

Introduction à JavaScript

(Navigator JavaScript 1, 1.1, 1.2)

Thomas LEDUC
Thomas.Leduc@lip6.fr
<http://quartz.dgs.jussieu.fr:8080/>

Novembre 1997

Table des matières

1	Une introduction	2
1.1	rapide introduction à l'objet :	4
1.1.1	Les objets définis par le développeur	4
1.1.2	Les objets pré-définis "built-in"	6
1.1.3	Les objets pré-définis liés au navigateur client	6
1.2	rapide introduction à la programmation événementielle	8
2	Premiers exemples	9
2.1	"Hello World" en V.F.	9
2.2	Des indiscretions en couleur - manipulation de chaînes de caractères	9
2.3	On peut aussi faire du JavaScript en attribut de balise hyper-lien	10
3	JavaScript et les tableaux	11
3.1	Tableaux uni- et bi-dimensionnels	11
3.2	Tableaux et date...	12
4	JavaScript et les "frames"	12
4.1	Des "frames" un peu vides générées "dynamiquement"	12
4.2	Des "frames" et du contenu	13
5	JavaScript et les animations	14
5.1	Le scrolling	14
5.2	L'horloge	15
5.3	Simuler l'effet GIF89	17
6	JavaScript et les menus	17
7	JavaScript et Java	18
8	JavaScript et la validation de formulaire	19
8.1	Le document HTML	19
8.2	Les résultats	25

9 En guise de conclusion	25
9.1 Gérer les évolutions successives	25
9.2 Ne pas confondre.	26
9.3 Les dernières évolutions en bref	26
9.4 Et un peu de documentation.	26

1 Une introduction

Constatation : le tandem *HTTP + CGI* surcharge beaucoup le travail du serveur et la bande passante du réseau d'interconnexion! Il faut, tant que faire se peut, chercher à alléger la charge du serveur en déportant au maximum les traitements sur le client! L'utilisation d'un langage très orienté client tel que **JavaScript** peut alors s'avérer très utile et efficace (pour la validation des champs d'entrée d'un formulaire, les opérations ne nécessitant pas de ressources strictement propre au serveur. . .).

Présentation : JavaScript est un langage de commandes interprétées, couche de programmation indépendante de l'HTML quoiqu'encapsulé dans le corps même des documents HTML. Il introduit les notions de programmations structurée, orientée objets et événementielle au sein du code HTML (qui est tout sauf un langage de programmation). Il n'a pas de raison d'être en dehors d'un navigateur (contrairement à Java qui est un vrai langage de programmation). JavaScript est interactif et "dynamique", contrairement à l'HTML et peut effectuer des traitements sur le poste client. Il est capable de manipuler les objets du client.

But de ce cours : prouver que l'on peut rendre un client Web relativement "intelligent" et autonome par l'utilisation du langage JavaScript.

Mise en garde : JavaScript est "un langage à la croissance fulgurante, emprunt de défauts de jeunesse corrigés au fil des versions successives. . . ¹" (attention aux problèmes de portabilité et de lisibilité)!

un peu de syntaxe

- deux façons de placer des commentaires dans un code :

```
// ce qui suit (jusqu'en fin de ligne) est un commentaire  
/* placer ici le commentaire sur plusieurs lignes si vous le souhaitez */
```

- JavaScript est sensible à la casse,
- un identificateur de variable est une chaîne de caractères commençant par une lettre de l'alphabet ou l'underscore (pour le reste de la chaîne, les chiffres sont autorisés),
- JavaScript est faiblement typé mais il reconnaît les types pré-définis suivant : nombres, chaîne de caractères, booléens, type à valeur unique null. . .
- littéraux (constantes non nommées) : 011 (constante octale de valeur 9 en base dix), 0xF (constante hexadécimale qui vaut 15 en base dix), -3.14E-2 (constante à virgule flottante), true et false (constantes booléennes), "une chaîne\n de \tplusieurs caracteres" ou 'je m\'appelle "Thomas", bonjour' (constantes chaîne de caractères).
- portée des variables : il est possible de redéfinir une variable locale à une procédure (bien que portant le nom d'une variable globale) en la précédant du mot réservé "var",
- expressions, opérateurs et instructions empruntés au langage C,

1. Extrait du livre : **Programmation JavaScript**, Cyrille LECOMTE, Thomas LEDUC, éditions Eyrolles, octobre 1996

- procédures et fonctions (cas particulier de procédure retournant un résultat) sont des sous-programmes ou blocs d'instructions relativement autonomes pouvant être appelés d'un autre endroit du programme. Les utiliser permet d'éviter la redondance de code mais aussi de le structurer et le rendre plus lisible. Syntaxe :

```

function un_Identificateur_De_fonction(liste des parametres) {
/* le bloc d'instructions */
}

```

Un interpréteur de JavaScript “en url” Il est possible d'évaluer de “petites” instructions JavaScript à l'aide de l'interpréteur livré “en standard” avec le navigateur Netscape (à partir des versions 2.xx) en remplaçant la chaîne de caractères d'un url standard par une chaîne basée sur le principe suivant :

```

mocha: Math.random();
livescript: age=prompt(‘‘Entrez-votre age :’’,0); (age >= 35) ?
                ‘‘Vous pouvez candidater aux senatoriales !’’ :
                ‘‘Vous ne pouvez pas candidater aux senatoriales !’’;
JavaScript: 'IBP'.link('http://www.ibp.fr')

```

Comment et où placer le script JavaScript ? Dans la mesure du possible il est préférable de placer le script (les procédures et fonctions au moins) entre les deux balises `< HEAD >` et `< /HEAD >` de votre document HTML en respectant l'une des syntaxes suivante :

- pour du code JavaScript standard (Netscape Navigator à partir de la version 2) :

```

<SCRIPT LANGUAGE=‘‘JavaScript’’>
// ... placer ici le code JavaScript 1.0 ...
</SCRIPT>

