VŠB TECHNICKÁ UNIVERZITA OSTRAVA | FAKULTA ELEKTROTECHNIKY A INFORMATIKY

Development of Internet Applications

JavaScript

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

Zdroj: http://petr.vaclavek.com

What is JavaScript

Scripting programming language (interpreted) developed for dynamical behavior of WWW pages on the client side.

Features

- A part of HTML source code (DOM)
- Multiplatform
- Depended on interpreter (web browser, V8/NodeJS, atc…)
- Objected oriented, class less (prototypes)
- Case-sensitive
- Similar syntax to C / C++ / Java / Python
- Weakly typed
- Nothing to do with Java

History of JavaScriptu

- Introduced in 1995 as part of Netspace Navigator (as a LiveScript).
- Microsoft in response to LiveScript introduced their own language called VBScriptu (only supported on Windows).
- In 1996 Microsoft introduced IE 3.0 with support of JScript (Microsoft implementation of ECMAScript Edition 3).
- In 1997 ECMAScript was standardized as a core for moder web browsers.
- ECMAScript is todays standard for JavaScript implementation (ESMAScript is standard and JavaScript is implementation of this standard).

What JavaScript can do

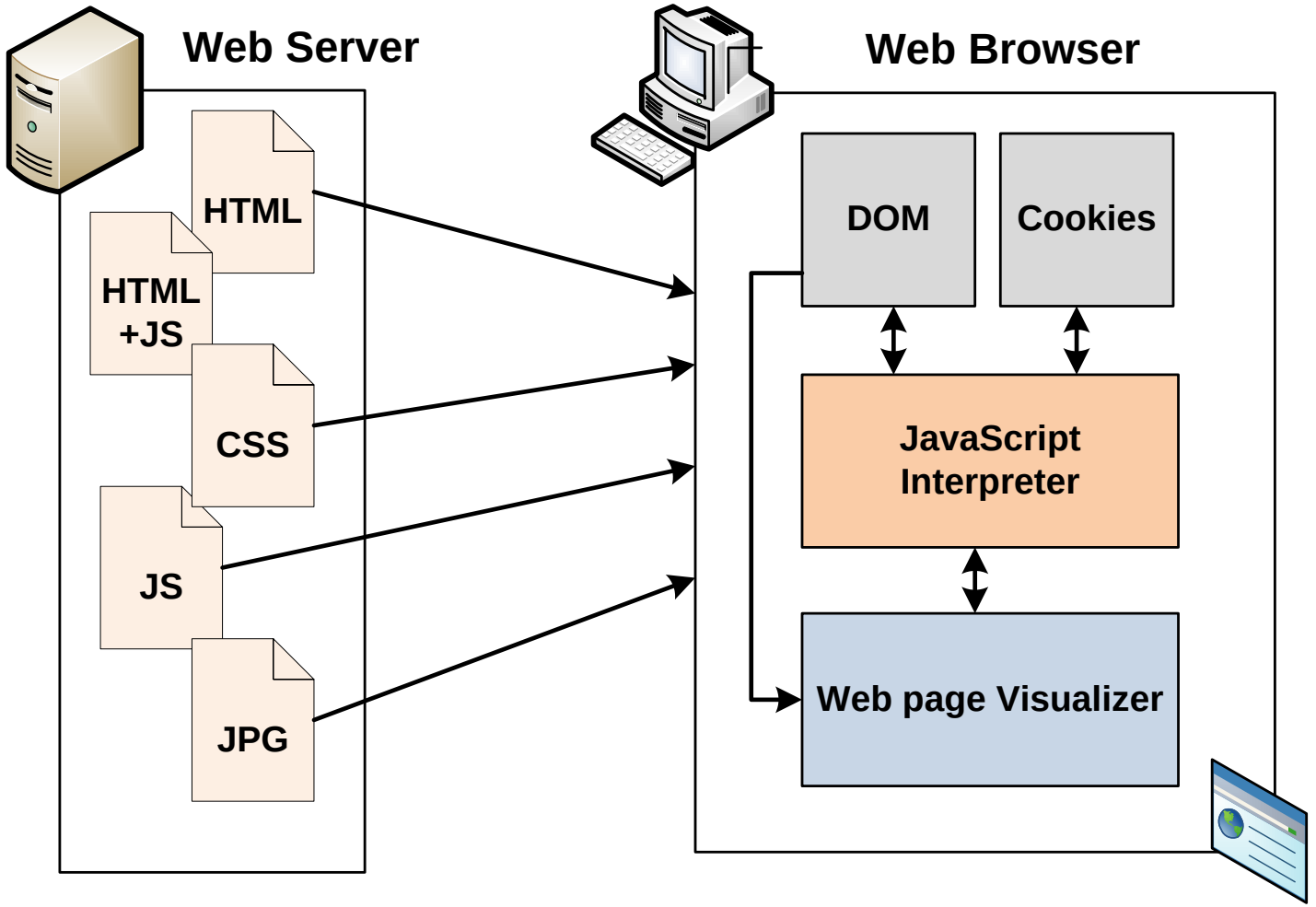
- Current JavaScript is powerful tool based on ideas of Perl, C/C++/Java or TCL.
- Influence of appearance and content of HTML document (DOM).
- Manipulation with images and other elements.
- Control web browser partially.
- Execute algorithms, calculations, etc.
- Control and manipulate with forms and theirs values.
- User events processing.
- Store and read a data in the form of Cookies.
- Collaborate with Flash, Java Applets and other plugins.

