Programmation Web

Sidi Mahmoud Kaber

ouldkaber@yahoo.fr

A.U. 2013/2014

Plan

- HTML… (Rappel)
- JavaScript / VBScript
- Applet Java

© Sidi Mahmoud Kaber

LE DOCUMENT HTML MINIMUM

<HTML>	Ceci est le début d'un document de type HTML.
</HTML>	Ceci est la fin d'un document de type HTML.
<HEAD>	Ceci est le début de la zone d'en-tête (prologue au document proprement dit contenant des informations destinées au browser).
</HEAD>	Ceci est la fin de la zone d'en-tête.
<TITLE>	Ceci est le début du titre de la page.
</TITLE>	Ceci est la fin du titre de la page.
<BODY>	Ceci est le début du document proprement dit.
</ BODY>	Ceci est la fin du document proprement dit.

© Sidi Mahmoud Kaber

LE TEXTE

Gras	[Bold]	<B>...</B> <STRONG>...</STRONG>	Début et fin de zone en gras
Italique	[Italic]	<I>...</I> <EM>...</EM>	Début et fin de zone en italique
Taille de caractère	[Font size]	<FONT SIZE=?>...</FONT>	Début et fin de zone avec cette taille
Couleur de caractère	[Font color]	<FONT COLOR="#$$$$$$"> </FONT>	Début et fin de zone en couleur
A la ligne	[Line break]	 	Aller à la ligne
Commentaires	[Comments]	<!-- *** -->	Ne pas afficher
Centrage	[Center]	<CENTER></CENTER>	Centrer

© Sidi Mahmoud Kaber

CODES DE QUELQUES COULEURS BASIQUES

Bleu	#0000FF	Vert	#00FF00
Blanc	#FFFFFF	Violet	#8000FF
Rouge	#FF0000	Jaune	#FFFF00
Gris clair	#C0C0C0	Noir	#000000

© Sidi Mahmoud Kaber

ALIGNEMENT ...

- Pour aligner du texte, on a utilisé l'attribut ALIGN ou le tag <CENTER>. Il existe une balise permettant d'aligner différents éléments. c'est le tag :

<DIV align=left>...</DIV>
<DIV align=center>...</DIV>
<DIV align=right>...</DIV>

© Sidi Mahmoud Kaber

EXEMPLE

```
<H1>Les mois du printemps</H1>
<UL><LI>avril
<LI>mai
<LI>juin</UL>
<P>
<H3>Les mois d'automne</H3>
<OL><LI>octobre
<LI>novembre
<LI>d&eacute;cembre</OL>
```

© Sidi Mahmoud Kaber

LES TITRES ET LES LISTES

En-têtes	[Heading]	<Hn></Hn> avec n=1 à 6	Afficher une en-tête de niveau n et sauter une ligne
Liste non-ordonnée	[Bullet list]	<UL></UL>	Afficher le texte sous forme d'une liste non-ordonnée.
Liste ordonnée	[Numbered list]	<OL></OL>	Afficher le texte sous forme d'une liste ordonnée.
Elément de liste	[List items]	<LI>	Voici un élément de la liste
Paragraphe	[Paragraph]	<P></P>	Saut de ligne, insérer une ligne vierge et commencer un paragraphe

© Sidi Mahmoud Kaber

LES LIENS

- <A HREF="2.HTM">Aller vers le document 2</A>

Point d'ancrage	<A NAME="***">...</A>	Ceci est une cible
Lien vers une ancre dans la même page	<A HREF="#***">...</A>	Lien vers la cible *** dans la même page
Lien vers une ancre dans une autre page	<A HREF="URL#***">...</A>	Lien vers la cible *** dans une autre page

© Sidi Mahmoud Kaber

LES IMAGES

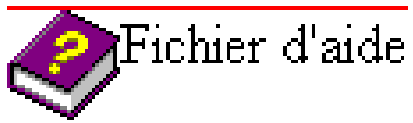
- <IMG SRC="Adresse de l'image">

Texte alternatif alt="****"	Pour les browser n'ayant pas l'option "image" activée
Dimensions width=? height=?	Hauteur et largeur (en pixels)
border=? (en pixels)	Bordure
align=top align=middle align=botton align=left align=right	Alignement

© Sidi Mahmoud Kaber

LES IMAGES

L'attribut Align

<IMG SRC="HELP.gif" align=TOP>Fichier d'aide	
<IMG SRC="HELP.gif" align=CENTER>Fichier d'aide	
<IMG SRC="HELP.gif" align=BOTTOM>Fichier d'aide	

© Sidi Mahmoud Kaber

LES ARRIÈRE-PLANS

```
<BODY BGCOLOR="#$$$$$$">
<BODY BGCOLOR="#000088">
<H1>Bonjour</H1>
</BODY>
```

```
BODY BACKGROUND="PAPER.gif">
<H1>Bonjour</H1>
</BODY>
```

© Sidi Mahmoud Kaber

LES ARRIÈRE-PLANS

Couleur de texte<BODY TEXT="#$$$$$$">

Couleur de lien<BODY LINK="#$$$$$$">

Lien visité<BODY VLINK="#$$$$$$">

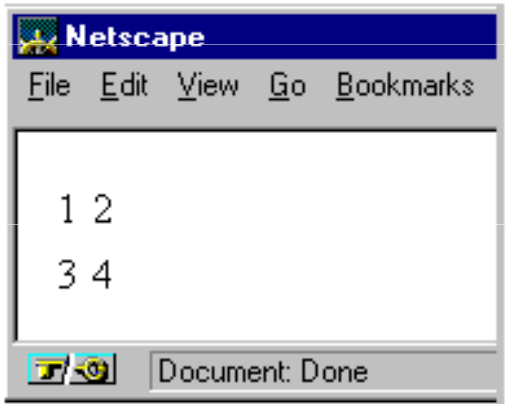
Lien actif<BODY ALINK="#$$$$$$">

© Sidi Mahmoud Kaber

LES TABLEAUX

Définition du tableau	[Table]	<TABLE></TABLE>	Début et fin de tableau
Définition d'une ligne	[Table Row]	<TR></TR>	Début et fin de ligne
Définition d'une cellule	[Table Data]	<TD></TD>	Début et fin de cellule

```
<TABLE>
<TR><TD>1</TD><TD>2</TD></TR>
<TR><TD>3</TD><TD>4</TD></TR>
</TABLE>
```

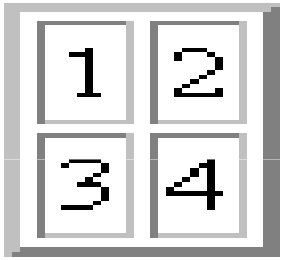


© Sidi Mahmoud Kaber

LES TABLEAUX

Bordure de cadre	[Border]	<TABLE border=?></TABLE>

```
<TABLE border=2>
<TR><TD>1</TD><TD>2</TD></TR>
<TR><TD>3</TD><TD>4</TD></TR>
</TABLE>
```



L'espace entre les cellules ou l'épaisseur des lignes du quadrillage	<TABLE cellspacing=?>
L'enrobage des cellules ou l'espace entre le bord et le contenu	<TABLE cellpadding=?>
La largeur de la table	<TABLE width=?> <TABLE width=%>

© Sidi Mahmoud Kaber

LES CELLULES DES TABLEAUX

Largeur d'une cellule	<TD width=?> en pixels
	<TD width=%> en pourcentage
Fusion de lignes	<TD rowspan=?>
Fusion de colonnes	<TD colspan=?>

```
<CENTER><TABLE width=60% border=1>
<TR>
<TD>cellule1</TD>
<TD>cel. 2</TD>
<TD>3</TD>
</TR>
</TABLE></CENTER>
```

cellule 1	cel 2	3

© Sidi Mahmoud Kaber

LES TABLEAUX

Je souhaite que la première ligne prenne toute la largeur de la ligne. La première cellule doit donc déborder sur 3 cellules horizontales.

cellule 1		
cellule 1	cel 2	3

```
<CENTER><TABLE width=60% border=1>
<TR>
<TD colspan=3>cellule 1</TD>
</TR>
<TR> <TD width=33%>cellule 1</TD> <TD width=33%>cel 2</TD>
<TD width=34%>3</TD> </TR>
</TABLE></CENTER>
```

© Sidi Mahmoud Kaber

LES TABLEAUX

La première colonne prenne toute la hauteur de la colonne. La première cellule doit donc déborder sur 2 cellules verticales.

cellule 1	cel 2	3
	cel 2	3

```
<CENTER><TABLE width=60% border=1>
<TR>
<TD width=33% rowspan=2>cellule 1</TD>
<TD width=33%>cel 2</TD>
<TD width=34%>3</TD>
</TR>
<TR>
<TD width=33%>cel 2</TD>
<TD width=34%>3</TD>
</TR>
</TABLE></CENTER>
```

© Sidi Mahmoud Kaber

LES FRAMES

On utilise les frames pour diviser l'écran en plusieurs fenêtres

Zone avec des fenêtres

- <FRAMESET>Début de zone avec des fenêtres
- </FRAMESET>Fin de zone avec des fenêtres

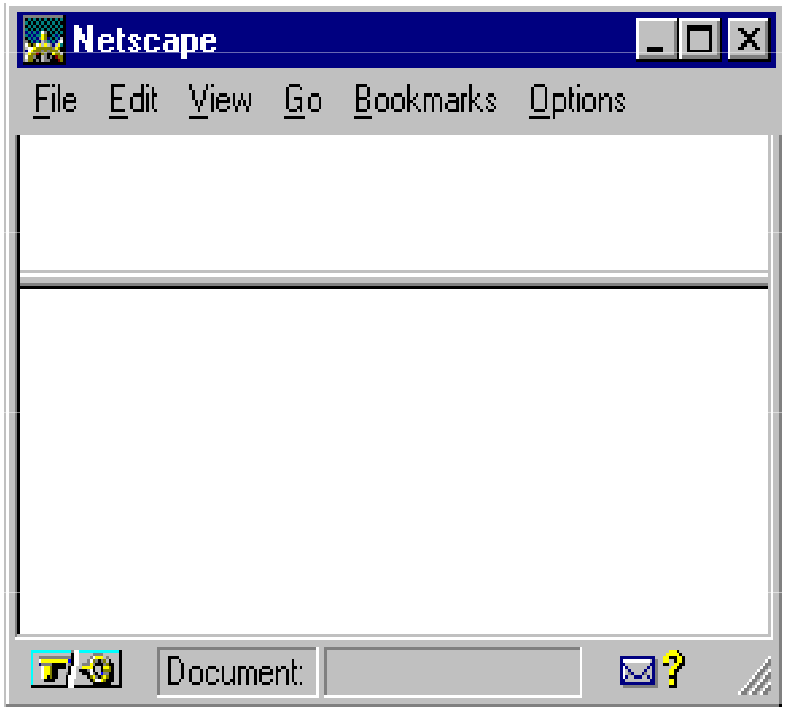
Agencement des fenêtres

- <FRAMESET ROWS="...">Fenêtres horizontales
- <FRAMESET COLS="...">Fenêtres verticales

© Sidi Mahmoud Kaber

LES FRAMES

```
<HTML>
<HEAD></HEAD>
<FRAMESET
   ROWS="30%,70%">
<FRAME>
<FRAME>
</FRAMESET>
</HTML>
```



© Sidi Mahmoud Kaber

Les frames

On utilise les frames pour diviser l'écran en plusieurs fenêtres

Zone avec des fenêtres

- <FRAMESET>Début de zone avec des fenêtres
- </FRAMESET>Fin de zone avec des fenêtres

Agencement des fenêtres

- <FRAMESET ROWS="...">Fenêtres horizontales
- <FRAMESET COLS="...">Fenêtres verticales

