

Vývoj Internetových Aplikací

JavaScript

Ing. Michal Radecký, Ph.D.

www.cs.vsb.cz/radecky

Co je to JavaScript

Zdroj: <http://petr.vaclavek.com>

- Skriptovací programovací jazyk (interpretovaný, zpracovává se přímo zdrojový kód) určený pro řešení dynamiky WWW stránek na straně klienta.
- Nejen pro webový frontend.
- Vlastnosti
 - Součást zdrojového kódu HTML (DOM)
 - Multiplatformní
 - Závislý na interpretačním prostředí (prohlížeči)
 - Objektivě orientovaný, ale beztrídí (prototypy)
 - Case-senzitivní
 - Syntaxí podobný jazykům typu C/C++/Java (někdy volnější, Python)
 - Beztypový
 - Není to Java

Historie JavaScriptu

- První představení (LiveScript) v roce 1995 jako součást Netscape Navigatoru a jako reakce na potřebu zajištění interaktivnosti webových stránek jinými prostředky než Java Applety.
- Rychlé rozšíření s ohledem na téměř žádné vstupní požadavky (syntaxe C/C++/Java, žádný kompilátor, atd.)
- Reakce Microsoftu formou VBScriptu, který byl podporován pouze na platformě Windows.
- V roce 1996 byla v rámci IE 3.0 uvedena portace JScript.
- V roce 1997 byla standardizována verze ECMAScript, která poskytovala jádro společné a podporované všemi prohlížeči.
- Prohlížeče od verze 4.0 podporovali DOM (Document-Object Model), nicméně opět v různé implementaci.
- V dnešní době již existuje standardizace W3C DOM a současné prohlížeče zvládají interpretaci JavaScriptu na úrovni tohoto modelu. JavaScript se používá i mimo WWW stránky.
- ECMAScript je standard pro implementaci JavaScriptu (každý rok nová specifikace)

