



JavaScript a DOM

PIA 2005/2006
Téma 7

Pojmy

- DOM
 - reprezentace HTML, XML dokumentů
 - API pro přístup k ní
- JavaScript
 - skriptovací jazyk interpretovaný na straně klienta
- DHTML
 - HTML + JavaScript + CSS

O tomto webu

CLICK FOR ENGLISH

The Purpose
WellStyled.com web site is intended to be a resource of useful CSS (Cascading Style Sheet), (X)HTML, Javascript and related techniques and tricks for webdesigners. It's based on several articles published since year 2002 on

Click!

WellStyled.com web si dal za cíl být zdrojem užitečných postupů a triků pro webdesignery, v oblasti CSS (kaskádových stylů), (X)HTML, Javascriptu a souvisejících. Jeho základ tvoří několik příspěvků publikovaných od roku 2002 na [webu autora](#)®. Tyto články byly přesunuty na WellStyled.com, přepsány a rozšířeny a jsou rovněž periodicky (i když



About this web site

The Purpose

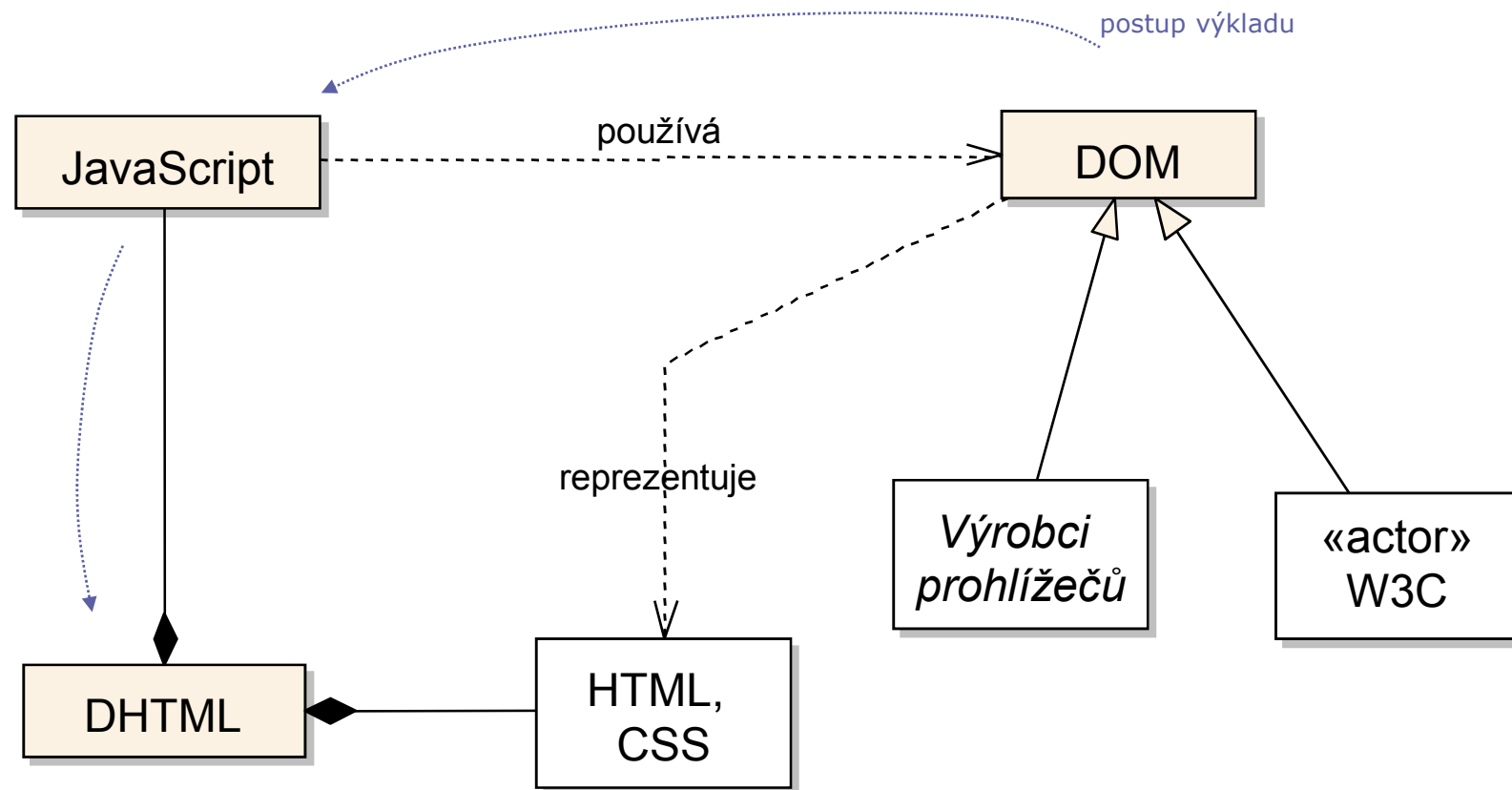
WellStyled.com web site is intended to be a resource of useful CSS (Cascading Style Sheet), (X)HTML, Javascript and related techniques and tricks for webdesigners. It's based on several articles published since year 2002 on the [author's own web](#)®. These articles has been transferred to WellStyled.com,