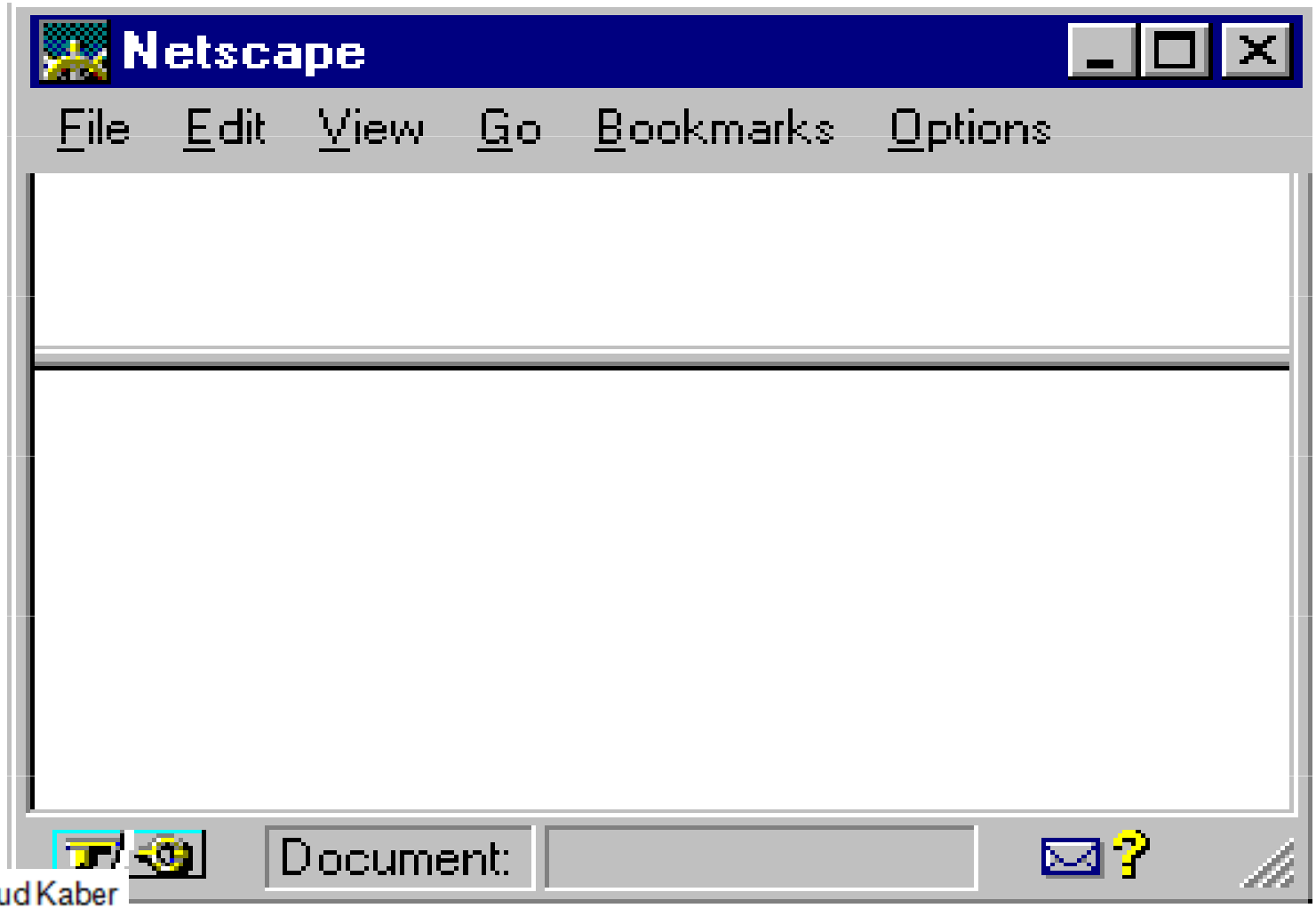
© Sidi Mahmoud Kaber

Frames ... Exemple 1

```
<HTML>
<HEAD></HEAD>
<FRAMESET ROWS="30%,70%">
<FRAME>
<FRAME>
</FRAMESET>
</HTML>
```

© Sidi Mahmoud Kaber

Frames ... Résultat



© Sidi Mahmoud Kaber

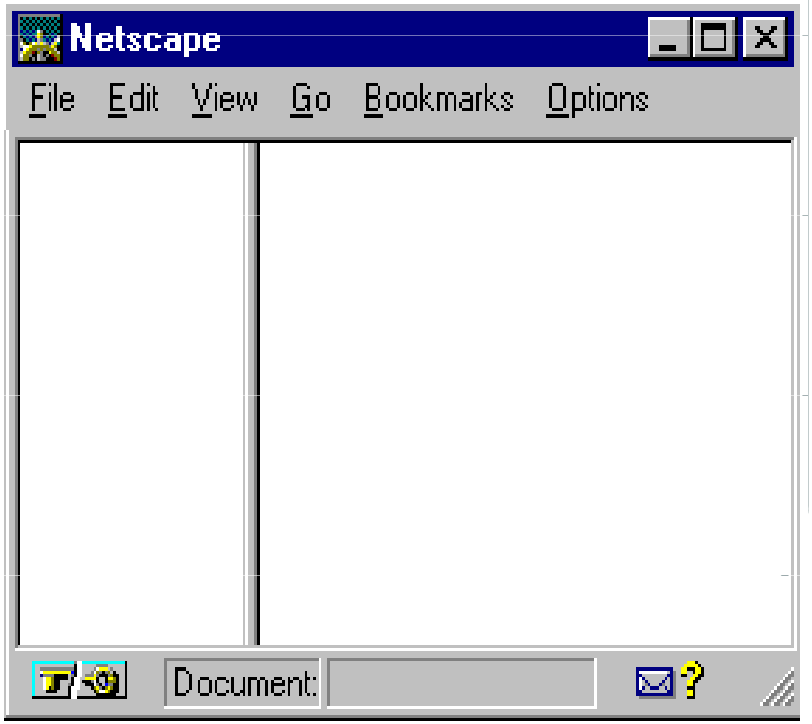
Attention !!!

- <FRAMESET></FRAMESET> remplace <BODY></BODY>
- L'attribut ROWS="hauteur1,hauteur2,...,hauteurN" définit la hauteur des différentes fenêtres en cas de division horizontale.
- La hauteur s'exprime en pixels ou en %. Dans ce cas,
- Le total doit être égal à100%;

© Sidi Mahmoud Kaber

Frames ... agencement vertical

```
<FRAMESET COLS="30%,70%">
<FRAME>
<FRAME>
</FRAMESET>
```



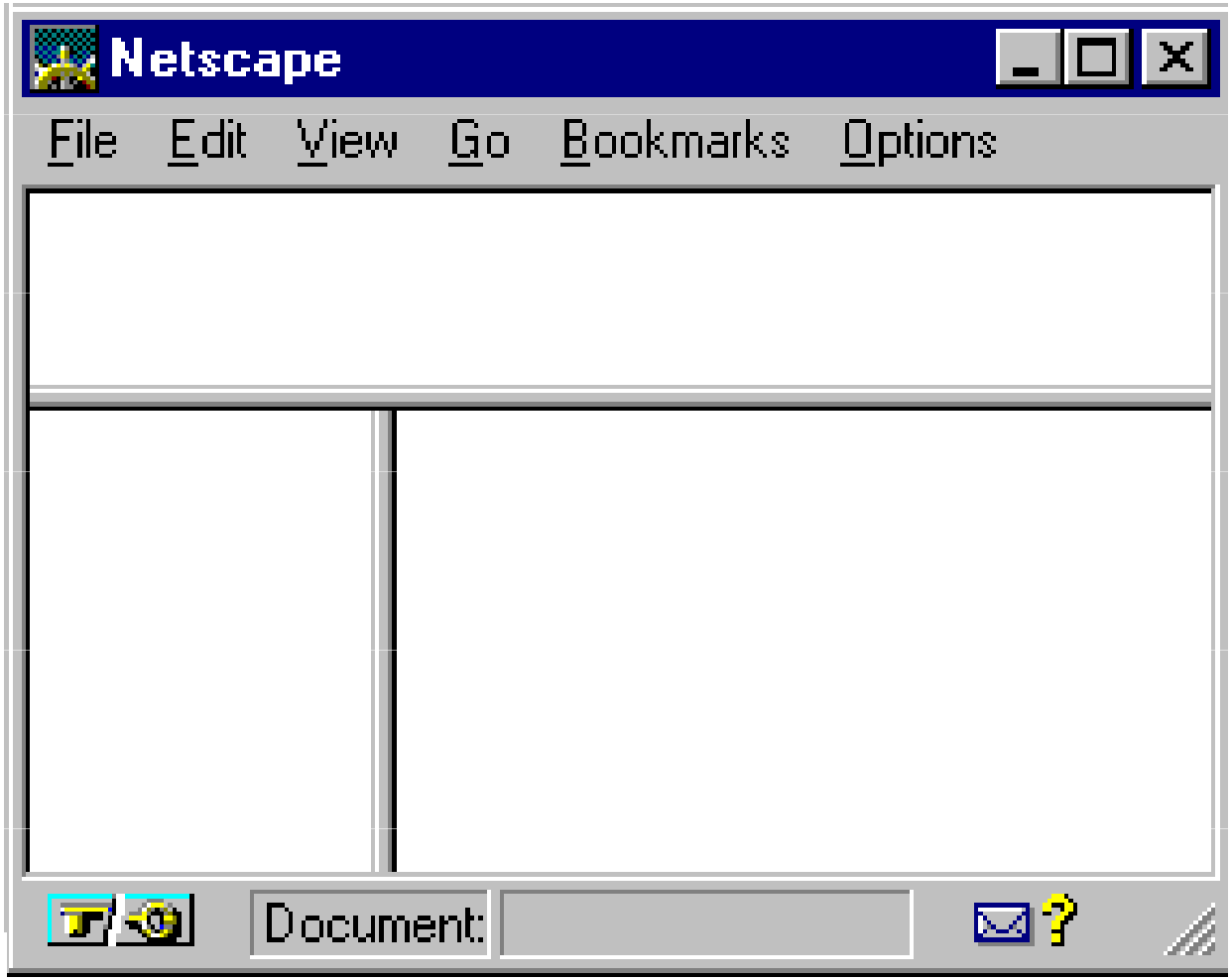
© Sidi Mahmoud Kaber

Frames ... agencement vertical

- L'attribut COLS="largeur1,largeur2,...,largeurN" définit la largeur des différentes fenêtres en cas de division verticale.
- La largeur s'exprime en pixels ou en %.
- Le total doit être égal à100%;

© Sidi Mahmoud Kaber

Frames ... Horizontales et Verticales



© Sidi Mahmoud Kaber

Frames ...Horizontales et Verticales

```
<FRAMESET ROWS="30%,70%">
<FRAME>
<FRAMESET COLS="30%,70%">
<FRAME>
<FRAME>
</FRAMESET>
</FRAMESET>
```

© Sidi Mahmoud Kaber

Frames ... les ascenseurs

Par l'attribut de la balise :

<FRAME SCROLLING="yes/no/auto >

vous pouvez indiquer si la fenêtre doit ou non posséder une barre de défilement.

© Sidi Mahmoud Kaber

Frames ... Border

- Par défaut, les cadres sont séparés par des bordures.
- Il est possible de supprimer ces bordures mais les attributs à utiliser diffèrent selon Netscape ou Internet Explorer.
- Netscape utilise l'attribut "border=0"
- Explorer, les attributs "frameborder=no" et "framespacing=0" (pour enlever l'espace entre les cadres).
- Le tout cohabite sans problème.
- La balise devient alors par exemple :

<FRAMESET COLS="30%,70%" border=0 frameborder=no framespacing=0>

© Sidi Mahmoud Kaber

Frames ... Exemple

```
<HTML>
<HEAD>
<TITLE>Tree Menu</TITLE>
<BASE TARGET="display">
</HEAD>

<FRAMESET Cols="210,430" frameborder="no" framespacing="0">
<FRAME Name="tree"  SRC="tree1.html" NORESIZE
    FRAMEBORDER="YES">
<FRAME Name="mypage"  SRC="initpage.html" NORESIZE
    FRAMEBORDER="YES">
</FRAMESET>
</HTML>
```

© Sidi Mahmoud Kaber

Frames ... Nom

Un autre attribut de cette balise <FRAME> est

NAME="NOM".