Co umí JavaScript

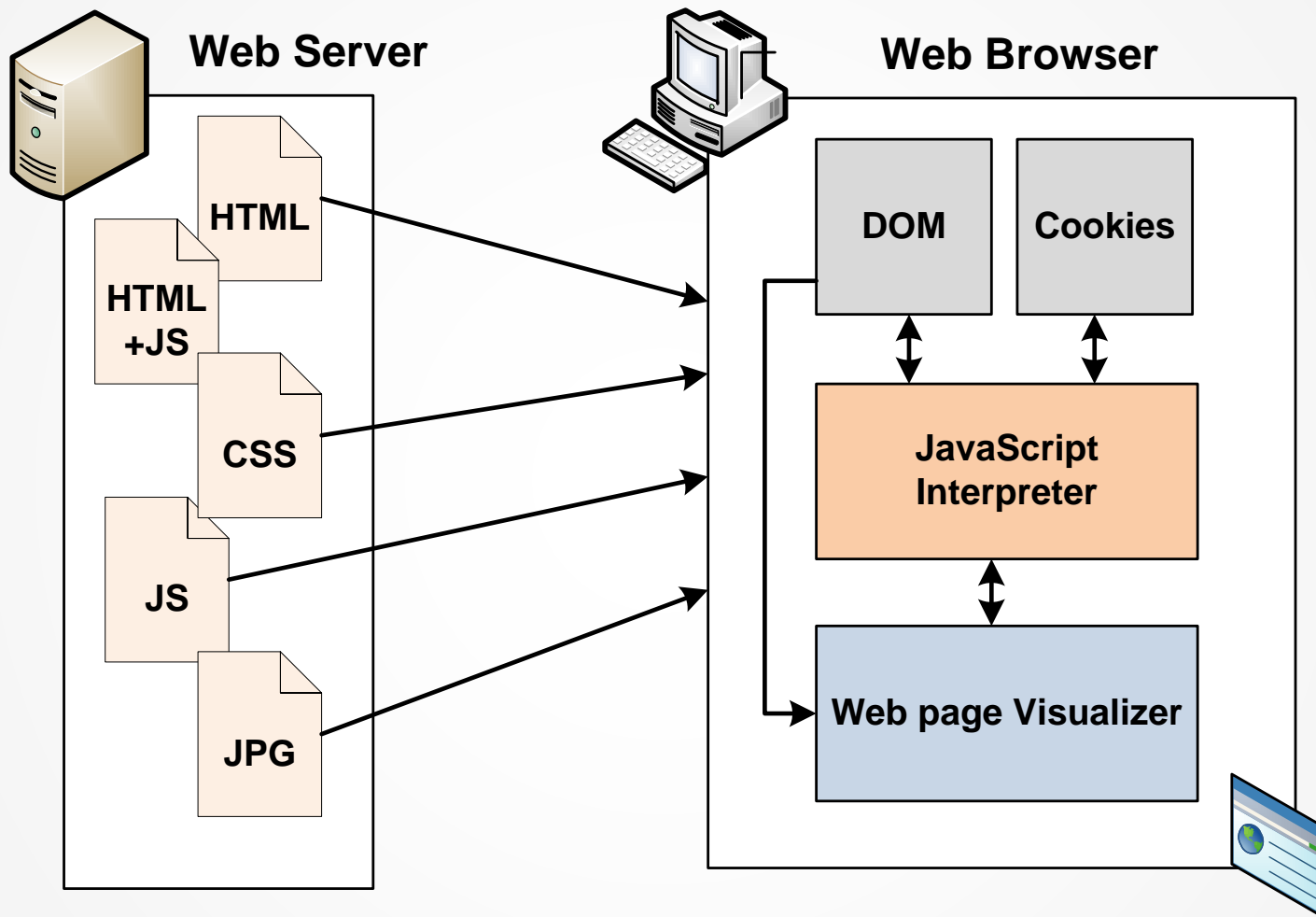
- Současný JavaScript se vyvinul do podoby silného nástroje s inspirací v jazycích Perl, C/C++/Java, Python či TCL.
- Ovlivňovat vzhled a obsah HTML dokumentu (DOM).
- Manipulovat s obrázky a dalšími prvky.
- Částečně řídit a ovlivňovat prohlížeč.
- Provádět algoritmy, matematické výpočty, atd.
- Ovládat a manipulovat s formuláři a jejich daty.
- Zpracovávat události od uživatele.
- Pracovat s úložišti v prohlížeči a dalšími API.
- Ukládat a číst data ve formě Cookies.
- Spolupracovat s externími pluginy.

- Pracovní prostor pro JavaScript je vždy omezen webovým prohlížečem. (V případě využití JavaScriptu pro dynamický web)

Co JavaScript neumí

- Kreslit vektorovou grafiku (?!)
- Přímou pracovat se sítí, pouze využívá možnosti prohlížeče (HTTP, WebSocket)
- Číst a zapisovat do souborů na lokálním počítači
- Samostatně zajistit zabezpečený přístup (autentikaci a autorizaci) na server
- Spouštět aplikace na úrovni OS
- Fungovat, pokud si to uživatel nepřeje.

Jak JavaScript funguje



JavaScript zůstává pořád v podobě zdrojového kódu, stejně jako se nemění zdrojový kód HTML stránky. Mění se pouze jeho vnitřní interpretace.

JavaScript se interpretuje až ve chvíli potřeby a postupně.

DOM (Document Object Model)

- Objektově orientovaná reprezentace XML nebo HTML dokumentu.
- Jedná se o API rozhraní pro objektový přístup k jednotlivým elementům webové stránky a jejich atributům, metodám, atd.
- Využívá se datová struktura stromu.
- Standard W3C DOM, dříve Intermediate DOM (Netscape, `document.layers`) a DHTML OM (Microsoft, `document.all`).
- Standard rozlišuje Levely (0-3), které určují sadu vlastností a funkcí, které DOM daného levelu musí splňovat.

