
          alert("Klikam w h2");
       });
     });
</script>
```

Zdarzenie zostanie obsłużone tylko wtedy gdy znacznik <h2> jest potomkiem znacznika o identyfikatorze “tresc”.

```
<script type="text/javascript">
     $(function(){
       $('tresc').find('h2').click(function(){
          alert("Klikam w h2");
       });
     });
</script>
```

Elementy można wybierać również, za pomocą metod wbudowanych takich jak “find()”

jQuery i selektory

W jQuery obowiązują również, wszystkie reguły dziedziczenia, łączenia selektorów oraz używania selektorów atrybutów:

```
$('a[href="www.uj.edu.pl"]')
```

```
<div id="tresc">
 <a href="www.uj.edu.pl"> Link do
strony UJ</h2>
</div>
```

jQuery i selektory

W jQuery obowiązują również, wszystkie reguły dziedziczenia, łączenia selektorów oraz używania selektorów atrybutów:

```
$('a[href="www.uj.edu.pl"]')
```

```
<div id="tresc">
 <a href="www.uj.edu.pl"> Link do
strony UJ</h2>
</div>
```

```
<script type="text/javascript">
     $(function(){
       $('a[href="www.uj.edu.pl"]').click(function(){
          alert("Klikam w link do strony UJ");
          return false;
       });
     });
</script>
  .
  .
  .
<div id="tresc">
   <a href="www.uj.edu.pl"> Link do strony UJ</h2>
</div>
```

UNIWERSYTET JAGIELLOŃSKI W KRAKOWIE

jQuery i zdarzenia

jQuery write less, do more.

Biblioteka jQuery potrafi również operować na zdarzeniach i funkcjach obsługi zdarzeń. Przypomniejmy typy najpopularniejszych zdarzeń:

UNIWERSYTET JAGIELLOŃSKI W KRAKOWIE

jQuery i zdarzenia

jQuery write less, do more.

Biblioteka jQuery potrafi również operować na zdarzeniach i funkcjach obsługi zdarzeń. Przypomniejmy typy najpopularniejszych zdarzeń:

Fun. obsługi zdarzenia	Metoda jQuery	Opis
onclick	click()	Kliknięcie elementu
odblclick	dblclick()	Podwójne kliknięcie elem.
onmousedown	mousedown()	Naciśnięcie przycisku myszy
onmouseover	mouseover()	Najechanie kursorem myszy na elem.

jQuery i zdarzenia

Biblioteka jQuery potrafi również operować na zdarzeniach i funkcjach obsługi zdarzeń. Przypomniejmy typy najpopularniejszych zdarzeń:

Fun. obsługi zdarzenia	Metoda jQuery	Opis
onclick	click()	Kliknięcie elementu
odblclick	dblclick()	Podwójne kliknięcie elem.
onmousedown	mousedown()	Naciśnięcie przycisku myszy
onmouseover	mouseover()	Najechanie kursorem myszy na elem.

Podobnie jak w samym JavaScript można używać tych zdarzeń do sterowanie aplikacją:

```
$(' selektor').zdarzenie();
```

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

UNIWERSYTET JAGIELLOŃSKI W KRAKOWIE

Modyfikacja elementów

jQuery write less, do more.

jQuery umożliwia w łatwy sposób manipulacje elementami struktury DOM każdego dokumentu hipertekstowego. Jedną z podstawowych operacji jaką można wykonać je możliwość zmodyfikowania wyglądu danego elementu przez nadanie mu odpowiedniego stylu CSS. Służy do tego metoda:

`.css()`

UNIWERSYTET JAGIELLOŃSKI W KRAKOWIE

Modyfikacja elementów

jQuery write less, do more.

jQuery umożliwia w łatwy sposób manipulacje elementami struktury DOM każdego dokumentu hipertekstowego. Jedną z podstawowych operacji jaką można wykonać je możliwość zmodyfikowania wyglądu danego elementu przez nadanie mu odpowiedniego stylu CSS. Służy do tego metoda:

.css()

Przykład wykorzystania:

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

Ta funkcja ustali wartość cechy "background" elementu "span" na odpowiedni kolor.

Modyfikacja elementów

jQuery umożliwia w łatwy sposób manipulacje elementami struktury DOM każdego dokumentu hipertekstowego. Jedną z podstawowych operacji jaką można wykonać je możliwość zmodyfikowania wyglądu danego elementu przez nadanie mu odpowiedniego stylu CSS. Służy do tego metoda:

.css()

Przykład wykorzystania:

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

Ta funkcja ustali wartość cechy "background" elementu "span" na odpowiedni kolor.

natomiast:

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

Ta funkcja odczyta tylko wartość ustawioną dla cechy "background" dla elementu "span".

Modyfikacja elementów

jQuery umożliwia w łatwy sposób manipulacje elementami struktury DOM każdego dokumentu hipertekstowego. Jedną z podstawowych operacji jaką można wykonać je możliwość zmodyfikowania wyglądu danego elementu przez nadanie mu odpowiedniego stylu CSS. Służy do tego metoda:

.css()

Przykład wykorzystania:

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

Ta funkcja ustali wartość cechy "background" elementu "span" na odpowiedni kolor.

natomiast:

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

UNIWERSYTET JAGIELLOŃSKI W KRAKOWIE

Modyfikacja elementów

jQuery write less, do more.

Poza operowaniem na pojedyńczych cechach CSS elementów mamy możliwość ustawiania “niejako w locie”, że dany element należy do danej klasy lub nie za pomocą metod:

`.addClass()` `.removeClass()`

UNIWERSYTET JAGIELLOŃSKI W KRAKOWIE

Modyfikacja elementów

jQuery write less, do more.

Poza operowaniem na pojedyńczych cechach CSS elementów mamy możliwość ustawiania "niejako w locie", że dany element należy do danej klasy lub nie za pomocą metod:

.addClass() .removeClass()

Przykład zastosowania:

```
<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>
```

UNIWERSYTET JAGIELLOŃSKI W KRAKOWIE

Modyfikacja elementów

jQuery write less, do more.

Poza operowaniem na pojedyńczych cechach CSS elementów mamy możliwość ustawiania "niejako w locie", że dany element należy do danej klasy lub nie za pomocą metod:

.addClass() **.removeClass()**

Przykład zastosowania:

```
<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>
```

Co w tym kontekście znaczy słowko "this" ??

UNIWERSYTET JAGIELLOŃSKI W KRAKOWIE

Modyfikacja elementów

jQuery write less, do more.

Poza operowaniem na pojedyńczych cechach CSS elementów mamy możliwość ustawiania "niejako w locie", że dany element należy do danej klasy lub nie za pomocą metod:

.addClass() **.removeClass()**

Przykład zastosowania:

```
<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>
```

Co w tym kontekście znaczy słowko "this" ??

Słówko "this" wskazuje na kontekst zdarzenia, czyli wskazuje na element DOM, który to zdarzenie wygnerował.

W tym konkretnym przykładzie:

```
$(this) = $('li')
```

Modyfikacja elementów

Poza operowaniem na pojedyńczych cechach CSS elementów mamy możliwość ustawiania “niejako w locie”, że dany element należy do danej klasy lub nie za pomocą metod:

.addClass()

.removeClass()

Przykład zastosowania:

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

<div id="tresc" style="background-color: red;">
    <p> Testowa tresc </p>
</div>
```

Modyfikacja elementów

Poza operowaniem na pojedyńczych cechach CSS elementów mamy możliwość ustawiania “niejako w locie”, że dany element należy do danej klasy lub nie za pomocą metod:

.addClass()

.removeClass()

Przykład zastosowania:

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

<div id="tresc" style="background-color: red;">
    <p> Testowa tresc </p>
</div>
```

Inny przykład użycia “this” :

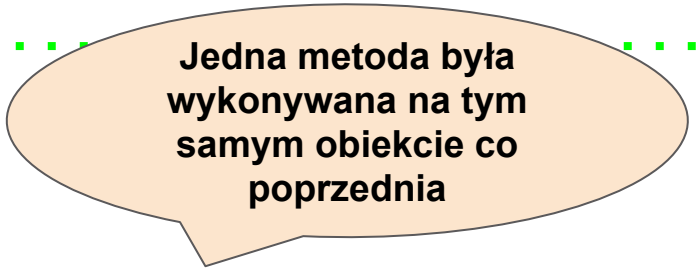
Uwaga:

kolor tła można ustawić za pomocą zbiorczej cechy “background”, natomiast odczytu możemy dokonać tylko z konkretnej pojedyńczej cechy takiej jak: “background-color”.

Wywołania łańcuchowe

W poprzednim przykładzie posłużyliśmy się dość dziwaczną, ale bardzo zgrabną konwencją zapisu:

```
$(function(){
  $('li').mouseover(function(){
     $(this).addClass('wazny');
  }).mouseout(function(){
     $(this).removeClass('wazny');
  });
});
```



Wywołania łańcuchowe

W poprzednim przykładzie posłużyliśmy się dość dziwaczną, ale bardzo zgrabną konwencją zapisu:

```
$(function(){
  $('li').mouseover(function(){
     $(this).addClass('wazny');
  }).mouseout(function(){
     $(this).removeClass('wazny');
  });
});
```

Jedna metoda była wykonywana na tym samym obiekcie co poprzednia

jest to tzw. "łączenie wywołań" lub "łańcuch wywołań". W ogólności przyjmuje on postać:

```
$(selektor).metoda1().metoda2().metoda3(). … .metodaN();
```

co odpowiada postaci:

```
$(selektor).metoda1();
$(selektor).metoda2();
$(selektor).metoda3();
.....
$(selektor).metodaN();
```

Wywołania łańcuchowe

W poprzednim przykładzie posłużyliśmy się dość dziwaczną, ale bardzo zgrabną konwencją zapisu:

```
$(function(){
  $('li').mouseover(function(){
     $(this).addClass('wazny');
  }).mouseout(function(){
     $(this).removeClass('wazny');
  });
});
```

Jedna metoda była wykonywana na tym samym obiekcie co poprzednia

jest to tzw. "łączenie wywołań" lub "łańcuch wywołań". W ogólności przyjmuje on postać:

```
$(selektor).metoda1().metoda2().metoda3(). … .metodaN();
```

co odpowiada postaci:

```
$(selektor).metoda1();
$(selektor).metoda2();
$(selektor).metoda3();
.....
$(selektor).metodaN();
```

Która z postaci zapisów jest lepszy ??

Wywołania łańcuchowe

W poprzednim przykładzie posłużyliśmy się dość dziwaczną, ale bardzo zgrabną konwencją zapisu:

```
$(function(){
  $('li').mouseover(function(){
     $(this).addClass('wazny');
  }).mouseout(function(){
     $(this).removeClass('wazny');
  });
});
```

Jedna metoda była wykonywana na tym samym obiekcie co poprzednia

jest to tzw. "łączenie wywołań" lub "łańcuch wywołań". W ogólności przyjmuje on postać:

```
$(selektor).metoda1().metoda2().metoda3(). … .metodaN();
```

co odpowiada postaci:

```
$(selektor).metoda1();
$(selektor).metoda2();
```

Która z postaci zapisów jest lepszy ??

Zapis łańcuchowy jest "lepszy" ponieważ w wyniku jego działania powstaje tylko jeden obiekt, który jest wynikiem przeszukania elementu pasującego do danego selektora. W drugim przypadku dokument byłby przeszukiwany wielokrotnie.

UNIWERSYTET JAGIELLOŃSKI W KRAKOWIE

Ukrywanie elementów

jQuery write less, do more.

Za pomocą CSS możemy ukrywać elementy strony dokumentu hipertekstowego używając cechy “display”:

```
<style type=”text/css”>
   div { display: none; }
</style>
```

Aby przywrócić widoczność elementu ustawiwmy wartość cechy na: “block” lub “inline”.

Ukrywanie elementów

Za pomocą CSS możemy ukrywać elementy strony dokumentu hipertekstowego używając cechy “display”:

```
<style type=”text/css”>
   div { display: none; }
</style>
```

Aby przywrócić widoczność elementu ustawiwmy wartość cechy na: “block” lub “inline”.

W jQuery również możemy wykonywać takie operacje za pomocą metod:

.hide()	.show()
(ukrywanie)	(pokazywanie)

Przykład zastosowania:

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

Ukrywanie elementów

Za pomocą CSS możemy ukrywać elementy strony dokumentu hipertekstowego używając cechy “display”:

```
<style type=”text/css”>
   div { display: none; }
</style>
```

Aby przywrócić widoczność elementu ustawiwmy wartość cechy na: “block” lub “inline”.

W jQuery również możemy wykonywać takie operacje za pomocą metod:

.hide()	.show()
(ukrywanie)	(pokazywanie)

Przykład zastosowania:

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

Dodatkowo argumentem obu funkcji może być opcjonalny paramter “slow” lub “fast” lub określenie w milisekundach szybkości wykonania danej czynności.

Ukrywanie elementów

Za pomocą CSS możemy ukrywać elementy strony dokumentu hipertekstowego używając cechy “display”:

```
<style type=”text/css”>
   div { display: none; }
</style>
```

Aby przywrócić widoczność elementu ustawiwmy wartość cechy na: “block” lub “inline”.

W jQuery również możemy wykonywać takie operacje za pomocą metod:

.hide()	.show()
(ukrywanie)	(pokazywanie)

Przykład zastosowania:

```
$(function(){
  $('button#hide').click(function(){
     $('p').hide("slow");
  });
  $('button#show).click(function(){
     $('p').show(2000);
  });
});
```

Dodatkowo argumentem obu funkcji może być opcjonalny paramter “slow” lub “fast” lub określenie w milisekundach szybkości wykonania danej czynności.

UNIWERSYTET JAGIELLOŃSKI W KRAKOWIE

Ukrywanie elementów

jQuery write less, do more.

W jQuery istnieje również metoda obsługująca obie operacje (ukrywania i pokazywania elementu):

.toggle()

(ukrywanie / pokazywanie)

Przykład użycia:

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

Ukrywanie elementów

W jQuery istnieje również metoda obsługująca obie operacje (ukrywania i pokazywania elementu):

.toggle()

(ukrywanie / pokazywanie)

Przykład użycia:

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

Również w tym przypadku możemy określić szybkość zachodzenia operacji tymi samymi parametrami co poprzednio.

Ukrywanie elementów

W jQuery istnieje również metoda obsługująca obie operacje (ukrywania i pokazywania elementu):

.toggle()

(ukrywanie / pokazywanie)

Przykład użycia:

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

Również w tym przypadku możemy określić szybkość zachodzenia operacji tymi samymi parametrami co poprzednio.

Ćwiczenie:

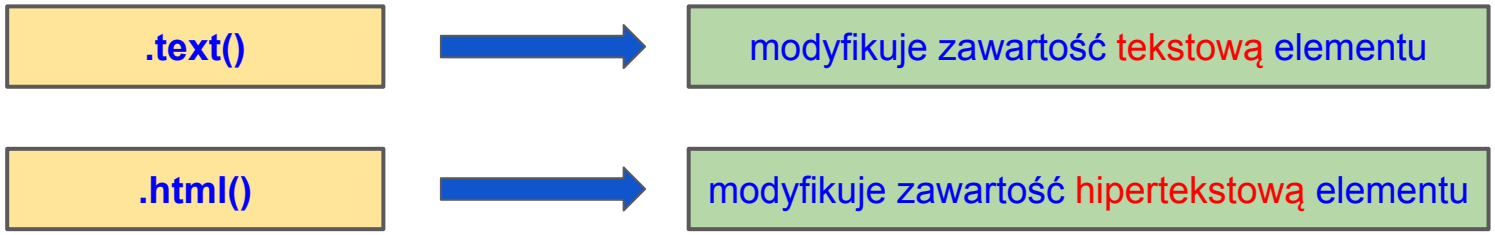
Napisać menu UL/LI z dwoma poziamami zagnieżdzenia, w którym drugi poziom menu będzie widoczny dopiero po najechaniu na niego kursorem myszy. Wykorzystać do tego metodę .toggle()

Odczyt i modyfikacja zawartości elementów

UNIWERSYTET JAGIELLOŃSKI W KRAKOWIE

jQuery write less, do more.

Modyfikacja zawartości elementów oraz i ich treści jest podobnie prosta jak w przypadku czystego JavaScriptu. Służą do tego celu dwie metody:

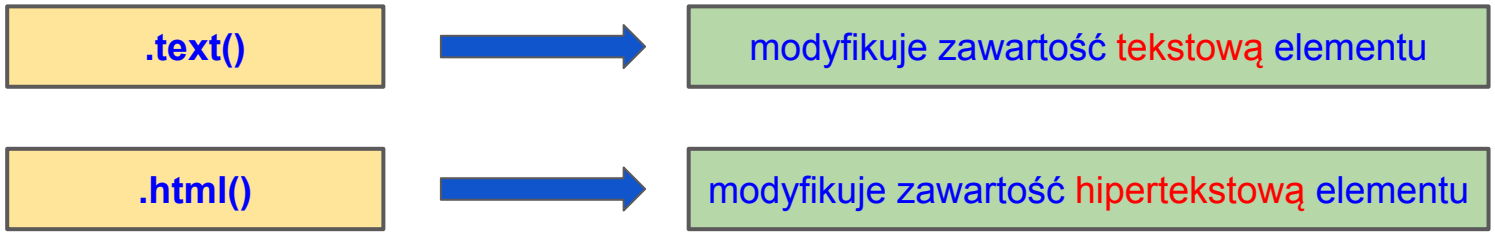


Odczyt i modyfikacja zawartości elementów

UNIWERSYTET JAGIELLOŃSKI W KRAKOWIE

jQuery write less, do more.

Modyfikacja zawartości elementów oraz i ich treści jest podobnie prosta jak w przypadku czystego JavaScriptu. Służą do tego celu dwie metody:



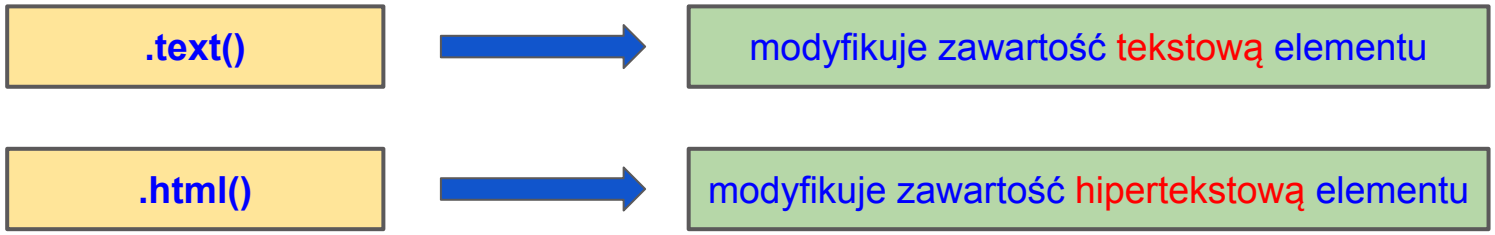