Work environment of JavaScript is always limited by web browser (in the case, that we use JavaScript within the web browser).

What JavaScript can not do

- Draw vector graphics (it is not 100% true in HTML 5)
- Work directly with network resources (only utilize web browser capabilities – HTTP, WebScoket).
- Read and write to the local files (it is not 100% true in HTML 5).
- Autonomously provide secure access to server (authentification and authorization).
- Execute applications on OS level.
- Operate if user don’t want it.

How JavaScript works



JavaScript is still in the form of source code (similar to HTML source code).

DOM (Document Object Model)

- Object-oriented representation of an XML or HTML document.
- It is an API for object-oriented access to individual elements of a web page and their attributes, methods, etc.
- The tree data structure is used.
- W3C DOM standard, formerly Intermediate DOM (Netscape, document.layers) and DHTML OM (Microsoft, document.all).
- The standard distinguishes Levels (0-3), which specify a set of features and functions that the DOM of a given level must satisfy.

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(±npages)
IE4,5
.current
.next forward URL
.previous back URL
N6
.home()
.innerHeight
.innerWidth
.java
.length =frames.length
.location... r.href, w.reload()
.hash (what's after the #)
.host
.hostname e.g. www.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", "alwaysLowered=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 <html> (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
.referer inbound link
.releaseEvents(mask) (+Op)
.routeEvent(event)
.scripts[ ] <script> Jscript
.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, used at
.defaultSelected reset time
.defaultValue
.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 Omni/Web

<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
.width .bottom .right
.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

JavaScript in page

```
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=windows-1250">
    <title>My first JavaScript powered page</title>
          <script type="text/javascript" src="library.js"></script>
    <script type="text/javascript">
                  alert („Hello world!");
    </script>
          <noscript>
                    This part is displayed if JavaScript is disabled.
          </noscript>
  </head>

  <body onload="alert(‚Loaded!')">
     Standard HTML content
  <a href="javascript: alert (one plus one is: '+(1+1));”>1+1=?</a>
  </body>
</html>
```

onLoad – is dispatched when page and all resources are loaded (images, styles, atc…)

JavaScript constructions

```
document.write("Hello");
document.write(„These 'are' quotas");
document.write(„These \"are\" quotas" + " – again”);
Console.log(a);
```