KLIKNUTÍM ZVOLÍ:

Češtině

Web WellStyled.com web si dal za cíl být zdrojem užitečných postupů a triků pro webdesignery, v oblasti CSS (kaskádových stylů), (X)HTML, Javascriptu a souvisejících. Jeho základ tvoří několik příspěvků

Souvislosti



Document object model

- API pro přístup k reprezentaci HTML, XML dokumentů
 - definuje logickou strukturu dokumentu
 - » strom, případně les
 - vlastnosti a metody pro upravování obsahu
- Části
 - jádro – základní typy
 - HTML, XML – specifické vlastnosti jazyků
 - stylesheety, events, traversal, ... – na 2. a vyšší úrovni

DOM – jádro

- `DOMImplementation`
 - informace o dostupnosti vlastností DOMu
- `Node`
 - základní strom uzlů
 - reference na dokument, rodiče, potomky, sourozence
 - manipulace s potomky
- `Dokument: Node`
 - *ML reprezentace elementu
 - jméno tagu, manipulace s atributy
 - normalizování textového obsahu
- `Text: CharacterData`
 - reprezentuje koncové uzly, např. textový obsah
- `Comment: CharacterData`

HTML DOM

- Cíle
 - specializace DOMu pro funkčnost spojenou s HTML
 - zpětná kompatibilita s „Level 0“
 - poskytování vhodných operací (skriptování)
- Vlastnosti
 - předpokládá Transitive nebo Frameset HTML 4.0
 - neumí
 - stylesheety, modifikace stylů
 - jmény událostí a jejich obsluha

HTML DOM rozhraní

- „Zkratková“ rozhraní a vlastnosti/metody
 - `HTMLDocument` → `title`; `images`; `write(text)`;
 - `HTMLMetaElement` → `name`; `content`; `httpEquiv`
 - `HTMLFormElement` → `action`; `submit()`
 - `HTMLUListElement` → `type`

 - běžné vlastnosti: `id`; `title`; `className`; ...
- Návaznost na skriptovací jazyky
 - ECMA script, Java

-
-
-
-
-
-
-
-
-
-
-
-

JavaScript



-
-
-
-
-
-
-
-

JavaScript

- Skriptovací jazyk interpretovaný na straně klienta
 - použití: dynamické HTML
 - » kontrola dat na straně klienta, přepisování stránek
- Vývoj
 - počátky: Netscape v NS2
 - následováno MS (JScript), NS
 - standardní stabilní jádro: ECMAScript (ECMA 262)
 - aktuální verze:
 - ECMA-262 Edition 4 = vývoj, Edition 3 = stabilní
 - JavaScript 1.7 = stabilní Firefox 2, 1.8 = vývoj