Name indique le nom de la fenêtre de telle sorte que cette frame puisse être utilisée comme cible d'un lien hypertexte

© Sidi Mahmoud Kaber

Frames ... Nom

Le fichier de frames devient :

```
<FRAMESET ROWS="30%,70%">
<FRAME SRC="A.htm">
<FRAMESET COLS="30%,70%">
<FRAME SRC="B.htm">
<FRAME SRC="C.htm" NAME="fenetreC">
</FRAMESET>
</FRAMESET>
```

© Sidi Mahmoud Kaber

Frames ... Nom

- Et on met un lien vers A.htm dans le fichier B.htm en désignant comme cible la frame C.

```
<HTML><BODY>
<A HREF="A.htm"
   TARGET="fenetreC"><H1>B</H1></A>
</BODY></HTML>
```

© Sidi Mahmoud Kaber

Frames ...

- L'attribut TARGET peut aussi prendre certaines valeurs prédéfinies :
  - _blank qui indique au browser qu'il doit créer une nouvelle fenêtre afin d'y afficher le fichier. Dans ce cas, vous ouvrer en fait un nouveau browser.
  - _self qui indique que le fichier sera chargé dans la même fenêtre que celle dans laquelle se trouve le lien.
  - _top qui implique l'affichage du fichier sur toute la surface de la fenêtre du browser.

© Sidi Mahmoud Kaber

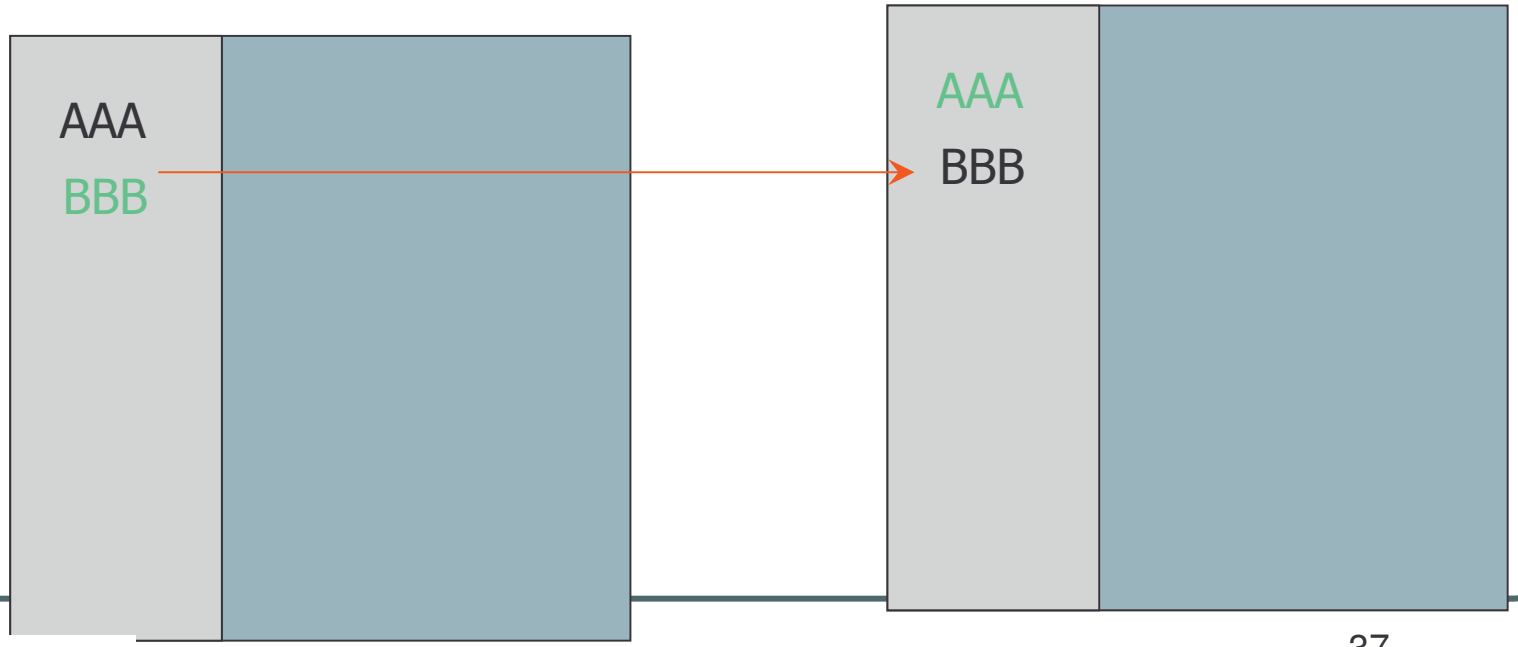
Frames ...

- la balise <NOFRAMES>...</NOFRAMES> est utilisée pour indiquer le texte que doivent afficher les browsers incapables de gérer les frames.
- Il est même indiqué de prévoir une page sans fenêtres pour que ces visiteurs puissent profiter du site.

© Sidi Mahmoud Kaber

Les Tableaux ...

- Possibilite de simulation des frames par l'utilisation des Tableaux



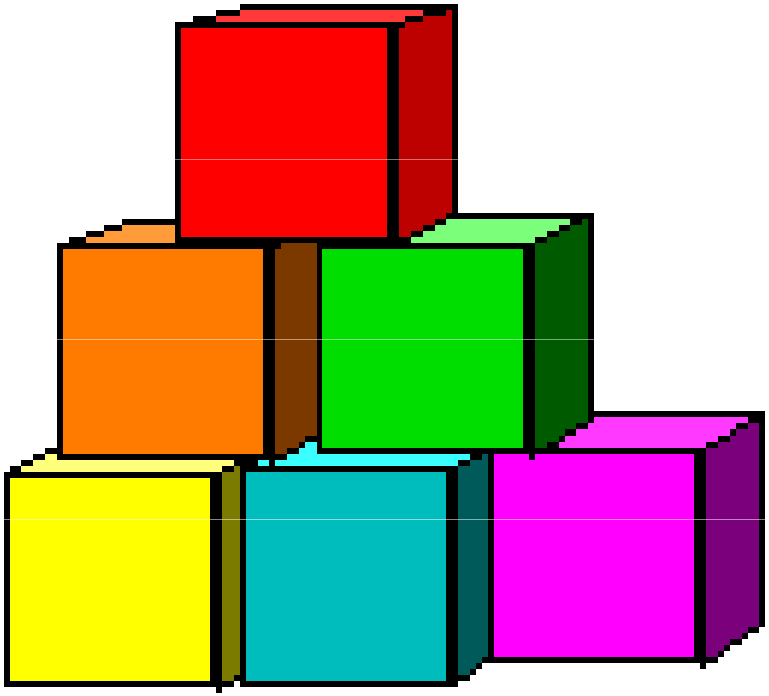
© Sidi Mahmoud Kaber

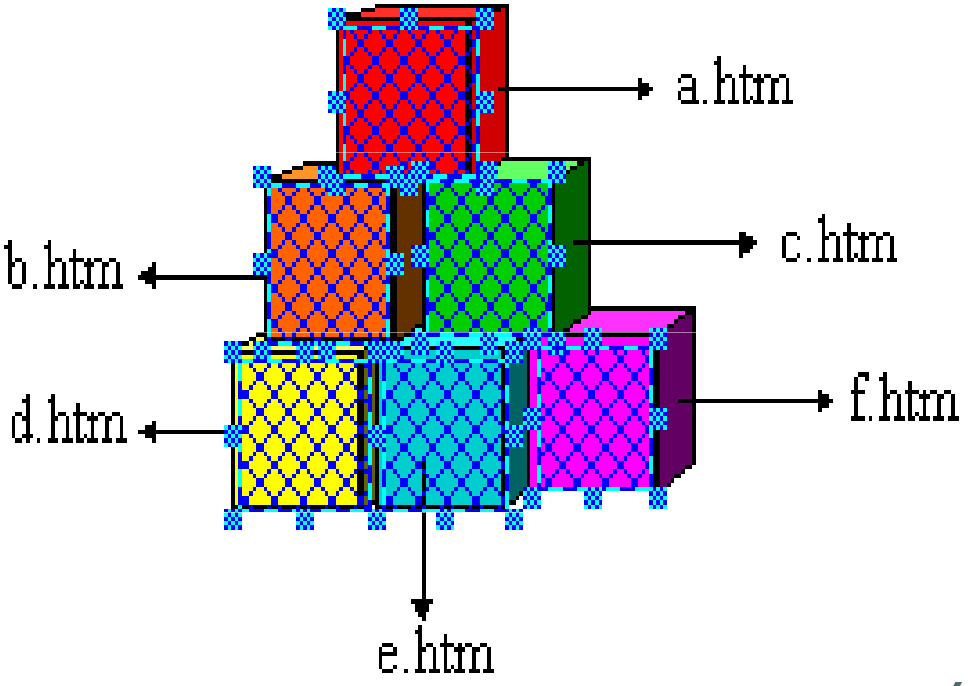
Frames et Tableaux

	AVANTAGES	INCONVENIENTS
FRAMES		
	Simplicité du code.	Référencement délicat sur certains moteurs de recherche (Hotbot).
	Menu toujours présent à l'écran.	Barre de défilement pas très esthétique.
	Riches possibilités en Javascript.	Manipulations plus difficiles pour les débutants.
	Code moins visible pour les débutants.	Non compatible avec certains (rares) navigateurs texte.
TABLEAUX		
	Présentation excellente.	Code entier à répéter pour chaque page à créer.
	Compatible avec tous navigateurs.	Lourdeur du code à télécharger avec certains navigateurs.
		Encore quelques différences entre Netscape et Internet Explorer.

© Sidi Mahmoud Kaber

Les Images Mappées ...





© Sidi Mahmoud Kaber

Les Images Mappées ...

- La balise MAP qui est gérée côté client. Elle simplifie la tâche des développeurs de page HTML.
- La balise **<MAP>** permet de définir la géométrie de l'image; elle contient également le nom de l'image dont on définit la géométrie.
- A l'intérieur du bloc **<MAP> </MAP>** on trouve des définitions de type **<AREA>** définissant chacune des zones sensibles.
- Ainsi la structure de la balise sera la suivante :
- **<MAP NAME**=Nom>
  <AREA SHAPE=valeur **COORDS**= "valeur_coordonnées" **HREF**=URL>
  <**AREA SHAPE**=valeur **COORDS**= "valeur_coordonnées" **HREF**=URL>
  <**AREA SHAPE**=valeur **COORDS**= "valeur_coordonnées" **HREF**=URL>
  <**AREA SHAPE**=valeur COORDS= "valeur_coordonnées" **HREF**=URL>
  </**MAP**>
  <**IMG SRC**=Nom de l'image **USEMAP**="#Nom">

© Sidi Mahmoud Kaber

Les Images Mappées ...

- Les paramètres prennent les valeurs suivantes :
- **SHAPE**= **rect|circle|poly|default** la zone de définition est un rectangle, un cercle, un polygone ou le reste de la figure qui n'a pas été décrite. Ainsi :
  - **rect** : est défini par les coins opposés (par exemple "130,10,170,90")
  - **circle** : est défini par son centre et par le rayon (par exemple "50,50,10")
  - **poly** : est défini par un ensemble de points (par exemple "250,10 210,90 290,90")
  - **default** : est défini par les points non définis précédemment.
- **COORDS**= des valeurs entre côtes et séparées par des virgules comme décrit précédemment.
- **HREF**= l'URL qui sera atteinte après le clic.

