VŠB TECHNICKÁ UNIVERZITA OSTRAVA | FAKULTA ELEKTROTECHNIKY A INFORMATIKY

Vývoj Internetových Aplikací

JavaScript

Ing. Michal Radecký, Ph.D.
www.cs.vsb.cz/radecky

Zdroj: http://petr.vaclavek.com

Co je to JavaScript

Skriptovací programovací jazyk (interpretovaný, zpracovává se přímo zdrojový kód) určený pro řešení dynamiky WWW stránek na straně klienta.

Nejen pro webový frontend.

Zákaldní vlastnosti

- Součást zdrojového kódu HTML (DOM)
- Multiplatformní
- Závislý na interpretačním prostředí (prohlížeči)
- Objektově orientovaný, ale beztřídní (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 formu 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

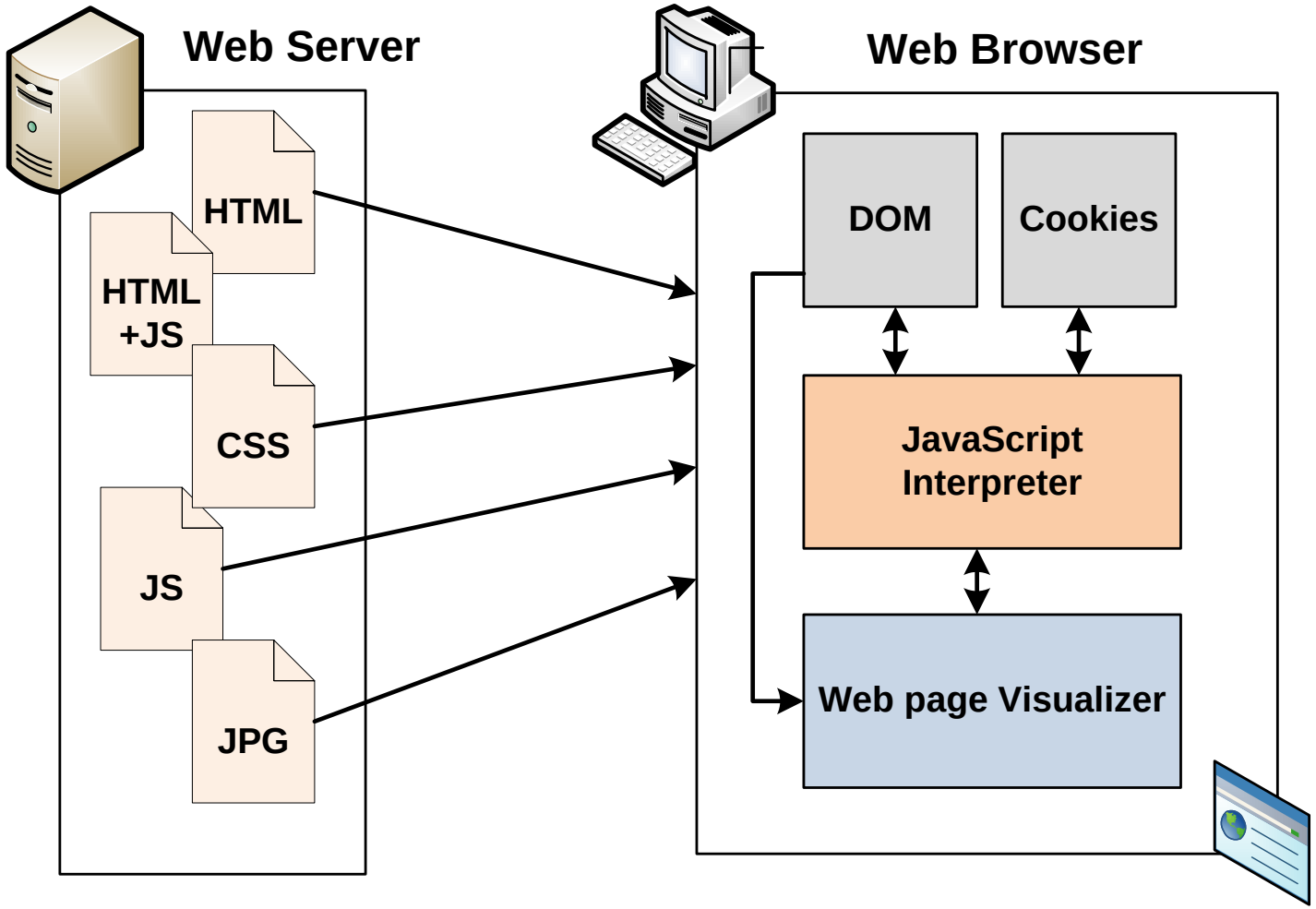
- Současný JavaScript se vyvinul do podoby silného nástroje s inspirací v jazycích Perl, C/C++/Java, Python či TCL.
- Ovlivňovat vzhled a obsah HTML dokumentu (DOM).
- Manipulovat s obrázky a dalšími prvky.
- Částečně řídit a ovlivňovat prohlížeč.
- Provádět algoritmy, matematické výpočty, atd.
- Ovládat 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 (?!)
- Přímo pracovat se sítí, pouze využívá možností 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.

window... the browser window

A page with frames forms a hierarchy of window objects.