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

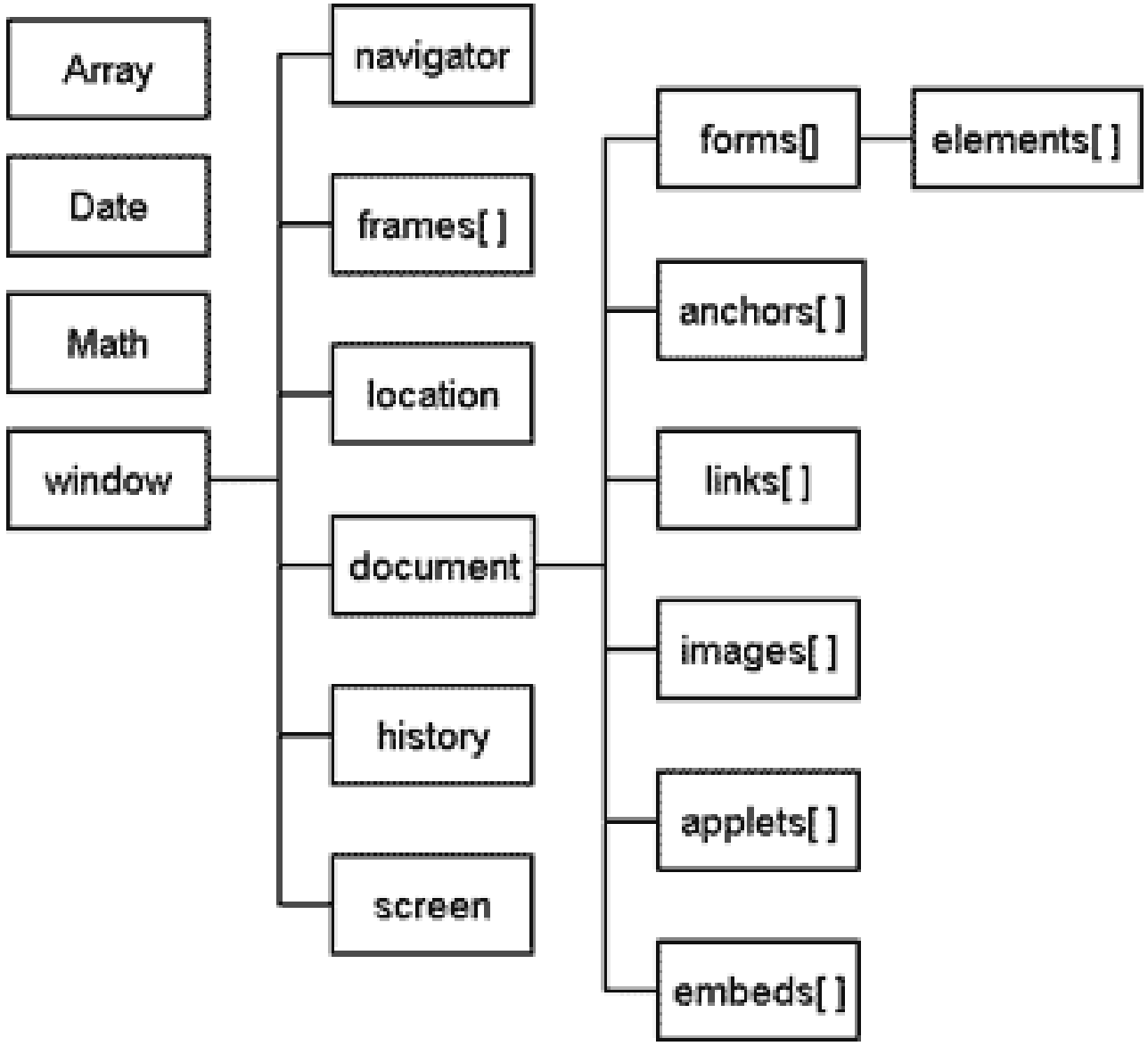
```
var array2 = ["carrot", "potatoes", "cauliflower"] //std. one-dimensional
for(i=0; i < array2.length; i++){
   document.write(array2[i] +" ")
}

array2["br”] = "potatoes”;

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

Zdroj: http://petr.vaclavek.com

Basic objects

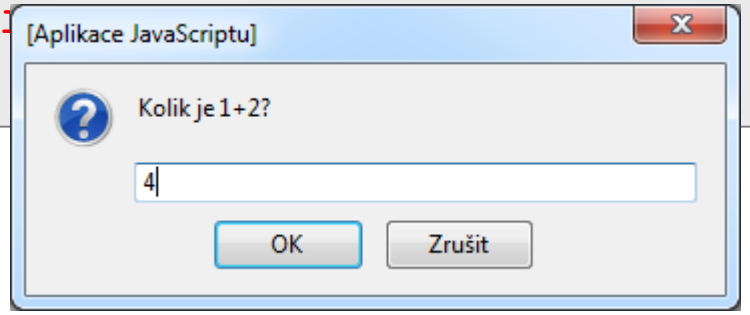


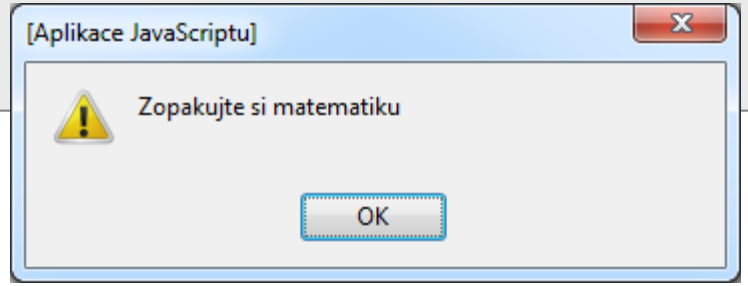
Zdroj: http://petr.vaclavek.com

window.

