

Support de cours Javascript

1. Introduction

JavaScript est un langage de programmation qui peut être inclus dans des pages HTML destinées aux navigateurs WWW les plus courants. Grâce à ce langage, il est possible d'écrire des pages interactives. Contrairement à l'utilisation du CGI, les réponses aux événements déclenchés par le lecteur de la page se font au niveau du navigateur, sans nécessité d'une transmission vers le serveur d'où provient la page.

JavaScript ne doit pas être confondu avec Java, qui est un langage beaucoup plus complexe permettant de créer des applications autonomes.

2. Les éléments du langage

2.1 Variables et assignement

Les **variables** sont les lieux de mémorisation des chiffres et des caractères.

En utilisant l'**assignement** on attribue une valeur à une variable. La structure générale de l'assignement est *target = source*.

Pour assigner une chaîne de caractères, on la place entre guillemets doubles ("...") ou simples ('...').

Exemples :

```
a = 3  
b = "hello world"  
c = 'hello from Mars'  
a = b  
a = 3*4*3+2
```

Il existe des notations abrégées pour certains types d'assignements. Par exemple :

Notation abrégée	Signification
$x += y$	$x = x + y$
$x -= y$	$x = x - y$
$x++$	$x = x + 1$
$x--$	$x = x - 1$

En Javascript, il n'y a pas besoin de déclarer le type de variables que l'on utilise, contrairement à des langages évolués tels que le langage C ou Java pour lesquels il faut préciser s'il s'agit d'entier (*int*), de nombre à virgule flottante (*float*), de caractères (*char*), etc.

En fait, Javascript n'autorise la manipulation que de 4 types de données :

- des nombres : entiers ou à virgules
- des chaînes de caractères (*string*) : une suite de caractères

- des booléens : des variables à deux états permettant de vérifier une condition :
- *false*: lors d'un résultat faux
- *true*: si le résultat est vrai
- des variables de type *null* : un mot caractéristique pour indiquer qu'il n'y a pas de données.

2.2 Opérations

2.2.1 Les opérateurs arithmétiques

Les 4 opérations de base sont disponibles en mode texte :

+
-
*
/

Exemples :

somme = somme + 1

b = c - d

somme = somme ^ 5

*resultat = (3 + 5) * (23 / 4) - 3*

2.2.2 Les opérateurs de comparaison

Les opérateurs de comparaison servent à exprimer des conditions. Attention à ne pas confondre l'opérateur de comparaison `==` avec le signe `=` d'assignation.

égal à	<code>==</code>
différent de	<code>!=</code>
supérieur à	<code>></code>
supérieur ou égal à	<code>>=</code>
inférieur ou égal à	<code><</code>
inférieur ou égal à	<code><=</code>

2.2.3 Les opérateurs logiques

ET logique	<code>&&</code>
OU logique	<code> </code>
NON logique	<code>!</code>

2.3 La séquence

Pour exécuter des instructions en séquence, il suffit d'écrire chaque instruction suivie d'un point-virgule :

```

<instruction> ;
<instruction> ;
<instruction> ;
...
<instruction> ;

```

Exemple :

```

a = 15;
b = 23;
c = 2 * a + b;

```

2.4 Les choix (sélecteurs)

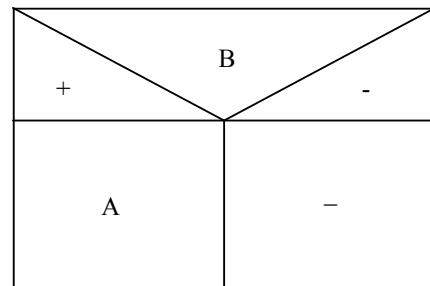
2.4.1 choix avec une alternative

Formulation générale :

```

if (<condition>) {
  <action>;
  <action>;
  ...
}

```



Exemple :

```

if (j == 5) {
  Somme = Somme + 1;
}

```

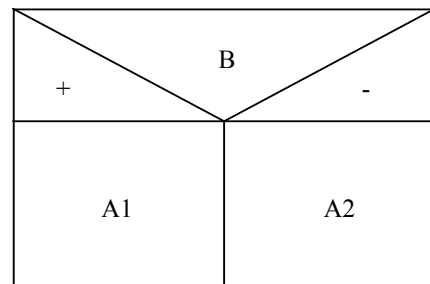
2.4.2 choix avec deux alternatives

Formulation générale :

```

if (<condition>) {
  <action>;
  <action>;
  ...
} else {
  <action>
  <action>
  ...
}