Access a window object via: window (the current frame) window.frames[ ] (inner) window.parent (outer) window.top (outermost)

All window members are global and vice versa: any global vars or functions become members of the "current" window.

.alert("message") popup
comment(), prompt()
.back()
.blur() keyboard focus off
.captureEvents(mask) (+Op)
.clearInterval(setIntervalID)
.clearTimeout(setTimeoutID)
.clientInformation...
.appCodeName
.appMinorVersion
.appName
.appVersion
.browserLanguage
.cookieEnabled
.cpuClass
.onLine
.platform
.systemLanguage
.userAgent
.userLanguage
.close()
.closed
.confirm("question") yes/no
.crypto
.defaultStatus
.document
.event
.external
.focus()
.forward()
.frameRate (-N6)
.frames[ ] sub-windows
.handleEvent(event) (-N6)
.history[ ] URLs in history
.history...
.back()
.forward()
.go(±pages)
IE4,5
.current
.next forward URL
.previous back URL
N6
.home()
.innerHeight
.innerWidth
.java
.length =frames.length
.location... =href, =reload()
.hash (what's after the #)
.host e.g. www.
.hostname visibone.com
.href complete URL
.pathname
.port '81' in 'imdb.com:81'
.protocol e.g. "http:"
.search (after the ?)
.reload(bypasscache)
.replace(url) load a page
.locationbar.visible
.menubar.visible
.moveBy(dx,dy)
.moveTo(x,y)
.name
.navigator...
.appCodeName (Mozilla)
.appName (e.g. Netscape)
.appVersion
.cookieEnabled
.javaEnabled()
.plugins[ ]
.mimeTypes[ ]
.language (e.g. en, en-US)
.platform (e.g. Win32)
.userAgent
.userLanguage (e.g. en-us)
.navigate(url) to a new page
.netscape
.offscreenBuffering (-N6)
.onblur losing user focus
.ondragdrop (N4 only)
.onerror JavaScript error
true skips browser popup
.onfocus user click or tab
.onhelp F1=help key

window.open(u,n,o,h)

var winfat=window.open("http://html-tags.info"); (with no options specified, almost all default to yes)

var winlean=window.open("http://html-tags.info", "htmltags","width=400"); (just specify 1 option, and most others default to no)

var wcustom=window.open("http://html-tags.info", "htmltags", "always.owered=no,"+ "alwaysRaised=no,"+ "channelmode=no,"+ "dependent=yes,"+ "directories=no,"+ "fullscreen=no,"+ "height=<npixels>,"+ "hotkeys=yes,"+ "innerHeight=<npixels>,"+ "innerWidth=<npixels>,"+ "left=<npixels>,"+ "location=no,"+ "menubar=no,"+ "outerHeight=<npixels>,"+ "resizable=no,"+ "screenX=<npixels>,"+ "screenY=<npixels>,"+ "scrollbars=no,"+ "status=no,"+ "titlebar=yes,"+ "toolbar=no,"+ "top=<npixels>,"+ "width=<npixels>,"+ "z-lock=no");

.onload <body onload>
.onresize desktop window
.onscroll browser scrollbar
.onunload new page coming
.open(url,name,options,ovrh) url,name act just like href and target in a hyperlink
call window.open() so as not to confuse with document.open() in events.
.opener (window that opened)
.outerHeight
.outerWidth
.Packages[ ]
.pageXOffset
.pageYOffset
.parent containing frame
.personalbar.visible
.print()
.prompt("question","default")
.releaseEvents(mask) (+Op)
.routeEvent(event)
.resizeBy(dwidth,dheight)
.resizeTo(width,height)
.screen...
.availHeight
.availLeft
.availTop
.availWidth
.bufferDepth
.colorDepth 24=true color
.height
.pixelDepth
.updateInterval
.width
.screenLeft
.screenTop
.screenX
.screenY
.scroll(x,y) scrollTo()
.scrollBy(dx,dy)
.scrollbars.visible
.scrollTo(x,y)
.screenLeft
.secure (-N6)
.self (the window itself)
.setInterval(function,ms)
.setTimeout(function,ms)
.setInterval("expression",ms)
.setTimeout("expression",ms)
.status
.statusbar.visible
.sun
.toolbar.visible
.stop()
.top outermost frame

document... web page

Access via: document (any frame).document

(Node members belong here) (form and image names too)