Umístění JavaScriptu

Zdroj: <http://petr.vaclavek.com>

```
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=windows-1250">
    <title>Můj první skript</title>
    <script type="text/javascript" src="knihovna.js"></script>
    <script type="text/javascript">
      <!-- Ukrytí před staršími prohlížeči
      alert ("No nazdar!");
      // Konec ukrytí -->
    </script>
    <noscript>
      Váš prohlížeč nepodporuje JavaScript, pokud by podporoval,
      zobrazil by dialogové okno s textem "No nazdar!".
    </noscript>
  </head>

  <body onload="alert('Načteno!')">
    Obsah mé první "živé" stránky.
    <a href="javascript: alert ('jedna plus jedna je: +(1+1) );">1+1=?</a>
  </body>
</html>
```

onLoad – spouští se, když je stránka kompletně načtena a zobrazena (vč. Obrázků, atd.), problémy s obrázky v CSS a JS

Konstrukce JavaScriptu

```
document.write("Ahoj");
document.write("Tohle 'jsou' uvozovky");
document.write("Tohle \"jsou\" uvozovky\" + \" - zase");
console.log(a);
```

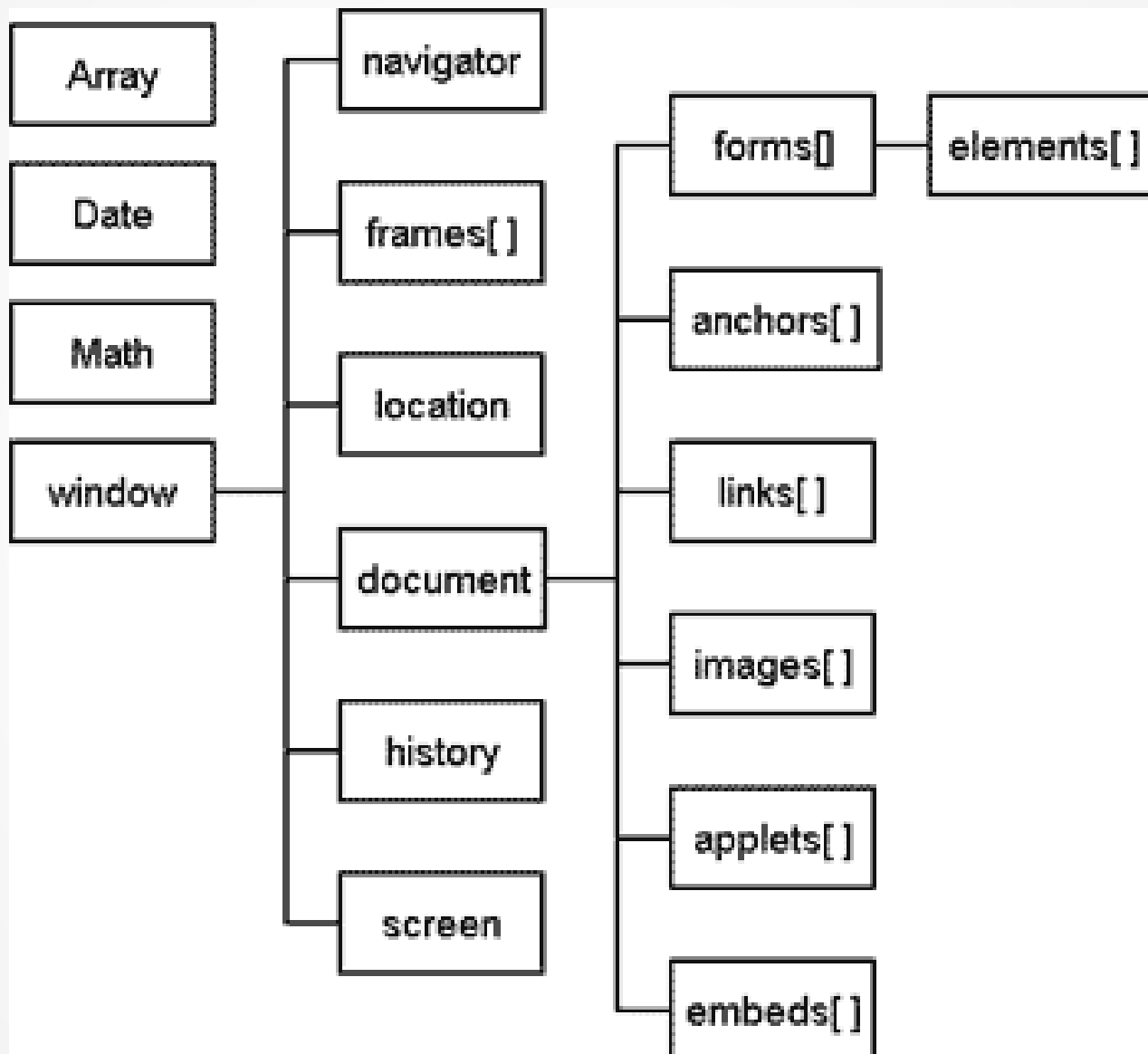
```
var p1 = 10;
var p2 = "10.5";
p3 = "ahoj";
var p4 = true;
document.write(p1 + p2); //1010.5
p2 = 10.5;
document.write(p1 + p2); //20.5
```

```
var pole2 = ["mrkev", "brambory", "kvetak"] //std. jednorozměrné
for(i=0;i<pole2.length;i++){
    document.write(pole2[i] +" ")
}

pole2["br"] = "brambory";

var pole = new Array("HTML", "DHTML", "XHTML");
document.write(pole.valueOf()); //HTML,XHTML,XHTML
document.write(pole.toString()); //"["HTML", "DHTML", "XHTML"]"
```