```
            var result = prompt („How much is 1+2?", "4");
    if (result){
      var conf = confirm („Confirm that 1+2=" + result + "?");


      if (conf){
        alert („Go to school");
      } else
        al
    }
```





Zdroj: http://petr.vaclavek.com

location. and history.

```
<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 (,Your refferer is <a
   href="'+document.referrer+'">'+document.referrer+'</a>');
      else
        document.write ('History does not contain any items or you work with local web page.');
      // -->
</script>
```

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.

- Individual element events (onclick, onmouseover, onload, onsubmit, etc.)
- DOM querying (recursively)
  - getElementById
  - getElementsByTagName
  - getElementsByClassName
  - querySelector, querySelectorAll
- DOM creation and modification
  - innerText, innerHTML
  - createElement, createTextNode
  - appendChild

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

Objects

```
var car = {   // anonymous object
    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) { // constructor
    this.name = carName;
    this.model = model;
    this.printMe = function() {
        return this.name + ' ' + this.model;
    };
}

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

Inheritence is not directly supported, there are prototype or variables relations.

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

Objects

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

// ensure that all objects created with Car have attribute spz
Car.prototype.spz = ,first';
document.write(hc.spz); // ,first'
document.write(sf.spz); // ,first'

// the prototype is not taken into account during assignment
hc.spz = ,second';
document.write(Car.prototype.spz); // 'second'
document.write(hc.spz); // ,second'
document.write(sf.spz); // ,second'
// of course, if we assign to the prototype...;)
Car.prototype.spz = ,third';
document.write(Car.prototype.spz); // ,third'
document.write(hc.spz); // ,second'
document.write(sf.spz); // ,third'
```

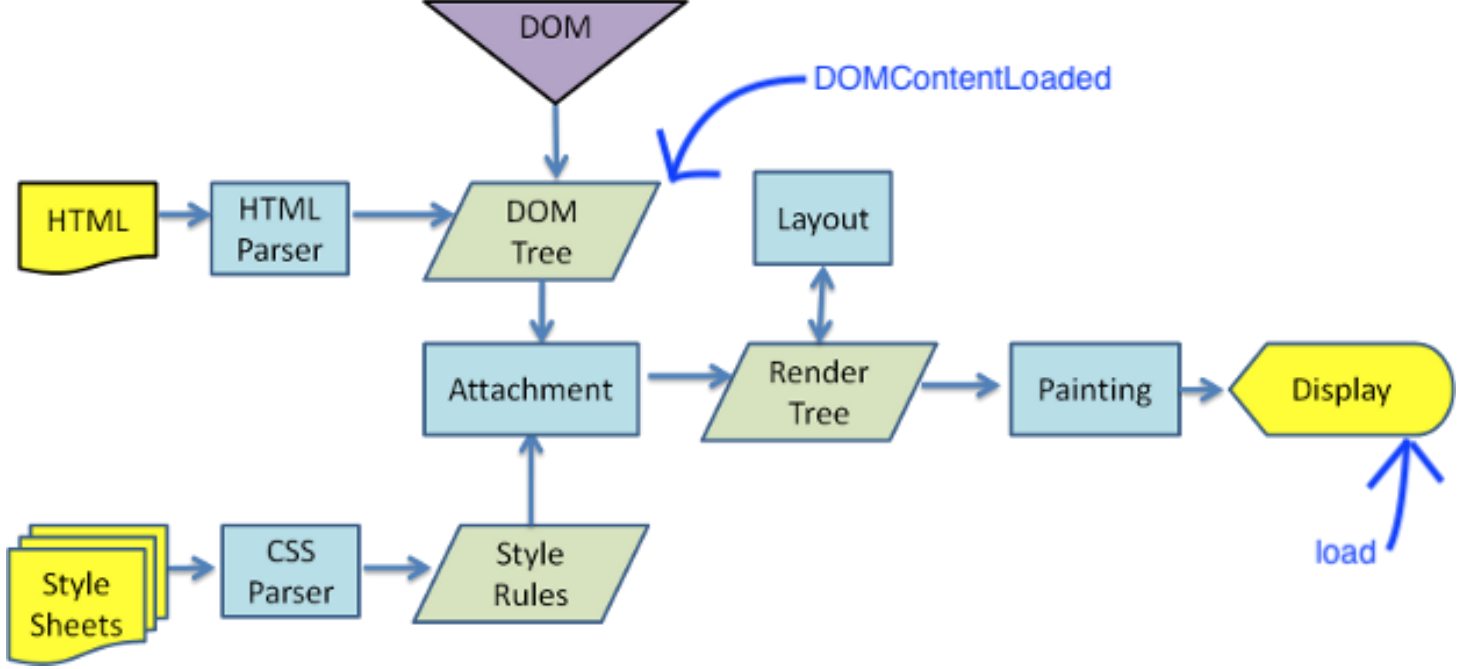
Prototyp is a part of each object that is the same for all objects created using the same description.
First, the particular object is checked, than the prototype.

Classes

- Available from ECMAScript 2015 (ES6), continuously evolving
- Only syntax sugar based on prototyping
- 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 events




```
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/

Asynchronous programming

- EventListener
  - Order of processing (bubbling vs. capturing)
  - stopPropagation (in DOM), preventDefault (default behavior of element)
  - callback funkce
- Callback funkce
  - Can performer functions that are not immediately finished
  - Anonymous functions, lambda syntax
  - Synchronous vs. Asynchronous
  - Higher-level functions
  - Hard errors processing
  - Chain of callback functions (callback hell)

Asynchronous programming

- Promise objects
  - Functional approach
  - Operation, that can successfully finish in the future (promise) or failed
  - Easy errors processing (from whole chain)
  - Multiple firing
- async/await
  - Automatization of Promise contructs
  - async – labels function as asynchronous
  - await – waits for resolving or rejecting