```
$(function(){
  $('div#tresc1').text('Nowa treść wpisana dynamicznie w jQuery');
  $('div#tresc2').html('Nowa treść wpisana dynamicznie w jQuery');
});
```

Odczyt i modyfikacja zawartości elementów

UNIWERSYTET JAGIELLOŃSKI W KRAKOWIE

jQuery write less, do more.

Modyfikacja zawartości elementów oraz i ich treści jest podobnie prosta jak w przypadku czystego JavaScriptu. Służą do tego celu dwie metody:




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

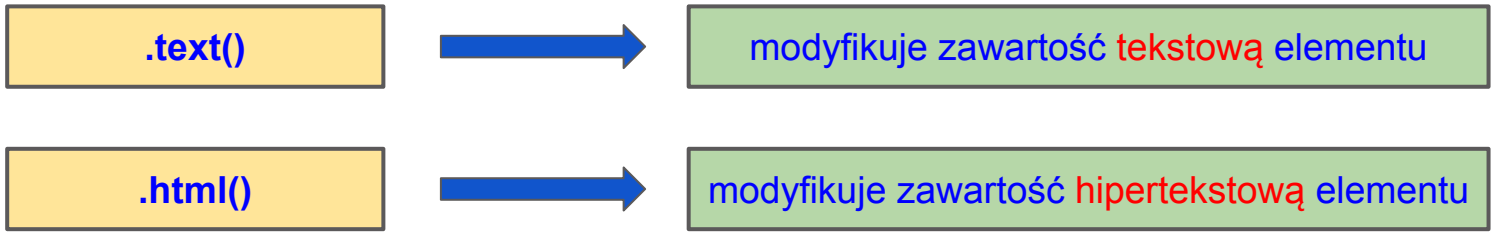
Czym będzie różnił się wynik działania obu funkcji ??

Odczyt i modyfikacja zawartości elementów

UNIWERSYTET JAGIELLOŃSKI W KRAKOWIE

jQuery write less, do more.

Modyfikacja zawartości elementów oraz i ich treści jest podobnie prosta jak w przypadku czystego JavaScriptu. Służą do tego celu dwie metody:




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

Czym będzie różnił się wynik działania obu funkcji ??

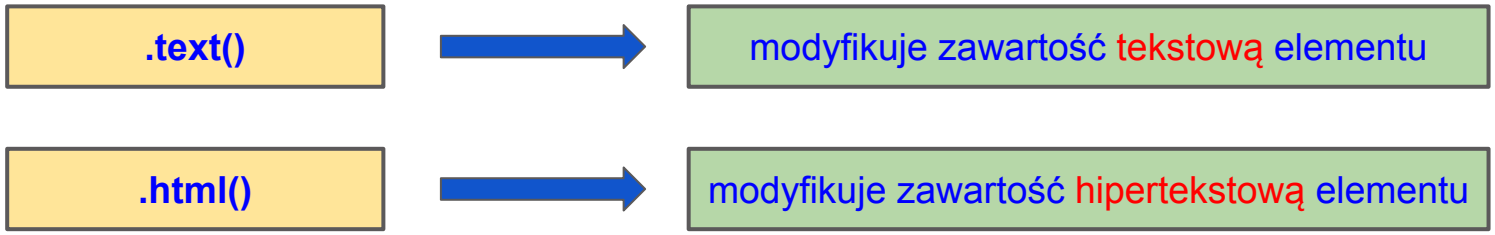
NICZYM !

Odczyt i modyfikacja zawartości elementów

UNIWERSYTET JAGIELLOŃSKI W KRAKOWIE

jQuery write less, do more.

Modyfikacja zawartości elementów oraz i ich treści jest podobnie prosta jak w przypadku czystego JavaScriptu. Służą do tego celu dwie metody:




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

Czym będzie różnił się wynik działania obu funkcji ??

NICZYM !

ALE ....

Odczyt i modyfikacja zawartości elementów

UNIWERSYTET JAGIELLOŃSKI W KRAKOWIE

jQuery write less, do more.

.text()	→	modyfikuje zawartość tekstową elementu
.html()	→	modyfikuje zawartość hipertekstową elementu

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

Odczyt i modyfikacja zawartości elementów

UNIWERSYTET JAGIELLOŃSKI W KRAKOWIE

jQuery write less, do more.

.text() → modyfikuje zawartość tekstową elementu

.html() → modyfikuje zawartość hipertekstową elementu

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