.activeElement
.alinkColor <body alink>
.all[ ] all elements
.anchors[ ] <a name>
.applets[ ] Java <applet> properties are public fields
.bgColor <body bgcolor>
.body... <body> (see Element, Node also)
.aLink (color)
.background (image)
.bgColor
.bgProperties
.bottomMargin
.leftMargin
.link (color)
.noWrap
.onafterprint
.onbeforeprint
.onbeforeunload
.onload
.onselect
.onunload
.rightMargin
.scroll (bars, yes or no)
.text (foreground color)
.topMargin
.vLink (color)
.captureEvents(mask) (+Op)
.charset
.clear() open() and close()
.close() after .write()'s
.cookie permanent store
r: nam=val; nam=val; ...
w: nam=val;expires=date
.createElement(tag) ◊ node
.createStyleSheet(url,index)
.designMode
.domain part of .URL IE5
.doctype <!DOCTYPE>
.documentElement <html>
.elementFromPoint(x,y)
.embeds[ ] <embed>
.expando catch name errors
.execCommand(command)
.fgColor <body text>
.fileCreatedDate
.fileModifiedDate
.fileSize
.fileUpdatedDate NT
.forms[ ] <form>
.frames[ ] window.frames
.getElementById(id)
.getElementsByName(name)
.getElementsByTagName(tag)
.getSelection() user-hilited
.handleEvent(event) (-N6)
.height
.ids[ ] (-N6)
.images[ ] <img>
.implementation
.hasFeature(feature,ver)
.lastModified date & time
.layers[ ] (-N6)
.linkColor <body link>
.links[ ] <a href>
.location window.location
.media
.mimeType
.namespaces
.namespaceURI
.onactivate
.onafterupdate
.onbeforeactivate
.onbeforedeactivate
.onbeforeeditfocus
.onbeforeupdate
.onblur (N6)
.oncellchange
.onclick
.oncontextmenu
.oncontrolselect
.ondataavailable
.ondatasetchanged
.ondatasetcomplete
.ondblclick (-N6)
.ondeactivate
.ondragstart
.onerrorupdate
.onfocus (N6)
.onfocusin
.onfocusout
.onhelp
.onkeydown
.onkeypress
.onkeyup
.onmousedown
.onmousemove (N6)
.onmouseout
.onmouseover
.onmouseup
.onmousewheel
.onpropertychange
.onreadystatechange
.onrowenter
.onrowexit
.onrowsdelete
.onrowsinserted
.onselectionchange
.onselectstart
.onunload
.open() prepare for .write()'s
.parentWindow outer frame
.plugins .embeds[ ]
.previousSibling
.protocol
.readyState
.referrer inbound link
.releaseEvents(mask) (+Op)
.routeEvent(event)
.scripts[ ] <script>
.text JavaScript source
.security
.selection
.styleSheets[ ] <style>
.cssRules
.item(index).cssText
.cssText
.deleteRule(index)
.disabled
.href
.id
.imports
.insertRule(cssrule,index)
.media
.owningElement
.ownerNode
.ownerRule
.pages
.parentStyleSheet
.readOnly
.title
.type
.title <title>
.uniqueID
.URL same as in address bar
.URLUnencoded
.vlinkColor <body vlink>
.width
.write(raw_html_string)
.writeln(raw_html_string)

⌘window.open (u,n,o,h) pops up a new browser window with a web page in it.

document.open()
document.write(string)
document.close()
rewrite from scratch new HTML content for an existing browser window.

style style –N4

Cascading Style Sheets with the names reworded slightly. Value syntax mostly the same.

❖CSS page of HTML Card

Netscape 4 supports merely what's in document.layers[ ]

document.styleSheets[ ]

dimensions need units, e.g. elem.style.top="200 px";

defaults are not exposed, members are empty if not set.

.accelerator
.background
.backgroundAttachment
.backgroundColor
.backgroundImage
.backgroundPositionX
.backgroundPositionY
.backgroundPosition
.backgroundRepeat
.behavior
.border
.borderBottom
.borderBottomColor
.borderBottomStyle
.borderBottomWidth
.borderCollapse
.borderColor
.borderLeft
.borderLeftColor
.borderLeftStyle
.borderLeftWidth
.borderRight
.borderRightColor
.borderRightStyle
.borderRightWidth
.borderStyle
.borderTop
.borderTopColor
.borderTopStyle
.borderTopWidth
.borderWidth
.bottom
.clear
.clip
.color
.cssText
.cursor
.direction
.display "none"=vanishes
.filter
.font
.fontFamily
.fontSize
.fontStyle
.fontVariant
.fontWeight
.height
.imeMode
.layoutFlow (IE5.5)
.layoutGrid
.layoutGridChar
.layoutGridLine
.layoutGridMode
.layoutGridType
.left
.letterSpacing
.lineBreak
.lineHeight
.listStyle
.listStyleImage
.listStylePosition
.listStyleType
.margin
.marginBottom
.marginLeft
.marginRight
.marginTop
.minHeight (IE5.5)
.overflow
.overflowX
.overflowY
.padding
.paddingBottom
.paddingLeft
.paddingRight
.paddingTop
.pageBreakAfter
.pageBreakBefore
.posBottom
.posHeight
.position animation
.posLeft
.posRight
.posTop
.posWidth
.right
.rubyAlign
.rubyOverhang
.rubyPosition
.scrollbar3dLightColor
.scrollbarArrowColor
.scrollbarBaseColor
.scrollbarDarkShadowColor
.scrollbarFaceColor
.scrollbarHighlightColor
.scrollbarShadowColor
.scrollbarTrackColor
.styleFloat
.tableLayout
.textAlign
.textAlignLast (IE5.5)
.textAutospace
.textDecoration
.textDecorationBlink
.textDecorationLineThrough
.textDecorationNone
.textDecorationOverline
.textDecorationUnderline
.textIndent
.textJustify
.textJustifyTrim
.textOverflow
.textTransform
.textUnderlinePosition (IE5.5)
.top
.unicodeBidi
.verticalAlign
.visibility transparent
.whiteSpace
.width
.wordBreak
.wordSpacing
.wordWrap (IE5.5)
.writingMode (IE5.5)
.zIndex
.zoom