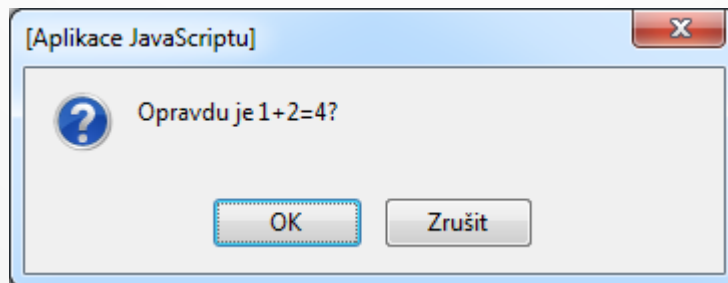
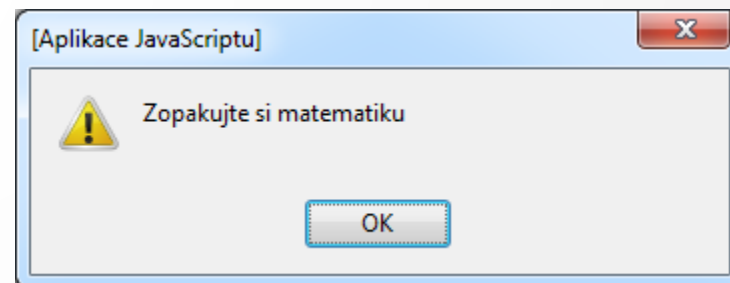
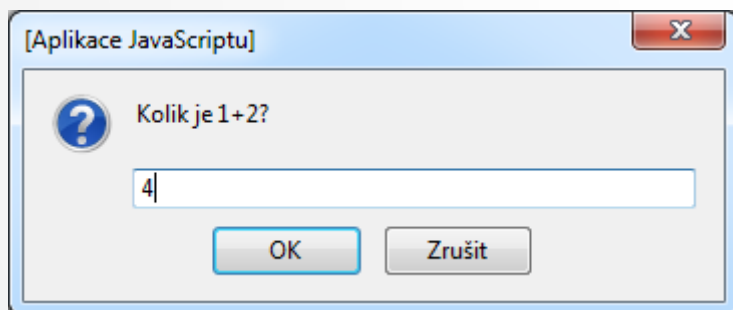
Základní objekty



window.