Základní charakteristiky

- Syntaxe na základě C/Java
- Netyповý jazyk
 - implicitní deklarace proměnných
 - základní typy: number (float), boolean, string
 - funkce
- Objektivě orientováno
 - třídy => objekty s vlastnostmi
- Bezpečnost
 - čtení pouze z počátku, žádný zápis souborů a událostí

Propojení JavaScriptu a HTML

- Možnosti uložení skriptů

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

- code

- `</script>`

- » pozor: ne-SGML text

- re-use kódu:

- `<script ... src="URL.js"></script>`

- kdekoli, nejlépe HEAD element

- Element `<noscript>`

Propojení JavaScriptu a HTML (2)

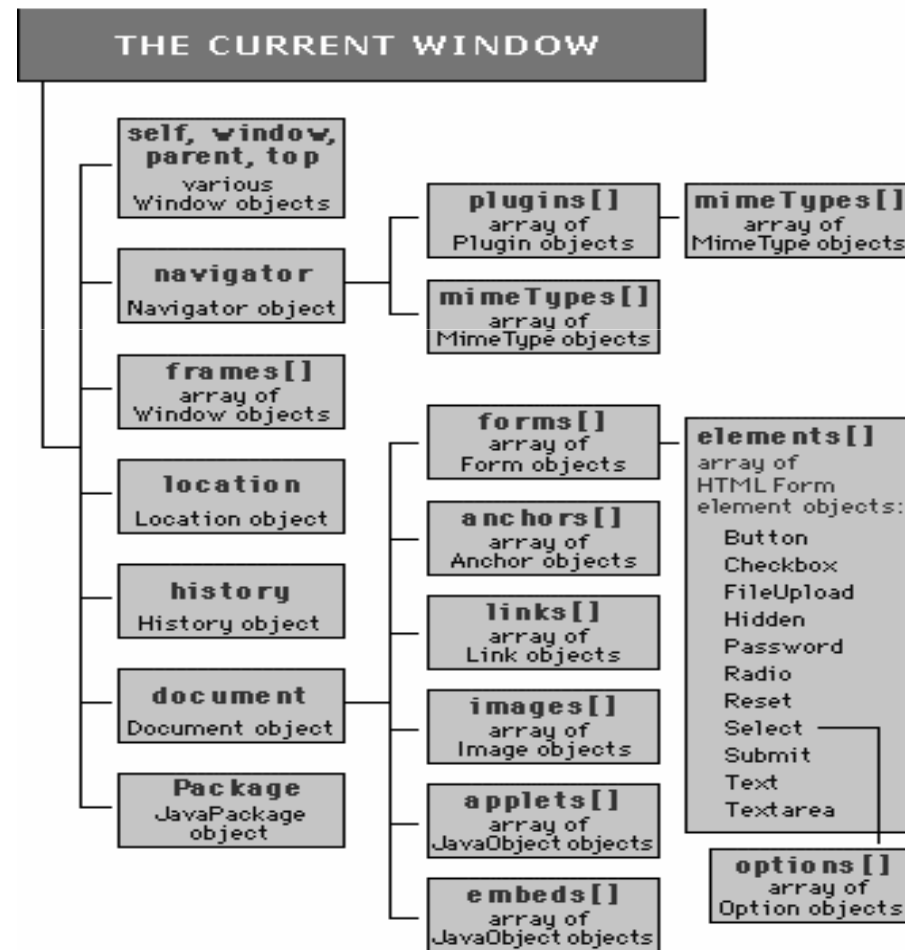
- Spouštění scriptů událostmi
 - klient (GUI) generuje události
 - `image:Abort`, `buttons:Click`, `document:Load`
 - dokument je pojmenovává jako události
 - `onAbort`, `onClick`, `onLoad`
 - a aktivuje přidruženou obsluhu (script)
 - `onEvent="code; return true"`
 - » *return true* umožní elementu normální práci
- Spouštění přes URL
 - běžně používané „javascript“ URI schema
 - ``