Element All the visible parts of an HTML page, including document.body

(in add'n to Node members)

.all[ ]
.behaviorUrns
.canHaveChildren
.canHaveHTML (IE5.5)
.children
.className CSS class(es)
.clientHeight
.clientWidth
.clientLeft .offsetLeft
.clientTop .offsetTop
.contentEditable (IE5.5)
.currentStyle
.dir
.disabled greyed-out (+Op) Input .disabled
.document
.filters
.getAttributeNode(name)
.getAttribute(name)
.getElementsByTagName(n)
.id XHTML says to use in tandem with .name
.innerHTML what's between start & end tags N6 body
.innerText subtags stripped
.isContentEditable (IE5.5)
.isDisabled (IE5.5)
.isMultiLine (IE5.5)
.isTextEdit
.lang
.language
.name .id
.offsetWidth
.offsetHeight
.offsetTop relative to
.offsetLeft parent element
.offsetParent
.onactivate (IE5.5)
.onafterupdate
.onbeforeactivate (IE5.5)
.onbeforecopy
.onbeforecut
.onbeforedeactivate (IE5.5)
.onbeforeeditfocus
.onbeforepaste
.onbeforeupdate
.oncellchange
.onclick
.oncontextmenu
.oncontrolselect (IE5.5)
.oncopy
.oncut
.ondataavailable
.ondatasetchanged
.ondatasetcomplete
.ondblclick (+N4)
.ondeactivate (IE5.5)
.ondrag
.ondragend
.ondragenter
.ondragleave
.ondragover
.ondragstart
.ondrop
.onerrorupdate
.onfilterchange
.onlayoutcomplete (IE5.5)
.onlosecapture
.onmousedown
.onmouseenter (IE5.5)
.onmouseleave (IE5.5)
.onmousemove (-N4)
.onmouseout
.onmouseover roll-overs
.onmouseup
.onmousewheel (IE5.5)
.onmove (IE5.5)
.onmoveend (IE5.5)
.onmovestart (IE5.5)
.onpage (IE5.5)
.onpaste
.onpropertychange
.onreadystatechange
.onresizeend (IE5.5)
.onresizestart (IE5.5)
.onrowenter
.onrowexit
.onscroll
.onselectstart
.outerHTML
.outerText
.parentElement
.parentTextEdit
.readyState
.removeAttribute(name)
.removeAttributeNode(attr)
.runtimeStyle
.scopeName
.scrollWidth
.scrollHeight
.scrollLeft (+Op for body)
.scrollTop (+Op for body)
.setAttribute(name,value)
.setAttributeNode(attr)
.sourceIndex
.style
.tagUrn
.tagName
.title

inputable elements <input> <select> <textarea> <a href>

Things you can click, tab to, or type on.

Access via: document.idform.idelement document.forms[ ].elements[ ] document.links[ ] document.getElementById() document.all[ ]

Loose association: members belong to at least 1 inputable. (in add'n to Element members)

.accessKey shortcut
.align (-N4)
.blur() focus elsewhere
.checked (checkbox, radio)
.click() simulated mouse
.cols (textarea) (-N4)
.dataFld
.dataFormatAs
.dataSrc
.defaultChecked read-only,
.defaultSelected used at
.defaultValue reset time
.disabled grayed-out
.focus()
.form
.length (of select's options)
.maxLength (typable text)
.multiple (select) (-N4)
.onblur
.onchange
.onclick
.ondblclick (+N4)
.onfocus
.onfocusin
.onfocusout
.onhelp
.onkeydown
.onkeypress
.onkeyup
.onselect (-N4)
.options[ ] (select)
.select() swipe visible text
.selected <option>
.selectedIndex (select, 0...)
.size (visible text width)
.readOnly
.recordNumber
.remove()
.rows (-N4) (textarea)
.tabIndex
.type
.value
.wrap (textarea)

Node In IE5 & N6, an HTML page is a hierarchical tree of nodes, with the document object at the root. document.createElement or document.createTextNode can make new nodes.

.nodeType 9=doc, 1=element, 2=attribute, 3=text, etc.
.nodeName id or name
.nodeValue
.childNodes[ ]
.firstChild
.lastChild
.nextSibling
.previousSibling
.parentNode Op
.ownerDocument root (-IE5)
.attributes element subnodes
.cloneNode(descendentstoo)
.hasChildNodes()
.insertBefore(nnew,nref)
.appendChild(nnew)
.removeChild(nnref)

Image <img>

Access: document.imgname (in add'n to Element members)

.align
.alt
.border
.complete
.dynsrc
.fileCreatedDate
.fileModifiedDate
.fileSize
.fileUpdatedDate
.height
.hspace
.isMap
.longDesc
.loop
.lowsrc
.mimeType
.onabort()
.onerror()
.onload()
.protocol
.src prefetch for rollovers: (new Image).src="x.gif"; then change on the fly
.start
.useMap
.vrml
.vspace
.width
.x Element .offsetLeft
.y Element .offsetTop

form element

<form> Access via document.forms[index] or document.formname

(Element and Node members are form members also)

.action
.elements[ ] inputables
.encoding enctype
.method
.onreset
.onsubmit
.target

anchor element

<a name> Access via document.anchors[index]

(Element and Node members are anchor members also)