Zdroj: <http://petr.vaclavek.com>

```
var result = prompt ("Kolik je 1+2?", "4");  
if (result){  
  var conf = confirm ("Opravdu je 1+2=" + result + "?");  
  
  if (conf){  
    alert ("Zopakujte si matematiku");  
  } else  
    alert ("Správně!");  
}
```



location. a history.

Zdroj: <http://petr.vaclavek.com>

```
<script type="text/javascript">
<!--
function delayer() {
    window.location = "http://www.cs.vsb.cz";
}
//-->
</script>
...
<body onLoad=„window.setTimeout('delayer()', 5000)“>
...
```

```
<a href="javascript:history.back();">Zpět</a>
<a href="javascript:history.forward();">Vpřed</a>
<script type="text/javascript">
    <!--
    if (document.referrer != '')
        document.write ('Přišli jste z adresy <a
href="' + document.referrer + '">' + document.referrer + '</a>');
    else
        document.write ('Historie neobsahuje žádné položky nebo je stránka
uložena na lokálním disku. ');
    // -->
</script>
```

navigator.

Zdroj: <http://www.javascriptkit.com>

```
<script type="text/javascript">

if (/MSIE (\d+)\.(\d+);/.test(navigator.userAgent)){ //test for MSIE x.x;
  var ieversion=new Number(RegExp.$1) // capture x.x portion and store as a number
  if (ieversion>=8)
    document.write("You're using IE8 or above")
  else if (ieversion>=7)
    document.write("You're using IE7.x")
  else if (ieversion>=6)
    document.write("You're using IE6.x")
  else if (ieversion>=5)
    document.write("You're using IE5.x")
}
else
  document.write("n/a")
</script>
```

Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 6.1; WOW64; Trident/4.0; SLCC2;
.NET CLR 2.0.50727; .NET CLR 3.5.30729; .NET CLR 3.0.30729; Media Center PC 6.0)

Document.

- Události jednotlivých elementů (onclick, onmouseover, onload, onsubmit, atd.)
- Dotazování v rámci DOMu (rekurzivně)
 - getElementById
 - getElementsByTagName
 - getElementsByClassName
 - querySelector, querySelectorAll
- Tvorba a modifikace DOMu
 - innerText, innerHTML
 - createElement, createTextNode
 - appendChild

Objekty

Zdroj: <http://www.augi.cz/programovani/javascript-ocima-programatora/>

```
var car = { //anonymní objekt
  name : "Honda",
  model : "Civic",
  owner : { name : "Jiri", surname : "Novak" },
  printMe : function() {
    return this.name + ' ' + this.model + ' owned by ' + this.owner.name + ' ' + this.owner.surname;
  },
};
```

```
function Car(carName, model) { //konstruktor
  this.name = carName;
  this.model = model;
  this.printMe = function() {
    return this.name + ' ' + this.model;
  };
}

var car1 = new Car("skoda", "fabia");
```

Dědičnost není přímo podporována, používá se prototype nebo vazby pomocí proměnných.

