Le Langage JavaScript

Didier DONSEZ
Université de Valenciennes
Institut des Sciences et Techniques de Valenciennes
donsez@univ-valenciennes.fr

Motivations

- Ajout du contrôle au niveau du client
  - Exemple :
    - contrôle des entrées d'un formulaire avant sa soumission.
- Scripts embarqués dans des documents HTML
  - Contrôle des ressources du client
    - documents, formulaires, applets, …
    - ⇒Moins d'intervention du Serveur WWW
  - Génération dynamique de documents DHTML
    - layers
    - ⇒Evite les GIFs animés et les animations Shockwave

Plusieurs propositions

- 4 langages de scripting
  - JavaScript *(Netscape)*
    - initialement LiveScript, rien à voir avec Java
  - VBScript, Jscript *(MicroSoft)*
    - syntaxe Visual Basic pour VBScript
    - JScript a une syntaxe compatible avec JavaScript
  - ECMAScript
    - *(ECMA : European Computer Manufacturers Association)*
- DOM : Document Object Model *(W3C)*
  - description « objet » d 'un document HTML ou XML
  - exploitable par les langages de scripting
    - mais IE et Navigator ont chacun leur modèle de document

L 'élément HTML <Script>

- Scripting local au document apparaît
  - dans l'élément HEAD ou dans l'élément BODY

```
<SCRIPT LANGAGE="JavaScript">
<!-- Commentaire HTML  qui dissimule le contenu du script aux browsers
  ... Instruction; ...
// qui ne supporte pas JS -->
</SCRIPT>
```

- Scripting externe
  - inclusion du code dans l'élément SCRIPT

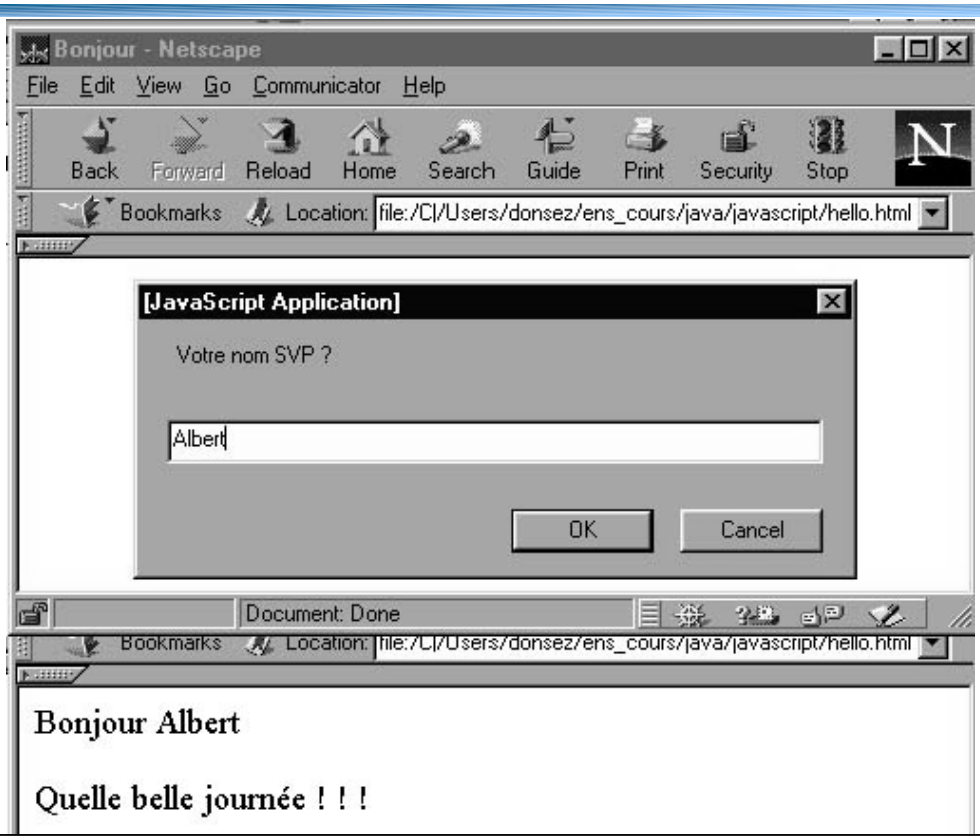
```
<SCRIPT LANGAGE="JavaScript" SCR="url/script.js">
// inclut le fichier script.js puis l'execute
// et reprends l'exécution des instructions suivantes
  ...
 Instruction;
</SCRIPT>
```

L 'élément HTML <Script>

```
<HTML><HEAD><TITLE>Bonjour</TITLE>
<SCRIPT LANGAGE="JavaScript">
<!-- Dissimule le contenu du script aux browsers
  var nom = window.prompt("Votre nom SVP ?","");
  document.write("Bonjour " + nom);
// qui ne supporte pas JS -->
</SCRIPT>
</HEAD><BODY><BR>
Quelle belle journée ! ! !
</BODY></HTML>
```

```
<HTML><HEAD><TITLE>Bonjour Script Externe</TITLE>
<SCRIPT LANGAGE="JavaScript"
SRC="http://www.mycomp.com/libscript.js">
// execute le contenu de libscript puis la suite
  document.write("Bonjour " + nom);
</SCRIPT>
</HEAD><BODY><BR>
Quelle belle journée ! ! !
</BODY></HTML>
```


Exemple



Constantes

- 4 types de base
  - entier
    - 127 (base 10), 0755 (base 8), 0xFA15 (base 16)
  - flottant
    - 0.123, -0.4e5, .67E-89
  - booléen
    - true, false
  - chaîne
    - "L'étoile" // L'étoile
    - 'filante' // filante
    - "'filante'" // 'filante'
    - \b, \f, \n, \r, \t, \"

