# IV Les principaux objets

Comme vu précédemment, un certain nombre d'objets sont prédéfins par la somme javascript (Voir le DOM). Certains sont plus utilisés que d'autres. Ce sont ceux-ci que nous allons découvrir.

# IV.1 L'objet window

Cet objet est le parent de tous les autres.

Par défaut, si vous utilisez une méthodes ou une propriété sans préciser l'objet auquel elle se rattache, c'est alors à l'objet window qu'elle est associée.

window.open() équivaut à open().

#### <u>a. Les propriétés</u>

#### 1. Récupération des propriétés

Ces propriétés peuvent être récupérées en utilisant la fonction for déjà vu dans le chapitre précédent. Il faut juste supprimer de l'affichage à l'ecran les éléments qui ne sont ps des propriétés comme les méthodes (function).

#### Attention

Les propriétés ne sont pas toutes gérées par les différents navigateurs. Certains respectent la norme (Mozilla, Netscape ..) et d'autres non (Internet Explorer). Il existe des différences importantes qui ne facilitent pas le travail du développeur.

Ces différences sont présentées dans le tableau ci-dessous. Les résultats proviennent de Netscape version 7.01, et IE version 6.0. Mozilla version 1.1b utilise le même moteur que Nestcape et le résultat est identique. Les tests avec Opera et Konqueror n'ont pas permis d'obtenir de résultats du fait certainement des privilèges appliquées sur ces deux navigateurs.

Netscape 7.01	Internet Explorer 6.0
window.closed	window.clientInformation
window.content	window.clipboardData
window.controllers	window.closed
window.crypto	window.defaultStatus
window.defaultStatus	window.document

window.directories	window.event
window.frames	window.external
window.history	window.frameElement
window.innerHeight	window.frames
window.innerWidth	window.history
window.length	window.Image
window.locationbar	window.length
window.menubar	window.location
window.name	window.name
window.opener	window.navigator
window.outerHeight	window.offscreenBuffering
window.outerWidth	window.onafterprint
window.pageXOffset	window.onbeforeprint
window.pageYOffset	window.onbeforeunload
window.parent	window.onblur
window.personalbar	window.onerror
window.pkcs11	window.onfocus
window.screen	window.onhelp
window.screenX	window.onload
window.screenY	window.onresize
window.scrollbars	window.onscroll
window.scrollX	window.onunload
window.scrollY	window.opener
window.self	window.Option
window.sidebar	window.parent
window.status	window.screenLeft
window.statusbar	window.screen
window.toolbar	window.screenTop
window.top	window.self
	window.status
	window.top
	window.window

Pour fournir un code portable il est de votre intérêt de ne pas privilégier l'un ou l'autre des navigateurs et par conséquent de n'utiliser dans la mesure du possible que les propriétés communes aux deux navigateurs. Le tableau ci-dessous indique ces propriétés communes.

Numéro du	Propriétés communes à Netscape
paragraphe	7.01 et Internet Explorer 6.0
2	window.closed
3	window.defaultStatus
4	window.frames
5	window.history
6	window.length
7	window.name
8	window.opener

9	window.parent
10	window.screen
11	window.self
12	window.status
13	window.top

#### Remarque

Il est à noter que bizaremment la propriété document n'est pas présente. Netscape et Mozilla ne la proposent pas même si c'est une propriété importante du DOM.

Afin d'obtenir ce tableau j'ai utilisé le programme suivant exécuté sous les différents navigateurs et sauvegardé au format texte. J'ai ensuite isolé les propriétés, ainsi que les objets que l'on retrouve aussi sous forme de propriétés. Une fois ceci effectué, j'ai effectué une différence entre les deux fichiers (Netscape et IE) et tout ceci via un script shell sous Unix.

Le format brut des résultats est sous la forme suivante :

```
...
window.parent = [object Window]
window.top = [object Window]
window.scrollbars = [object BarProp]
window.name =
window.scrollX = 0
window.scrollY = 0
window.scrollTo = function scrollTo() { [native code] }
window.scrollBy = function scrollBy() { [native code] }
window.getSelection = function getSelection() { [native code] }
window.scrollByLines = function scrollByLines() { [native code] }
```

Le programme javascript est le suivant :

<HTML> <HEAD> </HEAD> <BODY> <script langage="javascript"> for(var i in window)

```
document.write("window."+i+" = " + window[i] + "<BR>");
}
</script>
</BODY>
</HTML>
```

#### 2. closed

Cette propriété permet de tester l'état d'une fenêtre ouverte ou fermée. closed prend la valeur true si la fenêtre testée a été fermée. Cette propriété est souvent utilisée en même temps que la méthode open() qui permet elle d'ouvrir une fenêtre afin de savoir si la fenêtre en question est bien ouverte.

Exemple

	Code
1	<html></html>
2	<head></head>
3	
4	<body></body>
5	<script langage="javascript"></td></tr><tr><td>6</td><td>var fenetre=window.open("open.html","f2");</td></tr><tr><td>7</td><td>if(fenetre.closed)</td></tr><tr><td>8</td><td>{</td></tr><tr><td>9</td><td>alert("Fenetre fermee ->valeur = " + fenetre.closed);</td></tr><tr><td>10</td><td>}</td></tr><tr><td>11</td><td>else</td></tr><tr><td>12</td><td>{</td></tr><tr><td>13</td><td>fenetre.document.bgColor="blue";</td></tr><tr><td>14</td><td>alert("Fenetre ouverte ->valeur = " + fenetre.closed);</td></tr><tr><td>15</td><td>fenetre.close();</td></tr><tr><td>16</td><td>if(fenetre.closed)</td></tr><tr><td>17</td><td>{</td></tr><tr><td>18</td><td>alert("Fenetre fermee ->valeur = " + fenetre.closed);</td></tr><tr><td>19</td><td>}</td></tr><tr><td>20</td><td></td></tr><tr><td>21</td><td></script>
22	
23	

La ligne 6 permet l'ouverture d'une nouvelle fenêtre (objet fenetre) suite au chargement de la page qui contient le javascript.

La ligne 7 permet d'effectuer un test pour déterminer l'état de la nouvelle fenêtre.

Les lignes 13 à 18 permettent d'enchaîner un ensemble d'actions lorsque la nouvelle fenêtre est ouverte.

13 Mise en bleu du fond de cette nouvelle fenêtre

- 14 Avertissement de l'ouverture de la fenêtre
- 15 Fermeture de la nouvelle fenêtre
- 16 Nouveau test et affichage de l'état « fermé » de la nouvelle fenêtre

#### 3. DefautIstatus

Cette propriété concerne le contenu de la zone située en dessous de la fenêtre d'affichage. Vous pouvez y insérer des informations textes. C'est à cet endroit que vous voyez généralement apparaître l'adresse de I'URL avant le clic sur le lien hypertexte (Figure 1) ou l'information indiquant que le document est chargé (figure 2).

			🔁 <u>Mes pages façon nouvelle cuisine</u>
*	<u>1931</u>	Ý	http://www.cicrp.jussieu.fr/jacqueno/pagencuisine.html
Fia	ıre 1		

i iyule l

× 10	🤣 🚳	Document: Done (0.27 secs)
224× 1	W Est	Document. Done (0.27 secs)

#### Figure 2

Vous pouvez aussi y faire défiler du texte au moyen d'une petite astuce.

#### Exemple

	Code
1	<html></html>
2	<head></head>
3	<script langage="javascript"></script>

8	{
9	if(msg.length>0)
10	{
11	window.defaultStatus=msg;
12	window.setTimeout('scroll_status()',100);
13	msg=msg.substring(1,msg.length);
14	}
15	else
16	{
17	window.defaultStatus="> Jacquenod Frédéric";
18	}
19	}
20	
21	
22	<body onload="scroll_status()"></body>
23	
24	
25	

Ligne 4 et 5 déclaration d'une variable msg qui contient le texte qui va défiler.

Ligne 7 création d'une fonction qui est activée lors du chargement de la page html (ligne 22).

L'astuce est d'utiliser la propriété length de l'objet window et de supprimer la première lettre de la variable msg à chaque passage et ceci tant que la longueur de msg n'est pas nulle (ligne 11 à 13).

