

jQuery

Jan Bilak, 2008

Základní vlastnosti

- Open source (MIT, GPL) JavaScriptová knihovna
- Zjednodušuje
 - Práci s DOM
 - Obsluhu událostí
 - AJAX
 - Animace
- Pluginy
 - UI (draggable, droppables, selectables, ...)
 - ...
- Zakrývá rozdíly mezi prohlížeči
 - IE 6.0+, FF 2+, Safari 2.0+, Opera 9.0+
- Malá velikost (zkomprimovaná cca 15 kB)
- Jednoduchá stručná syntaxe
- Komunita, dokumentace + tutoriály, projekty

Použití

- `<script type="text/javascript" src="jquery.js">
</script>`
- Typické použití:

Najdi množinu elementů

`$("div").addClass("myclass")`

Vykonej na nich nějakou operaci

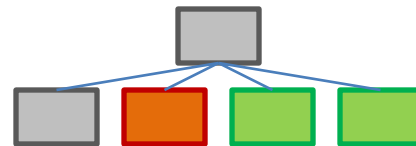
- `$ == jQuery`
 - `jQuery("div").addClass("myclass")`
- `$("div")` vrátí objekt – jQuery kolekci elementů

Použití objektu jQuery

- **`$(expression, [context])`**
 - `$("input:radio", document.forms[0])`
 - Najde všechny elementy input typu radio v prvním formuláři dokumentu
 - `$("div > p")`
- **`$(html)`**
 - `$("<div><p>Hello</p></div>").appendTo("body")`
- **`$(elements)`**
 - `$(document.body).css("background", "black")`
 - `$(myForm.elements).hide()`
- **`$(callback)`**
 - Zkratka za `$(document).ready(function(){...})`

Selektory

- **#id**
 - `$("#myDiv")`
- **element**
 - `"div"`
- **.class**
 - `".myClass"`
- *****
 - `"*"`
- **selector1, selector2, ..., selectorN**
 - `"div,span,p.myClass"`
- **ancestor descendant**
 - `"form input"`
- **Parent > child**
 - `"#main > *"`
- **Prev + next**
 - `"label + input"`
 - (elementy input, kterým těsně předchází label)
- **Prev ~ siblings**
 - `"#prev ~ div"`
 - (elementy div, které mají stejného rodiče jako **#prev** a následují po elementu **#prev**)



Filtry (1)

- **:first**
 - \$("tr:first")
- **:last**
 - \$("tr:last")
- **:not(selector)**
 - \$("input:not(:checked)")
- **:even**
 - \$("tr:even")
- **:odd**
 - \$("tr:odd")
- **:eq(index)**
 - \$("td:eq(2)")
 - Indexování od nuly
- **:gt(intex)**
 - \$("td:gt(4)")
- **:lt(index)**
 - \$("td:lt(4)")
- **:header**
 - \$(" :header")
 - Vybere všechny h1, h2, h3, ...
- **:animated**

Filtry (2)

- **:empty**
 - Prázdné elementy (ani text)
- **:has(selector)**
 - Obsahuje (kdekoli uvnitř)
- **:parent**
 - Jsou rodiči
(= neprázdné elementy)
- **:hidden**
- **:visible**
- **[attribute]**
- **[attribute=value]**
- **[attribute!=value]**
- **[attribute^=value]**
 - Začíná danou hodnotou
- **[attribute\$=value]**
 - Končí danou hodnotou
- **[attribute*=value]**
 - Obsahuje danou hodnotu
- Kombinování:
 - `$("input[id][name$='man']")`

Filtery (3)

- **:nth-child**
(index/even/odd/equation)
 - `$("ul li:nth-child(2)")`
 - `:nth-child(2)`
 - `:nth-child(even)`
 - `:nth-child(odd)`
 - `:nth-child(2n)`
 - `:nth-child(2n+1)`
- **:first-child**
- **:last-child**
- **:only-child**
 - Nemají sourozence

Formulářové filtry:

- **:input**
 - Input, textarea, select, button
- **:text**
- **:password**
- **:radio**
- **:checkbox**
- **:submit**
- **:image**
- **:reset**
- **:button**
- **:file**
- **:hidden**