© Sidi Mahmoud Kaber

Les Images Mappées ...

```
<BODY>
<MAP NAME="ww">
<AREA SHAPE=CIRCLE COORDS="255,165,81" HREF="a.html">
<AREA SHAPE=RECT COORDS="15,45,135,120" HREF="qqq">
<AREA SHAPE=POLY
    COORDS="180,15,240,60,270,30,210,15,195,15,180,15"
    HREF="qq.html">
</MAP>
</BODY>
```

```
<IMG SRC="cubes.gif" USEMAP="#ww"
```

© Sidi Mahmoud Kaber

Les Images Mappées ...

```
<HTML>
<BODY>
<CENTER>
<IMG SRC="cubes.gif" USEMAP="#cartons" HEIGHT=121 WIDTH=136>
</CENTER>
<MAP NAME="cartons">
<AREA SHAPE=RECT COORDS="37,9,72,40" HREF="a.htm">
<AREA SHAPE=RECT COORDS="18,46,46,79" HREF="b.htm">
<AREA SHAPE=RECT COORDS="61,43,93,78" HREF="c.htm">
<AREA SHAPE=RECT COORDS="9,84,36,119" HREF="d.htm">
<AREA SHAPE=RECT COORDS="48,85,77,116" HREF="e.htm">
<AREA SHAPE=RECT COORDS="89,81,123,115" HREF="f.htm">
</MAP>
</BODY>
</HTML>
```

© Sidi Mahmoud Kaber

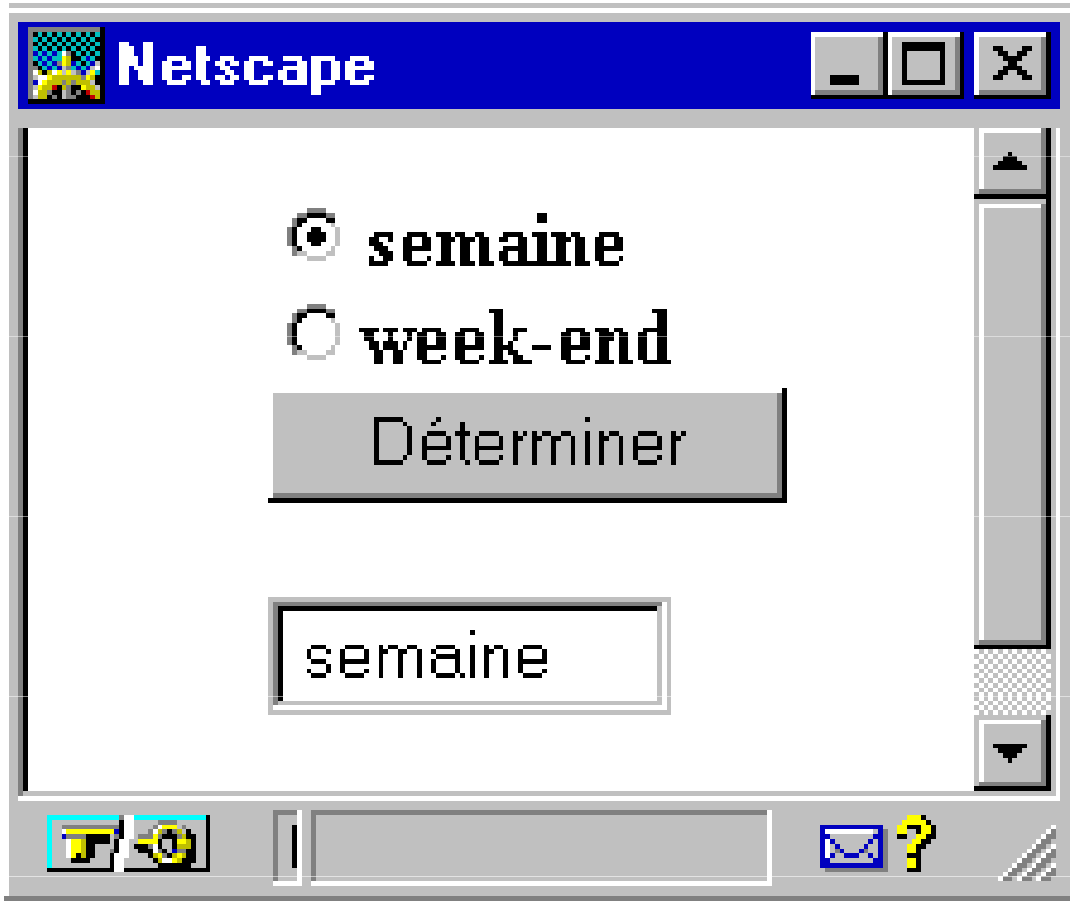
LES FORMULAIRES

LES FORMULAIRES

- Les formulaires peuvent prendre la forme :
- - d'une ligne de texte
- - de boutons radio
- - de cases à cocher
- - d'un menu déroulant