```

- pour du code JavaScript utilisant des instructions de la version 1.1 (Netscape Navigator à partir de la version 3) :

```

<SCRIPT LANGUAGE=‘‘JavaScript1.1’’>
// ... placer ici le code JavaScript 1.1 ...
</SCRIPT>

```

- pour du code JavaScript utilisant des instructions de la version 1.2 (Netscape Navigator et Communicator à partir de la version 4) :

```

<SCRIPT LANGUAGE=‘‘JavaScript1.2’’>
// ... placer ici le code JavaScript 1.2 ...
</SCRIPT>

```

Chargement de code JavaScript contenu dans un fichier externe voici un petit exemple en trois étapes :

- il s'agit d'abord de s'assurer que l'extension “.js” spécifique aux fichiers JavaScript, est supportée par votre serveur. Pour ce faire, ajoutez la ligne suivante au fichier “mime.types” de configuration de votre serveur :

```

application/x-javascript      js

```

puis, arrêtez-le et redémarrez-le afin qu'il prenne en compte cette modification.

- ensuite, créez votre fichier JavaScript (celui-ci ne doit pas contenir de balise HTML) comme dans l'exemple suivant :

```
/* contenu du fichier : monCode.js */
function afficher() {
document.write("bonjour tout le monde !");
}
afficher();
```

- enfin, spécifiez l'URL absolue ou relative du fichier JavaScript que vous venez de créer, en argument de balise SCRIPT dans votre document HTML :

```
<HTML>
<HEAD>
<TITLE> Exemple 12 </TITLE>
<SCRIPT SRC="http://quartz.dgs.jussieu.fr:8080/BB/monCode.js">
document.write("Fichier monCode.js introuvable !");
</SCRIPT>
</HEAD>
<BODY>
<HR>
<A HREF="http://quartz.dgs.jussieu.fr:8080/BB/monCode.js"> fichier
source </A>
<HR>
</BODY>
</HTML>
```

Principe des sites doublement lisibles par des navigateurs JavaScript et non-JavaScript compatibles Placer le code suivant dans l'en-tête du fichier destiné aux plates-formes ne pouvant ou "souhaitant" pas lire et interpréter du JavaScript :

```
<HEAD>
<TITLE> un titre quelconque </TITLE>
<SCRIPT LANGUAGE="JavaScript">
<!--
if (confirm("Voulez-vous vous rendre sur mon site JavaScript ?"))
    location = "placer ici l'url du site JavaScript";
//-->
</SCRIPT>
</HEAD>
```

1.1 rapide introduction à l'objet :

1.1.1 Les objets définis par le développeur

JS utilise en fait une restriction de la notion de classe des langages orientés objet (classiques). Ainsi, les classes de JS n'autorisent que l'instanciation (i.e. la création d'objets) et ni l'héritage ni la création de sous-classes.

Accéder aux propriétés d'un objet

monObjet.unePropriete

Syntaxe d'un constructeur de "classe"

```
function nomDeClasse([argument1,...,argumentN]) {
/* this correspond a l'auto-reference (i.e. reference
* vers l'objet courant */
```

```

    this.propriete1 = argument1;
    ...
    this.proprieteN = argumentN;
    /* les methodes */
    this.methode1 = methode1;
    ...
    this.methodeM = methodeM;
}

```

Syntaxe d'une instantiation En définissant une "classe" d'objets (par le biais de son constructeur), nous avons défini un ensemble d'entités de même caractéristiques et comportements. Il ne nous reste plus alors qu'à instancier un objet de cette classe pour pouvoir le manipuler réellement. Pour ce faire, nous procédons de la manière suivante :

```

monInstance = new nomDeClasse([argument1,...,argumentN]);

```

Un exemple d'utilisation un peu prématuré...

```

/* deux tableaux - constantes globales */

var jour = new Array("dimanche","lundi","mardi","mercredi",
                    "jeudi","vendredi","samedi");
var mois = new Array("janvier","fevrier","mars","avril",
                    "mai","juin","juillet","aout","septembre",
                    "octobre","novembre","decembre");

/* quatre methodes */
function franciserDate() {
var resultat = jour[eval(this.laDate.getDay())] + " "
    + this.laDate.getDate() + " "
    + mois[eval(this.laDate.getMonth())] + " "
    + this.laDate.getYear();
return resultat;
}

function franciserHeure() {
var resultat = "" + this.laDate.getHours()
    + ((this.laDate.getMinutes() < 10) ? ":0" : ":")
    + this.laDate.getMinutes()
    + ((this.laDate.getSeconds() < 10) ? ":0" : ":")
    + this.laDate.getSeconds();
return resultat;
}

function afficher() {
var resultat = "<CENTER>"
    + this.franciserDate().fontcolor('red')
    + "<BR>"
    + this.franciserHeure().fontcolor('blue')
    + "</CENTER><HR>";
document.write(resultat);
}

function reinitialiser(uneDate) {
this.laDate = uneDate;
}

/* le constructeur de classe */

```

```

function maDate() {
/* l'attribut */
this.laDate = new Date();
/* les methodes */
this.franciserDate = franciserDate;
this.franciserHeure = franciserHeure;
this.afficher = afficher;
this.reinitialiser = reinitialiser;
}

/* deux instantiations */
d1 = new maDate();
d1.afficher();
d2 = new maDate();
var an_deux_mille = new Date(2000,00,01);
d2.reinitialiser(an_deux_mille);
d2.afficher();

```

1.1.2 Les objets pré-définis “built-in”

Objet	méthodes	propriétés
Array	eval, join, reverse, sort, toString, valueOf	length, prototype
Boolean	eval, toString, valueOf	prototype
Date	eval, getDate, getDay, getHours, getMinutes, getMonth, getSeconds, getTime, getTimezoneOffset, getYear, pars, setDate, setHours, setMinutes, setMonth, setSeconds, setTime, setYear, toGMTString, toLocaleString, toString, UTC, valueOf	prototype
Function	eval, toString, valueOf	caller, prototype
Math	abs, acos, asin, atan, atan2, ceil, cos, eval, exp, floor, log, max, min, pow, random, round, sin, sqrt, tan, toString, valueOf	E, LN2, LN10, LOG2E, LOG10E, PI, SQRT1.2, SQRT2
Number	eval, toString, valueOf	MAX_VALUE, MIN_VALUE, NaN, NEGATIVE_INFINITY, POSITIVE_INFINITY, prototype
String	big, blink, bold, charAt, eval, fixed, fontcolor, fontsize, indexOf, italics, small, split, strike, sub, substring, sup, toLowerCase, toUpperCase, toString, valueOf	length, prototype