- **:enabled**
- **:disabled**
- **:checked**
- **:selected**

Práce s jQuery kolekcí

- **each(callback)**
 - `$("#div").each(function (i) {
 this.style.color = "blue";
});`
- **size()**
 - Počet elementů
- **length**
 - Počet elementů (totéž)
- **eq(position)**
 - Vybere jeden element a vrátí jako jednoprvkovou jQuery kolekci
- **get()**
 - Transformuje kolekci jQuery na pole DOM elementů
- **get(index)**
 - Vrátí jeden element
- **index(element)**
 - Hledá element v kolekci a vrátí pozici (nebo -1)
- **not(expr)**
 - Odstraní elementy, které splňují expr
- **filter(expr)**
 - Odstraní elementy, které nesplňují expr
- **filter(fn)**
- **is(expr)**
 - Alespoň jeden element z kolekce splňuje podmínku
→ true
- **slice(start, end)**
 - Vybere elementy s indexem v intervalu <start, end)

Atributy

- **attr(name)**
 - `var title = $("em").attr("title");`
- **attr(key, value)**
 - `$("#button").attr("disabled","disabled");`
- **attr(properties)**
 - `$("#img").attr({
 src: "/images/hat.gif",
 title: "jQuery",
 alt: "jQuery Logo"
});`
- **attr(key, fn)**
 - Pro výpočet hodnoty se použije funkce

Atributy

- **html()**
 - `alert($("#p").html())`
- **html(val)**
 - `$("#p").html("Pokus")`
- **text()**
- **text(val)**
- **val()**
- **val(val)**
- **addClass(class)**
 - `$("#my").addClass("oramuj")`
- **hasClass(class)**
- **removeClass(class)**
- **toggleClass(class)**

```
$("#single").val("Single2"); // combobox
```

```
$("#multiple").val(["Multiple2", "Multiple3"]); // listbox
```

```
$("#input").val(["checkbox2", "radio1"]); // checkbox a radiobox
```

CSS

- **css(name)**
 - Bere v úvahu první element
 - `$("#p").css("background-color")`
- **css(name, value)**
- **css(properties)**
 - `$("#p").css({ backgroundColor:"yellow", fontWeight:"bolder" })`
- **height()**
 - Vypočtená výška v px
- **height(val)**
 - Nastaví css vlastnost
- **width()**
- **width(val)**
- **offset()**
 - Vrací objekt se souřadnicemi oproti "pohledu" (např. oknu)
 - ```
var p = $("#p:last");
var offset = p.offset();
p.html("left: " + offset.left + ", top: " + offset.top);
```

# Práce s jQuery kolekcí

- **add( expr )**
  - Přidá do kolekce další elementy
  - `$("#div")`
    - `.css("border", "2px solid red")`
    - `.add("p")`
    - `.css("background", "yellow");`
- **children( [expr filter] )**
  - Množina přímých potomků
- **contents( )**
  - Množina přímých potomků včetně textových elementů
- **find( expr )**
  - `$("#p").find("span")` je ekvivalent `$("#p span")`
- **next( [expr filter] )**
  - Nejbližší praví sourozenci
- **nextAll( [expr filter] )**
  - Všichni praví sourozenci
- **parent( expr )**
  - Rodiče (nejbližší předci)
- **parents( expr )**
  - Předci
- **prev( [expr filter] )**
  - Nejbližší leví sourozenci
- **prevAll( [expr filter] )**
  - Všichni leví sourozenci
- **siblings( expr )**
  - Všichni sourozenci

# Vkládání dovnitř/vně

- **append( content )**
  - Přidá na konec (dovnitř) elementů nějaký obsah
  - Obsah: string, element, jQuery
  - `$( "p" ).append( "<b>Hello</b>" )`
    - Na konec odstavců přidá tučný text
- **appendTo( cíl )**
  - Prohozené operandy
  - `$( "span" ).appendTo( "p#foo" )`
    - Přidá na konec odstavce #foo všechny elementy span
- **prepend( content )**
- **prependTo( content )**
  - Totéž, ale na začátek (dovnitř)