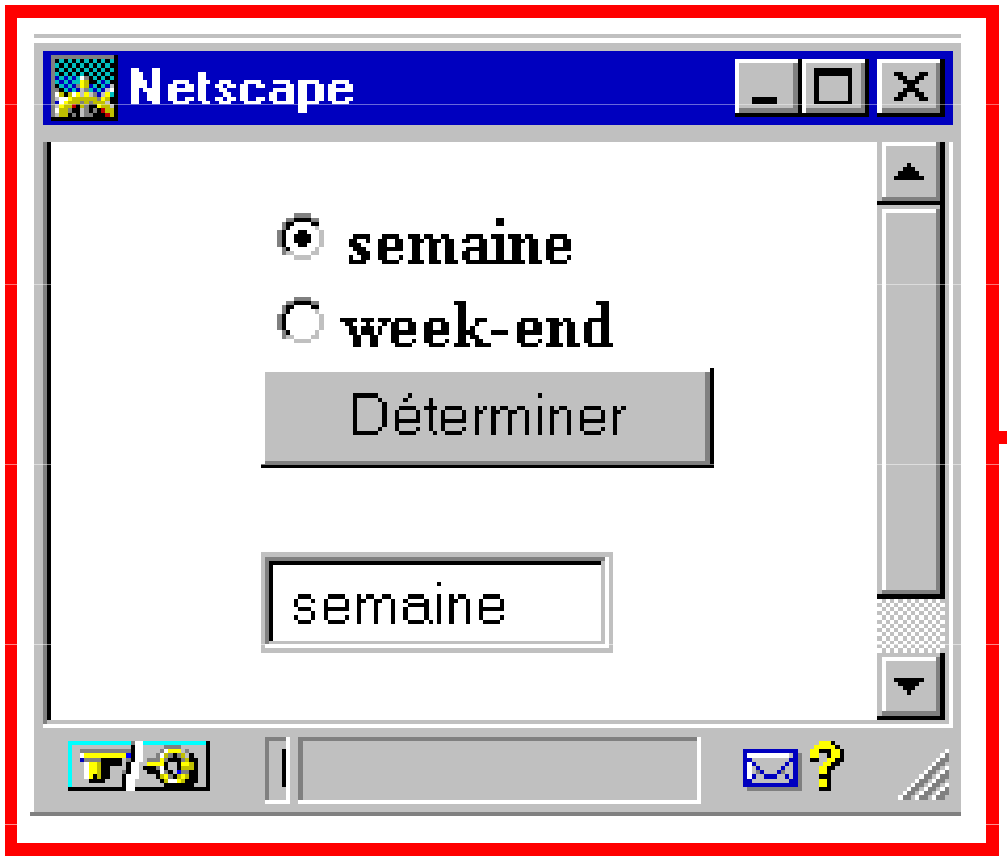
© Sidi Mahmoud Kaber

DÉFINITION D'UN FORMULAIRE



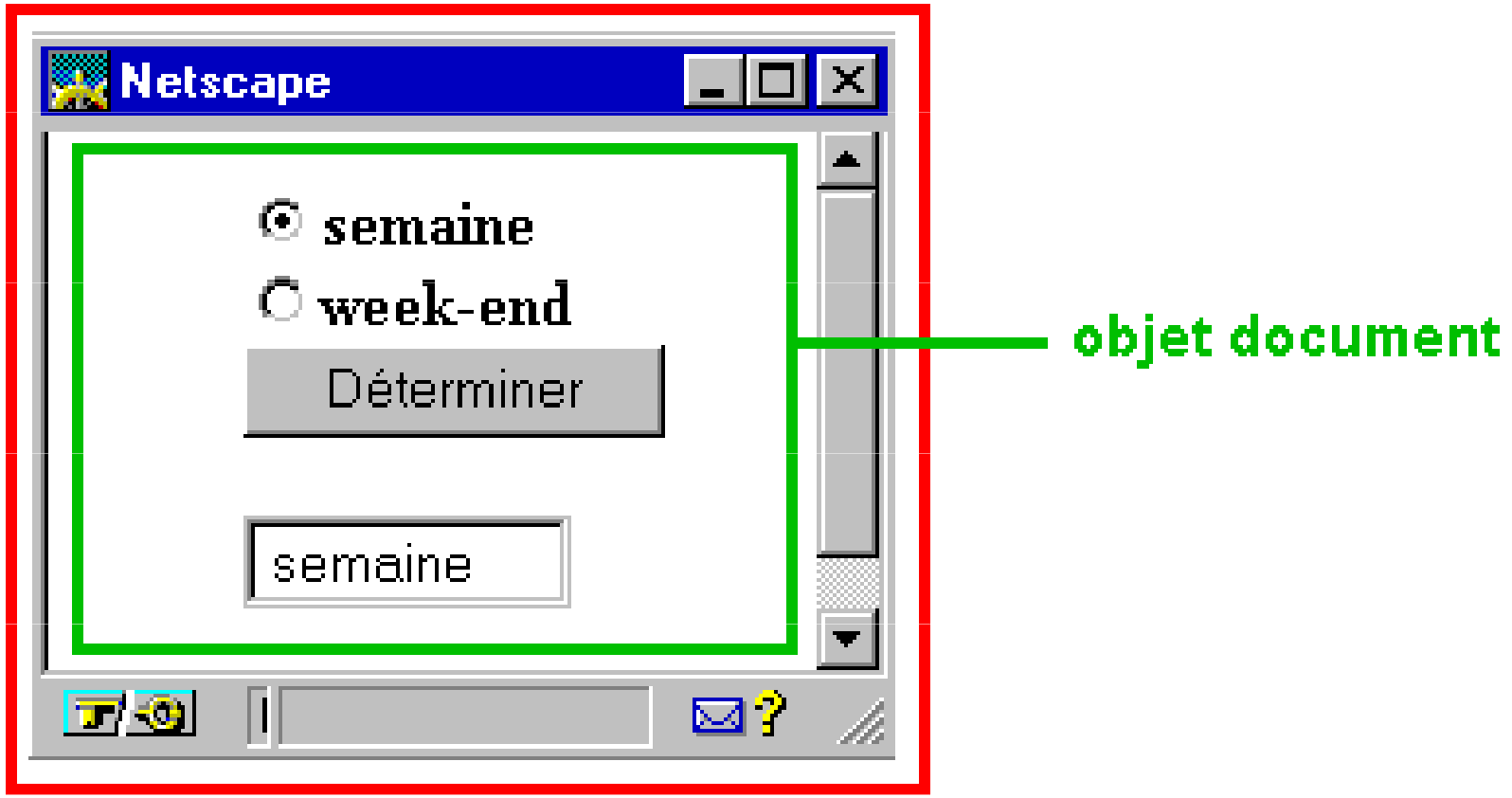
© Sidi Mahmoud Kaber

DÉFINITION D'UN FORMULAIRE



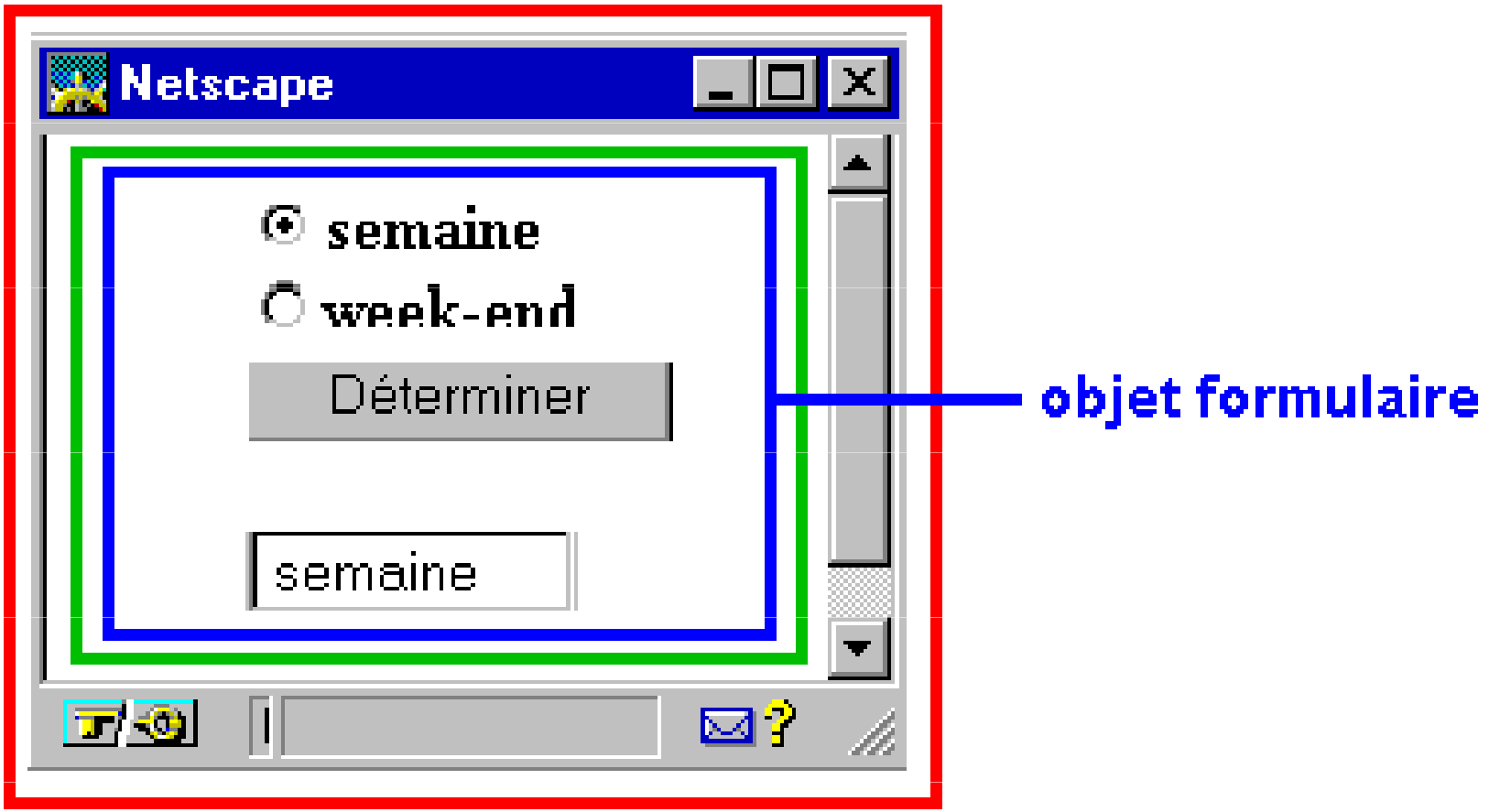
© Sidi Mahmoud Kaber

DÉFINITION D'UN FORMULAIRE



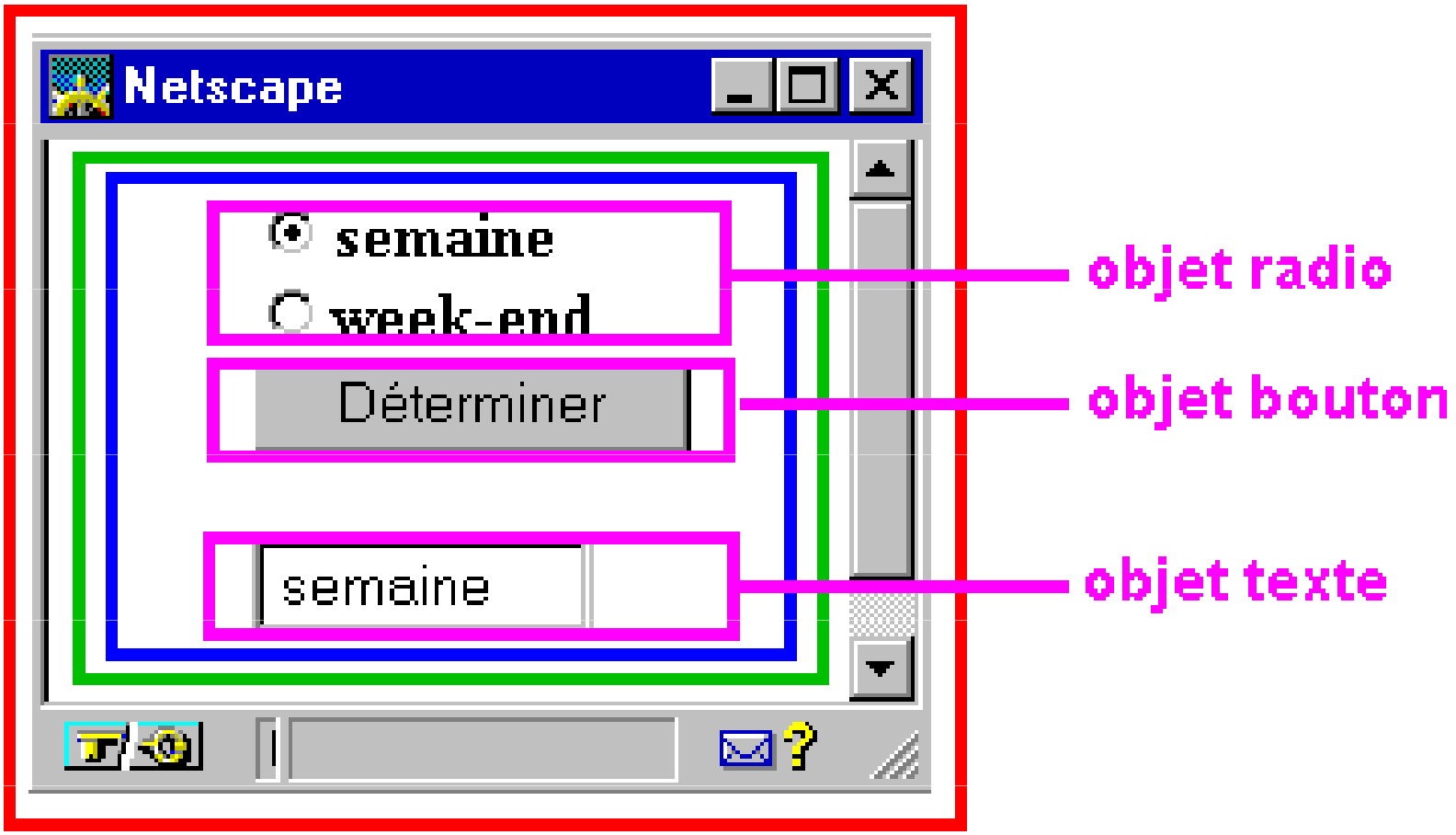
© Sidi Mahmoud Kaber

DÉFINITION D'UN FORMULAIRE



© Sidi Mahmoud Kaber

DÉFINITION D'UN FORMULAIRE



© Sidi Mahmoud Kaber

LES FORMULAIRES

Avant de pouvoir utiliser les différentes sortes de formulaires (ligne de texte, liste déroulante, cases à cocher...), il faut déclarer au browser qu'il devra gérer des formulaires et ce qu'il devra en faire.

<FORM method="post" action="URL d'expédition" enctype="text/plain">

... les formulaires proprement dit ...

</FORM>

© Sidi Mahmoud Kaber

LES FORMULAIRES

- L'attribut "method" : la façon dont les données seront transmises au serveur et exploitées par celui-ci
- L'attribut "action" spécifie l'adresse d'expédition des données.
- L'attribut "enctype" (optionnel) spécifie l'encodage utilisé pour le contenu du formulaire
- Dans le cas de l'utilisation en interne des formulaires par du Javascript, les attributs method, action et enctype sont inutiles car on ne fait pas appel au serveur.

© Sidi Mahmoud Kaber

LES FORMULAIRES : Ligne de texte

- INPUT type="text" indique un champ de saisie d'une seule ligne

```
<FORM>
<INPUT type="text" name="nom" size="50">
</FORM>
```

© Sidi Mahmoud Kaber

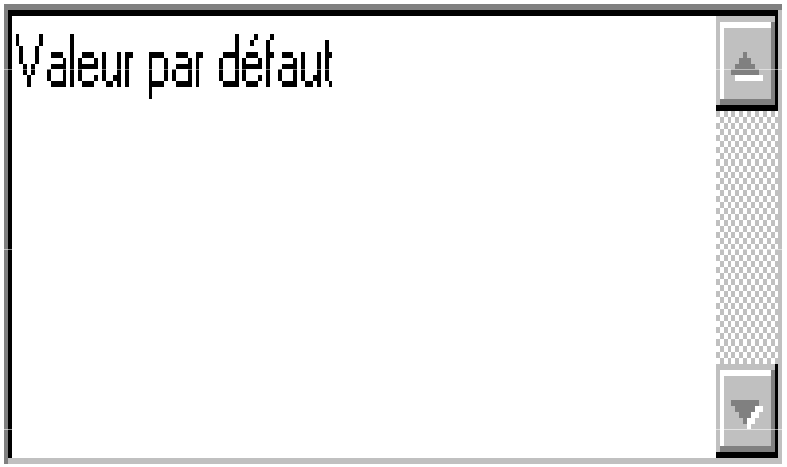
LES FORMULAIRES : Zone de saisie

La balise <TEXTAREA></TEXTAREA> introduit une zone de texte multilignes et non plus une simple ligne de texte. La syntaxe est :

<FORM>

<TEXTAREA name="nom" rows=4 cols=40 >Valeur par défaut</TEXTAREA>

</FORM>



© Sidi Mahmoud Kaber

LES FORMULAIRES : Zone de saisie

- L'attribut name permet de donner un nom au formulaire.
- L'attribut rows=x détermine le nombre de lignes de la zone de texte.
- L'attribut cols=y détermine le nombre de caractères visibles sur chaque ligne ou si vous préférez le nombre de colonnes.
- L'attribut wrap (optionnel) détermine la façon dont les sauts de ligne seront traités lors d'un changement de ligne.

© Sidi Mahmoud Kaber

LES FORMULAIRES : Liste déroulante

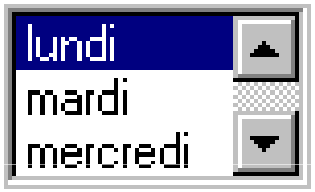
La balise <SELECT></SELECT> indique au browser l'usage d'une liste déroulante. Les éléments de la liste sont introduits par la balise <OPTION> ... (</OPTION> facultatif).

```
<FORM>
<SELECT name="nom" size="1">
<OPTION>lundi
<OPTION>mardi
<OPTION>mercredi
<OPTION>jeudi
<OPTION>vendredi
</SELECT>
</FORM>
```

© Sidi Mahmoud Kaber

LES FORMULAIRES : Liste déroulante

- L'attribut name="nom" définit le nom du formulaire.
- L'attribut size="x" détermine le nombre d'éléments de liste affiché dans la boite d'entrée. En fait, size="1" est optionnel car "1" est la valeur par défaut. Le même exemple avec size="3" donne :



© Sidi Mahmoud Kaber

LES FORMULAIRES : Liste déroulante

On peut présélectionner l'élément affiché dans la boite d'entrée (par défaut, le premier élément de la liste sera retenu)

```
<FORM>
<SELECT name="nom" size="1">
<OPTION>lundi
<OPTION>mardi
<OPTION selected>mercredi
<OPTION>jeudi
<OPTION>vendredi
</SELECT>
</FORM>
```

© Sidi Mahmoud Kaber

LES FORMULAIRES : Bouton d'option

Les boutons d'option, aussi appelés boutons radio, ont comme particularité qu'une seule option à la fois peut être activée (le "ou" exclusif).