Quelques exemples d'utilisation :

```

mocha: ValeurVraie = new Boolean(true);
livescript: n = new Number(Math.PI); Math.ceil(n);
javascript: moto = new String('xs 500'); moto.toUpperCase();
livescript: d = new Date(2000,00,01); m = new Date();
            delta = Math.floor((d.getTime() - m.getTime())/864E5);

```

A quoi correspond la valeur “delta” ?

1.1.3 Les objets pré-définis liés au navigateur client

Hiérarchie

```

navigator
|
|-- mimeTypes[]
|-- plugins[]

```

```
window
|
|-- document
|   |
|   |-- anchors[]
|   |-- applets[]
|   |-- forms[]
|   |   |
|   |   |-- button
|   |   |-- chekbox
|   |   |-- FileUpload
|   |   |-- hidden
|   |   |-- password
|   |   |-- reset
|   |   |-- select
|   |   |   |
|   |   |   |-- options[]
|   |   |-- submit
|   |   |-- text
|   |   |-- textarea
|   |-- images[]
|   |-- links[]
|-- frames[]
|-- history
|-- location
```

Objet	méthodes	propriétés
navigator	eval, javaEnabled, taintEnabled, toString,valueOf	appName, appVersion, mimeTypees, plugins, userAgent
mimeTypees	eval, toString, valueOf	description, enabledPlugin, type, suffixes
plugins	eval, refresh, toString, valueOf	description, filename, length, name
window	alert, blur, clearTimeout, close, confirm, eval, focus, open, prompt, scroll, setTimeout, toString, valueOf	closed, defaultStatus, frames, length, name, opener, parent, self, status, top
document	close, eval, open, toString, valueOf, write, writeln	alinkColor, anchors, applets, bgColor, cookie, domain, fgColor, forms, images, lastModified, linkColor, links, referer,title, URL, vlinkColor
frames	blur, clearTimeout, eval, focus, setTimeout, toString, valueOf	name, length, parent, self
history	back, eval, forward, go, toString, valueOf	current, length, next, previous
location	eval, reload, replace, toString, valueOf	hash, host, hostname, href, pathname, port, protocol, search
anchors	eval, toString, valueOf	length
applets	eval, toString, valueOf	length
forms	action, encoding, length, name, method, target (button, checkbox, FileUpload, hidden, password, radio, reset, submit, text, texarea)	eval, reset, submit, toString, valueOf
images	eval, toString, valueOf	border, complete, height, hspace, lowsrc, name, prototype, src, vspace, width
links	eval, toString, valueOf	hash, host, hostname, href, pathname, port, protocol, search, target
Button	blur, click, eval, focus, toString, valueOf	name, type, value
Chekbox	blur, click, eval, focus, toString, valueOf	checked, defaultChecked, name, type, value
FileUpload	blur, eval, focus, toString, valueOf	name, type, value
Hidden	eval, toString, valueOf	name, type, value
Password	blur, eval, focus, select, toString, valueOf	defaultValue, name, type, value
Reset	blur, click, eval, focus, toString, valueOf	name, type, value
Select	blur, eval, focus, toString, valueOf	length, name, options, selectedIndex, type
Submit	blur, click, eval, focus, toString, valueOf	name, type, value
Text	blur, eval, focus, select, toString, valueOf	defaultValue, name, type, value
Textarea	blur, eval, focus, select, toString, valueOf	defaultValue, name, type, value
options		defaultSelected, index, length, selected, selectedIndex, text, value

1.2 rapide introduction à la programmation événementielle

Un évènement, c'est une perturbation de l'environnement courant. Un clic de bouton de souris, le chargement d'une page, la modification du contenu d'une zone de saisie sont des événements par exemple. . . Mais un événement brut n'a pas grand intérêt, il faut en fait le gérer et générer une réponse adaptée. Nous verrons des exemples par la suite, présentons pour l'instant les événements et les cibles associées :

Événement	Cible associée
onAbort	images
onBlur	windows,frames,form elements
onClick	buttons,radio buttons,checkboxes,submit et reset buttons,links
onChange	text fields et textareas,select lists
onError	images,windows
onFocus	windows,frames,form elements
onLoad	document body
onMouseOut	area,links
onMouseOver	links
onSelect	text fields et textareas,
onSubmit	submit button
onUnload	document body

On les utilise de la manière suivante : `< BaliseHTML surEvenement = "code JavaScript" >`

```
<HEAD>
<TITLE> Exemple 0 </TITLE>
</HEAD>
<BODY>
<A HREF="http://quartz.dgs.jussieu.fr:8080/COURS/TL_JavaScript.dvi"
  onClick="alert('Vous allez charger ce cours !');">
cliquez-ici</A>
</BODY>
</HTML>
```

2 Premiers exemples

2.1 “Hello World” en V.F.

```
<HTML>
<HEAD>
<TITLE> Exemple 1 </TITLE>
<SCRIPT LANGUAGE="JavaScript">
<!--
document.write("Bonjour tout le monde&nbsp;!");
// -->
</SCRIPT>
<NOSCRIPT>
Vous ne pouvez ou voulez pas interpreter du JavaScript !
</NOSCRIPT>
</HEAD>
<BODY>
</BODY>
</HTML>
```