**after( ctx )**

Vložení před

**before( ctx )**

Vložení za

**insertAfter( c tx )**

**insertBefore( c tx )**

Prohozené operandy

`<p>Původní obsah Hello</p>`

`<p>Původní obsah</p> Hello`

# Obalení

- **wrap( html )**

- Obalí elementy, obal by neměl mít více sourozenců v žádné vrstvě

- `<a><b></b></a>` ... správně

- `<a></a><b></b>` ... špatně



- `$("#p").wrap("<div class='wrap'></div>")`

- **wrap( elem )**

- `$("#p#x").wrap(document.getElementById('content'));`

- `<p id="x">Test Paragraph.</p><div id="content"></div>`

- `<div id="content"><p>Test Paragraph.</p></div>`



- **wrapAll( html )**

- **wrapAll( elem )**

- Obalí všechny elementy do jednoho obalu

- **wrapInner( html )**

- **wrapInner( elem )**

- `$("#p").wrapInner("<b></b>");`



# Náhrady, odstraňování, kopírování

- **replaceWith( content )**
  - Nahradí elementy obsahem
  - `$("p").replaceWith("<b>Odstavec. </b>")`
    - Odstavec. Odstavec. Odstavec. ...
- **replaceAll( selector )**
  - Prohozené operandy
  - `("<b>Paragraph. </b>").replaceAll("p")`
- **empty()**
- **remove( expr )**
- **clone()**
  - `$("b").clone().prependTo("p#par")`
    - Vezme kopie tučných textů a vloží je na začátek p#par
- **clone( true )**
  - Včetně obsluh událostí



# Označení „externích“ odkazů

- `$("a").not("[href^=/]")  
.not("[href^=#]")  
.append("<span class='External'>?</span>");`
- `$("a:not([@href*=mysite])")  
.append("<span class='External'>?</span>");`

# Změna „stylu“ odrážek

- `$("#ul").addClass("Replaced");`  
`$("#ul > li").prepend("– ");`
- V CSS:  
`ul.Replaced { list-style : none; }`

# Události

- `$(document).ready( fn )`
  - Vykoná funkci po dokončení DOM stromu. Zkratka `$(fn)`.
- `bind( type, [data], fn )`
  - `$("#p").bind("mouseenter mouseleave", function(e){  
$(this).toggleClass("over");  
});`
  - Případná data předána do obsluhy jako `event.data`.
- `one( type, [data], fn )`
  - Obluha se vykoná max. jednou pro každý element.
- `trigger( type, data )`
  - Vyvolá událost.
- `triggerHandler( type, data )`
  - Jen uživatelské obsluhy – nikoli defaultní chování prohlížeče.
- `unbind( type, data )`


# Interakce

- **hover( over, out )**

- Při najetí kurzorem a opuštění kurzorem

- `$("#li").hover(`  
    `function () {`  
        `$(this).append($("#<span> ***</span>"));`  
    `},`  
    `function () {`  
        `$(this).find("span:last").remove();`  
    `}`  
`);`

```
Text
Text ***
Text
Text
```



- **toggle( fn, fn )**

- Obdoba při lichém a sudém kliknutí

# Konkrétní události

Vyvolání / přiřazení obsluhy:

- `blur( )`
- `blur( fn )`
- `change( )`
- `change( fn )`
- `click( )`
- `click( fn )`
- `dblclick( )`
- `dblclick( fn )`
- `error( )`
- `error( fn )`
- `focus( )`
- `focus( fn )`
- `keydown( )`
- `keydown( fn )`
- `keypress( )`
- `keypress( fn )`
- `keyup( )`
- `keyup( fn )`
- `load( fn )`
- `mousedown( fn )`
- `mousemove( fn )`
- `mouseout( fn )`
- `mouseover( fn )`
- `mouseup( fn )`
- `resize( fn )`
- `scroll( fn )`
- `select( )`
- `select( fn )`
- `ubmit( )`
- `submit( fn )`
- `unload( fn )`

# Změna stylesheetu (1/2)

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
 <head>
 <title>jQuery Demo</title>
 <link rel="stylesheet" type="text/css" media="screen" href="Nrm.css" />
 </head>
 <body>
 <h1>Choose A Style:</h1>

 Contrast
 Print
 Normal

 </body>
</html>
```

# Změna stylesheetu (2/2)

```
$("#StyleContrast").click(function() {
 $("link[media='screen']").attr("href", "Contrast.css");
});
```

```
$("#StylePrint").click(function() {
 $("link[media='screen']").attr("href", "Print.css");
});
```