Browser Object Model (BOM)

- Jak se mapuje DHTML na koncept DOMu
 - žádná formální definice objektového modelu
 - instance, ne rozhraní/třídy, hierarchie agregace
 - přizpůsobený pro specifické klienty (Netscape, Microsoft)
- DHTML vs. DOM
 - DHTML: obsluha událostí => “živé” dokumenty
 - DOM: jednoduché změny ve struktuře dokumentu

BOM

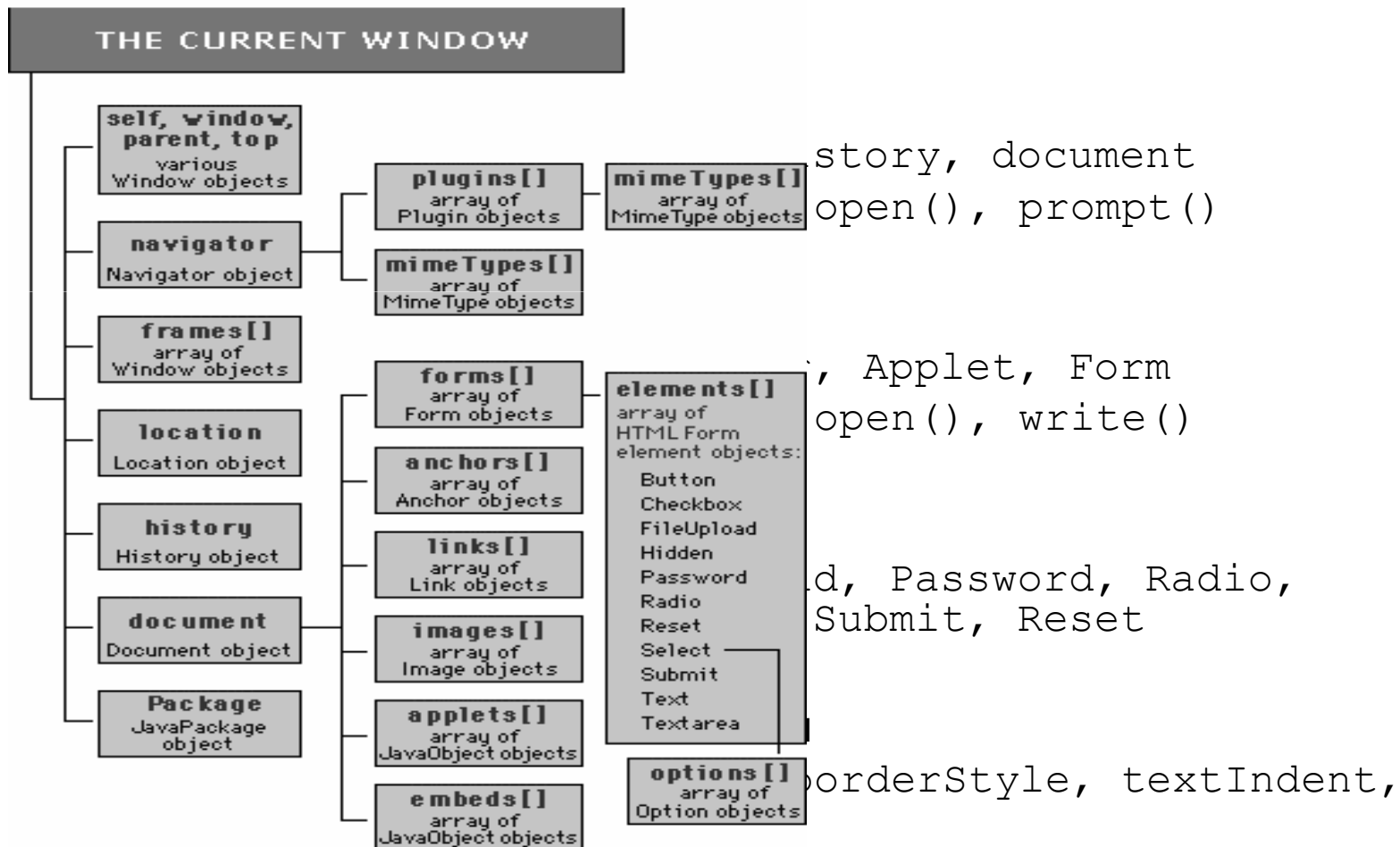
- Klíčové objekty navigace



Přístup k prvkům dokumentu

- Odkazování přes jména
 - element se jménem: ``
 - odkazován jak prvek pole: `document.images["Obr1"]`
 - nebo jako pojmenovaný atribut rodiče: `document.Obr1`
- Přes pole objektů, nejvíce v `document`:
 - `anchors[]`, `applets[]`, `forms[]`, `images[]`
 - `all[]` v MSIE
- Použití W3C DOM metod
 - » nutné pro XHTML
 - elementy s atributem *id*
 - metoda `getElementById()` na `Node`

JavaScript DOM objekty



-
-
-
-
-
-
-
-
-
-
-
-

Příklady



-
-
-
-
-
-
-
-

Formuláře

- DHTML pro
 - zvýšenou použitelnost formuláře
 - menší zatížení serveru
- Skripty

```
...  
<style type="text/css">  
  .hotkey { text-decoration: underline; }  
  .required { font-weight: bold; }  
</style>  
  
<script language="JavaScript"><!--  
  function setStatus(msg) { window.status = msg; }
```

pokračování ...

Formuláře (2)

```
function formCheck(form)
{
  var result = false;
  var name = form.lastname.value;
  if (name.match("^[A-Za-z]*$")) {
    alert("You must fill in your last name.");
    form.lastname.focus();
  }
  else {
    form.valid = "true";
    form.submit();
    result = true;
  }
  return result;
}
//-->
</script>
</head>
```


Formuláře (3)

- Stránka

```
<body onload="window.document.aform.firstname.focus()">
<form name="aform" action="..."
  onsubmit="return (formCheck (document.aform) );">
First name:
<input name="firstname" title="Your first name(s)"