2.2 Des indiscretions en couleur - manipulation de chaînes de caractères

Le script est ici placé dans le corps même du document HTML. Nous montrons quelques méthodes de l’objet “built-in” **String** :

```
<HTML>
<HEAD>
<TITLE> Exemple 2 </TITLE>
<NOSCRIPT>
Vous ne pouvez ou voulez pas interpreter du JavaScript !
</NOSCRIPT>
</HEAD>
```

```

<BODY>
<SCRIPT LANGUAGE="JavaScript">
<!--
maintenant = new Date();
var chaine = "Bonjour,<BR>"
    + "Vous utilisez le navigateur "
    + navigator.appName.fontcolor('red')
    + "<BR>"
    + "Numéro de version et plateforme: "
    + navigator.appVersion.bold()
    + "<HR SIZE=4 WIDTH=10% ALIGN=LEFT>"
    + "Vous avez chargé cette page &grave;&nbsp;: "
    + maintenant.toLocaleString().fixed().big();

document.bgColor='yellow';
document.write(chaine.fontSize(3).fontcolor('blue'));
// -->
</SCRIPT>
</BODY>
</HTML>

```

2.3 On peut aussi faire du JavaScript en attribut de balise hyper-lien

Nous utilisons ici une variable globale (d'identifiant "n") pour incrémenter un compteur et la notion d'image réactive à gestion locale (à même le client) :

```

<HTML>
<HEAD>
<TITLE> Exemple 3 </TITLE>
<SCRIPT LANGUAGE="JavaScript">
<!--
var n=0;
// -->
</SCRIPT>
<NOSCRIPT>
Vous ne pouvez ou voulez pas interpreter du JavaScript !
</NOSCRIPT>
</HEAD>
<BODY>

<MAP NAME='un_maillage'>
<AREA SHAPE="RECT"
    COORDS="0,0,81,65"
    HREF='livescript:alert("Un petit kawa ?");'>
<AREA SHAPE="RECT"
    COORDS="0,65,81,130"
    HREF='javascript:n++;
        alert("Voila "+n+" fois que vous cliquez sur ce logo!");'>
</MAP>

<IMG SRC='about:javalogo' BORDER=0 USEMAP='#un_maillage'>

</BODY>
</HTML>

```

3 JavaScript et les tableaux

3.1 Tableaux uni- et bi-dimensionnels

```
<HTML>
<HEAD>
<TITLE> Exemple 8 </TITLE>
<SCRIPT LANGUAGE="JavaScript">
<!--

/* constructeur d'objet de type tableau
 * pour les navigateurs non compatibles
 * JavaScript 1.1
 */

function tableau (Taille)
{
for(var i=0 ; i < Taille ; i++)
this[i]=0;
}

/* les variables globales */

var Ta = new tableau(3);
Ta[0] = "Alain";
Ta[1] = "Pierre";
Ta[2] = "Eric";

var Tb = new Array(3);
Tb[0] = "Alain";
Tb[1] = "Pierre";
Tb[2] = "Eric";

var Tc = new Array("Alain","Pierre","Eric");

var chaine = "Contenu des trois tableaux :<BR>";
for (i=0 ; i<3 ; i++)
    chaine += Ta[i] + " - " + Tb[i] + " - " + Tc[i] + "<BR>";

/* construction d'une matrice : A*/
var A = new Array(3);
for (i=0 ; i<3 ; i++)
{
    A[i] = new Array(12);
    for (j=0 ; j<12 ; j++)
        A[i][j] = i*12+j;
}

/* affichage du contenu de la matrice */
chaine += "<TABLE>";
for (i=0 ; i<3 ; i++)
{
    chaine += "<TR>";
    for (j=0 ; j<12 ; j++)
        chaine += "<TD>" + A[i][j] + "</TD>";
    chaine += "</TR>";
}

```

```

chaine += "</TABLE>";

document.write(chaine);

// -->
</SCRIPT>
<NOSCRIPT>
Vous ne pouvez ou voulez pas interpreter du JavaScript !
</NOSCRIPT>
</HEAD>
<BODY>
</BODY>
</HTML>

```

3.2 Tableaux et date...

```

<HTML>
<HEAD>
<TITLE> Exemple 9 </TITLE>
<SCRIPT LANGUAGE="JavaScript">
<!--
var jour = new Array("dimanche","lundi","mardi",
                    "mercredi","jeudi","vendredi","samedi");
var mois = new Array("janvier","fevrier","mars","avril","mai",
                    "juin","juillet","aout","septembre",
                    "octobre","novembre","decembre");

function dateFrancaise (uneDate) {
result = jour[eval(uneDate.getDay())] + " "
        + uneDate.getDate() + " "
        + mois[eval(uneDate.getMonth())] + " "
        + uneDate.getYear();
return result;
}

maintenant = new Date();
document.write(dateFrancaise(maintenant));
// -->
</SCRIPT>
<NOSCRIPT>
Vous ne pouvez ou voulez pas interpreter du JavaScript !
</NOSCRIPT>
</HEAD>
<BODY>
</BODY>
</HTML>

```

4 JavaScript et les “frames”

4.1 Des “frames” un peu vides générées “dynamiquement”

```

<HTML>
<HEAD>
<TITLE> Exemple 10 </TITLE>
<SCRIPT LANGUAGE="JavaScript">
<!--

```

```

var interface_ppale = "<FRAMESET COLS='*,5*'>"
    + "<FRAMESET ROWS='*,9*'>"
    + "<FRAME SRC='vide.html' NAME='f0'>"
    + "<FRAME SRC='vide.html' NAME='f1'>"
    + "</FRAMESET>"
    + "<FRAME SRC='vide.html' NAME='f2'>"
    + "</FRAMESET>";

document.write(interface_ppale);
// -->
</SCRIPT>
<NOSCRIPT>
Vous ne pouvez ou voulez pas interpreter du JavaScript !
</NOSCRIPT>
</HEAD>
<BODY>
</BODY>
</HTML>