Lorsque cette taille est nulle on affecte à la propriété defaultStatus une nouvelle valeur qui reste alors fixe dans la barre de status (voir copie d'écran).

			Capture écran
×	100	×	> Jacquenod Frédéric

#### Attention

Il existe une propriété status (voir paragraphe 12). La différence est minime mais existe. L'effet de la propriété DefaultStatus est permanente. Si un événement vient écrire par dessus le texte, une fois l'événement terminé, l'affichage réapparaît. Cela montre aussi que la propriété status a priorité sur la propriété DefaultStatus.

#### 4. frames

Cette propriété est un tableau qui contient les informations sur tous les codes en rapport avec votre fenêtre et donc avec l'objet window (voir aussi l'objet frame).

#### Attention

Il ne faut pas confondre la propriété frames avec l'objet Frame.

La création d'une frame permet de décomposer votre page en plusieurs sous-ensemble d'affichage. Chaque sous-ensemble est alors « indépendant ».

Pour cela les balises **frameset** et **frame** doivent être utilisées dans le fichier html chargé initialement pour effectuer ce découpage.

La fenêtre initiale est le parent des sous-fenêtres.

Ce parent est représenté par l'objet **top** qui dépend lui-même de l'objet window (Voir DOM).

La propriété frames correspond aux balises html frame.

#### Exemple

Nous découpons en trois zones une fenêtre afin d'y afficher des contenus suivant l'interaction de l'utilisateur sur le menu contenu dans la frame de gauche.

Le DOM de ce document est le suivant :

Тор

- MenuGauche
- contenuCentre
- ➤ menuBas

	Code (page de départ)
1	<html></html>
2	<head></head>
3	<frameset rows="90%,10%"></frameset>
4	<frameset cols="20%,80%"></frameset>
5	<frame name="menuGauche" src="menuGauche.html"/>
6	<frame name="contenuCentre" src="contenuCentre.html"/>
7	
8	<frame name="menuBas" src="menuBas.html"/>

- 9 </FRAMESET>
- 10 </HEAD>
- 11 **<BODY>**
- 12 </BODY>
- 13 </HTML>

	Code (menu gauche -> menuGauche.html)			
1	<hr/> HTML>			
2	<head></head>			
3				
4	<body></body>			
5	<u>haut droit</u> 			
6	<input type="BUTTON&lt;/td"/>			
7	onclick="window.top.frames[1].location='http://www.google.fr'"			
8	VALUE="Google !">			
9	 OU 			
10	<input type="BUTTON&lt;/td"/>			
11	onclick="top.frames[1].location='http://www.altavista.fr'" VALUE="Altavista !">			
12	 OU 			
13	<input type="BUTTON&lt;/td"/>			
14	onclick="top.contenuCentre.location='http://www.yahoo.fr'" VALUE="Yahoo !">			
15	  <u>bas droit</u> 			
16	<input type="BUTTON&lt;/td"/>			
17	onclick="top.frames[2].contenuBasCentre.location='http://www.lycos.fr'"			
18	VALUE="Lycos !">			
19	 OU 			
20	<input type="BUTTON&lt;/td"/>			
21	onclick="top.menuBas.contenuBasCentre.location='http://www.netscape.fr'"			
22	VALUE="Netscape !">			
23				
24				

Lorsque vous cliquez sur un des boutons du menu de gauche, la page d'accueil du moteur de recherche choisi s'affiche dans la frame de droite (contenuCentre).

Captures écran

×	lozilla {	Build	ID: 20	02072204}	$\odot$
ΈΙ	e <u>E</u> dit	<u>∨</u> iev	v <u>G</u> o	<u>B</u> ookmarks	<u>T</u> ools
▶:::::::	:: ▶::::::::				
	ioogle ! J		_	0	7
	Altavista !		a	itavis	sta
	J	- 1			
Υ	ahoo!				
<u> </u>			4	1	
P			<u> </u>		
	5750	/a	~		
225					

#### Remarque

Les propriétés, méthodes et objet rattachés au tableau frames sont les mêmes que pour window. En effet, la frame est elle-même une fenêtre. Il suffit donc de tout préfixer par le nom de cette frame pour atteindre les éléments de cette frame.

#### Remarque

Une frame peut elle-même à nouveau être décomposée en frame. Le schéma est alors un peu plus complexe. Par contre la méthode d'accès est la même.

#### Exemple

Reprenons l'exemple précédent en ajoutant un découpage en deux dans la frame du bas.

Le DOM de ce document est le suivant :

Тор

- menuGauche
- contenuCentre
- ➤ menuBas
  - menuBasGauche
  - contenuBasCentre

Ou graphiquement



	Code (page de départ)
1	<html></html>
2	<head></head>
3	<frameset rows="90%,10%"></frameset>
4	<frameset cols="20%,80%"></frameset>
5	<pre><frame name="menuGauche" src="menuGauche.html"/></pre>
6	<frame name="contenuCentre" src="contenuCentre.html"/>
7	
8	<frame name="menuBas" src="menuBas.html"/>
9	
10	
11	<body></body>
12	
13	

	Code (frame menu gauche -> menuGauche.html)				
1	<hr/> HTML>				
2	<head></head>				
3					
4	<body></body>				
5	<u>haut droit</u> 				
6	<input type="BUTTON&lt;/td"/>				
7	onclick="window.top.frames[1].location='http://www.google.fr'"				
8	VALUE="Google !">				
9	 OU 				

- 10 <INPUT TYPE=BUTTON
- 11 onclick="top.frames[1].location='http://www.altavista.fr" VALUE="Altavista !">
- 12 <BR>OU<BR>
- 13 <INPUT TYPE=BUTTON
- 14 onclick="top.contenuCentre.location='http://www.yahoo.fr'" VALUE="Yahoo !">
- 15 <BR><BR><U>bas droit</U><BR>
- 16 <INPUT TYPE=BUTTON
- 17 onclick="top.frames[2].contenuBasCentre.location='http://www.lycos.fr'"
- 18 VALUE="Lycos !">
- 19 **<BR>OU<BR>**
- 20 <INPUT TYPE=BUTTON
- 21 onclick="top.menuBas.contenuBasCentre.location='http://www.netscape.fr"
- 22 VALUE="Netscape !">
- 23 </BODY>
- 24 </HTML>

Code	(page frame	bas -> menuBas.html)	
------	-------------	----------------------	--

>
>

	Code (page frame base droite -> contenuBasCentre html)
1	
1	
2	
3	
4	<body></body>
5	<u>haut droit</u> 
6	<input type="BUTTON&lt;/td"/>
7	onclick="window.top.frames[1].location='http://www.google.fr'" VALUE="Google
8	!">
9	 OU 
10	<input type="BUTTON&lt;/td"/>
11	onclick="top.frames[1].location='http://www.altavista.fr" VALUE="Altavista !">
12	 OU 
13	<input type="BUTTON&lt;/td"/>
14	onclick="top contenuCentre location="http://www.vahoo.fr"" VALUE="Yahoo.l">
15	<pre>  <i l="">has droit </i></pre>
16	
17	onclick="top frames[2] contenuBasCentre location="http://www.lucos.fr"
10	
10	
19	
20	
21	onclick="top.menuBas.contenuBasCentre.location="http://www.netscape.fr"
22	VALUE="Netscape !">

23	
24	

Les boutons sous le texte « haut droit » permettrent l'affichage dans la frame située à ce niveau tandis que les bouton sous le texte bas effectuent l'affichage dans la frame bas droite.



#### 5. history

Cette propriété est aussi un tableau. Il contient l'historique de navigation associé à une fenêtre. C'est-à-dire les contenus de l'objet history. Ces informations ne sont accessibles qu'en lecture. Seuls les sites accédés sont présents dans cette liste. Un site qui n'a pu être atteint n'y est donc pas.

Pour obtenir les informations liées à cet objet, hormis length, il est

nécessaire de posséder des privilèges qui par défaut ne vont sont pas donnés.

Pour les obtenir, il faut utiliser les propriétés de l'objet **navigator**. Dans le cas de l'historique, il faut posséder le privilège <u>UniversalBrowserRead</u>.