```



Exemple :

```

if (j == 5) {
  Somme = Somme + 1 ;
} else {
  Somme = Somme - 1 ;
}

```

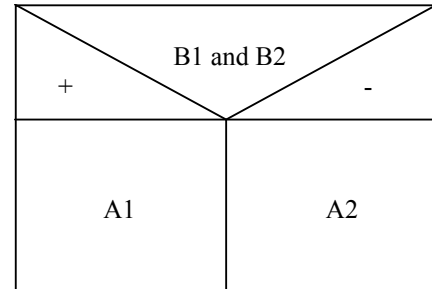
2.4.3 choix avec alternatives reliées

Formulation générale :

```
if ( <cond1> <opérateur> <cond2> ) {
    <action>
    <action>
    ...
}
```

Exemples :

```
if ( NP < 4000 && NP >= 3900 ) {
    Canton = "Valais";
}
if ( !(x < 5 && y > 7) ) {
    resultat = "correct";
}
```



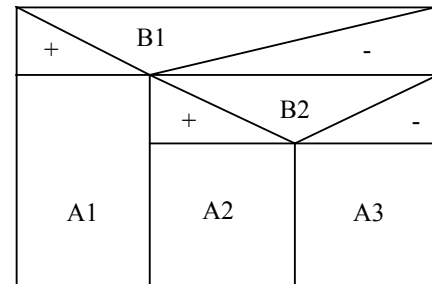
2.4.4 choix avec conditions imbriquées

Formulation :

```
if (<condition1>) {
    <action>
} else {
    if (<condition2>) {
        <action>
        <action>
        ...
    } else {
        <action>
        <action>
        ...
    }
}
```

Exemple :

```
if ( reponse == 1 ) {
    cadeau = "cigarettes";
} else {
    if ( reponse == 3 ) {
        cadeau = "fleurs";
    } else {
        cadeau = "chaussettes";
    }
}
```



2.5 Les boucles

Il est très utile de disposer d'une instruction permettant d'effectuer de manière répétitive une série d'opérations. JavaScript propose plusieurs types de boucles itératives. Pour chacune des boucles, la somme des nombres de 1 à 5 sera calculée.

2.5.1 while

Formulation générale :

```
while ( <condition> ) {  
    <action>;  
    ...  
}
```

Signification: Exécute les <actions> aussi longtemps que la <condition> est vraie. Si la condition est fausse au début, aucune instruction n'est exécutée.

Exemple :

```
chiffre = 0 ;  
somme = 0 ;  
while ( chiffre < 5 ) {  
    chiffre = chiffre + 1 ;  
    somme = somme + chiffre ;  
}
```

2.5.2 do ... while

Formulation générale :

```
do {  
    <action>  
    ...  
} while ( <condition> )
```

Signification: Exécute les <actions> aussi longtemps que la <condition> est vraie. Si la condition est fausse au début, les instructions seront quand même exécutées une seule fois.

Exemple :

```
chiffre = 0 ;  
somme = 0 ;  
do {  
    chiffre = chiffre + 1 ;  
    somme = somme + chiffre ;  
} while ( chiffre < 5 )
```

2.5.3 for

Formulation générale :

```
for ( <expression de depart> ; <condition de continuation> ; <incrementation> ) {  
    <action>  
    ...  
}
```

Exemple:

```
somme = 0 ;  
for ( chiffre = 1 ; chiffre <= 5 ; chiffre = chiffre + 1 ) {  
    somme = somme + chiffre ;  
}
```

Signification : La boucle *for* se déroule de la manière suivante :

Tant que la condition de continuation est vraie :

- en partant de l'expression de départ (*chiffre = 1*) on exécute le contenu des accolades;
- la variable (*chiffre*) est incrémentée (*chiffre = chiffre + 1*) et on exécute le contenu des accolades autant de fois que nécessaire;
- la boucle ne prend fin que lorsque la condition de continuation devient fausse.