.text betw. start & end tags
.x location on page (N4 only)
.y

Event object

details of the occurrence

.altKey
.altLeft (IE5.5)
.behaviorCookie (IE5.5)
.behaviorPart (IE5.5)
.bookmarks
.boundElements
.bubbles true=propagates
.button mouse button gen-erally: 0,1=left 2=right
.cancelBubble
.cancelable w/preventDefault
.charCode
.clientX click point within
.clientY browser window
.ctrlKey
.contentOverflow (IE5.5)
.ctrlLeft (IE5.5)
.currentTarget handler (+Op)
.data dropped URLs (N4)
.dataFld
.dataTransfer
.detail e.g. click count
.eventPhase
.fromElement
.getPreventDefault()
.height (N4 only)
.initEvent(type,c,c) (+Op)
.initKeyEvent(type,c,c,...)
.initMouseEvent(type,c,c,...)
.initUIEvent(type,c,c,...)
.isChar
.keyCode
.layerX local click point
.layerY or dimensions
.metaKey
.modifiers (N4 only)
.nextPage (IE5.5)
.offsetX
.offsetY
.originalTarget
.pageX click point within
.pageY browser window
.preventBubble()
.preventCapture()
.preventDefault() (+Op)
.propertyName
.qualifier
.rangeOffset
.rangeParent
.reason
.recordset
.relatedTarget (+Op)
.repeat
.returnValue
.screenX screen click point
.screenY
.shiftKey
.shiftLeft (IE5.5)
.srcElement
.srcFilter
.srcUrn
.stopPropagation() (+Op)
.target generating element
.timeStamp
.toElement
.type e.g. "click" for onclick
.view (window)
.wheelDelta (IE5.5)
.which mouse button gen-erally: 1=left 3=right
.width (N4 only)
.x local click point
.y

Besides all the diversity within an Event object, there are 3 ways just to get the thing:

1. "event" pseudo-parameter in the attribute code (N)
2. window.event (IE, Op)
3. parameter to an assigned handler function (N, Op)

The following HTML example neatly mixes ways 1 and 2:

```
<form id="frm" name="frm">
  <input id="btn" name="btn"
    type="button"
    onmousedown=
    "return mdown(event);" />
</form>
<script>
function mdown(e) {
  if (e.button == 2 ||
      e.which == 3) {
    alert("right-click");
  }
  else {
    alert("left-click");
  }
  return false;
  // false=browser ignore
} // true=browser handle it
</script>
```

This script uses way 3 or 2:

```
function mdown2(e) {
  if (e != null) {
    return mdown(e);
  }
  if (window.event != null) {
    return mdown(
      window.event);
  }
}
document.frm.btn
  .onmousedown=mdown2;
```

Layer Netscape 4 OmniWeb

<layer> and some others: <a>

if style="position:relative" or "position:absolute"

Access by name or number: document.layers.name document.layers[index]

.above
.background (e.g. "bg.gif")
.below
.bgColor (e.g. "#FFFFFF")
.captureEvents(mask)
.handleEvent(event)
.clip...
.height .left .top .right
.width .bottom
.document (HTML within) .open() .write() .close() to rewrite <layer> contents.
.hidden (true or false)
.layers[ ] contained layers
.left
.load(url,width)
.moveAbove(layerobj)
.moveBelow(layerobj)
.moveBy(dx,dy)
.moveTo(x,y) (within layer)
.moveToAbsolute(x,y) (page)
.name
.offset(dx,dy)
.pageX
.pageY
.parentLayer
.releaseEvents(mask)
.resizeBy(dw,dh) changes
.resizeTo(w,h) clipping
.routeEvent(event)
.siblingAbove other layers
.siblingBelow in z order
.src
.top
.visibility
.window (containing window)
.x
.y
.zIndex z-index (high=top)

Zdroj: http://petr.vaclavek.com

Umístění JavaScriptu

```
<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
```