#### Exemple

Je viens de naviguer sur plusieurs sites et je désire obtenir la liste. Je suis tenter de faire un petit programme comme suit :

	Code Version 1			
1	<html><head></head></html>			
2	<pre><script language="javascript"></pre></th></tr><tr><th>3</th><th>function go()</th></tr><tr><th>4</th><th>{</th></tr><tr><th>5</th><th>// Affichage de la liste des sites visites</th></tr><tr><th>6</th><th>// Affichage du nombre de sites contenus dans ma liste</th></tr><tr><th>7</th><th>document.write("window.history.length = " + window.history.length+ " ");</th></tr><tr><th>8</th><th>for(i=0;i<window.history.length;i++)</th></tr><tr><th>9</th><th></th></tr><tr><th>10</th><th colspan=4>document.write(history[i] + " ");</th></tr><tr><th>11</th><th>}</th></tr><tr><th>12</th><th>}</th></tr><tr><th>13</th><th></script></pre>			
14	<body></body>			
15	Pour obtenir la liste			
16	<input onclick="go()" type="BUTTON" value="Cliquez histoire de"/>			
17				

Explications

La fonction go est activée lorsque l'utilisateur clique sur le bouton cidessous.

Cette fonction récupère le nombre de lignes contenues dans l'objet history (Ligne 8) et les affiche via une boucle (Ligne 8 à 11).

#### Captures écran





	)	_		Ja	vaScri	pt Console		
100	<u>F</u> ile	<u>E</u> dit	<u>V</u> iew	<u>T</u> ools	<u>H</u> elp			
¥ uu	All	Errors	Warning	is Mess	ages	Clear		
1000								Evaluate
1		Error:	uncaug	ht excep	tion: Pe	rmission denied to call i	method Histo	ry.item

Mais n'ayant pas le privilège de lecture, une erreur apparaît au niveau de la console javascript.

Le source suivant permet d'obtenir le bon privilège et donc d'afficher la liste des sites visités.

1 <	<html><head></head></html>
2 <	<script language="javascript"></td></tr><tr><th>3 f</th><td>function go()</td></tr><tr><th>4 {</th><th>{</th></tr><tr><th>5 //</th><td>/ Affichage du nombre d'element dans la liste</td></tr><tr><th>6 d</th><td><pre>document.write("window.history.length = " + window.history.length+ " ");</pre></td></tr><tr><th>7 //</th><td>/ Modification des privileges sur l'objet navigator</td></tr><tr><th>8 n</th><td>netscape.security.PrivilegeManager.enablePrivilege("UniversalBrowserRead");</td></tr><tr><th>9 //</th><td>/ Affichage de la liste des sites visites</td></tr><tr><th>10 <b>f</b></th><td>for(i=0;i<window.history.length;i++)</td></tr><tr><th>11 {</th><td></td></tr><tr><th>12</th><td>document.write(history[i] + " ");</td></tr><tr><th>13 }</th><td></td></tr><tr><th>14 //</th><td>/ Ouverture d'une nouvelle fenetre</td></tr><tr><th>15 <u>f</u></th><td><sup>2</sup> = window.open("","","","");</td></tr><tr><th>16 💋</th><td>/ Affichage dans la nouvelle fenetre du 3eme site visite (0,1,2)</td></tr><tr><th>17 <u>f</u></th><td>2.document.location=history[2];</td></tr><tr><th>18 }</th><td></td></tr><tr><th>19 <</th><td></script>
20 <	<body></body>
21 <	<input onclick="go()" type="BUTTON" value="Cliquez histoire de"/>
22 <	

Explications

La ligne 8 permet d'activer le bon droit. Une fenêtre est ouverte pour demander à l'internaute si il accepte de donner ce privilège au programme chargé.

Ajout de l'ouverture d'une fenêtre contenant le 3<sup>ème</sup> site visités (la liste commence à 0).

Captures écran	
Internet Security O A script from "file://" has requested enhanced privileges. You should grant these privileges only if you are comfortable downloading and executing a program from this source. Do you wish to allow these privileges?	
Remember this decision	
Yes No	



#### 6. length

Cette propriété permet d'obtenir le nombre d'éléments existant pour l'objet ciblé.

Dans le cas de l'objet frame vu précédemment, cette propriété indique le nombre de découpages effectués dans la fenêtre cible. Si, il n'y a pas de

découpage, la valeur renvoyée est alors 0.

# Exemple

Reprise de l'exemple vu au point 3. Le DOM de ce document est le suivant :

Тор

- MenuGauche
- ➢ contenuCentre
- ➤ menuBas

	Code (page de départ)
1	<html></html>
2	<head></head>
3	<frameset rows="90%,10%"></frameset>
4	<frameset cols="20%,80%"></frameset>
5	<pre><frame name="menuGauche" src="menuGauche.html"/></pre>
6	<frame name="contenuCentre" src="contenuCentre.html"/>
7	
8	<frame name="menuBas" src="menuBas.html"/>
9	
10	
11	<body></body>
12	
13	

	Code (menu bas -> menuBas.html)	
1	<head></head>	
2	<frameset cols="20%,80%"></frameset>	
3	<frame name="menuBasGauche" src="menuBasGauche.html"/>	
4	<frame name="contenuBasCentre" src="contenuBasCentre.html"/>	
5		
6		

Code (menu bas gauche -> menuBasGauche.html)	
1	<html> <head> </head></html>
2	<body></body>
3	<script language="javascript"></script>

7	<pre>document.write(" window.top.contenuCentre.length = " +</pre>
8	window.top.contenuCentre.length);
9	
10	
11	

La frame menuBasGauche contient l'affichage du nombre de découpage effectué dans chaque fenêtre.

La fenêtre de départ (top) est découpée en 3 zones.

La fenêtre référencée par menuBas en 2 zones.

La fenêtre référencée par contenuCentre n'est pas redécoupée (0).

#### Captures écran

Netscape !

window.top.length = 3 window.top.menuBas.length = 2 window.top.contenuCentre.length = 0

#### <u>7. name</u>

Cette propriété contient le nom de la fenêtre. Ceci est vrai uniquement si la ce nom a été renseigné.

Pour le renseigner il existe plusieurs possibilités.

- Le renseigner directement dans la page html via la ligne : window.name="nom de la fenêtre";
- > Le renseigner au moment de l'ouverture d'une fenêtre
  - ✓ Lors du clic sur un lien hypertexte en utilisant l'option target
  - Lors de l'ouverture en utilisant la méthode open et en y insérant le nom

Exemple

Code (page de départ)

- 1 <HTML><HEAD>
- 2 <script language= javascript>

3	function ouvrir()
4	{
5	window.open("name3.html","fenetre3","","");
6	}
7	
8	
9	<body></body>
10	<script language="javascript"></td></tr><tr><td>11</td><td>document.write("window.name avant = " + window.name);</td></tr><tr><td>12</td><td>window.name="fenetre1";</td></tr><tr><td>13</td><td><pre>document.write(" window.name apres = " + window.name);</pre></td></tr><tr><td>14</td><td></script>
15	 
16	<a href="name2.html" onclick="ouvrir()" target="fenetre2">Cliquez</a>
17	et j'ouvre 2 fenetres
18	

Les lignes 3 à 6 permettent de créer une fonction nommée **ouvrir** qui lorsqu'elle est appelée ouvre une fenêtre référencée *fenetre3*. La ligne 11 permet d'écrire dans la fenêtre de départ le texte indiqué ainsi que le nom de la fenêtre active. A ce moment de l'exécution la fenêtre ne possède pas de nom.

La ligne 12 affecte à cette fenêtre sans nom le nom *fenetre1*. La ligne 13 effectue le même travail que la ligne 11. Par contre à ce moment de l'exécution, la fenêtre possède un nom (*fenetre1*). La ligne 16-17 permet d'insérer dans la page html, un lien hypertexte qui va, lorsque l'utilisateur clique dessus, ouvrir deux nouvelles fenêtres.

- L'une au moyen d'un code javascript (onClick) qui fait appel à la fonction *ouvrir*.
- ✓ L'autre simplement au moyen d'un code html (target).