2.6 Fonctions et procédures

On appelle fonction un sous-programme qui permet d'effectuer un ensemble d'instruction par simple appel de la fonction dans le corps du programme principal. Cette notion de sous-programme est généralement appelée fonction dans les langages autres que le Javascript (toutefois leur syntaxe est généralement différente...). Les fonctions et les procédures permettent d'exécuter dans plusieurs parties du programme une série d'instructions, cela permet une simplicité du code et donc une taille de programme minimale. Dans JavaScript, les fonctions et les procédures sont définies par le mot clé *function*.

La différence entre une fonction et une procédure est que la fonction retourne une valeur (numérique, booléen etc.), ce qui n'est pas le cas pour une procédure. Ce retour de valeur se fait par le mot clé *return*.

Avant d'être utilisée, une fonction doit être définie car pour l'appeler dans le corps du programme il faut que le navigateur la connaisse, c'est-à-dire qu'il connaisse son nom, ses arguments et les instructions qu'elle contient. La définition d'une fonction s'appelle "déclaration". La déclaration d'une fonction se fait grâce au mot clé *function* selon la syntaxe suivante::

```
function nom(parametre1,parametre2,...,parametreN) {  
    ....  
}
```

Exemple :

```
function carre(nombre) {  
    resultat = nombre * nombre ;  
    return(resultat)  
}
```

Pour exécuter une fonction, il suffit de faire appel à elle en écrivant son nom (en respectant la casse) suivie d'une parenthèse ouverte (éventuellement des arguments) puis d'une parenthèse fermée.

Exemple :

```
Nom_De_La_Fonction();
```

2.7 Méthodes

Une méthode est une fonction associée à un objet, c'est-à-dire une action que l'on peut faire exécuter à un objet. Les méthodes des objets du navigateur sont des fonctions définies à l'avance par les normes HTML, on ne peut donc pas les modifier, il est toutefois possible de créer une méthode personnelle pour un objet que l'on a créé soi-même.

Prenons par exemple une page HTML, elle est composée d'un objet appelé *document*. L'objet *document* a par exemple la méthode *write()* qui lui est associée et qui permet de modifier le contenu de la page HTML en affichant du texte. Une méthode s'appelle un peu comme une propriété, c'est-à-dire de la manière suivante :

```
window.objet1.objet2.methode()
```

Dans le cas de la méthode *write()*, l'appel se fait comme suit :

```
window.document.write()
```

3. JavaScript et HTML

3.1 La balise <SCRIPT>

Pour pouvoir fonctionner, le code JavaScript d'une page HTML doit en principe être placé à l'intérieur d'une balise <SCRIPT>.

Une page HTML peut contenir plusieurs balises <SCRIPT>, mais elles ne doivent pas être imbriquées.

On peut placer une balise <SCRIPT> soit dans l'entête (*HEAD*), soit dans le corps (*BODY*) de la page HTML. On placera de préférence une balise <SCRIPT> contenant les procédures et les fonctions dans l'entête, car cela permet qu'elles soient chargées avant le reste de la page.

Comme il existe d'autres langages utilisés de la même manière que JavaScript, il est nécessaire d'ajouter l'attribut *LANGUAGE="JavaScript"* dans la balise <SCRIPT>.

```
<SCRIPT LANGUAGE="JavaScript">  
...code JavaScript...  
</SCRIPT>
```

Il est aussi possible d'ajouter des scripts à une page HTML à partir d'un fichier. Dans ce cas, on ajoute à la balise <SCRIPT> un paramètre précisant le nom du fichier contenant les scripts :

```
<SCRIPT LANGUAGE="JavaScript" SRC="Nom_du_fichier.js">
```