Typage et Variable

- Pas de déclaration des Variables
  - nbr = 10;
  - fl =3.141;
  - str1 = "L'étoile";
  - str2 = 'brille';
- Portée des Variables
  - Local (uniquement dans le script ou la fonction)
    - var vloc = 0 ;
  - Global (en tout point du document)
    - vglob = 0 ;

Expressions et Opérateurs

- Expressions
  - **Arithmétique**
    - (3+4) * (56.7 / 89)
  - **Chaîne**
    - "L'étoile" + " " + "filante"
  - **Logique**
    - temp == 37
    - h2o = (temp<100) ? "eau" : "vapeur";
    - h2o = (temp>0) ? ((temp<100) ? "eau" : "vapeur") : "glace";
- Opérateurs
  - **Affectations**
    - +=, -=, *=, /=, %=, &=, |=, <<=, >>=
  - **Comparaisons**
    - ==, !=, <, <=, >, >=
  - **Arithmétiques**
    - %, ++, --
  - **Logiques**
    - &&, | |, !
  - **Bit**
    - &, |, ^ (XOR), <<, >>, >>>

Structures de Contrôle

- *bloc* { }
- if else, switch case, for, while, break, continue, do while, export, import, labeled

- Remarque : boucle for in
  - liste les propriétés d'un objet

```
function dumpobj(obj){
   var result= "", i= "";
   for(i in obj){ result += i + "  =  " + obj[i] + "<BR>"  ; }
   return result  ;
}
```

Fonctions

- Définition et Appel

```
function nomfonction( param1, ..., paramN) {
  // code JavaScript
  return variable_ou_valeur  ;
}
var res = nomfonction(var1, val2, varN)  ;
```

  - **Remarque : passage des paramêtres par valeur**

- Exemple

```
function isHumanAge(age) {
 if ((age < 0) || (age > 120)) { return false  ; } else { return true  ; }
}
if (  ! isHumanAge(age) ) {
  alert("Vous ne pouvez pas avoir " + age + " ans !");
}
```

- Arguments

```
function somme() {
  var argv = somme.arguments;      var argc = somme.arguments.length;
  var result=0  ; for(var i=0  ; i < argc  ; i++) { result += argv[i];  }  return result; }
somme(1,2,3) retourne 6 et  somme(2) est retourne 2
```


Fonctions prédéfinies

- **eval : évaluation d'expression**

```
angle = Math.PI / 2  ; expr = "1 - Math.cos(angle)"  ;
document.write( expr + " = " + eval(expr) )
```

- **parseInt, parseFloat**
  - convertion d'une représention chaîne d'un entier (dans une base) et d'un flottant
    - parseInt("12", 10)), parseInt("1100", 2)), parseInt("0xC", 16)) // retournent 12
    - parseFloat("123") // 123
    - parseFloat("456abc") // 456
    - parseFloat("1abc23") // 1
    - floatValue= parseFloat("abc789"); // NaN
    - if (isNaN(floatValue)) { notFloat() } else { isFloat() }
- **escape() , unescape()**
  - utilisées pour le codage des URL ou des Cookies

```
escape("#")          // retourne %23
unescape("%23")      // retourne #
```

Le modèle objet de JavaScript

- Modèle de Prototype
  - Différent du modèle de classe (Java,C++, ...)
- Contructeur
  - fonction qui initialise les propriétés d'un objet

```
function printPersonne() {
  document.write("Personne: "
    +this.nom+","+this.age);
}
function Personne(nom, age) {
  this.nom = nom  ;
  this.age = age  ;
  this.print = printPersonne;
  return this  ; // OBLIGATOIRE
}
```

```
function printVehicule() {
  document.write("Vehicule: "
   +this.marque+","+this.age);
}
function Vehicule(marque, age) {
  this.marque = marque  ;
  this.age = age  ;
  this.print = printVehicule;
  return this  ; // OBLIGATOIRE
}
```


Instanciation

- **new**
  - instancie un objet " vierge " et appelle un constructeur

```
pers1     = new Personne("Albert", 77);
pers2     = new Personne("Leo", 56);
voit3     = new Vehicule("Renault", 2);

// Construction dans JS1.2
voit3 = {nom:"Renault", age:2, print:printVehicule};
```

- **`this`**
  - donne accès aux propriétés de l'objet courant
- **`with`**
  - raccourci pour désigner les propriétés et les méthodes

```
with(document) {  write("La longueur du document est  :" + length)  ; }
// OU
document.write("La longueur du document est  :" + document.length)  ;
```

Accès aux propriétés d 'un objet (i)

- Propriété = Objet, Valeur ou Méthode
  - Opérateur de référence .
    - l'objet doit avoir les propriétés désignées
    - Dans une méthode, les propriétés de l'objet sont accessible par this.

```
pers1.age = 20;
voit3.age = 2;
function unAnDePlus() { this.age++  ; }

pers1.incrAge=unAnDePlus;          // ajout d'une nouvelle propriété
voit3.incrAge= pers1.incrAge;      // ajout d'une nouvelle propriété
pers1.incrAge();                   // appel de unAnDePlus() sur pers1
voit3. incrAge();                  // appel de unAnDePlus() sur voit3
pers1.print();                     // appel de printPersonne() sur pers1
voit3.print();                     // appel de printVehicule() sur voit3
```


Accès aux propriétés d 'un objet (ii)

- Opérateur [ ] et for in

```
function dumpobj(obj){
   var result= "", i= "";
   for(i in obj){ result += i + "  =  " + obj[i] + "<BR>"  ; }
   return result  ;
}

dumpobj(pers1);
dumpobj(pers1.incrAge);
dumpobj(voit3);
```

L 'héritage (i)