Les fichiers qui seront ouverts dans les deux nouvelles fenêtres contiennent le même code :

	Code des nouvelles fenêtres (name2.html et name3.html)
1	<html><head></head></html>
2	<body></body>
3	<script language="javascript"></td></tr><tr><td>4</td><td>document.write("window.name = " + window.name);</td></tr><tr><td>5</td><td></script>
6	
7	

#### Captures écran

> Avant de cliquer sur le lien hypertexte



> Après le clic sur le lien hypertexte



#### 8. opener

Cet objet renvoie, si il en possède un, le nom de la fenêtre parent en utilisant la propriété name.

Syntaxe nom = window.opener.name ;

#### Exemple

Modifions le code de la page html name2.html vu précédemment. On y ajoute l'affichage du nom de la fenêtre parent.

#### Code

- 1 <HTML><HEAD></HEAD><BODY>
- 2 <script language=javascript>
- 3 document.write("window.name = " + window.name + "<BR>");
- 4 document.write("window.opener.name = " + window.opener.name);
- 5 </script>
- 6 </BODY></HTML>

#### Captures écran

# window.name apres = fenetre1 Cliquez et j'ouvre 2 fenetres window.name = fenetre2

window.opener.name = fenetre1

#### 9. parent

Cet objet permet d'obtenir le nom de la frame d'origine (celle dans lequel est effectué le découpage) en l'associant avec la propriété name. Ce nom est celui associé à l'option name de la balise frame lors du découpage (<frame src="adresse" name="nom">).

#### Remarque

Cette propriété est aussi utilisée lors de l'ouverture de plusieurs fenêtres à l'écran. L'une est forcément le parent des autres.

Syntaxe nom = window.parent.name;

#### Exemple

Je modifie le fichier menBasGauche.html vu dans les paragraphes 4 (frames) et 6 (length) notamment en ajoutant l'affichage du nom du parent de la frame contenant ce fichier html.

	Code de la page menuBasGauche.html
1	<html> <head> </head></html>
2	<script language="javascript"></script>

3 document.write("window.top.length = " + window.top.length);

- 4 document.write("<br>window.top.menuBas.length = " +
- 5 window.top.menuBas.length);
- 6 document.write("<br>window.top.contenuCentre.length = " +
- 7 window.top.contenuCentre.length);
- 8 document.write("<br>window.parent.name = " + window.parent.name);
- 9 </script>
- 10 </BODY></HTML>

La ligne 8 permet d'afficher le nom de la frame parent de celle qui contient ce code html.

#### Captures écran

 $\mathbf{v}\mathbf{v}$ Netscape !

window.top.length = 3 window.top.menuBas.length = 2 window.top.contenuCentre.length = 0 window.parent.name = menuBas

#### <u>10. screen</u>

Screen est un objet de window qui permet d'obtenir les informations en rapport avec l'affichage X (graphique). Ses propriétés ne sont accessibles qu'en <u>lecture</u>.

#### Attention

Cela concerne les propriétés de votre écran et non de la fenêtre du navigateur.

#### Exemple

Le programme suivant permet de récupérer les informations provenant de cet objet.

	Code
1	<html><head></head><body></body></html>

2	<script language="javascript"></th></tr><tr><td>3</td><td>var i=0;</td></tr><tr><td>4</td><td>for(var prop in window.screen)</td></tr><tr><td>5</td><td>{</td></tr><tr><td>6</td><td>document.write(prop + "=" + window.screen[prop] + " ");</td></tr><tr><td>7</td><td>}</td></tr><tr><td>8</td><td></script>
9	

Les lignes 4 à 7 permettent de récupérer, au moyen de la boucle for, les propriétés de l'objet screen.

#### Capture écran

top=0
left=0
width=1152
height=864
pixelDepth=16
colorDepth=16
availWidth=1152
availHeight=864
availLeft=0
availTop=0

#### Attention

Toutes les propriétés ne sont pas disponibles sur tous les navigateurs. Celles présentées ici proviennent d'une utilisation de netscape. Seules les propriétés suivantes sont aussi disponibles sous IE :

- > width contient la taille en pixels de la largeur de l'écran
- height contient la taille en pixels de la hauteur de l'écran
- > colorDepth correspond au « nombre » de couleur par pixels
- availWidth correspond à la largeur effective de l'écran (moins les bandes d'états)
- availHeight correspond à la largeur effective de l'écran (moins les bandes d'états)

Remarque

Sous linux, qu'il y ait ou non les barres d'état, la taille maximale de l'écran est prise en compte. width et availWidth possèdent la même valeur (même conséquence pour height et availHeight, top et availTop ...).

Capture écran sur MacX (barre d'état en bas et en haut)

	🧉 Capture
3	000
	▶:000000
	top=0
1	left=0
2	width=1280
	height=854
	pixelDepth=32
	colorDepth=32
2	availWidth=1280
	availHeight=798
	availLeft=0
6	availTop=22

#### <u>11. self</u>

Quelque soit le nom de la fenêtre, vous pouvez utiliser pour la nommer l'objet self qui par défaut représente cette dernière.

#### Exemple

Dans l'exemple du paragraphe 4 (frames) il est ainsi possible de remplacer window.top.menuBas.length par self.top.menuBas.length.

Exemple

Code	
1	<html></html>
2	<head></head>
3	<script language="javascript"></script>

8	self.document.bgColor="yellow";
9	}
10	
11	
12	<body></body>
13	<form name="formulaire"></form>
14	<input name="B1&lt;/td" type="BUTTON"/>
15	onClick='ouvrir1();' VALUE="Ouvrir Google">
16	
17	
18	

La ligne 8 permet, lorsque l'utilisateur clique sur le bouton défini ligne 14 de modifier la couleur de fond de la fenêtre qui contient ce code. Ce code est identique à window.document.bgColor="yellow";

#### Captures écran

> Avant de cliquer sur le bouton



Une fois le clic effectué



#### 12. status

status permet d'afficher des informations dans la barre de statut (barre en bas) de votre navigateur.

Voir aussi defaultStatus paragraphe 3.

#### Exemple

	Code
1	<html></html>
2	<head></head>
3	
4	<body></body>
5	<script language="javascript"></td></tr><tr><td>6</td><td>window.defaultStatus="> Jacquenod Frédéric";</td></tr><tr><td>7</td><td></script>
8	<input type=button onMouseOver='window.status="coucou"</td>
9	value=ici>
10	
11	

Explications

Tant que la propriété status n'est pas renseignée, si la souris passe sur le bouton défini ligne 8, le message « coucou » se substitue au message par defaut «  $\rightarrow$  Jacquenod Frédéric ».

Une fois la propriété status renseignée, le passage sur le bouton de la souris ne provoque plus rien, le message est figé.

#### Captures écran

Avant le passage de la souris sur le bouton « ici »

• *******	<b>)</b> Back	Ŧ	Forward	Ŧ	3. Reload	Stop	4	•
	ici							
.38	6 FM		2 🕅	a Da	> 1	scauenor	1 Erádi	áric

> Après le passage de la souris sur le bouton « ici »

F	Bac	 k	For	ward	Ŧ	3. Reload	Stop	4	•
	ici								
-	¥ -	100 J	<b>%</b>		æ	coud	:ou		

#### <u>13. top</u>

Cette propriété est utilisée lors de la gestion des cadres (frames) ou de l'ouverture de multiples fenêtres.

Elle référence le cadre initial dont la fenêtre ou le sous-cadre est issu. Top et parent sont identiques seulement et seulement si il n'y a que deux niveaux de cadres ou de fenêtres (père -> fils).

Par contre, si il y a un « grand-père » (gp  $\rightarrow$  père  $\rightarrow$  fils), parent utilisé au niveau du cadre ou fenêtre « fils » (3<sup>ème</sup> niveau), renvoie le « père » tandis que top renvoie le « grand-père ».

#### Exemple

Reprenons l'exemple du paragraphe 4 (frames).

Je modifie le fichier menuBasGauche.html afin de voir la différence entre top et parent.