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();
```

Asynchronous programming

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`

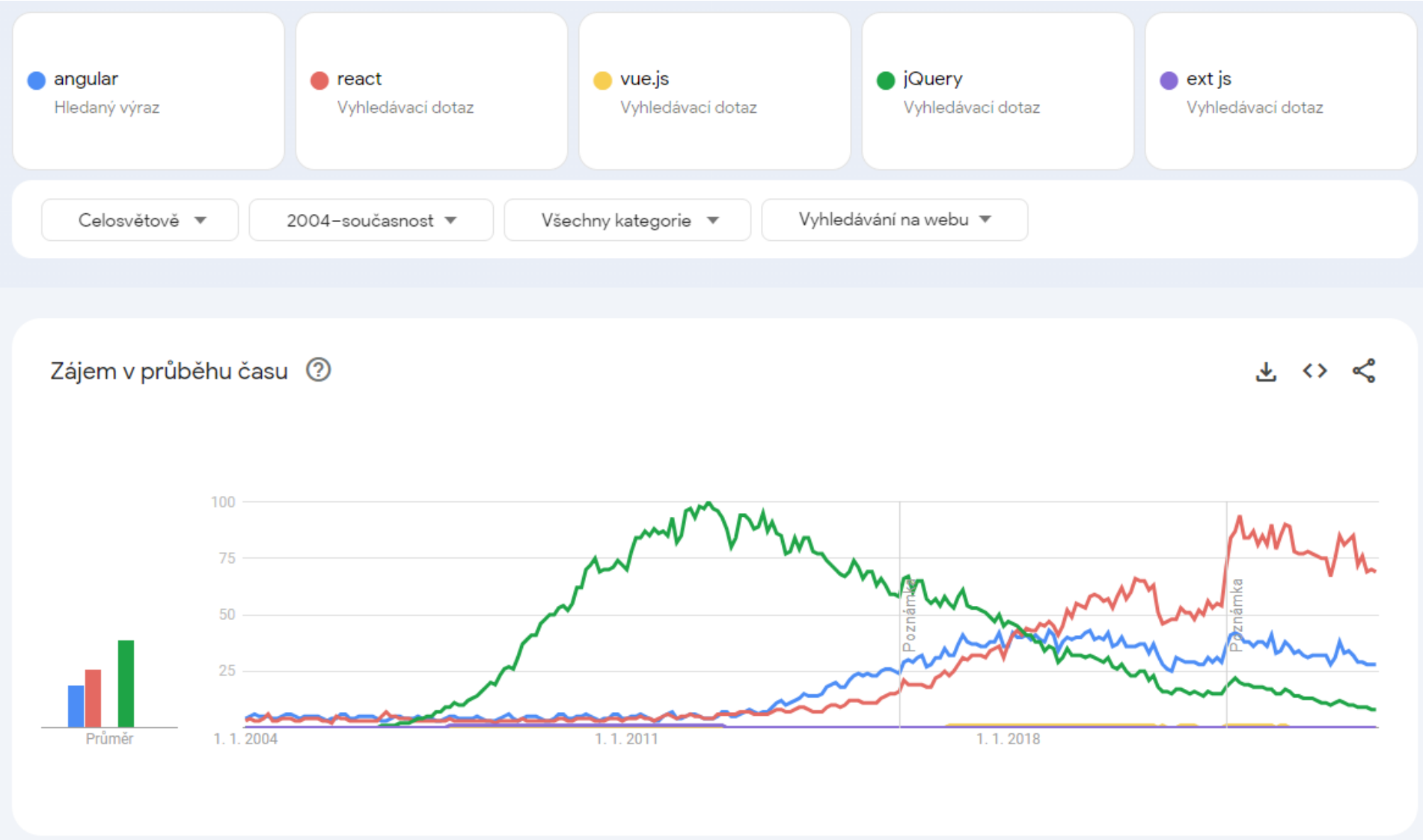
JavaScript frameworks

- They are JavaScript libraries which help with development of applications and make the work easier.
- The developer can be more focused on solving of problems, not on the optimization and debugging of the code for all web browsers.
- They are based on pure JavaScript and extend the objects, methods, etc. (by usage of prototype)

There are two basic groups

- JavaScript libraries – functionality extensions (Prototype, jQuery, MooTools, script.aculo.us,)
- RIA frameworks – complex solutions for RIA based on JS (Angular.js, Backbone, React.js, Embed, YUI, Dojo, extJS, GWT)

JavaScript frameworks



jQuery

```
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=utf-8">
  <script src="http://code.jquery.com/jquery-2.1.4.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 – dispatch when DOM is ready (do not wait for resources like images, styles, etc..)