- La propriété `prototype`

```
function Employee () {
   this.name = "";
   this.dept = "general";
}

function Manager () {
   this.reports = [];
}
Manager.prototype = new Employee;

function WorkerBee () {
   this.projects = [];
}
WorkerBee.prototype = new Employee;
```

```
paul = new Employee;
mark = new WorkerBee;
mark.name = "Doe, Mark";
mark.dept = "admin";
mark.projects = ["navigator"];
mark.bonus = 3000;
  // propriété ajoutée à mark uniquement
Employee.prototype.specialty = "none";
  // propriété ajoutée à mark et à paul

function Employee (name, dept) {
   this.name = name || "";
   this.dept = dept || "general";
}
noname = new Employee
carla = new Employee("Carla, Smith", "R&D");
```


L 'héritage (ii)

- Le test d'appartenance

```
function instanceOf(object, constructor) {
    while (object != null) {
        if (object == constructor.prototype) // constructeur de l'objet
            return true;
        object = object.__proto__;          // chainage de prototype
    }
    return false;
}
```

- Variable globale

```
var idCounter = 1;                          // variable globale
function Employee (name, dept) {
    this.name = name || "";                 // valeur par défaut
    this.dept = dept || "general";
    this.id = idCounter++;
}
```

Les objets Tableaux

- Ce sont des objets avec les propriétés qui sont les indices

```
function Array(nbElements) {
// constructeur
  this.length = nbElements;
  for( var e = 0  ; e < this.length  ; e++)
     { this[e] = 0  ; }
  return this  ;
}

marque = new Array(10);
marque[0]= "Renault";
marque[1]="Peugeot";
marque[2]="Citroen";
marque[10]="Ford"; marque.lenght++;
```

```
var i;

for( i=0;i <marque.length;i++ ){
  document.write(i + " = " + marque[i] + "<BR>")
}

document.write("<HR>");
for( i in marque ){
  document.write(i + " = " + marque[i] + "<BR>")
}
```


Les objets Tableaux (JS1.2)

- Prototype prédéfinie

- Constructeur

```
marque = Array("Renault", "Peugeot", "Citroen");
```

- Méthodes d'Array
  - **concat, pop, push, shift, unshift, slice, splice, sort**

- Tableau d'objets

```
var vehicules = new Array(4);
vehicules[1] = new Vehicule(marque[1], 2);
vehicules[2] = new Vehicule(marque[2], 4);
vehicules[3] = new Vehicule(marque[3], 8);
vehicules[4] = new Vehicule("Ford", 8);
for( i=1;i <= vehicules.length;i++ ){
  document.write(i + " = "); vehicules[i].print();
  document.write("<BR>");
}
```

Les objets Math et RegExp

- Prototypes prédéfinis

Math

constantes mathématiques : `PI`,…, fonctions mathématiques : `sin(x)`,`cos(x)`,`abs(x)`,`random()`...

```
circonference = Math.PI * rayon * rayon;        abscisse     = r * Math.sin(angle);
ordonnee     = r * Math.cos(angle);
with(Math) {
   circonference = PI * rayon * rayon;
   abscisse     = r * sin(angle);
   ordonnee     = r * cos(angle);
}
```

RegExp

expression régulière

```
re = /ab+c/
re = new RegExp("ab+c")
regexp = /pattern/[i|g|gi]
regexp = new RegExp("pattern", ["i"|"g"|"gi"])
```


Objets Function et Date

Function

- Définition des méthodes des objets
  - évite de passer par une fonction nommée

```
Employee.setdept = function(dept) {  this.dept = dept; }
```

Date

- référentiel 1er Janvier 1970

```
aujourdhui                  = new Date();
jour_an2000_moins_1sec  = new Date("December, 1999 23:59:59");
jour_an2000                 = new Date("January 1, 2000 00:00:00");
jour_an2000                 = new Date("January 1, 2000");
jour_an2000                 = new Date("January 1, 2000");
jour_avant_an2000           = new Date(99,11,31,0,0,0); // Jan(=0)
jour_avant_an2000           = new Date(99,11,31);       // Dec(=11)
msParJour = 24*60*60*1000;
nbjour_avant_an2000 = new Date();
nbjour_avant_an2000 =    Math.round(( jour_an2000.getTime() - aujourdhui.getTime()
                                    )/msParJour);
noel99= new Date();         noel99.setTime(Date.parse("Dec 25, 1999"));
```

Objets String (i)

- Création d'un objet sur affectation

```
objstr = "Le nouveau monde  !";
```

- Manipulation
  - les caractères sont indicés de 0 à objstr.length-1

```
"Le nouveau monde !".charAt(1)                  // "e"
"Le nouveau monde !".indexOf("e")               // 1
"Le nouveau monde !".indexOf("e",2)             // 7
"Le nouveau monde !".lastIndexOf("e")           // 15
"Le nouveau monde !".substring(3,7)             // "nouveau"
"Le Nouveau Monde !".toLowerCase()              // "le nouveau monde  !"
"Le Nouveau Monde !".toUpperCase()              // "LE NOUVEAU MONDE  !"
```

  - Mais aussi
    - charCodeAt, concat,fromCharCode, match, search, slice, split

Objets String (ii)

- Formatage HTML
  - insertion de balisage (tag) HTML :
    - <BIG>,<BLINK>, <B> …
    - big(), blink(), bold(), fixed(), fontsize(), fontcolor(), italics(), link(URL) ou anchor(URL), small(), strike(), sub(), sup()

```
"la " + "SNCF".anchor("http://www.sncf.fr")
        // "la <A HREF='http://www.sncf.fr'>SNCF</A>"
```

JavaScript, le navigateur et les documents HTML

- Description en objet JavaScript
  - du navigateur
  - des fenêtres
  - des documents qui les composent
- Consultation et Modification des propriétés de ces objets par scripting
- Future direction :
  - DOM