	Code
1	<html> <head> </head></html>
2	<body></body>
3	<script language="javascript"></td></tr><tr><td>4</td><td>document.write(top.document.location);</td></tr><tr><td>5</td><td></script>
6	<form></form>
7	<u>Modification de moi-meme</u> 
8	<input <="" b1.value="Fond bleu" name="b1" onclick='&lt;/td&gt;&lt;/tr&gt;&lt;tr&gt;&lt;td&gt;9&lt;/td&gt;&lt;td&gt;parent.menuBasGauche.document.bgColor="blue";&lt;/td&gt;&lt;/tr&gt;&lt;tr&gt;&lt;td&gt;10&lt;/td&gt;&lt;td&gt;parent.menuBasGauche.document.fgColor="green";&lt;/td&gt;&lt;/tr&gt;&lt;tr&gt;&lt;td&gt;11&lt;/td&gt;&lt;td&gt;b1.value="Enfoncé"&lt;/td&gt;&lt;/tr&gt;&lt;tr&gt;&lt;td&gt;12&lt;/td&gt;&lt;td&gt;&lt;/td&gt;&lt;/tr&gt;&lt;tr&gt;&lt;td&gt;13&lt;/td&gt;&lt;td&gt;onBlur=' td="" type="BUTTON" value="Fond bleu"/>
14	>
15	<INPUT TYPE="BUTTON" NAME="b2" VALUE="Fond rouge" onClick='</td>
16	top.menuBas.menuBasGauche.document.bgColor="red";

17	top.menuBas.menuBasGauche.document.fgColor="yellow"
18	b2.value="Enfoncé"
19	
20	onBlur='b2.value="Fond rouge"
21	>
22	
23	<hr/>
24	<u>Modification de la fenetre haut Gauche</u> 
25	<input <="" b3.value="Fond vert" name="b3" onclick='&lt;/td&gt;&lt;/tr&gt;&lt;tr&gt;&lt;td&gt;26&lt;/td&gt;&lt;td&gt;top.menuGauche.document.bgColor="green";&lt;/td&gt;&lt;/tr&gt;&lt;tr&gt;&lt;td&gt;27&lt;/td&gt;&lt;td&gt;top.menuGauche.document.fgColor="blue"&lt;/td&gt;&lt;/tr&gt;&lt;tr&gt;&lt;td&gt;28&lt;/td&gt;&lt;td&gt;b3.value="Enfoncé"&lt;/td&gt;&lt;/tr&gt;&lt;tr&gt;&lt;td&gt;29&lt;/td&gt;&lt;td&gt;&lt;/td&gt;&lt;/tr&gt;&lt;tr&gt;&lt;td&gt;30&lt;/td&gt;&lt;td&gt;onBlur=' td="" type="BUTTON" value="Fond vert"/>
31	>
32	
33	<input <="" b4.value="Fond jaune" name="b4" onclick='&lt;/td&gt;&lt;/tr&gt;&lt;tr&gt;&lt;td&gt;34&lt;/td&gt;&lt;td&gt;parent.parent.menuGauche.document.bgColor="yellow";&lt;/td&gt;&lt;/tr&gt;&lt;tr&gt;&lt;td&gt;35&lt;/td&gt;&lt;td&gt;parent.parent.menuGauche.document.fgColor="black"&lt;/td&gt;&lt;/tr&gt;&lt;tr&gt;&lt;td&gt;36&lt;/td&gt;&lt;td&gt;b4.value="Enfoncé"&lt;/td&gt;&lt;/tr&gt;&lt;tr&gt;&lt;td&gt;37&lt;/td&gt;&lt;td&gt;&lt;/td&gt;&lt;/tr&gt;&lt;tr&gt;&lt;td&gt;38&lt;/td&gt;&lt;td&gt;onBlur=' td="" type="BUTTON" value="Fond jaune"/>
39	>
40	
41	<hr/>
42	<u>Remise à zéro</u> 
43	<input onclick="&lt;/td&gt;&lt;/tr&gt;&lt;tr&gt;&lt;td&gt;44&lt;/td&gt;&lt;td&gt;top.menuGauche.document.bgColor=" td="" type="BUTTON" value="Reset" white";<=""/>
45	top.menuGauche.document.fgColor="black";
46	parent.menuBasGauche.document.bgColor="white";
47	parent.menuBasGauche.document.fgColor="black"
48	
49	
50	

Pour bien comprendre le fonctionnement de cette page HTML il faut bien avoir en tête le schéma de construction de la fenêtre et son découpage.



Il est possible d'utiliser le nom de la frame ou sa numérotation :

top.menusBas.menuBasGauche.document.bgcolor est identique à

top.frames[2].frames[0].document.bgcolor

Lorsque l'utilisateur clique sur les boutons, ceux-ci ont pour effet de modifier la couleur de certaines frames en fonctions de leur description (top. ... parent. ...).

Les lignes 15 à 21 par exemple

```
<INPUT TYPE="BUTTON" NAME="b2" VALUE="Fond rouge" onClick='
top.menuBas.menuBasGauche.document.bgColor="red";
top.menuBas.menuBasGauche.document.fgColor="yellow"
b2.value="Enfoncé"
```

```
onBlur='b2.value="Fond rouge"
```

>

Lorsque le bouton « Fond rouge » est enfoncé (clic), la couleur de fond de la frame menuBasGauche devient rouge (ligne 16), la couleur du texte devient jaune (ligne 17), le bouton voit son intitulé modifié en « Enfoncé » (ligne 18). Si le clic se porte sur un autre bouton, la ligne 20 permet alors de remettre le texte initial sur le bouton « Fond rouge ». Les autres boutons ont un comportement identique.

# Captures écran

<u>bas droit</u> Lycos ! OU Netscape !	
Modification de moi-meme Fond bleu Fond rouge Modification de la fenetre haut Gauche Fond vert Fond jaune Remise à zéro Reset	

Le tableau ci-dessous indique le nom de chaque frame par rapport à l'image ci-dessus.

menuGauche	contenuCentre

menuBasGauche	contenuBasCentre

Je clique sur les boutons « Fond bleu » puis « Fond rouge ». Le code javascript va permettre de changer la couleur de la frame menuBasGauche c'est à dire elle même.



Je clique sur les boutons « Fond vert » puis « Fond jaune » après avoir cliqué sur le bouton « fond rouge ».

Le code javascript va permettre de changer la couleur de la frame menuGauche.



# <u>b. Les méthodes</u>

#### 1. back() et forward()

Les deux méthodes permettent d'obtenir la page précédemment visitée dans la fenêtre indiquée (back()) ou la page suivante (forward()). Les informations utilisées sont celles déjà évoquées avec la propriété history.

D'ailleurs, ces méthodes sont aussi disponibles avec l'objet history sous la forme : window.history.back() et window.history.forward().

#### Exemple

La page suivante montre différentes possibilités d'utilisation de ces méthodes.

	Code		
1	<html><head></head></html>		
2	<script language="javascript"></script>		

8	{
9	window.forward();
10	}
11	
12	
13	<body></body>
14	<b></b>
15	<font color="red">Voir le passé</font> 
16	 
17	<a onclick="retour()">Vers le passé</a> 
18	<a href="javascript:retour();">Vers le passé</a> 
19	<pre><a href="javascript:window.back();">Vers le passé</a> </pre>
20	<form></form>
21	<input name="B1&lt;/td" type="BUTTON"/>
22	onClick="retour();"
23	VALUE="Vers le passé"
24	>
25	
26	 
27	 
28	<font color="green">Voir le futur</font> 
29	 
30	<a onclick="avance()">Vers le futur</a> 
31	<a href="javascript:avance();">Vers le futur</a> 
32	<a href="javascript:window.forward();">Vers le futur</a> 
33	
34	<input name="B2&lt;/td" type="BUTTON"/>
35	onClick='avance();'
36	VALUE="Vers le futur"
37	>
38	
39	

Les deux méthodes sont associées pour l'exemple, à un lien hypertexte directement (ligne 19) ou via une fonction prédéfinie (ligne 18). La ligne 17 montre que l'on peut aussi associer un événement à un texte. Elles sont aussi associées à un bouton de formulaire à travers l'événement onClick.

Pour que ces méthodes fonctionnent, il faut évidemment qu'il y ait eu une navigation, des pages visitées, dans la fenêtre concernée.