```

4.2 Des “frames” et du contenu

```

<HTML>
<HEAD>
<TITLE> Exemple 11 </TITLE>
<SCRIPT LANGUAGE="JavaScript">
<!--
var discours = "Bonjour,<BR>"
    + "Peut-&ecirc;tre utilisez-vous le navigateur<IMG SRC='about:logo' ALIGN=MIDDLE>";

var menu = "<FORM>"
    + "<INPUT TYPE=BUTTON VALUE='Discours' "
    + "onClick='top.ecrit(1,top.discours.fontSize(4),\"ceylan\");'>"
    + "<INPUT TYPE=BUTTON VALUE='La date' "
    + "onClick='d=new Date(); alert(d.toLocaleString());'>"
    + "<INPUT TYPE=BUTTON VALUE='En rouge' "
    + "onClick='document.bgColor=\"red\";'>"
    + "<INPUT TYPE=BUTTON VALUE='Exemple 1' "
    + "onClick='parent.window.frames[1].location=\"exemple-1.html\";'>"
    + "<INPUT TYPE=BUTTON VALUE='Recharger' "
    + "onClick='parent.location=\"exemple-11.html\";'>"
    + "<INPUT TYPE=BUTTON VALUE='Fermer' onclick='top.close();'>"
    + "</FORM>"
    + "<P><FONT SIZE=2>&#169 Copyright Thomas "
    + "<A HREF='mailto:leduc@masi.ibp.fr'>Leduc</A>, 1997</FONT></P>";

function ecrit(fr,ch,bg) {
    window.frames[fr].document.open();
    window.frames[fr].document.bgColor=bg;
    window.frames[fr].document.write(ch);
    window.frames[fr].document.close();
}

var interf_ppale = "<FRAMESET COLS='*,5*' "
    + "onLoad='ecrit(0,menu.fontSize(2),\"white\");'>"
    + "<FRAME SRC='vide.html' NAME='f0'>"
    + "<FRAME SRC='vide.html' NAME='f1'>"
    + "</FRAMESET>";

```

```

document.write(interf_ppale);
// -->
</SCRIPT>
<NOSCRIPT>
Vous ne pouvez ou voulez pas interpreter du JavaScript !
</NOSCRIPT>
</HEAD>
<BODY>
</BODY>
</HTML>

```

5 JavaScript et les animations

5.1 Le scrolling

```

<HTML>
<HEAD>
<TITLE> Exemple 17 </TITLE>
<SCRIPT LANGUAGE="JavaScript">
<!--

var timerID = null;
var enMarche = false;
var chaine = " Bonjour tout le monde ! ";
var longueur = chaine.length;

function stopper(i) {
if (enMarche == true)
    {
        clearTimeout(timerID);
        if (i != 0) frames[0].document.close();
    }
enMarche = false;
init = 0;
}

function lancer(i) {
stopper(i);
enMarche = true;
derouler(i,chaine);
}

var init = 0;

function deroulerAux(i,ch) {
var resultat = "";
if (i <= 1)
    {
        resultat = ch.substring(1,longueur) + ch.charAt(0);
    }
else if (i == 2)
    {
        for(var j=0 ; j < longueur ; j++)
            resultat += chaine.charAt((init+j)%longueur) + "<BR>";
        init++;
    }
}

```

```

    }
return resultat;
}

function derouler(i,ch) {
var ch_locale = deroulerAux(i,ch);
if (i == 0)
    window.status = ch_locale;
else
    {
        frames[0].document.open();
        frames[0].document.write(ch_locale.fontcolor('red').fontsize(5));
    }
var cmd = "derouler(" + i + "," + ch_locale + ")";
timerID = setTimeout(cmd,1000);
}

function interfacier() {
var interf = "<FORM>"
+ "<SELECT NAME=choix>"
+ "<OPTION>barre des status"
+ "<OPTION>horizontal"
+ "<OPTION>vertical"
+ "</SELECT>"
+ "<INPUT TYPE=BUTTON VALUE=lancer "
+ "onClick='parent.lancer(this.form.choix.selectedIndex);'"
+ "<INPUT TYPE=BUTTON VALUE=stopper "
+ "onClick='parent.stopper(this.form.choix.selectedIndex);'"
+ "</FORM>";
frames[1].document.open();
frames[1].document.write(interf.fontsize(2).fixed());
frames[1].document.close();
}

// -->
</SCRIPT>
<NOSCRIPT>
Vous ne pouvez ou voulez pas interpreter du JavaScript !
</NOSCRIPT>
</HEAD>

<FRAMESET ROWS='8*,*' onLoad='interfacier();'>
<FRAME SRC='vide.html' NAME='f0' SCROLLING='NO'>
<FRAME SRC='vide.html' NAME='f1'>
</FRAMESET>

</HTML>

```

5.2 L'horloge

```

<HTML>
<HEAD>
<TITLE> Exemple 15 </TITLE>
<SCRIPT LANGUAGE="JavaScript">
<!--
var jour = new Array("dimanche","lundi","mardi",
                    "mercredi","jeudi","vendredi",

```

```

        "samedi");
var mois = new Array("janvier","fevrier","mars","avril",
                    "mai","juin","juillet","aout","septembre",
                    "octobre","novembre","decembre");

function dateFrancaise (uneDate) {
result = jour[eval(uneDate.getDay())] + " "
  + uneDate.getDate() + " "
  + mois[eval(uneDate.getMonth())] + " "
  + uneDate.getYear();
return result;
}

var timerID = null;

function horloge() {

var maintenant = new Date();
var heures = maintenant.getHours();
var minutes = maintenant.getMinutes();
var resultat = "" + heures + ((minutes < 10) ? ":0" : ":") + minutes;

var corps = "<CENTER>"
  + top.dateFrancaise(maintenant).fontcolor('blue')
  + "<BR>"
  + resultat.fontcolor('red')
  + "</CENTER>";

// LANCEMENT DE L'HORLOGE
if (maintenant.getSeconds()==0)
  timerID = setTimeout("top.horloge()",60000);
else {
  var delta = eval(60-maintenant.getSeconds()) * 1000;
  timerID = setTimeout("top.horloge()",delta);
}

/* generation interface */
frames[0].document.open();
frames[0].document.write(corps.fontSize(3).fixed());
frames[0].document.close();
}

// -->
</SCRIPT>
<NOSCRIPT>
Vous ne pouvez ou voulez pas interpreter du JavaScript !
</NOSCRIPT>
</HEAD>

<FRAMESET COLS='*,4*' onLoad='horloge();'>
<FRAMESET ROWS='*,9*'>
<FRAME SRC='vide.html' NAME='f0' SCROLLING='NO'>
<FRAME SRC='vide.html' NAME='f1'>
</FRAMESET>
<FRAME SRC='vide.html' NAME='f2'>
</FRAMESET>

</HTML>