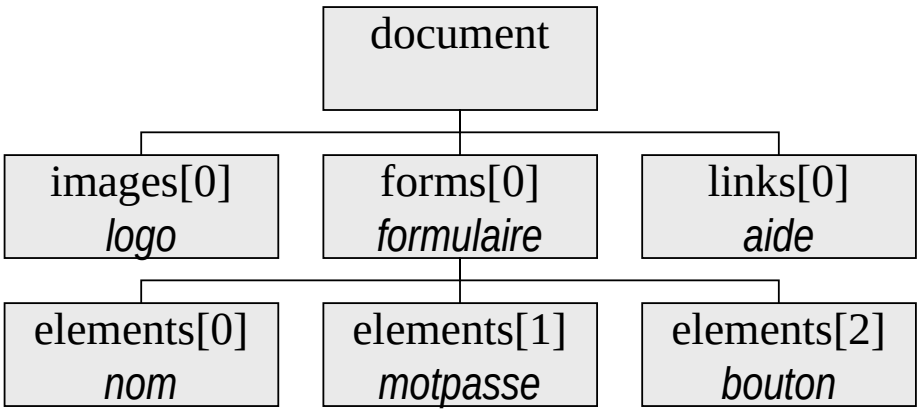
Document « Simple »

- HTML

```
<HTML><BODY>
<CENTER><IMG SRC="bienvenu.gif"
    HEIGHT=69 WIDTH=184
    name="logo"></CENTER>
<FORM name="formulaire">Nom:
<BR> <INPUT type="text"
    name="nom" value="">
<BR>Mot de Passe:
<BR> <INPUT type="password"
    name="motpasse" value="">
<CENTER><INPUT type="button"
    value="Valider" name="bouton"
    onClick="verif(this.parent)"></FORM>
</CENTER>
<DIV ALIGN=right><I><A
    HREF="aide1.htm »
    name="aide">Aide</A></I></DIV>
</BODY></HTML>
```

- Affichage

La représentation arborescence d'objets d'un document



- Exemple (script relatif au document)

```
passwd = document.forms[0].elements[1].value;
if( passwd==  'toto') {        document.images[0].src = "iconlog.gif"; }
else {                         document.logo.src = "iconnotlog.gif"; }
```

- Propriété parent
  - référence l'objet englobant

Les formulaires

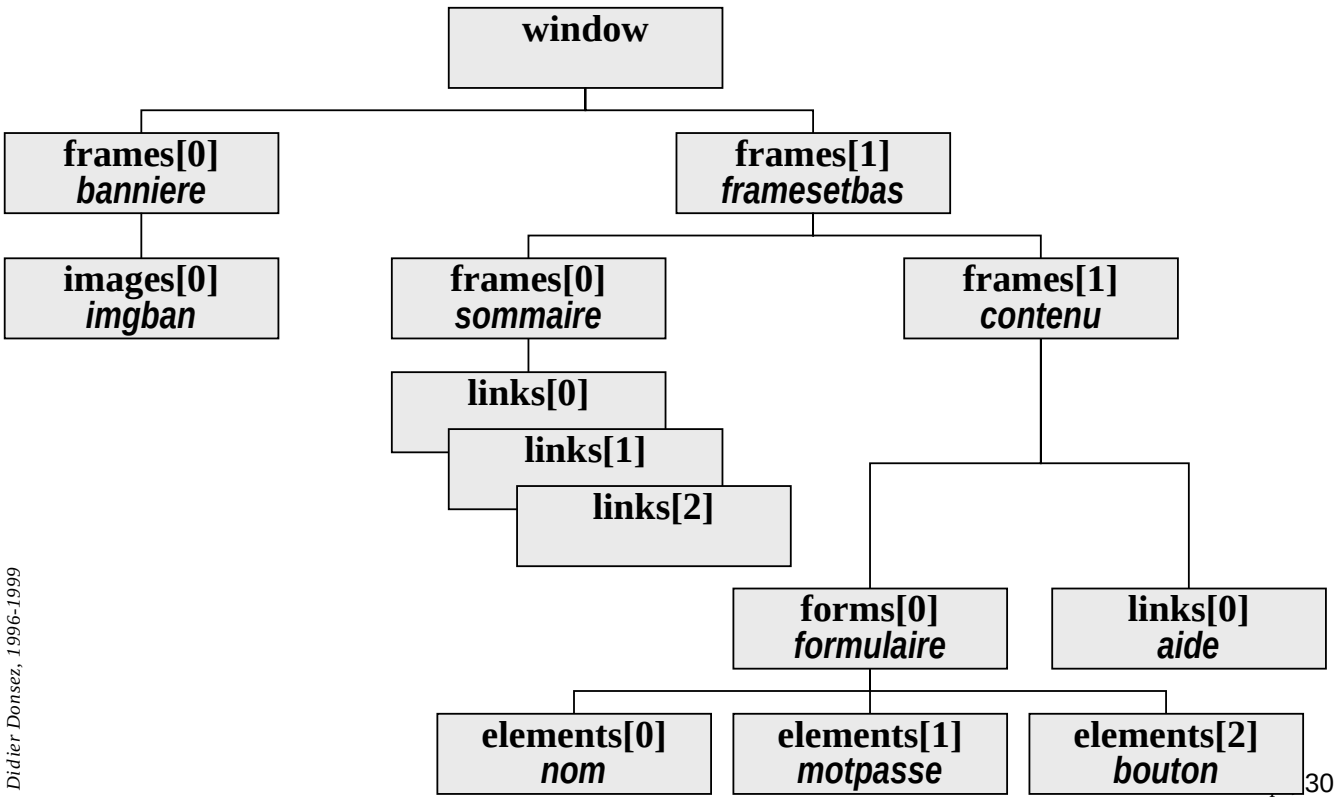
- l'objet Form
  - action
  - elements[ ]
  - encoding
  - attribute
  - method
  - target
  - submit()
  - onSubmit
- Les éléments
  - un type d'objet pour chaque type d'entrée
  - Hidden
  - Password
  - Radio
  - Checkbox
  - Select
  - Text
  - Textarea
  - Submit
  - Reset