Captures écran	
·	



#### 2. blur() et focus()

Ces deux méthodes permettent de rendre inactif (blur) ou actif (focus) une fenêtre. Lorsqu'une fenêtre (objet window ou frame) possède le « focus », les événements sont alors destinés à cette fenêtre. Lorsque ces méthodes sont utilisées sur une fenêtre, il se peut que celle qui perd le focus se retrouve cachée par les autres fenêtres présentes à l'écran. Ces dernières viennent alors en avant.

#### Exemple

Une fenêtre nommée f1 est créée au moment du chargement de la page. Les méthodes blur et focus lui sont associées.

	Code
1	<html></html>
2	<head></head>
3	<script language="javascript"></td></tr><tr><td>4</td><td>function ouvrir()</td></tr><tr><td>5</td><td>{</td></tr><tr><td>6</td><td>f1 = window.open("http://www.google.com","google",</td></tr><tr><td>7</td><td>"height=100,width=100,screenx=50","");</td></tr><tr><td>8</td><td>}</td></tr><tr><td>9</td><td></script>

10	
11	<body onload="ouvrir()"></body>
12	<form name="formulaire"></form>
13	<input name="B1&lt;/td" type="BUTTON"/>
14	onClick='f1.blur();B1.value="Selectionné";B2.value="Focus"
15	VALUE="Blur">
16	 
17	<input name="B2&lt;/td" type="BUTTON"/>
18	onClick='f1.focus();B2.value="Selectionné";B1.value="Blur"
19	VALUE="Focus">
20	   
21	

22 </BODY></HTML>

#### **Explications**

La fonction ouvrir (lignes 4 à 8) est activée au moment du chargement de la page html contenant ce code grâce à l'utilisation de l'événement onLoad inclus dans la balise BODY (ligne 11).

Cette focntion ouvre un fenêtre appelée f1 qui contient la page du site <u>www.google.com</u>. Cette fenêtre a une taille de 100 pixels sur 100 pixels (sans les barres d'état).

Au moment du chargement, c'est cette fenêtre qui possède le focus. Les boutons définis dans la page (B1 et B2) parent, permettent de donner le focus ou de l'enlever à cette fenêtre f1. Ceci a pour effet de l'envoyer en avant ou en arrière par rapport à la fenêtre parent.

#### Captures écran



#### 3. close() et open()

Ces deux méthodes ont déjà été vues. Elles permettent d'ouvrir et de fermer une fenêtre.

L'ouverture d'une fenêtre est possible en incluant un certain nombre de paramètres.

#### Syntaxe

[windowVar=][window].open("URL","windowName",["windowFeatures"])

**windowVar** est la variable qui permet de référencer la fenêtre. C'est elle qui sera ensuite utilisée avec les méthodes et propriétés.

**URL** contient l'adresse de la page à ouvrir dans cette nouvelle fenêtre. **WindowName** contient un identifiant qui sera utilisé avec l'option target des balises. Ce nom ne doit contenir que des caractères alphanumériques et/ou underscore (\_).

**WindowFeatures** contient les options associées à l'ouverture de la fenêtre.

toolbar[=yes|no]|[=1|0]  $\rightarrow$  barre de navigation (back, forward .. .)

location[=yes|no]|[=1|0]  $\rightarrow$  barre contenant l'URL

directories[=yes|no]|[=1|0]  $\rightarrow$  barre des favoris

status[=yes|no]|[=1|0]  $\rightarrow$  barre de status tout en bas

menubar[=yes|no]|[=1|0]  $\rightarrow$  barre de menus tout en haut

scrollbars[=yes|no]|[=1|0]  $\rightarrow$  ascenseurs horizontaux et verticaux

resizable[=yes|no]|[=1|0]  $\rightarrow$  redimensionnement de la fenêtre

width=pixels  $\rightarrow$  largeur de la fenêtre au moment de l'ouverture

height=pixels  $\rightarrow$  hauteur de la fenêtre au moment de l'ouverture

#### Exemple

Cet exemple montre simplement comment ouvrir et fermer une fenêtre tut en modifiant le contenu des boutons qui permettent ces actions.

	Code
1	<html></html>
2	<head></head>
3	<script language="javascript"></td></tr><tr><td>4</td><td>function ouvrir()</td></tr><tr><td>5</td><td>{</td></tr><tr><td>6</td><td>f1 = window.open("http://www.google.com","google",</td></tr><tr><td>7</td><td>"height=100,width=100,screenx=50","");</td></tr><tr><td>8</td><td>}</td></tr><tr><td>9</td><td></script>

10	
11	<body></body>
12	<form name="formulaire"></form>
13	<input name="B1&lt;/td" type="BUTTON"/>
14	onClick='ouvrir();B1.value="Selectionné";B2.value="Close"
15	VALUE="Open">
1§	 
17	<input name="B2&lt;/td" type="BUTTON"/>
18	onClick='f1.close();B2.value="Selectionné";B1.value="Open"
19	VALUE="Close">
20	 
21	 
22	 
23	
24	

Lors de l'ouverture (Ligne 14) ou de la fermeture (Ligne 17) le texte des boutons changent tout en effectuant l'ouverture ou la fermeture de la fenêtre nommée *google* et dont la variable est *f1*.



Il peut être utile lors de la création de fenêtres, de ne pas laisser à l'écran les celles précédemment ouvertes. Pour cela il est possible d'utiliser l'événement onUnLoad en lui associant la méthode close sur une fenêtre déjà ouverte.

#### Exemple

Voici une petite astuce qui permet de tester l'existence des fenêtres ouvertes.

Il peut être intéressant lors de la création de fenêtres, de ne pas laisser à l'écran les fenêtres précédemment ouvertes. Pour cela il est possible d'utiliser l'événement onUnLoad en lui associant la méthode close sur une fenêtre déjà ouverte.

	Code		
1	<html></html>		
2	<head></head>		
3	<script language="javascript"></script>		

34	if(f2_ouverte)
35	f2.close();
36	}
37	
38	
39	
40	<body onload="initialisation()" onunload="fermeture()"></body>
41	<u><b>Selectionnez votre  moteur de recherche</b></u>
42	  
43	<form name="formulaire"></form>
44	<input name="B1&lt;/td" type="BUTTON"/>
45	onClick='ouvrir1();B1.value="Selectionné";B2.value="Ouvrir
46	Altavista"
47	VALUE="Ouvrir Google">
48	 
49	 
50	<input name="B2&lt;/td" type="BUTTON"/>
51	onClick='ouvrir2();B2.value="Selectionné";B1.value="Ouvrir Google"
52	VALUE="Ouvrir Altavista">
53	   
54	
55	

Lors du chargement de la page html, deux variables contenues dans la fonction *initialisation* (Ligne 4 à 8) sont positionnées à la valeur false. Lorsque la fenêtre correspondante est ouverte, la variable prend la valeur true.

Lors du rechargement de la page principale, (reload) ou lors du chargement d'une autre page html dans la page principale, l'événement onUnLoad est activé (Ligne 40) et ferme automatiquement la fenêtre fille ouverte au moyen de la fonction *fermeture*.

Cette fonction teste la valeur des variables initialisée au départ. Celle qui possède la valeur true indique que la fenêtre est ouverte et qu'il faut donc la fermer au moyen de la méthode close (Ligne 32 à 36).

Lors de l'ouverture des fenêtres fille, qui se produit au moment ou l'utilisateur clique sur les boutons de la page principale, les couleurs sont modifiées ainsi que les textes associés aux boutons. Les fonctions ouvrir1 et ouvrir2 sont exécutées grâce à l'événement onClick (Ligne 45 et 51).





#### d. moveBy() et moveTo()

Ces deux méthodes permettent de déplacer la fenêtre à différents endroits de l'écran.

La fenêtre a une position par rapport au point d'origine de l'écran à savoir (x,y) = (0,0) qui est le point haut gauche de l'écran.

Il est donc possible via les deux méthodes, de déplacer la fenêtre par rapport à ce point fixe (moveTo()) ou par rapport à la position courante (moveBy()).

Ces deux méthodes prennent deux arguments qui sont les valeurs numériques x et y exprimées en pixels. moveBy(x,y) permet un déplacement par rapport à la position courante de la fenêtre de x pixels sur l'axe des x (horizontal) et de y pixels sur l'axe des y (vertical). On parle de déplacement relatif.