Objekty

Zdroj: <http://www.augi.cz/programovani/javascript-ocima-programatora/>

```
var hc = new Car();
var sf = new Car("Skoda", "Fabia");

// zajistime, aby vsechny objekty vytvorene pomoci Car mely polozku spz
Car.prototype.spz = 'prvni';
document.write(hc.spz); // 'prvni'
document.write(sf.spz); // 'prvni'

// pri prirazeni se prototype neuplatnuje
hc.spz = 'druha';
document.write(Car.prototype.spz); // 'prvni'
document.write(hc.spz); // 'druha'
document.write(sf.spz); // 'prvni'

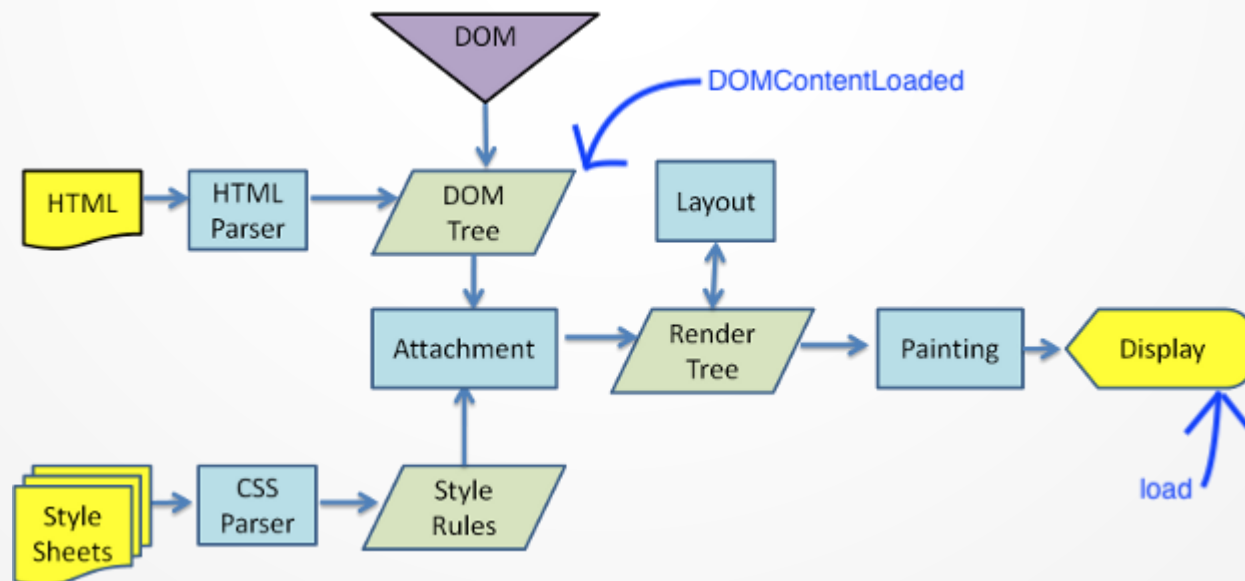
// samozrejme kdyz priradime do prototype...;)
Car.prototype.spz = 'treti';
document.write(Car.prototype.spz); // 'treti'
document.write(hc.spz); // 'druha'
document.write(sf.spz); // 'treti'
```

Prototyp je část objektu, který mají společný všechny objekty vytvořené podle stejného vzoru. Nejprve se prohledává prostor konkrétního objektu, pak prototype.

DOM události

Zdroj: <http://www.augi.cz/programovani/javascript-ocima-programatora/>

```
window.addEventListener("load", (event) => {  
  console.log("page is fully loaded");  
});  
  
document.addEventListener("readystatechange", (event) => {  
  console.log("page DOM is ready in different states");  
});  
  
document.addEventListener("DOMContentLoaded", (event) => {  
  console.log("page DOM is loaded with deferred scripts etc.");  
});
```



Asynchronní programování

- Události – EventListener
 - nemožnost ovlivnit pořadí zpracování, pouze částečně
- Callback funkce
 - řetězení callback funkcí (callback hell), okamžité vykonávání předávaných metod, pouze jedno vyvolání
- Promise objekty, async/await
 - funkcionální přístup
 - budoucí příslib návratu/hodnoty, nezávislé na čase, snadnější zachytávání chyb (z celého řetězce), násobné vykonávání
 - async/await – automatizace Promise konstrukcí