Les frames

- HTML

```
<FRAMESET ROWS="100,*">
   <FRAME NAME="banniere"
                SRC="bann.htm">
   <FRAMESET COLS="100,*"
     NAME="framesetbas">
     <FRAME NAME="sommaire"
                SRC="somm.htm">
     <FRAME NAME="contenu"
                SRC="cont1.htm">
   </FRAMESET>
</FRAMESET>
```

La représentation arborescence d'objets d'un frameset



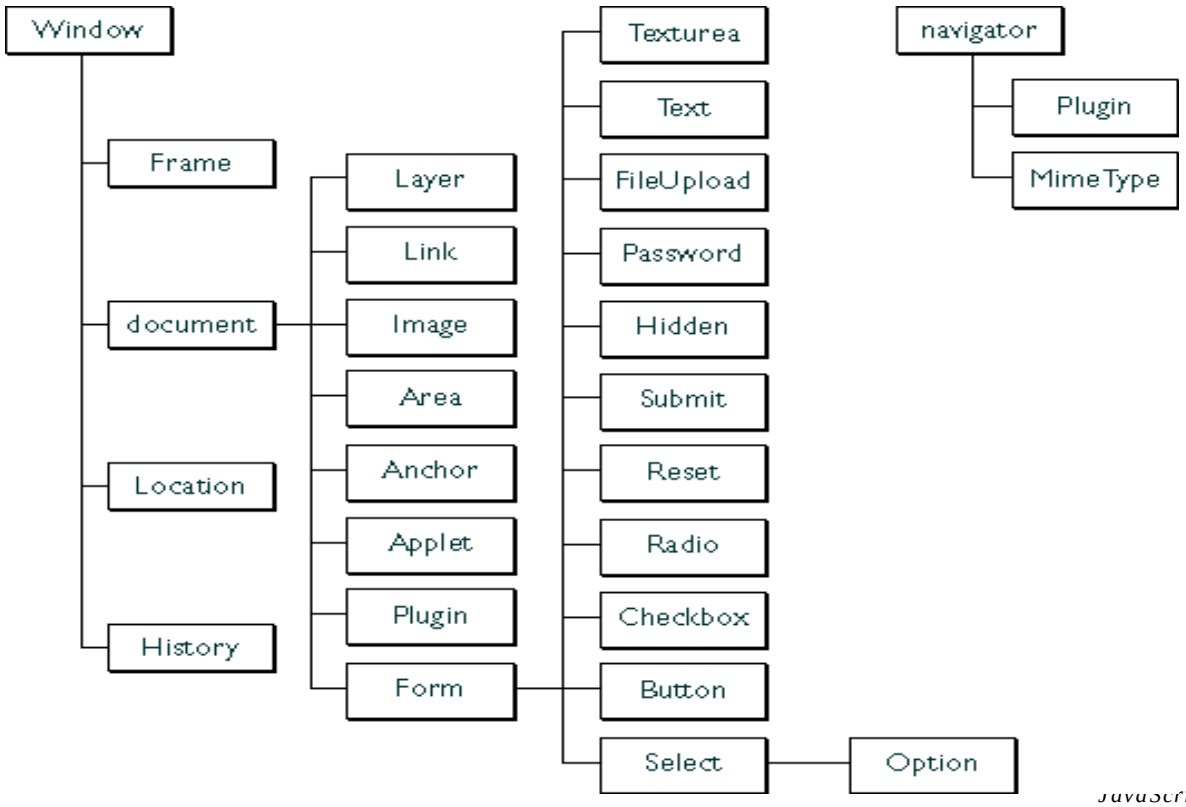
Exemple

```
<HTML><HEAD>
    <BASE TARGET="contenu">
<SCRIPT LANGUAGE="JavaScript">
function loaddoc(urlbanndoc, urlsommdoc, urldocfirst) {
    parent.parent.banniere.location.href= urlbanndoc;
    parent.sommaire.location.href= urlsommdoc;
    parent.contenu.location.href= urldocfirst;
}
</SCRIPT>
</HEAD><BODY>
 <A HREF="cont1.htm"> Accueil</A><BR>
 <A HREF="cont2.htm" TARGET="_top">Présentation</A><BR>
 <A HREF="javascript:loaddoc('banndoc.htm',  'sommdoc.htm', 'docdeb.htm')"
      TARGET="_self">Documentation</a>
</BODY>
</HTML>
```


L 'objet Window

- frames[ ]
- parent
- top
- self, window
- alert(message)
- close()
- confirm(message)
- open(url,name,features)
- toolbar=[yes,no,1,0]
- location=[yes,no,1,0]
- directories=[yes,no,1,0]
- status=[yes,no,1,0]
- menubar=[yes,no,1,0]
- scrollbars=[yes,no,1,0]
- resizable=[yes,no,1,0]
- width=pixels
- height=pixels
- prompt(message,response)
- id=setTimeout(expression,time)
- clearTimeout(id)

L 'objet Window



Création de fenêtre

- Création d 'une fenêtre

```
myWin= open("nouv.htm", "Nouvelle fenêtre",
    "width=400,height=300,status=no,toolbar=no,menubar=no");
```

- Création à la volée

```
<html><head><script language="JavaScript">
function openOnTheFly() {
  myWin= open("", "displayWindow", "width=500,height=400,"
                    +"status=yes,toolbar=yes,menubar=yes");
  myWin.document.open();
  myWin.document.writeln("<html><head><title>On-the-fly</title></head><body>");
  myWin.document.writeln("Ce document HTML est généré par un script<BR>");
  myWin.document.writeln("<form><input type=button value='Close' onClick='Close()'>");
  myWin.document.writeln("</form></body></html>");
  myWin.document.close();
}
</script></head><body>
<form><input type=button value="On-the-fly" onClick="openOnTheFly()"></form>
</body></html>
```