#### Remarque

x et y peuvent posséder des valeurs négatives. Si vous déplacez la fenêtre vers la gauche, x sera négatif, si vous la déplacez vers le haut, y sera négatif.

**moveTo(x,y)** permet un déplacement de la fenêtre par rapport au point de coordonnées (x,y) = (0,0). C'est le repère absolu de votre écran. On parle de déplacement absolu.

#### **Attention**

L'axe des x est l'axe horizontal il va de gauche à droite. Les valeurs de cet axe sont donc postives.

L'axe des y est vertical, il va de haut en bas. Les valeurs de cet axe sont donc négatives.

#### Attention

Le déplacement de votre fenêtre est limité par les « bords » de votre écran. Si vous donnez une valeur supérieure à la distance entre votre fenêtre et ces « bords », le déplacement n'ira pas au-delà de ces « bords ». De plus, les sommets de votre fenêtre bloquent en position (x,y) = (0,0). Par contre vous pouvez la déplacer en indiquant des coordonnées négatives en x et y ou en coordonnées non définies (Supérieures à la taille maximale de votre écran). En clair le coin hautgauche de la fenêtre, ne peut se déplacer de plus de pixels que n'en possède votre écran moins la taille de votre fenêtre. déplacement max = taille max écran – taille fenêtre.

# Exemple

Cet exemple propose le déplacement de la fenêtre selon les 2 méthodes de x et y pixels.

x et y sont saisis par l'utilisateur. Deux possibilités lui sont proposées. Un mouvement absolu au moyen de la fonction mouvement\_moveTo() (Ligne 16) ou un mouvement relatif au moyen de la fonction mouvement\_moveBy() (Ligne 11).

	Code	
1	<html><head></head></html>	
2	<script language="javascript"></script>	

3	xDepart=window.screenX;		
4	yDepart=window.screenY;		
5	function initialisation()		
6	{ · · · · · · · · · · · · · · · · · · ·		
7	document.formulaire.x.value = window.screenX:		
8	document.formulaire.v.value = window.screenY:		
9	}		
10			
11	function mouvement, moveBv()		
12	r		
12	1 Livindow movoBv(dooumont formulairo x voluo dooumont formulairo x voluo):		
13	window.moveby(document.iormulaire.x.value,document.iormulaire.y.value),		
14	}		
10	for ation means at many Te()		
16	function mouvement_movero()		
17			
18	window.movelo(document.formulaire.x.value,document.formulaire.y.value);		
19	}		
20			
21			
22			
23	<body onload="initialisation()"></body>		
24	<form name="formulaire"></form>		
25	<script language="javascript"></td></tr><tr><td>26</td><td>document.write("<U>Les valeurs de X et Y initiales sont</U> :   ");</td></tr><tr><td>27</td><td>document.write("X : " + xDepart + "   ");</td></tr><tr><td>28</td><td>document.write("Y : " + vDepart + "  ");</td></tr><tr><td>29</td><td>document.write("<U>Les caracteristiques de l'écran sont</U> :   "):</td></tr><tr><td>30</td><td>document.write("width : " + window.screen.width + "&nbsp:&nbsp:&nbsp:"):</td></tr><tr><td>31</td><td>document.write("height : " + window.screen.height + " "):</td></tr><tr><td>32</td><td></script>		
33	 		
34	<pre></pre>		
35	<pre> </pre>		
36			
37	Déplacement en X : ZINDUT TVDE-TEXT SIZE-6 NAME-ys		
20			
20	Déplement on V : ZINDUT TYDE-TEXT SIZE-6, NAME-vs		
39			
40			
41	<pre>SBR&gt;</pre>		
42			
43	<pre> </pre>		
44			
45	<input name="move1&lt;/td" type="BUTTON"/>		
46	onClick='moveBy(document.formulaire.x.value,document.formulaire.y.value);'		
47	VALUE=moveBy>		
48			
49	<input name="move2&lt;/td" type="BUTTON"/>		
50	onClick='mouvement_moveTo(document.formulaire.x.value,document.formulair		
51	e.y.value)'		
52	VALUE=moveTo>		

#### 53 </FORM> 54 </BODY></HTML>

#### Explications

Au moment du chargement de la page html, les valeurs initiales correspondantes aux valeurs de l'écran sont insérées dans les champs de saisis de x et y. D'autres informations sont aussi affichées (Ligne 26 à 31).

Les mouvements sont associés à l'événement onClick (Ligne 46 et 50) grâce aux fonctions définies dans la partie en-tête de la page html.

Captures écran

			💌 Netscape 🔍 🔍
			<u>Eile Edit V</u> iew <u>G</u> o <u>B</u> ookmarks <u>T</u> ools <u>W</u> indow <u>F</u>
	Netscape     File Edit View Co Bookmarke Teels Window	umme I	Les un laure de V et Vinitieles sout :
		линна (	Les valeurs de X et 1 initiales sont :
_	Les valeurs de X et Y initiales sont :		X:31 Y:39
L	X:31 Y:39		Les caracteristiques de l'écran sont :
	Les caracteristiques de l'écran sont :		width : 1152 height : 864
К	width : 1152 height : 864		Entrez une valeur Numerique
	Entrez une valeur Numerique		Déplacement en X : 1152
1.2	Déplacement en X : 31		
LIN	Déplacement en Y : 39		Choisissez le type de déplacement :
	Choisissez le type de déplacement :		, moveBy moveTo
	moveBy moveTo		
'N	etscape 💿 🗢		
2	<u>ào B</u> ookmarks <u>T</u> ools <u>W</u> indow <u>H</u>		
_			
e X et Y initiales sont :			
sti	ques de l'écran sont :		
h	height : 864		
aleur Numerique			
en X : -115 en Y : 0			
ty	type de déplacement :		
mo	veTo		

#### Attention

Si vous créez une fenêtre à partir de la fenêtre initiale (window.open ...), vous pouvez être tentés de la déplacer avec les mêmes fonctions que vues ci-dessus. Hélas vous ne pourrez pas car, il faut posséder les privilèges UniversalBrowser-Privilege. Sans ces privilèges, vous aurez l'erreur suivante dans la console javascrit :

۲	JavaScript Console	
<u> </u>	dit <u>V</u> iew <u>T</u> ools <u>H</u> elp	
All Err	rors Warnings Messages   Clear	
		Evaluate
X	Error: uncaught exception: Permission denied to call method Window	/.moveBy
$\mathbf{X}$	Error: uncaught exception: Permission denied to call method Window	/.moveTo

#### <u>e. print()</u>

Cette méthode permet d'imprimer la page en cours. Il suffit d'associer cette méthode à l'événement onClick.

Vous pouvez aussi proposer l'impression d'autres fenêtres dans un système de frame.

Exemple

Reprenons l'exemple de frame utilisé depuis le début.

	Code
1	<html> <head> </head></html>
2	<body></body>
3	<form></form>
4	<u>Impression de moi-meme</u>
5	<input <="" name="b1" td="" type="BUTTON" value="Imp"/>
6	onClick='window.print()'>
7	
8	<u>Impression de mon cousin</u>
9	<input <="" name="b2" td="" type="BUTTON" value="imp"/>
10	onClick='top.menuBas.contenuBasCentre.print()'>
11	
12	
13	

La méthode print est utilisée pour imprimer la fenêtre courante, celle qui contient les boutons sur lesquels l'utilisateur doit appuyer en ligne 5. En ligne 9 c'est l'impression de la fenêtre du bas qui sera imprimée. Ces deux actions sont associées à l'événement onClick.

#### Captures écran

Netscape !	
Impression de moi-meme imprimer Impression de mon cousin imprimer	Partie Contenu bas centre

#### f. resizeBy() et resizeTo()

Ces deux méthodes permettent de redimensionner la fenêtre courante. Elles nécessitent deux arguments x et y. Ces arguments peuvent être des valeurs positives ou négatives.

Le coin haut gauche de votre écran correspond au point de coordonnées (x,y)=(0,0).

resizeBy(x,y) permet de redimensionner la fenêtre en ajoutant (valeur positive) ou en enlevant (valeur négative) x et y pixels à la valeur x et y initiale.

resizeTo(x,y) permet de redimensionner la fenêtre en prenant comme nouvelle valeur de x et y celles indiquées comme argument de la méthode.