```
$("#StyleNormal").click(function() {
 $("link[@media='screen']").attr("href", "Nrm.css");
});
```

# Animace

- `animate( params, [duration], [easing], [callback] )`
  - `$("#go").click(function(){  
    $("#block").animate({  
        width: "70%",  
        opacity: 0.4,  
        marginLeft: "0.6in",  
        fontSize: "3em",  
        borderWidth: "10px"  
    }, 1500 );  
});`
  - `easing` ... průběh změny (standardně lineární, možnost nadefinovat jiný průběh)
  - `callback` ... fn po dokončení



# Animace a fronta

- `show( )`
- `show( speed, callback )`
- `hide( )`
- `hide( speed, callback )`
- `toggle( )`
- `slideDown( speed, callback )`
- `slideUp( speed, callback )`
- `slideToggle( speed, callback )`
- `fadeIn( speed, callback )`
- `fadeOut( speed, callback )`
- `fadeTo( speed, opacity, callback )`
- `animate( params, duration, easing, callback )`
- `animate( params, options )`
- `stop( )`
- `queue( )`
  - vrací aktuální frontu
- `queue( callback )`
  - přidá do fronty zavolání funkce
- `queue( queue )`
  - nahradí současnou frontu jinou
- `dequeue( )`
  - vyjme funkci z fronty a spustí ji

# Ajax příklady

Načte a spustí JavaScript:

```
$.ajax({
 type: "GET",
 url: "test.js",
 dataType: "script"
});
```

# Ajax příklady

Odešle data na server a informuje uživatele.

```
$.ajax({
 type: "POST",
 url: "some.php",
 data: "name=John&location=Boston",
 success: function(msg){
 alert("Data Saved: " + msg);
 }
});
```

# Ajax příklady

Přečte html a vloží do stránky na konec do #result.

```
$.ajax({
 url: "test.html",
 cache: false,
 success: function(html){
 $("#results").append(html);
 }
});
```

# Ajax – zkratky

- **jQuery.load( url, [data], [callback] )**
  - `$("#target").load("data.html");`
- **jQuery.get( url, [data], [callback] )**
  - `$.get("test.php", { name: "John", time: "2pm" } );`
  - `$.get("test.php", function(data){alert("Data: " + data);});`
- **jQuerygetJSON( url, data, callback )**
  - `$.getJSON("data.php",  
function(data){  
$.each(data.items, function(i,item){  
$("<img/>").attr("src",  
item.media.m).appendTo("#images");  
});  
});`
- **jQuery.getScript( url, callback )**
  - `$.getScript("test.js");`
- **jQuery.post( url, data, callback, type )**
  - `$.post("test.cgi", { name: "John", time: "2pm" } );`

**BONUS**

Praktická ukázka

**UI**

DEMO

# Plug-iny

Příklad dvou plug-in funkcí:

```
jQuery.fn.extend({
 check: function() {
 return this.each(function() { this.checked = true; });
 },
 uncheck: function() {
 return this.each(function() { this.checked = false; });
 }
});
```

Jejich volání:

```
$("#input[@type=checkbox]").check();
$("#input[@type=radio]").uncheck();
```



# Plug-iny

Přidání funkcí do namespace jQuery:

```
jQuery.extend({
 min: function(a, b) { return a < b ? a : b; },
 max: function(a, b) { return a > b ? a : b; }
});
```

# Zde zapiš hledané slovo...

```
<form action="#" method="post">
 <input id="SearchText" name="text" />
 <button>Search</button>
</form>
```

## JavaScript:

```
populateElement('#SearchText',
 'Zde zapiš hledané slovo...');

$('#SearchText').populateElement('Zde zapiš
hledané slovo...');
```

[DEMO](#)

# Zde zapiš hledané slovo...

```
function populateElement(selector, defvalue) {
 if($.trim($(selector).val()) == "") {
 $(selector).val(defvalue);
 }

 $(selector).focus(function() {
 if($(selector).val() == defvalue) {
 $(selector).val("");
 }
 });