```


5.3 Simuler l'effet GIF89

```
<HTML>
<HEAD>
<TITLE> Exemple 4 </TITLE>
<SCRIPT LANGUAGE="JavaScript">
<!--

// pre-chargement des images
img = new Array();
for(i=0 ; i<=9 ; i++)
{
  img[i] = new Image();
  img[i].src = "http://quartz.dgs.jussieu.fr:8080/icons/GIF/dg" + i + ".gif";
}
// animation en image :
var iter=0;
function animation() {
  document.anim.src = img[iter].src;
  if (iter < 9)
    iter++;
  else iter = 0;
}

// -->
</SCRIPT>
<NOSCRIPT>
Vous ne pouvez ou voulez pas interpreter du JavaScript !
</NOSCRIPT>
</HEAD>
<BODY>
<IMG NAME='anim'
      SRC='about:logo'
      onLoad='setTimeout("animation();",1000)''>
</BODY>
</HTML>
```

6 JavaScript et les menus

Le listing du fichier principal qui permet de définir deux frames :

```
<HTML>
<FRAMESET COLS='205,*' FRAMESPACING='0'
          BORDER='1' FRAMEBORDER='1'
          BORDERCOLOR="#000000" SCROLLING='auto'>
  <FRAME SRC='exemple-19menu.html' NAME='menu'>
  <FRAME SRC='page0.html' NAME='info'>
</FRAMESET>
<NOFRAME>
<BODY BGCOLOR="#FFFFFF" FONT="#0000FF">
Votre navigateur ne supporte pas les <I>frames</I>, il faudrait
envisager de le mettre &agrave; jour...
</BODY>
</NOFRAME>
</HTML>
```

Le listing du fichier des menus qui permet de gérer les événements se rapportant à l'image des menus (et de précharger les images qui vont-êtré utilisées à l'aide d'instructions en JavaScript 1.1) :

```
<HTML>

<SCRIPT LANGUAGE="JavaScript1.1">
<!--
// pre-chargement des IMAGES
img = new Array();
for(i=0 ; i<=7 ; i++)
{
  img[i] = new Image();
  img[i].src = "menu" + i + ".gif";
}
// -->
</SCRIPT>

<SCRIPT LANGUAGE="JavaScript">
<!--
function modifierImage(i) {
top.frames[0].document.images[0].src = "menu" + i + ".gif";
return true;
}
// -->
</SCRIPT>

<BODY>
<MAP NAME="mapMenu">
  <AREA COORDS="0,0,162,19" HREF="page1.html" SHAPE="rect" TARGET="info"
  onMouseOut="modifierImage(0);" onMouseOver="modifierImage(1);">
  <AREA COORDS="0,0,162,38" HREF="page2.html" SHAPE="rect" TARGET="info"
  onMouseOut="modifierImage(0);" onMouseOver="modifierImage(2);">
  <AREA COORDS="0,0,162,57" HREF="page3.html" SHAPE="rect" TARGET="info"
  onMouseOut="modifierImage(0);" onMouseOver="modifierImage(3);">
  <AREA COORDS="0,0,162,76" HREF="page4.html" SHAPE="rect" TARGET="info"
  onMouseOut="modifierImage(0);" onMouseOver="modifierImage(4);">
  <AREA COORDS="0,0,162,95" HREF="page5.html" SHAPE="rect" TARGET="info"
  onMouseOut="modifierImage(0);" onMouseOver="modifierImage(5);">
  <AREA COORDS="0,0,162,114" HREF="page6.html" SHAPE="rect" TARGET="info"
  onMouseOut="modifierImage(0);" onMouseOver="modifierImage(6);">
  <AREA COORDS="0,0,162,133" HREF="page7.html" SHAPE="rect" TARGET="info"
  onMouseOut="modifierImage(0);" onMouseOver="modifierImage(7);">
</MAP>

<IMG SRC="menu0.gif" HEIGHT=136 WIDTH=162 USEMAP="#mapMenu">
</BODY>
</HTML>
```

NOTE : il importe bien-sûr de créer les fichiers page0.html, ..., page7.html et les images menu0.gif, ..., menu7.gif correspondantes.

7 JavaScript et Java

Dans l'exemple qui suit, nous utilisons la console Java de notre navigateur. Ceci peut nous être utile pour "débuguer" nos codes JavaScript (c'est une alternative à la méthode des "alert()").

```
<HTML>
```

```

<HEAD>
<TITLE> Exemple 7 </TITLE>
<SCRIPT LANGUAGE="JavaScript">
<!--
var laSortieStandard = java.lang.System.out;
var laSortieErreur = java.lang.System.err;
laSortieStandard.println("Ceci est un message informatif !");
laSortieErreur.println("Ceci est un message d'erreur !");
// -->
</SCRIPT>
<NOSCRIPT>
Vous ne pouvez ou voulez pas interpreter du JavaScript !
</NOSCRIPT>
</HEAD>
<BODY>
</BODY>
</HTML>

```

8 JavaScript et la validation de formulaire

8.1 Le document HTML

```

<HTML>
<HEAD>
<TITLE> Exemple 6 </TITLE>
<SCRIPT LANGUAGE="JavaScript">
<!--

function estUneChaine(ch) {
    var resultat = false;
    if (ch.length > 0)
        {
            resultat = true;
            for (i=0 ; i<ch.length ; i++)
                if ("abcdefghijklmnopqrstuvwxy-,ABCDEFGHIJKLMNopqrstuvwxyz"
                    .indexOf(ch.charAt(i)) < 0)
                    resultat = false;
        }
    return resultat;
}

function estUnEntier(n) {
    var resultat = false;
    if (n.length > 0)
        {
            resultat = true;
            for (i=0 ; i<n.length ; i++)
                if ("1234567890".indexOf(n.charAt(i)) < 0)
                    resultat = false;
        }
    return resultat;
}

function valider_titre(form) {
    if (form.titre[2].checked == true)
        {