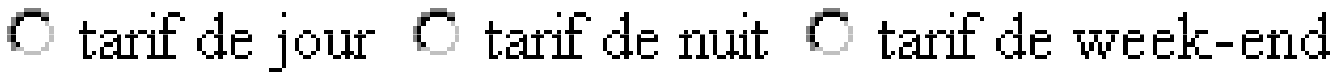
La syntaxe de base est :

```
<FORM>
<INPUT type="radio" name="nom du groupe"
  value="valeur du bouton">
</FORM>
```

© Sidi Mahmoud Kaber

LES FORMULAIRES : Bouton d'option

```
<FORM>
<INPUT type= "radio" name="tarif" value="jour"> tarif de jour
<INPUT type= "radio" name="tarif" value="nuit"> tarif de nuit
<INPUT type= "radio" name="tarif" value="week-end"> tarif de week-
    end
</FORM>
```



L'attribut "checked" (optionnel) permet de présélectionner un bouton radio

© Sidi Mahmoud Kaber

LES FORMULAIRES :
Case à cocher

Plusieurs choix simultanés peuvent être réalisés.

La syntaxe de base est :

```
<FORM>
<INPUT type="checkbox" name="nom"
   value="valeur attachée au bouton">
</FORM>
```

© Sidi Mahmoud Kaber

LES FORMULAIRES :
Case à cocher

```
<FORM>
<INPUT type="checkbox" name="choix1" value="1">
   glace vanille
<INPUT type="checkbox" name="choix2" value="2">
   chantilly
<INPUT type="checkbox" name="choix3" value="3">
   chocolat chaud
<INPUT type="checkbox" name="choix4" value="4">
   biscuit
</FORM>
```

☐ glace vanille ☐ chantilly ☐ chocolat chaud ☐ biscuit

© Sidi Mahmoud Kaber

LES FORMULAIRES : Bouton de commande

Avec l'introduction des langages de scripts (Javascript et VBscript) l'usage du bouton de commande présente un intérêt certain

<FORM>

<INPUT type="button" name="nom" value="texte du bouton" onclick="fonction Javascript">

</FORM>

© Sidi Mahmoud Kaber

LES FORMULAIRES : Bouton de commande

- <FORM>
- <INPUT type="button" name="nom" value= "Bouton de test" onclick="alert('Test réussi !')">
- </FORM>

© Sidi Mahmoud Kaber

LES FORMULAIRES : Submit et Reset

Le bouton **Submit** a la tâche spécifique de transmettre toutes les informations contenues dans le formulaire à l'URL désignée dans les attributs ACTION et METHOD du tag <FORM>.

la syntaxe Html est :

```
<FORM>
<INPUT TYPE="submit" NAME="nom"
   VALUE="texte du bouton">
</FORM>
```

© Sidi Mahmoud Kaber

LES FORMULAIRES : Submit et Reset

- Soit par exemple :
- <FORM>
- <INPUT TYPE="sub
  mit" NAME="nom"
  VALUE=" Envoyer ">
- </FORM>

© Sidi Mahmoud Kaber

LES FORMULAIRES : Submit et Reset

Le bouton Reset permet d'annuler les modifications apportées aux contrôles d'un formulaire et de restaurer les valeurs par défaut.

la syntaxe Html est :

```
<FORM>
<INPUT TYPE="reset" NAME="nom"
   VALUE="texte du bouton">
</FORM>
```

© Sidi Mahmoud Kaber

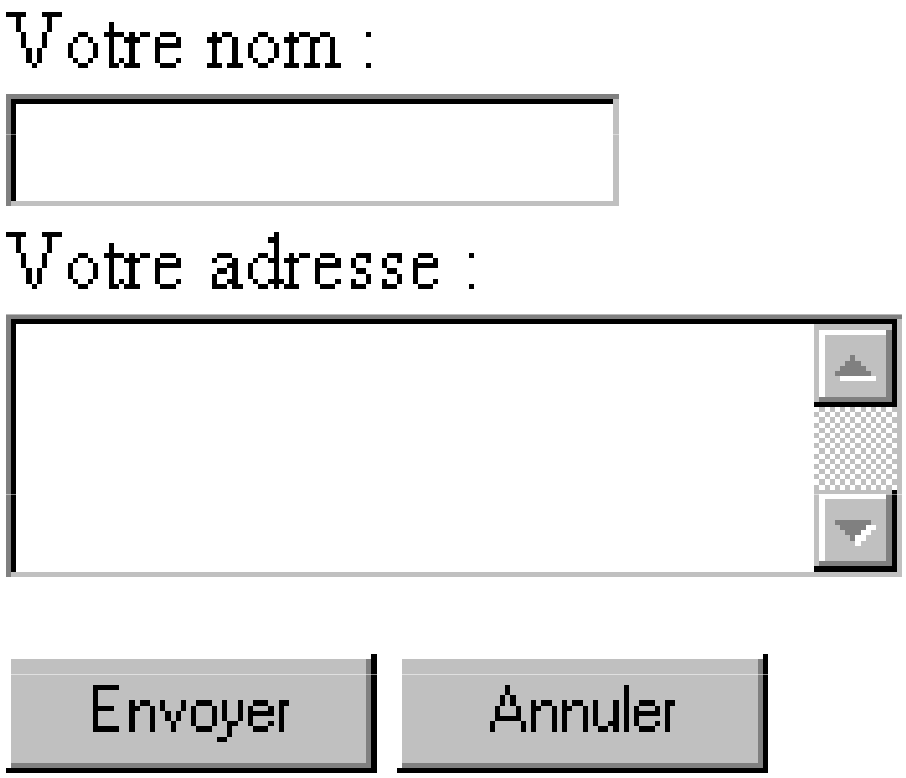
LES FORMULAIRES : Submit et Reset

Soit par exemple :

```
<FORM>
<INPUT TYPE="reset" NAME="nom" VALUE=" Annuler ">
</FORM>
```

© Sidi Mahmoud Kaber

EXEMPLE



© Sidi Mahmoud Kaber

EXEMPLE

```
<FORM method="post" action="mailto:votre_adresse_E-mail">
Votre nom :<BR>
<INPUT type="text" name="nom"><BR>
Votre adresse :<BR>
<TEXTAREA name="adresse" ROWS=2 COLS=35>
</TEXTAREA><BR>
<INPUT type="submit" value="Envoyer">
<INPUT type="reset" value="Annuler">
</FORM>
```

© Sidi Mahmoud Kaber

JAVASCRIPT

Introduction

Qu'est-ce que ce langage ?

- C'est un *langage de Programmation Interprété , structuré et Orienté Objet.*
- Il est inspiré du langage « C ».
- Il permet de résoudre les problèmes que HTML ne sait pas résoudre.
- Il est interprété directement par le Navigateur du client.
- Il s'écrit directement dans la Page HTML.

© Sidi Mahmoud Kaber

Pourquoi apprendre JavaScript ?

- Pour rendre un site *Interactif* :
  - Agir, par exemple, à un clique de sourie
  - Afficher de boite d'alerte ou de dialogue
  - Modification de paramètres d'une pages WEB, couleur du texte, couleur de fond

© Sidi Mahmoud Kaber

Pourquoi apprendre JavaScript ?

- **Avec JavaScript vous pourrez *programmer*, c'est à dire écrire des *algorithmes***
  - Effectuer des calculs et afficher les résultats obtenus
  - Vérifier la cohérence de données saisies dans un formulaire
  - Évaluer des questionnaires à choix multiple

© Sidi Mahmoud Kaber

Caractéristiques

- **Javascript** est un langage structuré (boucles conditionnelles, structures de test, fonctions, etc..).
- **Javascript** sait gérer les évènements principaux de la souris (déplacements, clicks , etc..).
- **Javascript** sait gérer les temporisations.

© Sidi Mahmoud Kaber

La balise <script> ... </script>

- Les scripts sont insérés dans une page HTML en utilisant la balise <script> de la manière suivante:

```
<script language="JavaScript">.
    instructions en Javascript.
</script>.
```

- Dès que le navigateur rencontre la balise <script> il passe la main à l'interpréteur du langage appelé.

© Sidi Mahmoud Kaber

Où mettre les scripts ?

- Si l'on préfère inclure le code JavaScript stocké dans un autre fichier, on peut l'indiquer par la ligne comme :

```
<script src= "nom_fichier.js" >
</script>
```

© Sidi Mahmoud Kaber

Où mettre les scripts ?

- Au début, nous placerons les instructions de JavaScript dans le conteneur <body>...</body>.
- Pour les fonctions globales pour tout le document, nous les placerons le plus souvent dans le conteneur <head>...</head>.

© Sidi Mahmoud Kaber

Premier exemple

```
<HTML>
<HEAD>
<TITLE>
  Mon 1er Programme en JavaScript
</TITLE>
</HEAD>
<BODY>
<SCRIPT language= "JavaScript">
document.write("Bonjour ..JavaScript..! ");
alert("Bonjour ..JavaScript..! ")
// voilà un commentaire au programme js
</SCRIPT>
</BODY>
</HTML>
```

© Sidi Mahmoud Kaber

Premier exemple ... Suite

L'instruction writeln()

```
<PRE>
<SCRIPT LANGUAGE="Javascript">
<--
document.writeln("Ligne 1");
document.writeln("Ligne 2");
//-->
</SCRIPT>
</PRE>
```

© Sidi Mahmoud Kaber

Premier exemple ... Suite

- str="Something"; (str est une variable)
- document.write("<BIG>"+str+"</BIG>");
- document.write('<BIG>Something</BIG>');
- document.write(str.big());
- document.write("Something".big());

© Sidi Mahmoud Kaber

Premier exemple ... Suite

- str="Something";
- document.write("<SMALL>"+str +"</SMALL>");
- document.write("<SMALL>Something" +"</SMALL>");
- document.write(str.small());
- document.write("Something".small());

© Sidi Mahmoud Kaber

Premier exemple ... Suite

- str="Something";
- document.write('<BLINK>'+str+'</BLINK>');
- document.write("<BLINK>Something</BLINK>");
- document.write(str.blink());
- document.write("Something".blink());

© Sidi Mahmoud Kaber

Premier exemple ... Suite

- str="Some words";
- document.write("<B>"+str+"</B>");
- document.write("<B>Some words</B>");
- document.write(str.bold());
- document.write("Some words".bold());

© Sidi Mahmoud Kaber

Premier exemple ... Suite

- str="Something";
- document.write("<TT>"+str+"</TT>");
- document.write("<TT>Something</TT>");
- document.write(str.fixed());
- document.write("Something".fixed());

© Sidi Mahmoud Kaber

Premier exemple ... Suite

- str="Something";
- document.write("<I>"+str+"</I>");
- document.write("<I>Something</I>");
- document.write(str.italics());
- document.write("Some word".italics());

© Sidi Mahmoud Kaber

Premier exemple ... Suite

- str1="Some words";
- str2="red";
- document.write("<FONT COLOR='red'>" +str1+"</FONT>");
- document.write("<FONT COLOR='red'>" +"Something</FONT>");
- document.write(str1.fontcolor(str2));
- document.write(str1.fontcolor("red"));

© Sidi Mahmoud Kaber

Premier exemple ... Suite

- str="Something";
- x=3;
- document.write("<FONT SIZE=3>" +str+"</FONT>");
- document.write("<FONT SIZE=3>" +"Something</FONT>");
- document.write(str.fontsize(3));
- document.write(str.fontsize(x));

© chafik ALOULOU

Premier exemple ... Suite

- str="Something";
- document.write("<STRIKE>"+str +"</STRIKE>");
- document.write("<STRIKE>Something" +"</STRIKE>");
- document.write(str.strike());
- document.write("Something".strike());

© Sidi Mahmoud Kaber

Premier exemple ... Suite

- str="Something";
- document.write("_{"+str+"}");
- document.write("<SUB>Something"
  +"</SUB>");
- document.write(str.sub());
- document.write("Something".sub());

© Sidi Mahmoud Kaber

Premier exemple ... Suite

- str="Something";
- document.write("^{"+str+"}");
- document.write("^{Something}");
- document.write(str.sup());
- document.write("Something".sup());

© Sidi Mahmoud Kaber

formatage de document

- document.bgColor
  - document.bgColor="white";
  - document.bgColor="#FFFFFF";
- document.fgColor
  - document.fgColor="black";
  - document.fgColor="#000000";
- document.alinkColor
  - document.alinkColor="white";
  - document.alinkColor="#FFFFFF";

© Sidi Mahmoud Kaber

formatage de document

- document.linkColor
  - document.linkColor="white";
  - document.linkColor="#FFFFFF";
- document.vlinkColor
  - document.linkColor="white";
  - document.linkColor="#FFFFFF";

© Sidi Mahmoud Kaber

JAVASCRIPT

La Syntaxe

Commentaires

- **Commentaires JavaScript : (2 marqueurs)**

```
//commentaire JS sur une ligne
/* commentaire JS sur
    plusieurs lignes */
```

- **Commentaires HTML : (1 seul marqueur)**

```
<!--commentaire HTML sur une ligne -->
<!-- commentaire HTML
            sur plusieurs lignes -->
```

© Sidi Mahmoud Kaber

Les mot-clés

- Un *mot-clé* est un symbole réservé au langage JavaScript.
- Par exemple, if est un mot-clé (il indique le début d'un test).
- On ne peut pas, donc, utiliser un mot-clé en dehors de son contexte.
- On ne peut pas, par exemple, choisir if comme nom d'une variable.