```
<div id="tresc1">
   &lt;span&gt; To jest treść &lt;/span&gt;
</div>

<div id="tresc2">
   <span> To jest treść</span>
</div>
```

Treść w metodzie ".text()" zostanie sparsowana i wszystkie znaki specjalne zostaną zmienione na ich odpowiedniki zapisane za pomocą endji html.

Odczyt i modyfikacja zawartości elementów

UNIWERSYTET JAGIELLOŃSKI W KRAKOWIE

jQuery write less, do more.

.text() → modyfikuje zawartość tekstową elementu

.html() → modyfikuje zawartość hipertekstową elementu

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

```
<div id="tresc1">
   &lt;span&gt; To jest treść &lt;/span&gt;
</div>

<div id="tresc2">
   <span> To jest treść</span>
</div>
```

Treść w metodzie ".text()" zostanie sparsowana i wszystkie znaki specjalne zostaną zmienione na ich odpowiedniki zapisane za pomocą endji html.

Zatem za pomocą metody .text() dodajemy treści, a za pomocą metody .html() możemy dodawać nowe zagnieżdzone struktury znacznikowe.

Odczyt i modyfikacja atrybutów elementów

UNIWERSYTET JAGIELLOŃSKI W KRAKOWIE

jQuery write less, do more.

Bardzo często w dynamicznym dokumencie hipertekstowym zachodzi potrzeba manipulacji wartościami atrybutów konkretnego znacznika. Dostęp do atrybutów znaczników uzyskujemy za pomocą metody:

```
.attr( nazwa atrybutu , wartosc atrybutu )
```

Odczyt i modyfikacja atrybutów elementów

UNIWERSYTET JAGIELLOŃSKI W KRAKOWIE

jQuery write less, do more.

Bardzo często w dynamicznym dokumencie hipertekstowym zachodzi potrzeba manipulacji wartościami atrybutów konkretnego znacznika. Dostęp do atrybutów znaczników uzyskujemy za pomocą metody:

`.attr(` *nazwa atrybutu , wartosc atrybutu* `)`

Jako argument tej metody podajemy nazwę atrybutu który chemy zamodyfikować:

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

<img src="" alt="obrazek"/>
```

Odczyt i modyfikacja atrybutów elementów

UNIWERSYTET JAGIELLOŃSKI W KRAKOWIE

jQuery write less, do more.

Bardzo często w dynamicznym dokumencie hipertekstowym zachodzi potrzeba manipulacji wartościami atrybutów konkretnego znacznika. Dostęp do atrybutów znaczników uzyskujemy za pomocą metody:

`.attr(` *nazwa atrybutu , wartosc atrybutu* `)`

Jako argument tej metody podajemy nazwę atrybutu który chemy zamodyfikować:

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

<img src="" alt="obrazek"/>
```

Jak taką funkcjonalność wykorzystać w przypadku elementów img ???

Odczyt i modyfikacja atrybutów elementów

UNIWERSYTET JAGIELLOŃSKI W KRAKOWIE

jQuery write less, do more.

Bardzo często w dynamicznym dokumencie hipertekstowym zachodzi potrzeba manipulacji wartościami atrybutów konkretnego znacznika. Dostęp do atrybutów znaczników uzyskujemy za pomocą metody:

`.attr( nazwa atrybutu , wartosc atrybutu )`

Jako argument tej metody podajemy nazwę atrybutu który chemy zamodyfikować:

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

<img src="" alt="obrazek"/>
```

Jak taką funkcjonalność wykorzystać w przypadku elementów img ???

Można stworzyć prosta galerię

Odczyt i modyfikacja atrybutów elementów

UNIWERSYTET JAGIELLOŃSKI W KRAKOWIE

jQuery write less, do more.

`.attr( nazwa atrybutu , wartosc atrybutu )`

```
$(function(){
  $('td img').click(function(){
     var adres = $(this).atrr('src').replace('maly', 'duzy');
     $('div img').atrr('src', adres);
   });
});

<tr>
   <td><img src="maly/obraz-1.png"/></td>
   <td><img src="maly/obraz-2.png"/></td>
   <td><img src="maly/obraz-3.png"/></td>
   <td><img src="maly/obraz-4.png"/></td>
</tr>

<div>
   <img src="duzy/obraz-1.png" alt="obrazek" />
</div>
```

Odczyt i modyfikacja atrybutów elementów

UNIWERSYTET JAGIELLOŃSKI W KRAKOWIE

jQuery write less, do more.

`.attr( nazwa atrybutu , wartosc atrybutu )`

```
$(function(){
  $('td img').click(function(){
     var adres = $(this).atrr('src').replace('maly', 'duzy');
     $('div img').atrr('src', adres);
   });
});

<tr>
   <td><img src="maly/obraz-1.png"/></td>
   <td><img src="maly/obraz-2.png"/></td>
   <td><img src="maly/obraz-3.png"/></td>
   <td><img src="maly/obraz-4.png"/></td>
</tr>

<div>
   <img src="duzy/obraz-1.png" alt="obrazek" />
</div>
```

Metoda .replace() zastępuje w stringu wybrany wzorzec (podany jako argument 1) i zastępuje go podanym jako argument 2 tekstem.

UNIWERSYTET
JAGIELLOŃSKI
W KRAKOWIE

KONIEC WYKŁADU 7