3.2 Affichage et introduction des données : *alert* et *prompt*

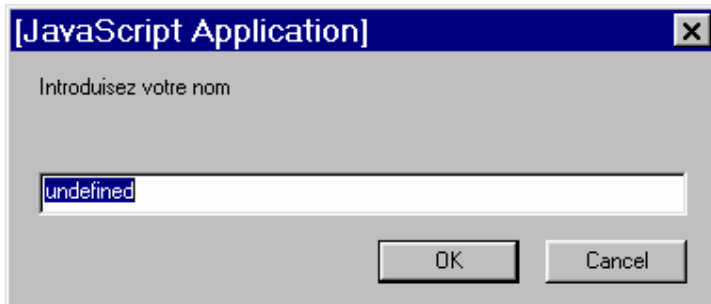
La fonction *alert* sert à afficher une valeur. Voici un exemple:

```
alert ("Hello World !");
```



La fonction *prompt* sert à lire une valeur. Exemple:

```
x = prompt("Introduisez votre nom");
```



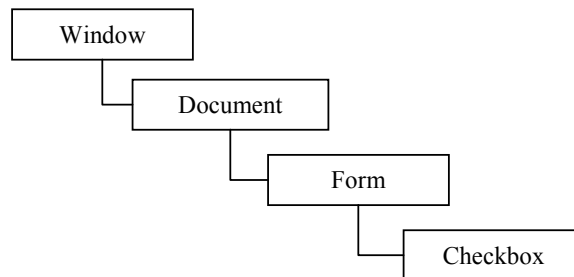
Après avoir cliqué sur **OK**, la variable *x* contient la chaîne de caractères qui a été introduite.

3.3 Les objets

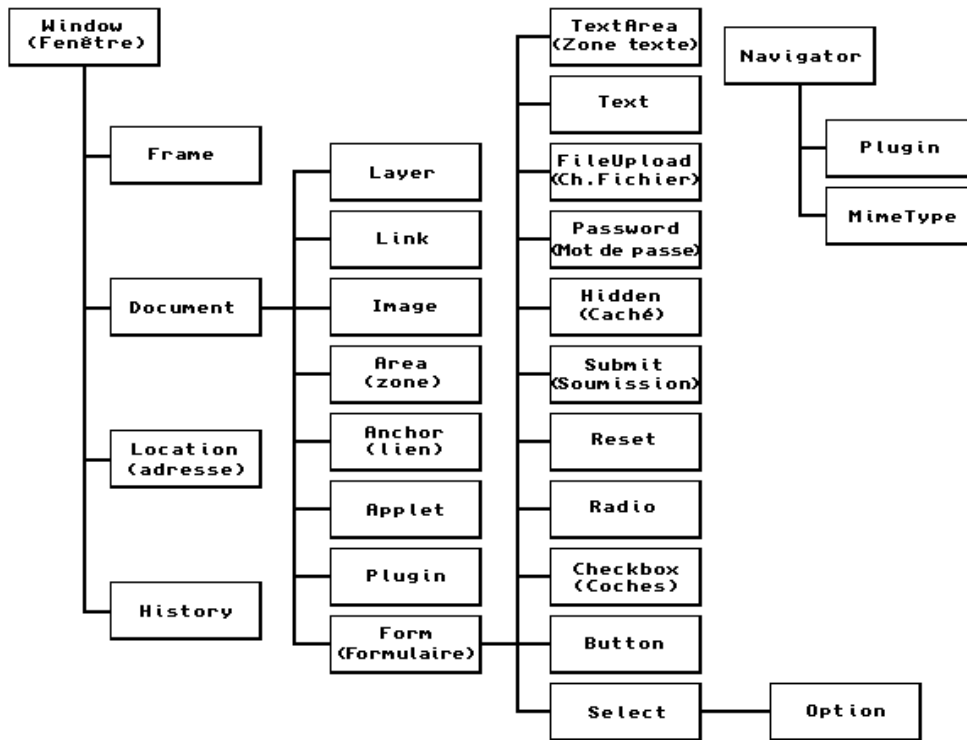
Javascript traite les éléments qui s'affichent dans votre navigateur comme des objets, c'est-à-dire des éléments classés selon :

- une hiérarchie pour pouvoir les désigner précisément
- auxquels on associe des propriétés

Par exemple, pour atteindre un bouton à l'intérieur d'un formulaire, la hiérarchie est :



La hiérarchie des objets est la suivante :



L'accès aux objets se fait par une notation par points. D'autre part, comme il peut y avoir plusieurs formulaires et plusieurs boutons dans chaque formulaire, certains objets sont automatiquement numérotés. Ainsi, accéder au premier bouton du premier formulaire d'une page web se fait par la notation suivante :

```
window.document.forms[0].checkbox[0]
```

Le [0] représente ici le premier élément (le [1], le deuxième, et ainsi de suite).