© Sidi Mahmoud Kaber

Les mots-clés (suite)

abstract	else	instanceof	short
boolean	extends	int	static
break	false	width	super
byte	final	interface	switch
case	finally	long	synchronized
catch	float	native	this
chat	for	new	throw
class	function	null	throws
const	goto	package	true
continue	if	private	try
default	implements	protected	var
do	import	public	void
Double	in	return	while

© Sidi Mahmoud Kaber

Les variables

- Une *variable* est une case mémoire à laquelle on donne un nom.
- Le contenu de cette case mémoire peut varier au cours de l'exécution du programme.
- On peut ranger une valeur dans la variable, par une instruction *d'affectation*.
- On peut aussi lire son contenu à tout moment

© Sidi Mahmoud Kaber

Comment déclarer une Variable?

- Une variables est, donc, définie par:
  - un **nom** (identificateur), un **type** , une **valeur** .

- `var Toto;`

  //Le type de variable n'est pas défini en la

  //déclarant ..elle le sera en lui affectant une valeur.

© Sidi Mahmoud Kaber

Comment déclarer une Variable?

- La valeur d'une variable peut être définie lors de sa déclaration ou au cours du programme.
- `var Toto` = 38 ; //donc Toto est une variable de type
  //numérique..avec laquelle on pourra faire des calculs.
- `var Toto = 'Fils'` ; //donc Toto est une variable
  //de type chaîne de caractères.
- `var PrUnit = 123.8` ; //donc Toto est une variable
  // de type numérique
- `var AxeH, AxeV, Vites = 25` ;

© Sidi Mahmoud Kaber

Les variables (suite)

- Les noms de variables répondent à quelques contraintes :
  - Ils ne doivent pas commencer par un chiffre
  - Ils peuvent commencer par une **Lettre**,le signe **$**, le caractère de soulignement (_)
  - ne pas utiliser : **%** * @ qui sont réservés à certains opérateurs du Langage..

© Sidi Mahmoud Kaber

Les variables (suite)

- Attention les **majuscules** et les **minuscules** sont bien différenciées..Donc :
  - La variable `Toto` n'est pas la même que `TOTO` ou `ToTo` ou `toto`

- En général il est intéressant de choisir un nom de variable qui rappelle sa fonction dans le programme .

© Sidi Mahmoud Kaber

Visibilité des variables

- On appelle ‘visibilité’ des variables la possibilité d’utiliser des variables depuis diverses parties du programme.
- Une variable peut être dite **globale** si elle est déclarée dans la balise JavaScript et en dehors de toute fonction.
- Alors elle peut être utilisée dans toute la page HTML et à l’intérieur des fonctions éventuelles.
- Une Variable est dite **locale** si elle est déclarée à l’intérieur d’une fonction.

© Sidi Mahmoud Kaber

Les types de données

- ***Peut on mettre n'importe quelle sorte de données dans une variable?***
- La réponse est non, car il se poserait alors des problèmes de cohérence:
- Un *type de données* est un ensemble de valeurs et un ensemble d'opérateurs sur ces valeurs.
- Par exemple, le type *entier* est constitué de l'ensemble des entiers et des opérateurs +, ,*, /, etc.
- Un autre type très utilisé est le type *chaîne de caractères* permettant de traiter les messages.
- Une chaîne est un ensemble de caractères composé de lettres, symboles ou chiffres.

© Sidi Mahmoud Kaber

Les types de variables

- **Booléen** : 2 valeurs Vrai ou Faux
  - `var flag1 = true;//flag1` **est initialisé à vrai**
  - `var flag2 = false;//flag2` **est initialisé à faux**

- **Numérique :** on pourra faire des calculs avec
  - `var Tva = 18.6 ;`
  - `var quantite = 150 ;`

© Sidi Mahmoud Kaber

Les types de variables

- **Chaîne de Caractères :**
  - `var Nom = "Durand" ;`
  - `var Age = '25' , age = 25 ;`

    `//Age` **n'est pas** `age..!`

- *Remarque :* Le mot clé **VAR** permet de déclarer plusieurs variables en les séparant par une virgule

© Sidi Mahmoud Kaber

Deuxième exemple

```
<HTML>
<HEAD><TITLE> 2ème Programme en JavaScript</TITLE>
</HEAD>
<BODY>
<!-- du code HTML
<SCRIPT language="JavaScript">
   var age= 18, nom='Durand';
age*=2;
   document.write('Bonjour Mr. ' + nom + 'vous
                  avez' + age + 'ans');
  // blablabla-->
</SCRIPT>
<BODY>
</HTML>
```

© Sidi Mahmoud Kaber

Les Opérateurs

- **Les Opérateurs d'affectation de JavaScript**

```
x = 12; // affectation simple

x += 20; // équivalent à x = x + 20;

x -= 5; // équivalent à x = x - 5;

x *= 4; // équivalent à x = x * 4;
```

© Sidi Mahmoud Kaber

Les Opérateurs

x /= 3; // équivalent à **x = x / 3;**

Y %= 2; // équivalent à **Y = Y % 2;**
reste de la division entière de Y/2 (le Modulo)

I++; // permet d'incrémenter **i** de 1 .. c'est donc
équivalent à **i = i + 1;**

i--;// permet de décrémenter **i** de 1.. c'est
équivalent à **i = i – 1;**

© Sidi Mahmoud Kaber

Les Opérateurs de Comparaison

M1 > M2 *// signifie* ***M1*** *plus grand que* ***M2***

N3 < 12 *// signifie* ***N3*** *plus petit que 12*

X >= 25 *// signifie* ***X*** *plus grand ou égal à 25*

X <= 63 *// signifie* ***X*** *plus petit ou égal à 63*

© Sidi Mahmoud Kaber

Les Opérateurs de Comparaison

X == Y *// signifie* ***X*** *égal à* ***Y***

m != 41 *// signifie* ***m*** *différend de 41*

exp1 && exp2 *// ET logique entre* `exp1` *et* `exp2`

e5 || e4 *// OU logique entre* `e5` *et*

© Sidi Mahmoud Kaber

Operations ...

Signe	Nom	Signification	Exemple	Résultat
+	plus	addition	x + 3	14
-	moins	soustraction	x - 3	8
*	multiplié par	multiplication	x*2	22
/	divisé	par division	x /2	5.5
%	modulo	reste de la division par	x%5	1
=	a la valeur	affectation	x=5	5

© Sidi Mahmoud Kaber

Afficher du texte

- **Document.write()**

```
<html>
<head><title>Programme Out1 </title>
</head>
<body>
<script language="JavaScript">
   document.write('Bienvenue à  JavaScript ');
</script>
</body>
</html>
```

© Sidi Mahmoud Kaber

Afficher le contenu des variables ?

- L'information est dans la mémoire de l'ordinateur. Pour la voir, il faut l'appeler à l'écran.

```
<script language="JavaScript">
 var  jour = 21;
  var mois = 'juin';
  document.write(' : Date = '+ jour + '  ' + mois);
</script>
```

© Sidi Mahmoud Kaber

JAVASCRIPT

Les conditions
Les Boucles

Structure conditionnelle

```
if (condition vraie) {
instructions1;
}
else {
instructions2;
instructions3;
instructions4;
}
```

© Sidi Mahmoud Kaber

Structure conditionnelle… Exemple

```
if (X > Y ) //si X > Y alors
 {
  alert(" X supérieur à Y ! ");
 }
else
 {
  alert("Y supérieur ou égal à X !
  ");
 }
```

© Sidi Mahmoud Kaber

L'expression for

```
for (valeur initiale ; condition ; progression) {
        instructions;
        }
```

Prenons un exemple concret

```
        for (i=0; i<10; i++) {
        document.write(i + "<BR>")
```

© Sidi Mahmoud Kaber

While

```
while (condition vraie){
        continuer à faire quelque chose
        }
```

Prenons un exemple.

```
        compt=1;
        while (compt<=5) {
        document.write ("ligne : " + compt + "<BR>");
        compt++;
        }
        document.write("fin de la boucle");
```

© Sidi Mahmoud Kaber

Break

L'instruction **break** permet d'interrompre prématurément une boucle **for** ou **while**

```
compt=1;
while (compt<5) {
if (compt == 4)
break;
document.write ("ligne : " + compt + "<BR>");
compt++;
}
document.write("fin de la boucle");
```

© Sidi Mahmoud Kaber

Continue

L'instruction **continue** permet de sauter une instruction dans une boucle **for** ou **while** et de continuer ensuite le bouclage (sans sortir de celui-ci comme le fait **break**).

```
compt=1;
while (compt<5) {
if ((compt == 2)||(compt == 3)){
compt++
continue;}
document.write ("ligne : " + compt + "<BR>");
compt++;
}
document.write("fin de la boucle");
```

© Sidi Mahmoud Kaber

JAVASCRIPT

Les Fonctions

Lire des informations avec prompt()

- Cette méthode ouvre une boîte de dialogue avec une zone saisie et 2 boutons : `OK` et `Annuler`. Elle permet :
  - d'envoyer un message, mais surtout,
  - de recevoir des informations.

```
<script language="JavaScript">
 annee= prompt('En quelle année sommes-nous ? ',
  1999);
 alert('Vous avez répondu : ' + annee)
</script>
```

© Sidi Mahmoud Kaber

```
<Script language="javascript">
X= prompt('donner la valeur de X ', 5);
Y= prompt('donner la valeur de X ', 5)
XX=Number(X)
YY=Number(Y)
if (X > Y ) //si X > Y alors
 {
  alert(" X supérieur à Y ! ");
 }
else
 {
  alert("Y supérieur ou égal à X ! ");
 }
  <Script>
```

© Sidi Mahmoud Kaber

Lire l'information avec comfirm()

Cette méthode ouvre une boîte de dialogue avec 2 boutons : OK Annuler. Elle permet :

d'envoyer une information et de recevoir un booléen.

Syntaxe :

confirm(...) {...} else {...}

```
<script language="JavaScript">
    if (confirm('Je vais dire sur quel bouton vous avez appuyé : '))
        alert(' vous avez appuyé sur OK ')
    else alert(' vous avez appuyé sur Annuler ') ;
</script>
```

© Sidi Mahmoud Kaber

Lire l'information avec comfirm()

Cette méthode ouvre une boîte de dialogue avec 2 boutons : OK Annuler. Elle permet :

d'envoyer une information et de recevoir un booléen.