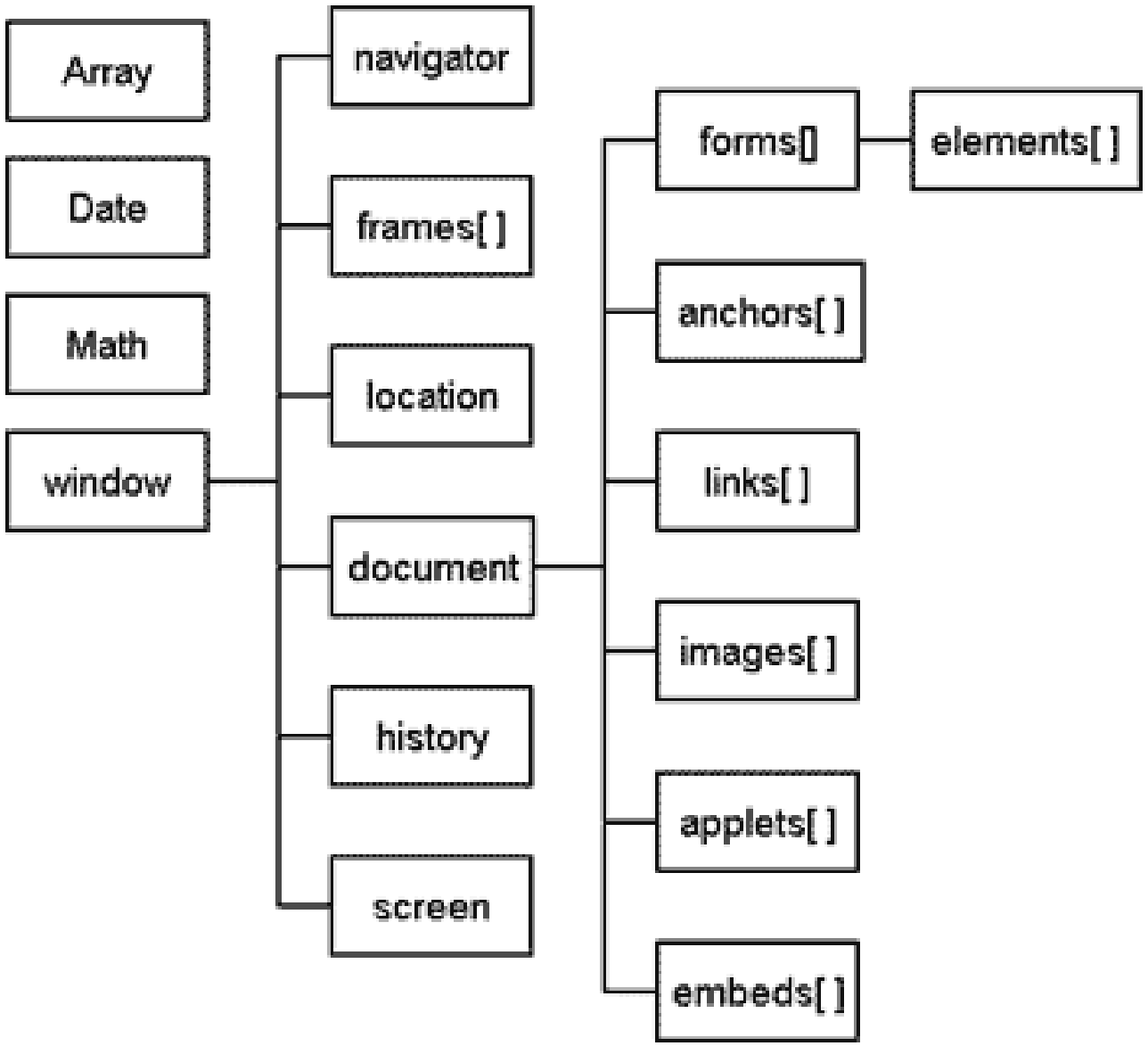
```
var pole2 = ["mrkev", "brambory", "kvetak"] //std. jednorozměrné
for(i=0;i<pole2.length;i++){
   document.write(pole2[i] +" ")
}

pole2[“br”] = “brambory”;

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

Základní objekty

Zdroj: http://petr.vaclavek.com

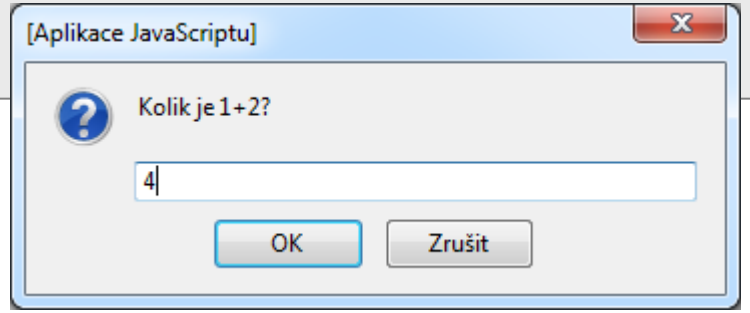


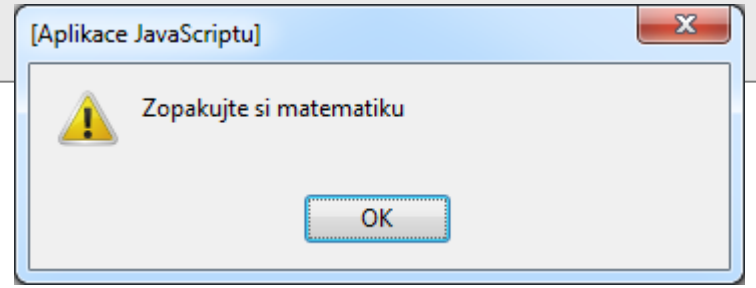
Zdroj: http://petr.vaclavek.com

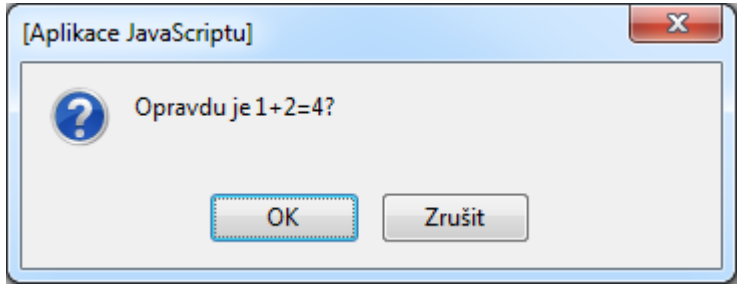
window.

```
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!");
}
```







Zdroj: http://petr.vaclavek.com

location. a history.