Il est aussi possible d'utiliser le nom de l'objet directement. Ainsi, si le formulaire s'appelle *enquete* (évités les accents dans les noms d'objets) et le bouton s'appelle *courrier*, il sera possible d'y accéder de la manière suivante :

```
window.document.enquete.courrier
```

Enfin, il est aussi possible d'utiliser la notation :

```
window.document.form["enquête"].buton["courrier"]
```

3.4 Les événements

Les événements sont des actions de l'utilisateur qui vont pouvoir donner lieu à une interactivité. L'événement par excellence est le clic de souris, car c'est le seul que le HTML gère. Grâce au Javascript il est possible d'associer des fonctions, des méthodes à des événements tels que le passage de la souris au-dessus d'une zone, le changement d'une valeur dans un champ, etc.

Ce sont les gestionnaires d'événements qui permettent d'associer une action à un événement. La syntaxe d'un gestionnaire d'événement est la suivante :

```
onEvenement="Action_Javascript_ou_Fonction()";
```

Les gestionnaires d'événements sont associés à des objets, et leur code s'insère dans la balise de ceux-ci. Chaque événement ne peut pas être associé à n'importe quel objet. Ainsi par exemple, l'événement *OnChange* ne peut pas s'appliquer à un lien hypertexte.

Le tableau ci-dessous présente les événements par liste alphabétique :

<i>Evénement</i>	Objets affectés	Description
<i>onAbort</i>	Images	S'exécute en cas d'arrêt de chargement, par appui sur le boutons stop ou par un clic prématuré sur un lien
<i>onBlur</i>	Fenêtres et tous les éléments de formulaire	S'exécute quand on quitte la fenêtre ou un objet de formulaire
<i>onChange</i>	Champs texte, zones texte, listes de sélection	S'exécute quand un élément de formulaire est modifié
<i>onClick</i>	Boutons, boutons radio, boutons submit et reset, liens	S'exécute quand on clique dans ou sur un élément
<i>onDragDrop</i>	Fenêtres	S'exécute quand on pose un élément à l'intérieur de la fenêtre du navigateur à l'aide la souris
<i>onError</i>	Images, fenêtres	S'exécute quand le chargement de l'image ou de la fenêtre provoque une erreur
<i>onFocus</i>	Fenêtres et tous les éléments de formulaire	S'exécute quand on sélectionne la fenêtre ou l'objet formulaire
<i>onKeyDown</i>	Documents, images, liens, zones texte	S'exécute quand une touche du clavier est pressée
<i>onKeyPress</i>	Documents, images, liens, zones texte	S'exécute quand on appuie et maintient une touche du clavier
<i>onKeyUp</i>	Documents, images, liens, zones texte	S'exécute quand on relâche une touche du clavier
<i>onLoad</i>	Documents	S'exécute quand le document se charge
<i>onMouseDown</i>	Documents, boutons, liens	S'exécute quand on clique avec le bouton de la souris
<i>onMouseMove</i>	rien par défaut	S'exécute quand on bouge la souris
<i>onMouseOut</i>	Cartes, liens	S'exécute quand le pointeur de la souris sort d'une zone de sélection graphique ou un lien
<i>onMouseOver</i>	liens	S'exécute quand le pointeur de la souris passe sur un lien
<i>onMouseUp</i>	Documents, boutons, liens	S'exécute quand on relâche le bouton de la souris

<i>onMove</i>	Fenêtres	S'exécute quand l'utilisateur ou un script bouge une fenêtre
<i>onReset</i>	Formulaires	S'exécute quand on "resete" un formulaire
<i>onResize</i>	Fenêtres	S'exécute quand l'utilisateur ou un script change la taille d'une fenêtre
<i>onSelect</i>	Champs ou zones texte	S'exécute quand on sélectionne une zone ou un champ texte (clavier ou souris)
<i>onSubmit</i>	Formulaire	S'exécute au moment de l'envoi d'un formulaire
<i>onUnload</i>	Documents	S'exécute quand on quitte le document