 $(selector).blur(function() {
 if($.trim($(selector).val()) == "") {
 $(selector).val(defvalue);
 }
 });
}
```

# Zapiš hledané slovo...

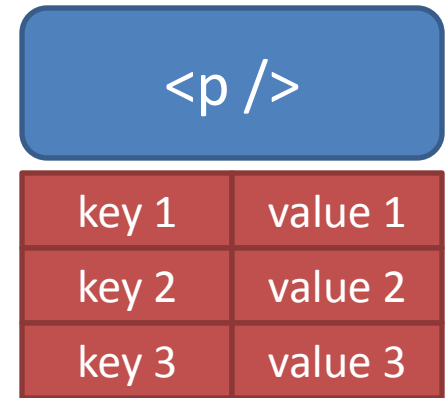
```
$.fn.extend({
 populateElement : function(defvalue)
 {
 if($.trim($(this).val()) == "") {
 $(this).val(defvalue);
 }
 $(this).focus(function() {
 if($(this).val() == defvalue) {
 $(this).val("");
 }
 });
 $(this).blur(function() {
 if($.trim($(this).val()) == "") {
 $(this).val(defvalue);
 }
 });
 }
});
```

Použití pluginu:

```
$(function(){
 $('#SearchText')
 .populateElement(
 'Zde zapiš hledané slovo...'
);
});
```

# Datová cache elementů

- `data( name, value )`
- `data( name )`
- `removeData( name )`



```
$("#div#menu").data("oldColor", "red");
var c = $("#div#menu").data("oldColor");
```

# Řetězení

- **andSelf( )**
  - Přidá předchozí výběr
  - `$("#div").find("p").andSelf()`
- **end( )**
  - Zruší poslední "destruktivní" operaci v řetězci operací
  - `$("#p").find("span").css("background", "yellow").end().css("font-style", "italic")`
    - Zruší důsledek operace find ... kolekce bude obsahovat elementy p

# Utils

- **jQuery.browser**
  - `if ($.browser.safari) { alert("this is safari!"); }`
- **jQuery.browser.version**
- **jQuery.boxModel**
  - false ... IE QuirksMode, true ... jinak

# Utils

- **jQuery.each( object, callback )**
  - Pro každý prvek zavolá callback s parametrem
- **jQuery.extend( target, object1, objectN )**
  - Merge vlastností – upravuje a vrací původní modifikovaný objekt
- **jQuery.grep( array, callback, invert )**
  - Filtruje podle rozhodovací funkce
  - ```
var arr = [ 1, 9, 3, 8, 6, 1, 5, 9, 4, 7, 3, 8, 6, 9, 1 ];  
arr = jQuery.grep(arr, function(n, i){  
    return (n != 5 && i > 4);  
});
```
- **jQuery.makeArray(obj)**
 - Přeměna objektů "podobných poli" (mají length a vlastnosti 0, 1, ...) na pole.
- **jQuery.map(array, callback)**
 - Pole po jednom zpracovaných prvků původního pole pomocí funkce
- **jQuery.inArray(value, array)**
 - Hledání v poli (vrací index nebo -1)
- **jQuery.unique(array)**
 - Ruší duplicity.
- **jQuery.isFunction(obj)**
 - Je parametr funkce?
- **jQuery.trim(str)**
 - Odstraňuje bílé znaky z konců řetězce

Ajax možnosti

- **async** - Boolean
- **beforeSend** – Function
- **cache** – Boolean
- **complete** – Function
- **contentType** – String
(default application/x-www-form-urlencoded)
- **data** – String nebo key/value páry
{ name: "John", time: "2pm" }
- **dataType** – String
(xml, html, script, json, jsonp, text)
- **error** – Function
- **global** – Boolean
(použít globální obsluhy událostí Ajaxu)
- **ifModified** – Boolean
- **jsonp** – String
- **password** – String
- **processData** – Boolean
- **success** – Function
- **timeout** – Number
- **type** – String (GET, POST)
- **url** – String
- **username** – String

JSON

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

Ajax – globální nastavení

- Události:
 - `ajaxComplete(callback)`
 - `ajaxError(callback)`
 - `ajaxSend(callback)`
 - `ajaxStart(callback)`
 - `ajaxStop(callback)`
 - `ajaxSuccess(callback)`
- Ostatní:
 - `jQuery.ajaxSetup(options)`
 - `serialize()`
 - Serializuje name/value elementů input do stringu `k1=v1&k2=v2&...`
 - `serializeArray()`
 - Totéž do JSON

Interoperabilita

- **jQuery.noConflict();**
 - Vrací \$ původní význam.
- **jQuery.noConflict(extreme)**
 - Kompletně odstraňuje význam \$ a jQuery a přesouvá je do zadaného "namespace":
 - ```
var dom = {};
dom.query = jQuery.noConflict(true);
```