```

```

        form.numSS.value = "1.";
        return true;
    }
    else if ((form.titre[0].checked == true) ||
        (form.titre[1].checked == true))
    {
        form.numSS.value = "2.";
        return true;
    }
    else return false;
}

function valider_nom(form) {
    if (estUneChaine(form.nom.value))
    {
        form.nom.value = form.nom.value.toUpperCase();
        return true;
    }
    else
    {
        window.status = "nom invalide !";
        form.nom.value = "";
        return false;
    }
}

function valider_prenom(form) {
    if (estUneChaine(form.prenom.value))
    {
        form.prenom.value = form.prenom.value.toLowerCase();
        return true;
    }
    else
    {
        window.status = "prenom(s) invalide(s) !";
        form.prenom.value = "";
        return false;
    }
}

function valider_jour(form) {
    if (estUnEntier(form.jour.value))
    {
        if ((form.jour.value > 0) && (form.jour.value < 32))
        {
            window.status = "jour valide !";
            return true;
        }
        else
        {
            window.status = "date invalide !";
            form.jour.value = "";
            return false;
        }
    }
    else
    {

```

```

        window.status = "date invalide !";
        form.jour.value = "";
        return false;
    }
}

function valider_mois(form) {
    var resultat=false;
    valider_titre(form);
    valider_an(form);
    for(i=0 ; i<12 ; i++)
        if (form.mois.options[i].selected == true)
            {
                if (i<9)
                    form.numSS.value += "0";
                form.numSS.value += i+1 + ".";
                resultat=true;
            }
    return resultat;
}

function valider_an(form) {
    if (estUnEntier(form.an.value))
        {
            if ((form.an.value > 1880) && (form.an.value < 1997))
                {
                    window.status = "Annee valide !";
                    valider_titre(form);
                    form.numSS.value += form.an.value.substring(2,4) + ".";
                    return true;
                }
            else
                {
                    // alert("Date invalide !");
                    window.status = "date invalide !";
                    form.an.value = "";
                    return false;
                }
        }
    else
        {
            // alert("Date invalide !");
            window.status = "date invalide !";
            form.an.value = "";
            return false;
        }
}

function valider_commune(form) {
    if (estUneChaine(form.commune.value))
        {
            form.commune.value = form.commune.value.toUpperCase();
            return true;
        }
    else
        {

```

```

        window.status = "commune invalide !";
        form.commune.value = "";
        return false;
    }
}

function valider_dept(form) {
    if (estUnEntier(form.dept.value))
    {
        if ((form.dept.value >= 0) && (form.dept.value < 96))
        {
            valider_titre(form);
            valider_an(form);
            valider_mois(form);
            form.numSS.value += form.dept.value + ".";
            return true;
        }
        else
        {
            window.status = "departement invalide !";
            form.dept.value = "";
            return false;
        }
    }
    else
    {
        window.status = "departement invalide !";
        form.dept.value = "";
        return false;
    }
}

function valider_numSSbref(form) {
    var chaine = form.numSSbref.value;
    var resultat = false;

    if (((chaine.length == 6) && (estUnEntier(chaine))) ||
        ((chaine.length == 7) && (chaine.charAt(3) == ".") &&
        (estUnEntier(chaine.substring(0,3))) &&
        (estUnEntier(chaine.substring(4,7)))))
    {
        valider_titre(form);
        valider_an(form);
        valider_mois(form);
        valider_dept(form);
        form.numSS.value += chaine.substring(0,3) + "."
            + chaine.substring(4,7);

        resultat=true;
    }
    else
    {
        window.status = "numero invalide !";
        form.numSSbref.value = "";
    }
    return resultat;
}

```

```

function valider_numSS(form) {
    var chaine = form.numSS.value;
    var resultat = false;
    if ((chaine.length == 18) && (chaine.charAt(1) == ".") &&
        (chaine.charAt(4) == ".") && (chaine.charAt(7) == ".") &&
        (chaine.charAt(10) == ".") && (chaine.charAt(14) == "."))
        resultat=true;
    else alert("Matricule invalide !");
    return resultat;
}

```

```

function valider(form) {
    resultat = (valider_titre(form) && valider_nom(form) &&
        valider_prenom(form) && valider_an(form) &&
        valider_mois(form) && valider_jour(form) &&
        valider_commune(form) && valider_dept(form) &&
        valider_numSSbref(form));
    if (resultat==false)
        alert("Formulaire a revoir !");
    else
        {
            form.numSS.value = prompt(
                "Confirmez-vous votre identifiant SS : ",
                form.numSS.value);
            if (valider_numSS(form)==true)
                alert("Le formulaire est soumis !");
            else resultat=false;
        }
    return resultat;
}

```

```

// -->
</SCRIPT>
<NOSCRIPT>
Vous ne pouvez ou voulez pas interpreter du JavaScript !
</NOSCRIPT>
</HEAD>
<BODY>
Remplissez le formulaire ci-dessous sans accents&nbsp;nbsp;nbsp;!
<HR>
<FORM METHOD=POST
    ENCTYPE="text/plain"
    ACTION="mailto:leduc@masi.ibp.fr?subject=Formulaire">

<INPUT TYPE=RADIO
    NAME="titre"
    VALUE="madame"
    onClick="valider_titre(this.form);">Mme
<INPUT TYPE=RADIO
    NAME="titre"
    VALUE="mademoiselle"
    onClick="valider_titre(this.form);">Mlle
<INPUT TYPE=RADIO
    NAME="titre"
    VALUE="monsieur"
    onClick="valider_titre(this.form);">Mr
<BR>