Le tableau ci-dessous résume les objets et les événements associés :

Objet	Événements associables
Lien hypertexte	<i>onClick, onMouseOver, onMouseOut</i>
Page du navigateur	<i>onLoad, onUnload</i>
Bouton, Case à cocher, Bouton radio, Bouton Reset	<i>onClick</i>
Liste de sélection d'un formulaire	<i>onBlur, onChange, onFocus</i>
Bouton Submit	<i>onSubmit</i>
Champ de texte et zone de texte	<i>onBlur, onChange, onFocus, onSelect</i>

Exemple :

```
<html> <head>
<title>Ouverture d'une boite de dialogue lors d'un click</title> </head>
<body>
<a href="javascript:;" onClick="window.alert('Message d'alerte a utiliser avec
moderation');">Cliquez ici!</a>
</body>
</html>
```

Annexe : Propriétés des objets

A.1 Objets fenêtre

Object	Propriétés	Methodes
window (fenêtre)	<ul style="list-style-type: none"> ■ clear (fermé) ■ closed (fermé) ■ defaultStatus (texte ligne status) ■ document (document) ■ frames (cadres) ■ history (historique) ■ innerHeight (hauteur interne) ■ innerWidth (largeur interne) ■ length (longueur) ■ location (adresse) ■ locationbar (barre d'adresse) ■ menubar (barre de menu) ■ name (nom) ■ opener (source ouverture) ■ outerHeight (hauteur externe) ■ outerWidth (largeur externe) ■ pageXOffset (décalage horizontal) ■ pageYOffset (décalage vertical) ■ parent (parent) ■ personalbar (barre adresses personnelles) ■ scrollbars (glissères) ■ self (sois-même) ■ status (état) ■ statusBar (ligne d'état) ■ toolbar (barre d'outils) ■ top (haut) ■ window (fenêtre) 	<ul style="list-style-type: none"> ■ alert (message d'alerte) ■ back (arrière) ■ blur (désélection) ■ captureEvents (événement) ■ clearInterval (efface compteur) ■ clearTimeout (efface délai) ■ close (fermeture) ■ confirm (panneau de demande) ■ disableExternalCapture (pas de capture externe) ■ enableExternalCapture (capture externe (souris)) ■ find (cherche) ■ focus (sélection) ■ forward (avant) ■ handleEvent (événement) ■ home (page par défaut) ■ moveBy (avance ou recule de) ■ moveTo (avance ou recule à) ■ open (ouvre) ■ print (imprime) ■ prompt (panneau de demande) ■ releaseEvents (événement) ■ resizeBy (change taille de) ■ resizeTo (change taille à) ■ routeEvent (événement) ■ scroll (fait défiler) ■ scrollBy (fait défiler de) ■ scrollTo (fait défiler à) ■ setInterval (valeur intervalle du compteur) ■ setTimeout (valeur du délai) ■ stop (stop)

<p>location (adresse)</p>	<ul style="list-style-type: none"> ■ hash (référence dans page, ex: #ref1) ■ host (machine hôte) ■ hostname (nom de la machine hôte) ■ href (document) ■ pathname (chemin) ■ port (port utilisé, ex: 80, 8000...) ■ protocol (protocole: HTTP, FTP, etc...) ■ search (recherche) 	<ul style="list-style-type: none"> ■ reload (relit) ■ replace (remplace)
<p>history (historique)</p>	<ul style="list-style-type: none"> ■ current (courant) ■ length (longueur) ■ next (suivant) ■ previous (précédent) 	<ul style="list-style-type: none"> ■ back (arrière) ■ forward (avant) ■ go (vas à)
<p>screen (écran)</p>	<ul style="list-style-type: none"> ■ availHeight (Hauteur disponible) ■ availWidth (largeur disponible) ■ colorDepth (nombre de plans couleurs) ■ height (hauteur) ■ pixelDepth (profondeur du point) ■ width (largeur) 	