#### Remarque

La fenêtre ne peut avoir une valeur x et y inférieure à 100 pixels sur 100 pixels ou plus précisément la taille ne peut être inférieure à la taille de la fenêtre avec uniquement les menus.

#### Remarque

Si vous désirez utiliser ces méthodes sur des fenêtres autres que la fenêtre courante qui contient le code html, vous devez posséder les privilèges adéquats.

# Exemple

	Code
1	<html></html>
2	<head></head>
3	
4	<script language="javascript"></td></tr><tr><td>5</td><td>xDepart = 600;</td></tr><tr><td>6</td><td>yDepart = 600;</td></tr><tr><td>7</td><td>function initialisation()</td></tr><tr><td>8</td><td>{ · · · · · · · · · · · · · · · · · · ·</td></tr><tr><td>9</td><td>document.formulaire.x.value = xDepart;</td></tr><tr><td>10</td><td>document.formulaire.y.value = yDepart;</td></tr><tr><td>11</td><td>}</td></tr><tr><td>12</td><td></td></tr><tr><td>13</td><td>function mouvement_resizeBy()</td></tr><tr><td>14</td><td>{</td></tr><tr><td>15</td><td>window.resizeBy(document.formulaire.x.value,document.formulaire.y.value);</td></tr><tr><td>16</td><td>}</td></tr><tr><td>17</td><td></td></tr><tr><td>18</td><td>function mouvement_resizeTo()</td></tr><tr><td>19</td><td>{</td></tr><tr><td>20</td><td>window.resizeTo(document.formulaire.x.value,document.formulaire.y.value);</td></tr><tr><td>21</td><td>}</td></tr><tr><td>22</td><td></script>
23	
24	
25	<body onload="initialisation()"></body>
26	<form name="formulaire"></form>
27	<script language="javascript"></td></tr><tr><td>28</td><td>document.write("<U>Les valeurs initiales sont</U> :   ");</td></tr><tr><td>29</td><td>document.write("largeur : " + xDepart + "   ");</td></tr><tr><td>30</td><td>document.write("hauteur : " + yDepart + "  ");</td></tr><tr><td>31</td><td>document.write("<U>Les caracteristiques de l'écran sont</U> :   ");</td></tr><tr><td>32</td><td>document.write("width : " + window.screen.width + "   ");</td></tr><tr><td>33</td><td>document.write("height : " + window.screen.height + " ");</td></tr><tr><td>34</td><td></script>
35	 
36	<u>Entrez une valeur Numerique</u>
37	 
38	 
39	Redimensionnement en X : <input name="x" size="6" type="TEXT"/>
40	 

- 41 Redimensionnement en Y : <INPUT TYPE=TEXT SIZE=6 NAME=y>
- 42 **<BR>**
- 43 **<BR>**
- 44 <U>Choisissez le type de Redimensionnement :</U>
- 45 <BR>
- 46 **<BR>**
- 47 <INPUT TYPE=BUTTON NAME=redim1 onClick='
- 48 mouvement\_resizeBy(document.formulaire.x.value,document.formulaire.y.value
- 49 VALUE=resizeBy>
- 50
- 51 <INPUT TYPE=BUTTON NAME=redim2 onClick='
- 52 mouvement\_resizeTo(document.formulaire.x.value,document.formulaire.y.value
- 53 VALUE=resizeTo>
- 54 </FORM>
- 55 </BODY>
- 56 </HTML>

La fonction *initialisation* (Ligne 7) est chargée en même temps que la page html au moyen de l'événement onLoad inséré dans la balise body (Ligne 25). Elle permet d'insérer dans les zones de saisie du formulaire les valeurs en x et y de 600 pixels (bouton Ligne 39 et 41 x et y). Cette initialisation ne redimensionne pas la taille de la fenêtre à son ouverture. Les lignes 28 à 33 permettent d'obtenir les informations sur la taille de la fenêtre courante et sur la taille de l'écran.

Lorsque l'utilisateur saisit les valeurs x et y dans les zones du formulaire, il a le choix entre utiliser la méthode resizeBy et resizeTo au moyen des boutons lignes 48 et 51. Lors du clic sur un de ces boutons, l'événement onClick fait appel aux fonctions *mouvement\_resizeBy* et

mouvement\_resizeTo définies en ligne 13 et 18.

Les valeurs x et y passées en argument sont celles saisies dans les zones texte Ligne 39 et 41.

#### Captures écran

Ecran de la fenêtre courante

Les valeurs initiales sont :		
largeur : 600 hauteur : 600		
Les caracteristiques de l'écran sont :		
width : 1152 height : 864		
Entrez une valeur Numerique		
Redimensionnement en X : 600 Redimensionnement en Y : 600		
Choisissez le type de Redimensionnement :		
resizeBy		

Dans le cas où les valeurs de x et y sont inférieures à 100 la fenêtre a cette aspect.



Cas où vous faites appel à ces méthodes sur une autre fenêtre que celle contenant le code sans posséder les privilèges



<u>c. Les propriétés et les événements</u>

Les propriétés permettent lors d'un événement en rapport avec la fenêtre courante de l'intercepter et d'effectuer une action.

Il existe un grand nombre de propriétés liées à des événements.

Le document proposé par Mozilla les recense.

Ce document est consultable à l'adresse :

http://www.mozilla.org/docs/dom/domref/dom el ref html

La plupart du temps, ces événements et leurs actions vont être inclus dans la balise <body>. Ceci permet de les « charger » à l'initialisation et de les activer.

Ils peuvent néanmoins être associés à d'autres contenus html (a, img ...).

Voici une liste de ces événements.

<u>onblur</u>	Returns the event handling code for the blur event.
onclick	Returns the event handling code for the click event.
ondblclick	Returns the event handling code for the dblclick event.
<u>onfocus</u>	Returns the event handling code for the focus event.
onkeydown	Returns the event handling code for the keydown event.
onkeypress	Returns the event handling code for the keypress event.
<u>onkeyup</u>	Returns the event handling code for the keyup event.
onmousedown	Returns the event handling code for the mousedown event.
onmousemove	Returns the event handling code for the mousemove event.
onmouseout	Returns the event handling code for the mouseout event.
onmouseover	Returns the event handling code for the mouseover event.
onmouseup	Returns the event handling code for the mouseup event.
<u>onresize</u>	Returns the event handling code for the resize event.

Un événement seul ne fait rien, il faut lui associer une action.

#### Remarque

Il est aussi possible de gérer les événements en utilisant le gestionnaire d'événements au moyen de la syntaxe **document.captureEvents**.

# Exemple

Voyons comment il est possible de savoir que l'utilisateur redimensionne la fenêtre courante et d'effectuer une action à ce moment.

»" +		
xDepart + "sur" + yDepart + " ");		
<d[>" + "\.</d[>		
•),		
'x' +		
~		
+ yDiff		
yDiff );		
+ vDiff)∙		
- XDIII),		

alert('Vous avez diminue la largeur de la fenetre de -> ' + xDiff);		
}		
yNouveau = window.innerHeight;		
xNouveau = window.innerWidth;		
}		
<body onresize="mouvement()"></body>		

Dans les calculs, la taille des menus est comptabilisée lors du premier mouvement ce qui pose un petit problème de calcul sur le premier mouvement si celui-ci n'est pas assez important.

La taille de la fenêtre au départ est de 350x250. Cette taille est définie au moyen de la méthode resizeTo (Ligne 7).

Les informations sont affichées dans la fenêtre (Ligne 8 et 10).

La fonction *mouvement* permet d'effectuer des actions dès que la fenêtre courante est redimensionner. Cette fonction est activée dès que l'événement onresize est intercepté (Ligne 48).

A ce moment des messages d'alerte apparaissent à l'écran indiquant l'action effectuée par l'utilisateur, la différence de taille de la fenêtre en hauteur et largeur par rapport aux valeurs précédentes ainsi que la nouvelle taille effective de la fenêtre.

#### Captures écran

Diminution de la hauteur de la fenêtre



Diminution de la hauteur et de la largeur puis augmentation de la hauteur de la fenêtre