```
<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>
```

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

Zdroj: http://www.javascriptkit.com

navigator.

```
<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

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

Objekty

```
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.

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

Objekty

```
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'
```

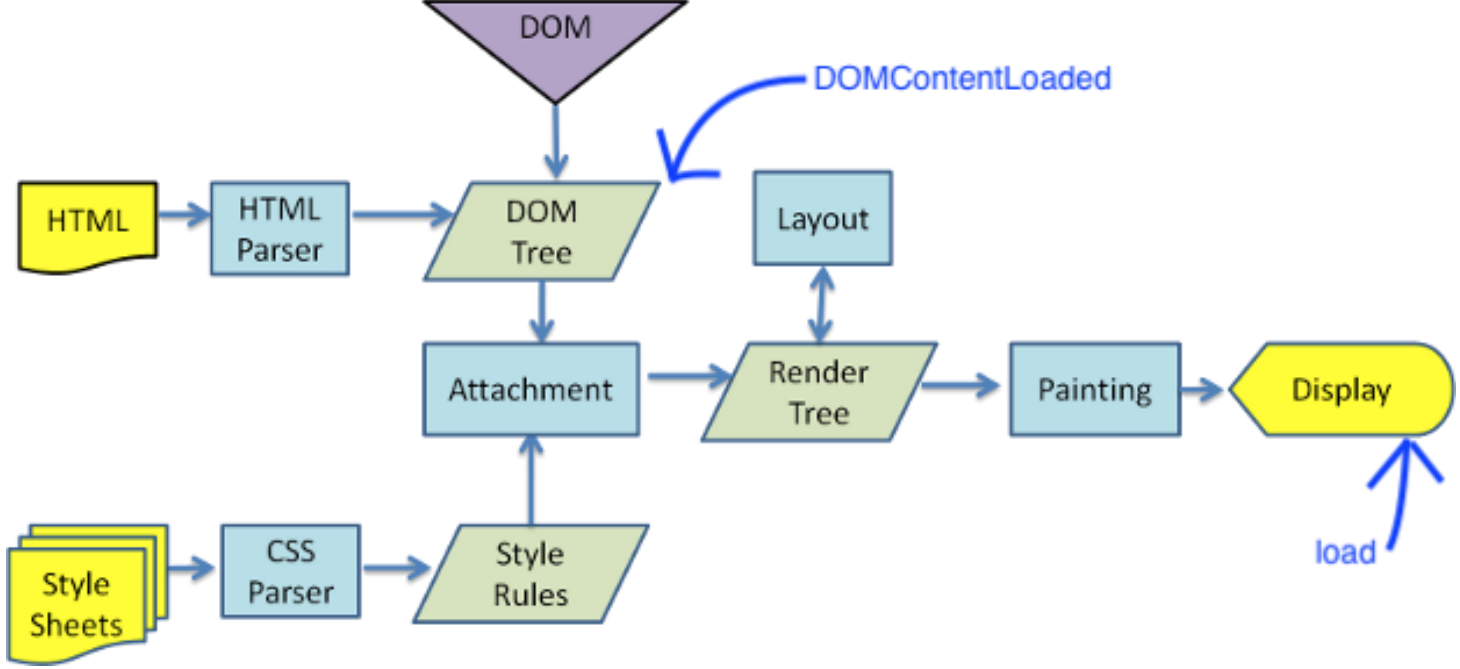
Prototyp je část objektu, který mají společný všechny objekty vytvořené podle stejného vzoru. Nejprve se prohledává prostor konkrétního objektu, pak prototype.

Třídy

- Možné používat od ECMAScript 2015 (ES6), postupně se rozvíjí
- Jen syntaktické řešení nad prototypovým děděním
- Constructor, extends, super, get/set, static, …

```
class Car {
  constructor(name, year) {
    this.name = name;
    this.year = year;
  }
  age(x) {
    return x - this.year;
  }
}
const myCar = new Car("Ford", 2014);
document.getElementById("demo").innerHTML=
"My car is " + myCar.age(2024) + " years old.";
```

DOM události




```
window.addEventListener("load", (event) => {
  console.log("page is fully loaded with resources");
});

document.addEventListener("DOMContentLoaded", (event) => {
  console.log("page DOM is loaded without resources etc.");
});

document.addEventListener("readystatechange", (event) => {
  console.log("page DOM in different states");
});
```

Zdroj: https://javascript.info/callbacks/

Asynchronní programování

- Události – EventListener
  - pořadí zpracování (bubbling vs. capturing)
  - stopPropagation (zastavení propagace skrz DOM), preventDefault (neprovedení defaultního chování)
  - callback funkce
- Callback funkce
  - umožňují provádět operace, které nejsou okamžitě dokončeny
  - Anonymní funkce, lambda zápis
  - Synchronní vs. Asynchronní
  - Funkce vyššího řádu
  - Problematické zpracování chyb
  - Řetězení callback funkcí (callback hell)

Asynchronní programování

- Promise objekty
  - funkcionální přístup
  - Operace, která může být úspěšně dokončena v budoucnosti (příslib) nebo selhat
  - snadnější zpracování chyb (z celého řetězce)
  - násobné vykonávání
- async/await
  - Automatizace Promise konstrukcí
  - async – označuje funkci jako asycnhronní
  - await – čeká na vyřešení nebo odmítnutí Promise

Zdroj: https://javascript.info/callbacks/