A.2 Objets document

Object	Propriétés	Methodes
document (document)	<ul style="list-style-type: none"> ■ alinkColor (couleur liens actif <BODY ALINK="">) ■ anchors (liens) ■ applets (programmes Java externes <APPLET>) ■ bgColor (couleur de fond <BODY BGCOLOR="">) ■ cookie (cookies (reconnaissance)) ■ domain (domaine) ■ embeds (programmes externes <EMBED>) ■ fgColor (couleur du texte <BODY TEXT="">) ■ formName (nom formulaire) ■ forms (formulaires) ■ images (images) ■ lastModified (dernière modification) ■ layers (couches) ■ linkColor (couleur des liens <BODY LINK="">) ■ links (documents liés <LINK="">) ■ plugins (programmes externes) ■ referrer (adresse page précédente) ■ title (titre) ■ URL (adresse) ■ vlinkColor (couleur liens visités <BODY VLINK="">) 	<ul style="list-style-type: none"> ■ captureEvents (événements) ■ close (fermeture) ■ getSelection (texte sélectionné) ■ handleEvent (piège événements) ■ open (ouverture) ■ releaseEvents (arrêt piège événements) ■ routeEvent (événements) ■ write (écrit des caractères) ■ writeln (écrit des lignes)
link (attaches)	<ul style="list-style-type: none"> ■ hash (référence dans page, ex: #ref1) ■ host (machine hôte) ■ hostname (nom de la machine hôte) ■ href (document attaché) ■ pathname (chemin) ■ port (port utilisé, ex: 80, 8000...) ■ protocol (protocole: HTTP, FTP, etc...) 	<ul style="list-style-type: none"> ■ HandleEvent (événement)

	<ul style="list-style-type: none"> ■ search (recherche) ■ target (destination) ■ text (texte) 	
images (images)	<ul style="list-style-type: none"> ■ border (bord) ■ complete (chargement fini) ■ clear (efface) ■ height (hauteur) ■ hspace (espace horizontal) ■ lowsrc (?) ■ name (nom) ■ prototype (prototype) ■ src (nom du fichier) ■ vspace (espace vertical) ■ width (largeur) 	<ul style="list-style-type: none"> ■ HandleEvent (événement)
layers (couches)	<ul style="list-style-type: none"> ■ above (dessus) ■ background (image de fond) ■ bgColor (couleur de fond) ■ below (dessous) ■ clip.bottom (coordonée du bas pour découpe) ■ clip.height (coordonée de la hauteur pour découpe) ■ clip.left (coordonée gauche pour découpe) ■ clip.right (coordonée droite pour découpe) ■ clip.top (coordonée du haut pour découpe) ■ clip.width (coordonée de la largeur pour découpe) ■ document (nom du document) ■ name (nom de la couche) ■ pageX (?) ■ pageY (?) ■ parentLayer (couche parent) ■ siblingAbove (?) ■ siblingBelow (?) ■ src (nom de fichier) ■ top (haut) 	<ul style="list-style-type: none"> ■ captureEvents événements ■ handleEvent (événement) ■ load (charge) ■ moveAbove (souris dessus) ■ moveBelow (souris dessous) ■ moveBy (déplace de) ■ moveTo (déplace à) ■ moveToAbsolute (déplace à (valeur absolue)) ■ releaseEvents (événement) ■ resizeBy (change taille de) ■ resizeTo (change taille à) ■ routeEvent (événement)

	<ul style="list-style-type: none"> ■ visibility (affichage) ■ zIndex (?) 	
--	--	--