JavaScript frameworky

- Jedná se o JavaScriptové knihovny, které zjednodušují a rozšiřují možnosti standardního JavaScriptu. Díky JS frameworku se může vývojář více soustředit na samotné řešení, nikoliv na optimalizaci a ladění pro různé prohlížeče, apod.
- Jedná se v podstatě o skripty (napsané v JS), které rozšiřují stávající objekty, metody, apod.
- Obvykle mají možnost využití a výběru z velkého množství pluginů, které již řeší obvyklé aktivity a funkce (animace, AJAX, DOM, atd.)
- Dvě základní skupiny
 - JavaScriptové knihovny – rozšíření funkcionality (Prototype, jQuery, MooTools, script.aculo.us,)
 - RIA frameworky – komplexní řešení RIA pomocí JS (extJS, React.js, Angular.js)

JavaScript frameworky

<http://trends.builtwith.com/javascript/javascript-library>
<http://www.google.com/trends/explore?hl=en-US#q=ember%20js%2C%20angular%20js%2C%20backbone%20js%2C%20react%20js%2C%20knockout%20js&cmpt=q&tz=Etc%2FGMT-2>

angular
Hledaný výraz

React
Webový framework

Vue.js
Open source

Ext JS
Software

jQuery
Software

Celosvětově ▾

2004–současnost ▾

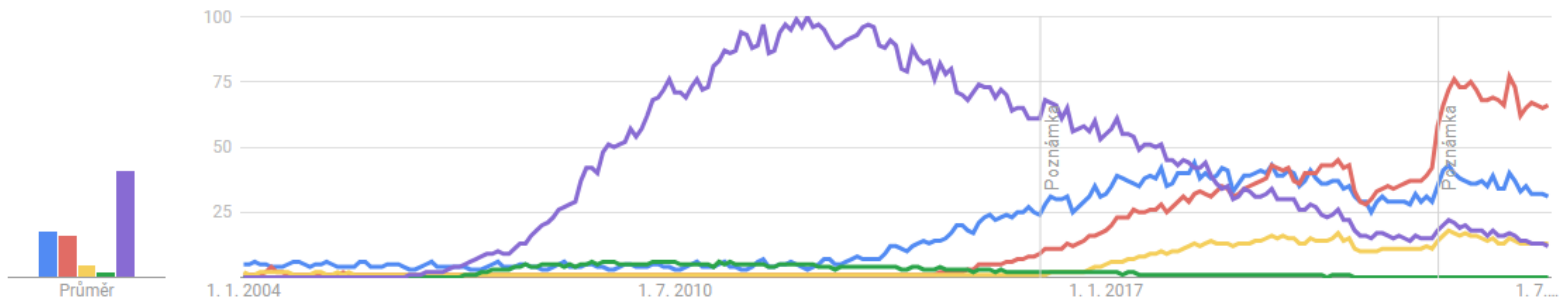
Všechny kategorie ▾

Vyhledávání na webu ▾

! Poznámka: Toto porovnání obsahuje vyhledávací dotazy a témata, která se měří odlišně.

[DALŠÍ INFORMACE](#)

Zájem v průběhu času ?



jQuery

```
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=utf-8">
  <script src="http://ajax.googleapis.com/ajax/libs/jquery/1.3/jquery.min.js" type="text/javascript"></script>
  <script type="text/javascript">
    $(document).ready(function() {
      $("a").click(function(event) {
        alert("As you can see, the link no longer took you to jquery.com");
        event.preventDefault();
      });
    });

  </script>
</head>
<body>
  <a href="http://jquery.com/">jQuery</a>
</body>
```

```
$(document).ready(function() {
  $("#orderedlist li:last").hover(function() {
    $(this).addClass("green");
  }, function() {
    $(this).removeClass("green");
  });
});
```

\$(document).ready – spouští se, když je DOM kompletně načten a připraven (nečeká na obrázky, atd.)