  onfocus="setStatus('Enter your first name')" onblur="setStatus('') ">
<span class="required"><span class="hotkey">L</span>ast name:
<input name="lastname" accesskey="l"
  title="Your last name, without spaces. Required"
  onfocus="setStatus('Enter your last name')"
  onblur="setStatus('') "></span>
<input type="hidden" name="valid" value="false">
<input type="submit" name="submit" value="Submit">
</form>
</body>
```

Formuláře: Rendering

- Výsledek



The screenshot shows a web form with two input fields. The first field is labeled "First name:" and contains the text "Premek". The second field is labeled "Last name:" and is empty. Below the fields is a "Submit" button. A modal dialog box is overlaid on the form, titled "[JavaScript Application]". The dialog contains a yellow warning triangle icon and the text "You must fill in your last name." Below the message is an "OK" button.

AJAX & AHAH

- Async JavaScript and XML
Async HTTP and HTML
 - XMLHttpRequest objekt
 - překreslení části stránky přes DOM
- Coolness vs. Back

```
<select name="vyber" id="vyber"
  onchange="vyberClanek();" >
  <option value="0">-Vyber-</option>
  <option value="clanek1.htm">Článek 1
</option>
  <option value="clanek2.htm">Článek 2
</option>
</select>

<div id="text"> </div>
```

```
function vyberClanek() {
  var url = document.
  getElementById("vyber").value;
  if (url != 0) {
    httpRequest = new XMLHttpRequest();
    httpRequest.open("GET", url, true);
    httpRequest.onreadystatechange=
    function() { processRequest(); } ;
    httpRequest.send(null);
  }
}

function processRequest() {
  if (httpRequest.readyState == 4) {
    if (httpRequest.status == 200) {
      var kam =
      document.getElementById("text");
      kam.innerHTML =
      httpRequest.responseText;
    }
    else
      alert("Chyba nacistani stanky");
  }
}
```