Les timers

- 2 méthodes
  - ID=setTimeout(expr,delai);
  - clearTimeout(ID);
- Exemple : *un message déroulant dans la Statusbar*

```
<html><head> <script language="JavaScript">
var scrtxt =  "Bienvenue sur ce site !";
var length = scrtxt.length; var width = 100; var pos = -(width + 2);
function scroll() {
  var scroller = ""; pos++;  if (pos == length) { pos = -(width + 2); }
  if (pos < 0) {
    for (var i = 1; i <= Math.abs(pos); i++) { scroller = scroller + " ";}
    scroller = scroller + scrtxt.substring(0, width - i + 1);
  } else { scroller = scroller + scrtxt.substring(pos, width + pos); }
  window.status = scroller; setTimeout("scroll()", 100);
}
</script>
</head><body onLoad="scroll()"><h1>Bienvenue</h1></body></html>
```


Les layers

- Propriétés
  - name="layerName", left=xPosition, top=yPosition z-index=layerIndex, width=layerWidth, clip="x1_offset, y1_offset, x2_offset, y2_offset", above="layerName", below="layerName", Visibility=show|hide|inherit, bgcolor="rgbColor", background="imageURL »
- Exemple

```
<html><head><script language="JavaScript">
function move() {
  if (pos < 0) direction= true; if (pos > 200) direction= false;
  if (direction) pos++ else pos--;
  document.layers["layer0"].left= pos; setInterval('move()', 20);
}
</script>
</head><body onLoad="setInterval('move()', 20)">
<ilayer name=layer0 left=0>
<font size=+1 color="#0000ff"><i>Bienvenue</i></font>
</ilayer></body></html>
```

Les layers imbriquées *(nested)*

```
<html><head>
<script language="JavaScript">
var pos= 0; var direction= false;
function moveNclip() {
  if (pos<-180) direction= true;
  if (pos>40) direction= false;
  if (direction) pos+= 2 else pos-= 2;
  document.layers["clippingLayer"].layers["imgLayer"].top= 100 + pos;
  setInterval('moveNclip()', 20);
}
</script>
</head><body onLoad="setInterval('moveNclip()', 20);">
<ilayer name="clippingLayer" z-index=0 clip="20,100,200,160" top=0 left=0>
  <ilayer name="imgLayer" top=0 left=0>
  <img name=davinci src="davinci.jpg" width=209 height=264>
  </ilayer></ilayer></body></html>
```


Cookies

```
<HTML><HEAD><SCRIPT>
function register(name) {
   var today = new Date(); var expires = new Date();
   expires.setTime(today.getTime() + 1000*60*60*24*365)
   setCookie("TheCoolJavaScriptPage", name, expires)
}
</SCRIPT></HEAD>
<BODY>
<H1>Register Your Name with the Cookie-Meister</H1><P>
<SCRIPT>
var yourname = getCookie("TheCoolJavaScriptPage")
if (yourname != null)          document.write("<P>Welcome Back, ", yourname)
else                           document.write("<P>You haven't been here in the last year...")
</SCRIPT>
<P> Enter your name. When you return to this page within a year,
you will be greeted with a personalized greeting. <BR>
<FORM onSubmit="return false">
Enter your name: <INPUT TYPE="text" NAME="username" SIZE= 10><BR>
<INPUT TYPE="button" value="Register" onClick="register(this.form.username.value);
    history.go(0)">
</FORM>
```

Les événements dans le document

- Différents événements

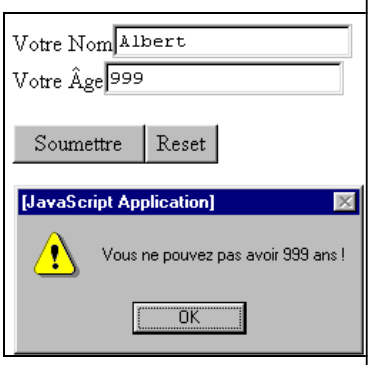
Evénement	Action de l'Utilisateur	Balise HTML
onBlur	*désélection d'un champ de saisie*	*INPUT*
onFocus	*activation d'un champ de saisie*	*INPUT*
onSelect	*sélection d'un champ de saisie*	*INPUT*
onChange	*changement de valeur d'un champ* *changement de valeur d'une sélection*	*INPUT* *SELECT*
onMouseOver	*passage de la souris sur un lien*	*A*
onClick	*clique sur un lien* *clique sur un élément du formulaire*	*A* *FORM*
onSubmit	*soumission du formulaire*	*FORM*
onLoad	*chargement de la page*	*BODY*
onUpload	*sortie du document*	*BODY*

- Handler

```
< TAG PARAMETRES ÉVÉNEMENT=FONCTION >
<INPUT    TYPE="BUTTON" NAME="monButton" VALUE="Appuyer"
          onClick="document.alert('Bien cliqué !')">
```


Exemple de handler

```
<HTML><HEAD><TITLE>Gestion d'événement</TITLE>
<SCRIPT LANGAGE="JavaScript">
function VerifAge(form) {
  if ((form.sonAge.value < 0) || (form.sonAge.value > 120)) {
    alert("Vous ne pouvez pas avoir " + form.sonAge.value+" ans !");
    form.sonAge.value = 0;
  }}
</SCRIPT></HEAD><BODY>
<FORM NAME="formulaire">
Votre Nom<INPUT TYPE=TEXT NAME="sonNom"><BR>
Votre Âge<INPUT TYPE=TEXT NAME="sonAge"
      onChange="VerifAge(formulaire)"> <P>
<INPUT TYPE=SUBMIT VALUE="Soumettre">
<INPUT TYPE=RESET>
</FORM></BODY></HTML>
```