A.3 Objets formulaire

Object	Propriétés	Methodes
form (formulaire)	<ul style="list-style-type: none"> ■ action (action) ■ elements (éléments) ■ encoding (codage) ■ length (longueur) ■ method (méthode) ■ name (nom) ■ target (cible) 	<ul style="list-style-type: none"> ■ handleEvent (événement) ■ reset (reset) ■ submit (soumission)
hidden (champ caché)	<ul style="list-style-type: none"> ■ form (formulaire) ■ name (nom) ■ type (type) ■ value (valeur) 	
text (champ texte)	<ul style="list-style-type: none"> ■ defaultValue (valeur par défaut) ■ form (formulaire) ■ name (nom) ■ type (type) ■ value (valeur) 	<ul style="list-style-type: none"> ■ blur (désélection) ■ focus (sélection) ■ handleEvent (événement) ■ select (actif)
textarea (zone texte)	<ul style="list-style-type: none"> ■ defaultValue (valeur par défaut) ■ form (formulaire) ■ name (nom) ■ type (type) ■ value (valeur) 	<ul style="list-style-type: none"> ■ blur (désélection) ■ focus (sélection) ■ handleEvent (événement) ■ select (actif)
password (mot de passe)	<ul style="list-style-type: none"> ■ defaultValue (valeur par défaut) ■ form (formulaire) ■ name (nom) ■ type (type) ■ value (valeur) 	<ul style="list-style-type: none"> ■ blur (désélection) ■ focus (sélection) ■ handleEvent (événement) ■ select (actif)
fileupload (champ fichier)	<ul style="list-style-type: none"> ■ form (formulaire) ■ name (nom) ■ type (type) ■ value (valeur) 	<ul style="list-style-type: none"> ■ blur (désélection) ■ focus (sélection) ■ handleEvent (événement) ■ select (actif)
button	<ul style="list-style-type: none"> ■ form (formulaire) 	<ul style="list-style-type: none"> ■ blur (désélection)

(bouton)	<ul style="list-style-type: none"> ■ name (nom) ■ type (type) ■ value (valeur) 	<ul style="list-style-type: none"> ■ click (cliqué) ■ focus (sélection) ■ handleEvent (événement)
submit (Bouton submit)	<ul style="list-style-type: none"> ■ form (formulaire) ■ name (nom) ■ type (type) ■ value (valeur) 	<ul style="list-style-type: none"> ■ blur (désélection) ■ click (cliqué) ■ focus (sélection) ■ handleEvent (événement)
reset (bouton reset)	<ul style="list-style-type: none"> ■ form (formulaire) ■ name (nom) ■ type (type) ■ value 	<ul style="list-style-type: none"> ■ blur (désélection) ■ click (cliqué) ■ focus (sélection) ■ handleEvent (événement)
radio (bouton radio)	<ul style="list-style-type: none"> ■ checked (enfoncé) ■ defaultChecked (enfoncé par défaut) ■ form (formulaire) ■ name (nom) ■ type (type) ■ value (valeur) 	<ul style="list-style-type: none"> ■ blur (désélection) ■ click (cliqué) ■ focus (sélection) ■ handleEvent (événement)
checkbox (bouton à cocher)	<ul style="list-style-type: none"> ■ checked (coché) ■ defaultChecked (coché par défaut) ■ form (formulaire) ■ name (nom) ■ type (type) ■ value (valeur) 	<ul style="list-style-type: none"> ■ blur (désélection) ■ click (cliqué) ■ focus (sélection) ■ handleEvent (événement)
select (sélection)	<ul style="list-style-type: none"> ■ form (formulaire) ■ length (longueur) ■ name (nom) ■ options (menus) ■ selectedIndex (menu sélectionné) ■ type (type) 	<ul style="list-style-type: none"> ■ blur (désélection) ■ focus (sélection) ■ handleEvent (événement)
options (menu)	<ul style="list-style-type: none"> ■ defaultSelected (menu sélectionné par défaut) ■ selected (menu sélectionné) ■ text (texte) ■ value (valeur) 	

A.4 Objet navigateur

Object	Propriétés	Methodes
navigator (navigateur) (browser) (baladeur)	<ul style="list-style-type: none"> ■ appCodeName (nom de code de l'application) ■ appName (nom de l'application) ■ appVersion (version de l'application) ■ language (langue) ■ mimeTypes (formats connus) ■ platform (machine utilisée) ■ plugins (programmes externes) ■ userAgent (nom du navigateur) 	<ul style="list-style-type: none"> ■ javaEnabled (Java actif) ■ plugins.refresh (relecture des programmes externes) ■ preference (préférences) ■ taintEnabled (fonction taint active)
MimeType (Formats connus)	<ul style="list-style-type: none"> ■ description (description) ■ enabledPlugin (programmes autorisés) ■ suffixes (suffixes) ■ type (type) 	
Plugin (Programmes externes)	<ul style="list-style-type: none"> ■ description (description) ■ filename (nom du programme) ■ length (longueur du programme) ■ name (nom du plugin) 	