```

```

Nom :
<INPUT TYPE=TEXT
    NAME="nom"
    onChange="valider_nom(this.form);"> <BR>
Prénom(s) :
<INPUT TYPE=TEXT
    NAME="prenom"
    onChange="valider_prenom(this.form);"> <BR>
Date de naissance :<BR>
année :
<INPUT TYPE=TEXT
    NAME=an
    SIZE=4
    VALUE=19
    onChange="valider_an(this.form);">
mois :
<SELECT NAME=mois
    onChange="valider_mois(this.form);">
    <OPTION VALUE=1> janvier
    <OPTION VALUE=2> février
    <OPTION VALUE=3> mars
    <OPTION VALUE=4> avril
    <OPTION VALUE=5> mai
    <OPTION VALUE=6> juin
    <OPTION VALUE=7> juillet
    <OPTION VALUE=8> août
    <OPTION VALUE=9> septembre
    <OPTION VALUE=10> octobre
    <OPTION VALUE=11> novembre
    <OPTION VALUE=12> décembre
</SELECT>
jour :
<INPUT TYPE=TEXT
    NAME=jour
    SIZE=2
    onChange="valider_jour(this.form);">
<BR>
Commune de naissance :
<INPUT TYPE=TEXT
    NAME=commune
    onChange="valider_commune(this.form);"> <BR>
Département de naissance :
<INPUT TYPE=TEXT
    NAME=dept
    SIZE=3
    onChange="valider_dept(this.form);"> <BR>
Entrez vos 6 derniers chiffres de numéro de SS :
<INPUT TYPE=TEXT
    NAME="numSSbref"
    SIZE=7
    onChange="valider_numSSbref(this.form);"> <BR>
<HR>
<INPUT TYPE=HIDDEN
    NAME="numSS"> <BR>
<INPUT TYPE=RESET
    VALUE=annuler>
<INPUT TYPE=SUBMIT

```



```
        VALUE=Soumettre
        onClick="return valider(this.form);">
</FORM>

</BODY>
</HTML>
```

8.2 Les résultats

```
From leduc@masi.ibp.fr Sat Jun 28 15:04:02 1997
Sender: leduc@masi.ibp.fr
Date: Sat, 28 Jun 1997 15:00:58 +0100
From: Thomas LEDUC <leduc@masi.ibp.fr>
X-Mailer: Mozilla 3.01 (X11; I; SunOS 4.1.3_U1 sun4m)
Mime-Version: 1.0
To: leduc@masi.ibp.fr
Subject: Formulaire
Content-Type: text/plain
Content-Disposition: inline; form-data
Content-Length: 131
```

```
titre=monsieur
nom=LEDUC
prenom=thomas
an=1970
mois=10
jour=26
commune=PARIS
dept=75
numSSbref=111111
numSS=1.70.10.78.111.111
```

9 En guise de conclusion

9.1 Gérer les évolutions successives

Un petit exemple...

```
<HTML>
<HEAD>
<TITLE> Exemple 18 </TITLE>

<SCRIPT LANGUAGE="JavaScript">
function afficherVersion() {
    var jsVersion;
    if (navigator.userAgent.indexOf("4.0") != -1)
        jsVersion = "1.2";
    else if (navigator.userAgent.indexOf("3.0") != -1)
        jsVersion = "1.1";
    else
        jsVersion = "1.0";
    return jsVersion;
}
</SCRIPT>
</HEAD>

<BODY>
```

```

<SCRIPT LANGUAGE="JavaScript">
document.write("Votre navigateur supporte le JavaScript " + afficherVersion());
</SCRIPT>

</BODY>
</HTML>

```

9.2 Ne pas confondre...

- ne pas confondre le “Navigator JavaScript” et le “LiveWire JavaScript”. Le premier est un langage côté client (“client-side”) et le second est côté serveur (“server-side”). De plus le “LiveWire JavaScript” doit-être compilé en byte-code et le serveur Web correspondant doit supporter un runtime LiveWire JavaScript (c’est le cas du Netscape Server à partir de la version 2). Le LiveWire permet, entre-autres choses, de générer dynamiquement des documents HTML à partir de requêtes SQL sur des bases de données (un peu comme le ferait les CGI). Il existe cependant un dénominateur commun pour le JavaScript côté serveur et le JavaScript côté client : c’est le “core JavaScript” (mêmes mots-clefs, syntaxe, grammaire, expressions, variables, littéraux, modèle objet et mêmes objets et fonctions dits “built-in”). En fait, ce sont surtout les objets plus spécifiques comme ceux liés au navigateur client par exemple, qui diffèrent. En pratique le LiveWire-JavaScript est encapsulé directement dans le code du document HTML entre les balises :

```

< SERVER > ... placer ici votre code LiveWire – JavaScript... < /SERVER >

```

- ne pas confondre JavaScript et Java qui n’ont quasiment rien à voir...

9.3 Les dernières évolutions en bref

Manipulation des expressions régulières, manipulation et paramétrage du navigateur client au sein même du display, gestion de la sécurité et des objets “Layers”...

9.4 Et un peu de documentation...

- le site de référence du JavaScript 1.2 côté client :
<http://developer.netscape.com/library/documentation/communicator/jsguide/js1.2.htm>
- le site de référence du JavaScript 1.1 côté client (attention il est “out of date”) :
<http://home.netscape.com/eng/mozilla/3.0/handbook/javascript/>
- le site de référence du JavaScript 1 côté client (attention il est encore plus “out of date”) :
<http://home.netscape.com/eng/mozilla/Gold/handbook/javascript/>
- la bible de JavaScript (LiveAudio, LiveConnect, les JavaScript Mail Filters, la sécurité) :
<http://developer.netscape.com/library/documentation/communicator/jsguide4/index.htm>
- le site de référence du JavaScript côté serveur :
<http://developer.netscape.com/library/documentation/enterprise/wrijsap/index.htm>
- “toute” la documentation mise en ligne par la *Netscape Communications Corporation* :
<http://developer.netscape.com/library/documentation/index.html>
- le forum `news:comp.lang.javascript`