```
let myPromise = new Promise(function(myResolve, myReject) {
  let x = 0; // The producing code (this may take some time)
  if (x == 0) {
    myResolve("OK");
  } else {
    myReject("Error");
  }
});

myPromise.then(
  function(value) {myDisplayer(value);},
  function(error) {myDisplayer(error);}
);

async function myDisplay() {
  let myPromise = new Promise(function(resolve) {
    resolve("I love You !!");
  });
  document.getElementById("demo").innerHTML = await myPromise
}
myDisplay();
```

Asynchronní programování

Vlastnost	Callbacky	Promises	Async/Await
Styl kódu	Zanořené funkce (callback hell)	Řetězení pomocí `then()`	Lineární kód jako synchronní
Správa chyb	Náročné, chyby se musí spravovat v každém callbacku	Použití `catch()`	Použití `try/catch`
Čitelnost kódu	Méně čitelný a přehledný	Čitelnější díky `then` / `catch`	Nejčistší, vypadá jako synchronní
Komplexita	Zvyšuje se s rostoucím počtem callbacků	Snadnější než callbacky	Nejsnazší a nejpřehlednější
Podpora chycení více chyb	Obtížná	Možné pomocí `catch()`	Snadné díky `try/catch`

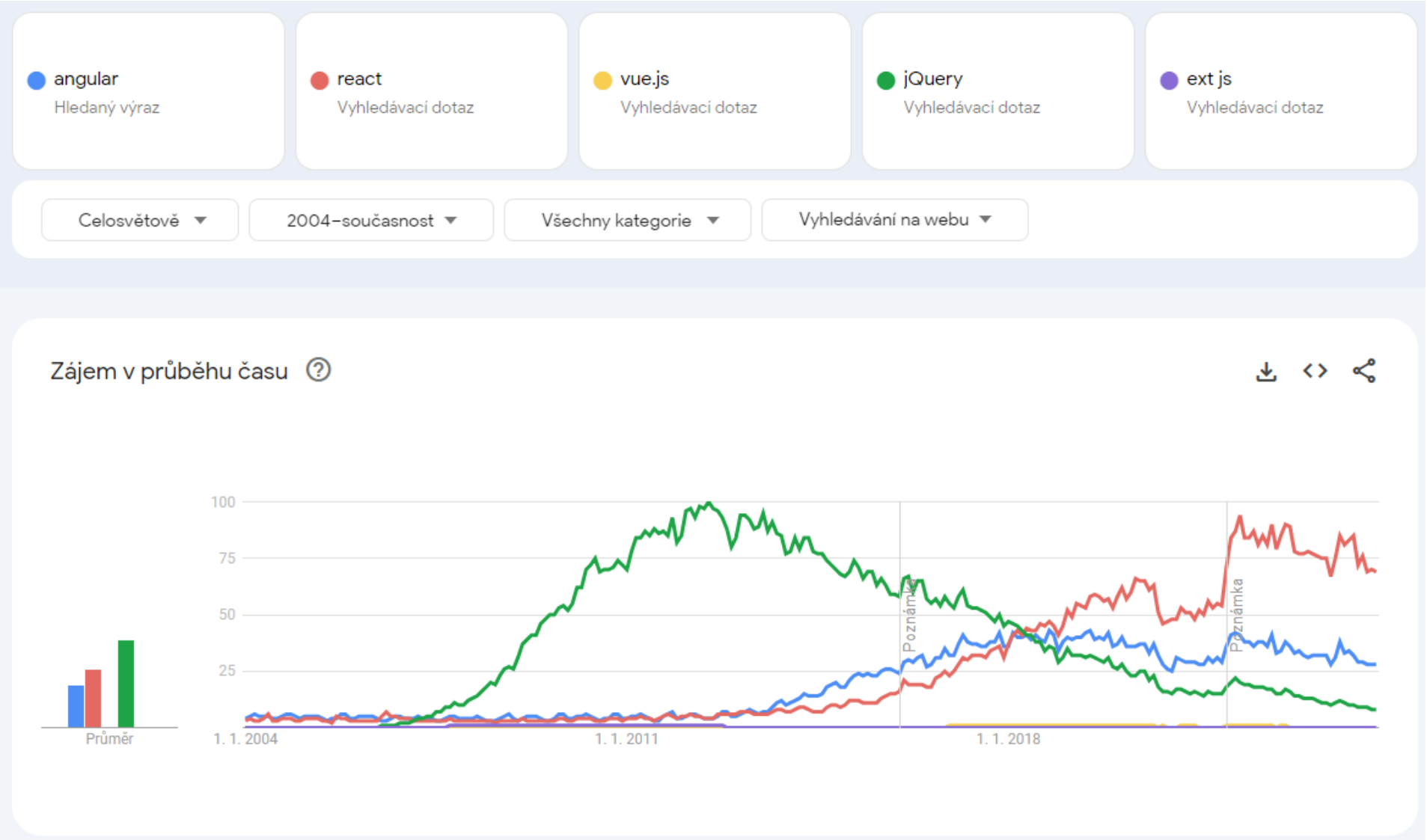
JavaScript frameworky

- 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



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.)