Les événements dans les Fenêtres

- Propriétés de l 'objet Event
  - target, type, data (url des objets droppé)
  - height, width
  - layerX, layerY
  - modifiers: ALT_MASK, CONTROL_MASK, SHIFT_MASK, and META_MASK.
  - pageX, pageY, screenX, screenY
  - which (touche ASCII ou numéro du bouton de la sourie
- Capture des événements dans la fenêtre

```
window.captureEvents(Event.CLICK);
window.onclick= displayCoords;
function displayCoords(e) {
      if(confirm("x: " + e.pageX + " y: " + e.pageY){
            window.releaseEvents(Event.CLICK); }
```


Exemple de capture

```
<html><head>
<script language="JavaScript">

window.captureEvents(Event.MOUSEDOWN | Event.MOUSEUP);
window.onmousedown= startDrag;
window.onmouseup= endDrag;
window.onmousemove= moveIt;

function startDrag(e)   { window.captureEvents(Event.MOUSEMOVE);}
function moveIt(e)      { status= "x: " + e.pageX + " y: " + e.pageY; }
function endDrag(e)     { window.releaseEvents(Event.MOUSEMOVE); }

</script></head><body>

Push the mouse button and move the mouse. The coordinates are being
displayed on the statusbar.
</body></html>
```

JavaScript et les Plug-ins

- Pilotage d 'un document inclus par un plug-in
  - QuickTime, RealPlayer, LiveAudio, Cosmo
- Plug-ins installés sur le navigateur
  - **navigator.plugins[] ou navigator.components[]**

Exemple avec un Plug-in

```
function rembobiner() { // d 'après Mac Farlane Chap 4
   // ... rembobine un film quicktime à la première image
}
var connu = true;
if ( ! navigator.plugins["QuickTime Plug-In"].enabledPlugin )
   connu = false;                    // le plug-in souhaité est absent
else if ( ! navigator.plugins["QuickTime Plug-In"]["video/quicktime"])
   connu = false;                    // il n'est pas configuré pour notre type MIME
if ( connu ) {
 document.write('<EMBED SRC="film.mov" HEIGHT=260 WIDTH=320'
          + 'CONTROLLER=true LOOP=false AUTOPLAY=TRUE>');
 document.write('<INPUT TYPE="BUTTON" VALUE="Play" ONCLICK="rembobiner()">');
} else { // utilise seulement les paramètres HTML standards
 document.write('<EMBED SRC="film.mov" HEIGHT=260 WIDTH=320');}
 document.write('Better with <IMG SRC="installez_quicktime.gif" HEIGHT=260
   WIDTH=320>');
}
```

LiveConnect : *Communication JavaScript avec Java*

- Une fonction JavaScript peut contrôler une applet (i.e. appeler une méthode de l'applet)

- Exemple

```
<html> <head><title>Cercle</title></head><body>
<applet code=Cercle.class id=IdCercle width=60 height=60 >
   <param name=r value=0> <param name=v value=0> <param name=b value=0>
</applet>
<script language="JavaScript">
 function setROUGE()        { document.IdCercle.setCercleColor(255,0,0); }
 function setVERT()         { document.IdCercle.setCercleColor(0,255,0); }
 function setBLEU()         { document.IdCercle.setCercleColor(0,0,255); }
</script>
<hr>
<form>
<input type=button="cmdROUGE" value="ROUGE" onClick="setROUGE()">
<input type=button="cmdVERT" value="VERT" onClick="setVERT()">
<input type=button="cmdBLEU" value="BLEU" onClick="setBLEU()">
</form></body></html>
```


LiveConnect : *Communication Java vers JavaScript (i)*

- **netscape.javascript.JSObject, netscape.javascript .JSException**
- package Java permettant à une APPLET chargée dans une page d'accèder aux objets JavaScript définis dans la page.

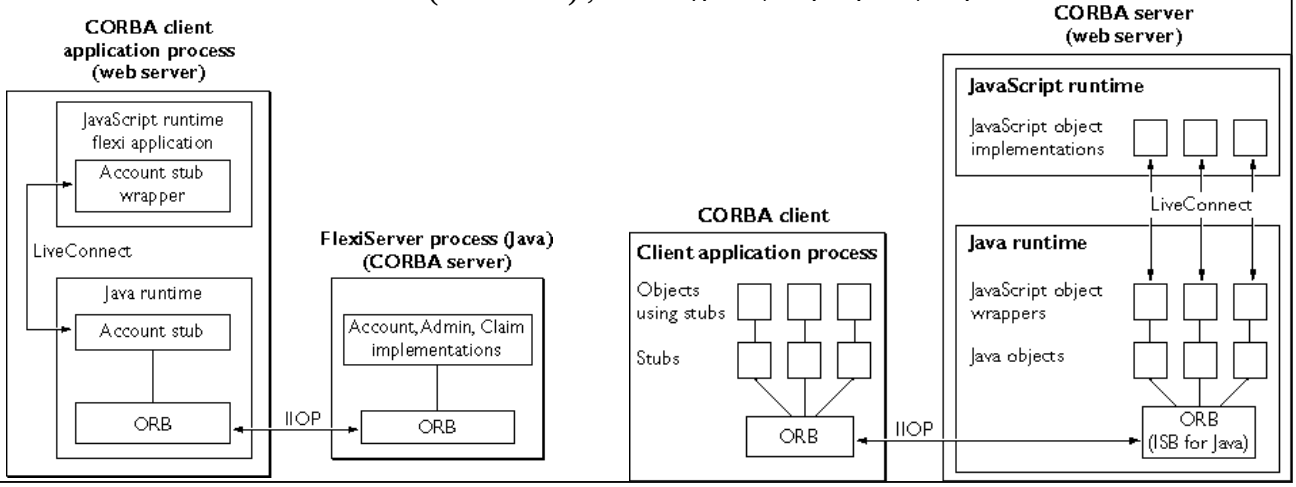
```
<SCRIPT>
function whatIuse() {
  alert("Vous surfez avec " + navigator.appName
        + " " +  navigator.appVersion)
}
</SCRIPT>

<APPLET CODE="myapplet.class" MAYSCRIPT>
```

LiveConnect :
Communication Java vers JavaScript (ii)

```
import netscape.javascript.* (JSObject et JSException)
public class myApplet extends Applet {
  public void init() {
    JSObject win        = JSObject.getWindow(this);
    JSObject doc        = win.eval("document");
    JSObject myForm     = win.eval("document.testForm");
    JSObject doc2       = (JSObject) win.getMember("document");
    JSObject myForm2            = (JSObject) doc2.getMember("testForm");
    JSObject check2             = (JSObject) myForm2.getMember("testChk");
    Boolean isChecked           = (Boolean) check2.getMember("checked");
  }
 public boolean mouseUp(Event e, int x, int y) {
    win.eval("alert(\"Hello world!\");");      // par .eval
    String args[] = {"Hello"," World Bis!"};
    win.call("alert", args);                   // méthode de window
    String args2[] = {""};                     // pas d'arguments
    win.call("whatIuse", args2);               // fonction utilisateur
    return true;
} }
```


LiveConnect
JavaScript coté Serveur

- Server Side JavaScript *(sur les serveurs NetScape)*
  - les scripts JavaScript sont des éléments du document HTML (tag <SERVER> … </SERVER>.
  - Ils sont exécutés par le serveur avant d 'envoyer le document résultant au client.
- Le SSJ dialogue avec l 'environnement du serveur
  - bases de données (LiveWire) , messagerie, objet java, objet CORBA …



Exemple SSJ

```
<HTML><BODY><H1>Liste des Employés</H1>
<SERVER>
pool = new DbPool ("ORACLE", "serveurDRH", "dirdrh", "motpasse", "", 5);
conn = pool.connection ("Ma connection", 60);
write ("<HR>");conn.SQLTable("select * from "); write ("<HR>");
ce = conn.cursor ("select * from employe");
if ( ce && (conn.majorErrorCode() == 0) ) {
  write ("<TABLE>");
  while (ce.next()) {
    write ("<TR><TD>" + ce.name + "</TD>" +  "<TD>" + ce.salary + "</TD></TR>");
  }
  write ("</TABLE>");
  ce.close();
}
</SERVER>
<HR>
</BODY></HTML>
```


JavaScript Bean

- Même principe que les JavaBeans
  - introspection pour déploiement
    avec des outils graphiques
    - Netscape ' Visual JavaScript, Arcadia ' Infuse
- Spécification .jsb :

```
<JSB>
  <JSB_DESCRIPTOR ...>
  <JSB_ICON ...>
  <JSB_INTERFACE ...>
  <JSB_PROPERTY ...>
  <JSB_EVENT ...>
  <JSB_METHOD ...>
    <JSB_PARAMETER ...>
  </JSB_METHOD>
  <JSB_CONSTRUCTOR ...> ... <JSB_CONSTRUCTOR ...>
</JSB>
<JSB_LISTENER> ... </JSB_LISTENER>
```

Exemple de JavaScript Bean

```
<JSB> (MacFarlane Chap 10)
  <JSB_DESCRIPTOR NAME="ObjetDePanique">
  <JSB_PROPERTY NAME="message_de_panique" TYPE="string">
  <JSB_PROPERTY NAME="niveau_de_panique" TYPE="string">
  <JSB_METHOD NAME="paniquer" TYPE="void"> </JSB_METHOD>
  <JSB_CONSTRUCTOR>
    function paniquer() {
      if ( this.niveau_de_panique == "faible" )
        alert(message_de_panique);
       else
        top.location.href = "javascript:'<HTML><H1>"+message_de_panique+"</H1></HTML>'"
    }
  function ObjetDePanique(params) {
      this.message_de_panique = params.message_de_panique;
      this.niveau_de_panique   = params.niveau_de_panique;
      this.paniquer      = paniquer;
    }
  </JSB_CONSTRUCTOR>
```


Netscape et Proxy

- Fichier proxy.pac
  - configure dynamiquement l 'accès aux proxys pour un navigateur

```
// fichier proxy.pac (d 'après Mac Farlane Chap 2)
function FindProxyForURL(url, host) {
 if (isPlainHostName(host) || isInNet(host, "192.123.234.0") )
   return "DIRECT";
 else if ( url.substring(0, 6) == "https:" || url.substring(0, 6) == "snews:")
   return "SOCKS sockshost:1081";
 else {
   if ( Math.random() < 0.5 )
      return "PROXY proxy1:1080 ; proxy2:1080";
   else
      return "PROXY proxy2:1080 ; proxy1:1080";
 }
}
```

Des outils

- MicroSoft
  - VisualDev
- MacroMedia
  - DreamWeaver
- Netscape
  - VisualJS
  - DebugJS
    - .jar à installer dans Communicator

Conclusion

- Scripting Client
  - DHTML
- Scripting Serveur
  - BD, Java, Corba

Bibliographie

- NetScape et MicroSoft
  - les références et des démonstrations HTML
- Tutorial VOODOO
  - DHTML et JavaScript
- Les livres *(Attention ! ils deviennent vite obsolètes)*
  - Jeff Rouyer , Dynamic HTML Web Magic, Ed New Riders, ISBN: 1568304218, 07/98, pp 296 +CD-ROM
    - http://www.htmlguru.com/magic
    - orienté Animation et Graphisme en DHTML
  - Nigel McFarlane, JavaScript, Ed Eyrolles, 1998, ISBN 2-212-09034-X
  - D. Charnay, P. Chaleat, "Programmation HTML et JavaScript", Ed. Eyrolles, ISBN : 2-212-09024-2
  - Arman Danesh, “ JavaScript ”, (Simon & Schuster Macmillan) en français ISBN 2-7440-0167-8