Syntaxe :

confirm(...) {...} else {...}

```
<script language="JavaScript">
function Confirmer()
    {
       if (confirm('Je vais dire sur quel bouton vous avez appuyé : '))
        alert(' Sur OK \n Continuez avec :')
      else alert(' Sur Annuler \n Sortez avec Ok !') ;
    }
</script>
```

© Sidi Mahmoud Kaber

Fonctions

- Une fonction est un groupe de ligne(s) de code de programmation destiné à exécuter une tâche bien spécifique et que l'on pourra, si besoin,l’ utiliser à plusieurs reprises.
- De plus, l'usage des fonctions améliorera grandement la lisibilité de votre script.

© Sidi Mahmoud Kaber

Fonctions

- En Javascript, il existe deux types de fonctions :
  - les fonctions propres à Javascript. On les appelle des "méthodes". Elles sont associées à un objet bien particulier comme c'était le cas de la méthode Alert() avec l'objet window.
  - les fonctions écrites par vous-même pour les besoins de votre script. C'est à celles-là que nous nous intéressons maintenant.

© Sidi Mahmoud Kaber

Déclaration des fonctions

- Pour déclarer ou définir une fonction, on utilise le mot (réservé) function.
- La syntaxe d'une déclaration de fonction est la suivante :

```
function nom_de_la_fonction(arguments) {
... code des instructions ...
}
```

© Sidi Mahmoud Kaber

Déclaration des fonctions

- Le nom de la fonction suit les mêmes règles que celles qui régissent le nom de variables (nombre de caractères indéfini, commencer par une lettre, peuvent inclure des chiffres...).
- Javascript est sensible à la case.
- fonction() ne sera pas égal à Fonction().
- Tous les noms des fonctions dans un script doivent être uniques.

© Sidi Mahmoud Kaber

Appel d'une fonction

- L'appel d'une fonction se fait par le nom de la fonction (avec les parenthèses).
- Soit par exemple nom_de_la_fonction();
- Il faut que l'interpréteur trouve la fonction bien définie avant d'être appelée.

© Sidi Mahmoud Kaber

Exemple

```
<HTML>
<Script language="Javascript">
var x,y;
function calcul_somme(a,b)
{
var somme;
somme=a+b
document.write(somme)
}
function calcul_produit(a,b)
{
var produit;
produit=a*b
document.write(produit)
}
x=prompt("Donner la valeur de x);
y=prompt("Donner la valeur de y);
x=Number(x);
y=Number(y);
if(confirm("calculer la somme?"))
calcul_somme(x,y);
if(confirm("calculer le produit?"))
calcul_produit(x,y);
</script>
</HTML>
```

© Sidi Mahmoud Kaber

Exemple

```
<HTML>
<Script language="Javascript">
var x,y;
function calcul_somme(a,b)
{
var somme;
somme=a+b
document.write(somme)
}
function calcul_produit(a,b)
{
var produit;
produit=a*b
document.write(produit)
}
```

```
x=prompt("Donner la valeur de x);
y=prompt("Donner la valeur de y);
x=Number(x);
y=Number(y);
if(confirm("calculer la somme?"))
calcul_somme(x,y);
```

document.write(somme)

```
if(confirm("calculer le produit?"))
calcul_produit(x,y);
```

document.write(produit)

```
</script>
</HTML>
```

© Sidi Mahmoud Kaber

Exemple

```
<HTML>
<Script language="Javascript">
var x,y;
var somme, produit;
function calcul_somme(a,b)
{
var somme;
somme=a+b
}
function calcul_produit(a,b)
{
var produit;
produit=a*b
}
```

```
x=prompt("Donner la valeur de x);
y=prompt("Donner la valeur de y);
x=Number(x);
y=Number(y);
if(confirm("calculer la somme?"))
calcul_somme(x,y);

document.write(somme)

if(confirm("calculer le produit?"))
calcul_produit(x,y);

document.write(produit)

</script>
</HTML>
```

© Sidi Mahmoud Kaber

Les fonctions dans <HEAD>...<HEAD>

- Il est prudent de placer toutes les déclarations de fonction dans l'en-tête de la page c.-à-d .dans la balise <HEAD> ... <HEAD>.

- Ainsi, les fonctions seront prises en compte par l'interpréteur avant qu'elles soient appelées dans le <BODY>

© Sidi Mahmoud Kaber

Exemple

```
<HTML>
<HEAD>
<SCRIPT LANGUAGE="Javascript">
A=“Bienvenue à ma page’’
B=‘Message 2’’
function message(W) {
alert(W);
}
</SCRIPT>
</HEAD>
<BODY onLoad="message(A)"   onUnload="message(B)">
</BODY>
</HTML>
```

© Sidi Mahmoud Kaber

Passer une valeur à une fonction

- On peut passer des valeurs ou paramètres aux fonctions Javascript. La valeur ainsi passée sera utilisée par la fonction.

- Pour passer un paramètre à une fonction, on fournit un nom d'une variable dans la déclaration de la fonction.

Exemple

- Ecrire une fonction qui affiche une boite d'alerte dont le texte peut changer.

© Sidi Mahmoud Kaber

Passer une valeur à une fonction

Dans la déclaration de la fonction, ecrire :

```
function Exemple(Texte) {
alert(texte);
}
```

Le nom de la variable est Texte et est définie comme un paramètre de la fonction.

Dans l'appel de la fonction, on lui fournit le texte :

```
Exemple("Salut à tous");
```

© Sidi Mahmoud Kaber

Passer plusieurs valeurs à une fonction

- On peut passer plusieurs paramètres à une fonction.
- Comme c'est souvent le cas en Javascript, on sépare les paramètres par des virgules.

```
function nom_de_la_fonction(arg1, arg2, arg3)
{
... code des instructions ...
}
```

© Sidi Mahmoud Kaber

Passer plusieurs valeurs à une fonction

Le premier exemple devient pour la déclaration de fonction :

function Exemplebis(Texte1, Texte2){...}

et pour l'appel de la fonction

Exemplebis("Salut à tous", "Signé Luc")

© Sidi Mahmoud Kaber

Retourner une valeur

Pour renvoyer un résultat, il suffit d'écrire le mot clé return suivi de l'expression à renvoyer.

Par exemple :

```
function cube(nombre) {
var cube = nombre*nombre*nombre
return cube;
}
```

© Sidi Mahmoud Kaber

Retourner une valeur

- Précisons que l'instruction **return** est facultative et qu'on peut trouver plusieurs return dans une même fonction.

- Pour exploiter cette valeur de la variable retournée par la fonction, on utilise une formulation du type

document.write(cube(5)).

© Sidi Mahmoud Kaber

Variables locales et variables globales

Une variable déclarée dans une fonction par le mot clé var aura une portée limitée à cette seule fonction. On ne pourra donc pas l'exploiter ailleurs dans le script. On l'appelle donc variable locale.

```
function cube(nombre) {
var cube = nombre*nombre*nombre
}
```

la variable cube dans cet exemple est une variable locale. Si vous y faites référence ailleurs dans le script, cette variable sera inconnue pour l'interpréteur Javascript (message d'erreur).

© Sidi Mahmoud Kaber

Variables locales et variables globales

- Si la variable est déclarée sans utiliser le mot var, sa portée sera globale -- et pour être tout à fait précis, une fois que la fonction aura été exécutée--.

- Pour la facilité de gestion des variables, il faut les déclarer en début de script (comme dans la plupart des langages de programmation).

© Sidi Mahmoud Kaber

Exercices

- Ecrire un programme qui affiche sur la page même la somme des 100 premiers entiers.
- Ecrire une version sans fonction et une deuxième avec fonction.
- Modifier le programme en ajoutant un bouton qui permet après son clique d'afficher cette somme sur un composant alert.
- Modifier ce programme afin d'afficher la somme sur une zone text.

© Sidi Mahmoud Kaber

Reponse

```
<HTML>
<BODY>
<Script Language= "JavaScript">
....
</Script>
<FORM>
<INPUT TYPE="button" value="Calculer la somme"
   onClick="calculS()">
</FORM>
</BODY> </HTML>
```

© Sidi Mahmoud Kaber

```
<FORM name="AA">
Donner la borne Inferieure: <INPUT TYPE=text  name="E1">
   <BR>
Donner la borne Superieure: <INPUT TYPE=text
   name="E2">
<BR>
Le resultat :<INPUT TYPE=text  name="resultat"> <BR>
<INPUT type="button" name="nom" value= "Calculer somme"
   onclick="calculer()">
</FORM>
</html>
```

© Sidi Mahmoud Kaber

Exercices

- Ecrire un programme qui affiche dans un tableau la somme impaire des 100 premiers entiers.
- Ecrire un programme qui affiche la somme des entiers entre deux valeurs saisies au clavier (utilisation de boutons).
- Ecrire un programme qui affiche le nombre d'années bissextiles entre 1900 et 2004.

© Sidi Mahmoud Kaber

```
<html>
<script Language='Javascript'>
Nbre_annee=0;
For(i=1900;i<=2004;i++)
{
if (i%4==0)
  Nbre_annee=Nbre_annee+1;
}
alert(Nbre_annee)
</script>
</html>
```

© Sidi Mahmoud Kaber

Exercices

- Ecrire une page Web contenant deux boutons :
  - Le premier permet d'afficher sur un composant (alert) la somme des 10 premiers entiers.
  - Le deuxième permet d'afficher sur un composant (alert) le produit des 6 premiers entiers.
- Réaliser une fonction qui double un nombre saisie à partir d'une zone texte. Le résultat doit être affiché sur une zone texte.

© Sidi Mahmoud Kaber

```
<html>
<FORM name=“F1">
<INPUT TYPE=text  name=“saisie">
<INPUT TYPE=text  name=“resultat">
<INPUT type="button" name="nom" value=
  “doubler" onclick="calculer()">
<INPUT type=“reset" name="nom" value=
  “Initialiser" >

</FORM>
```

© Sidi Mahmoud Kaber

```
<script language="JavaScript">
function calculer()
{
F1.resultat.value=2*(F1.saisie.value)
}
</script>
</html>
```

© Sidi Mahmoud Kaber

Exercices

- Ecrire une page Web contenant deux boutons :
  - Le premier permet de lire une valeur entière *n* à partir d'un composant d'entrée (prompt) et affiche la somme des *n* premiers entiers sur un composant de sortie (alert). Donner une solution itérative et une solution récursive.
  - Le deuxième permet le retour à la page précédente du navigateur.
  - (onClick="window.history.go(-1))

© Sidi Mahmoud Kaber

JAVASCRIPT

Les événements

Les événements

- En Html classique, il y a un événement qui est bien connu: C'est le clic de la souris sur un lien pour passer sur une autre page Web.
- Hélas, c'est à peu près le seul.
- Javascript va en ajouter une bonne dizaine, pour mieux gérer le site.

© Sidi Mahmoud Kaber

Les événements

Description	Evénement
Lorsque l'utilisateur clique sur un bouton, un lien ou tout autre élément.	Clik
Lorsque la page est chargée par le browser ou le navigateur.	Load
Lorsque l'utilisateur quitte la page.	Unload
Lorsque l'utilisateur place le pointeur de la souris sur un lien ou tout autre élément.	MouseOver
Lorsque le pointeur de la souris quitte un lien ou tout autre élément. Attention : Javascript 1.1 (donc pas sous MSIE 3.0 et Netscape 2).	MouseOut

© Sidi Mahmoud Kaber

Les événements

Description	Evénement
Lorsque un élément de formulaire a le focus c-à-d devient la zone d'entrée active.	Focus
Lorsque un élément de formulaire perd le focus c-à-d que l'utilisateur clique hors du champs et que la zone d'entrée n'est plus active.	Blur
Lorsque la valeur d'un champ de formulaire est modifiée.	Change
Lorsque l'utilisateur sélectionne un champ dans un élément de formulaire.	Select
Lorsque l'utilisateur clique sur le bouton Submit pour envoyer un formulaire.	Submit

© Sidi Mahmoud Kaber

Les gestionnaires d'événements

- Pour être efficace, il faut qu'à ces événements soient associées les actions prévues par vous. C'est le rôle des gestionnaires d'événements. La syntaxe est

onévénement="fonction()"

- Par exemple, onClick="alert('Vous avez cliqué sur cet élément')".

- De façon littéraire, au clic de l'utilisateur, ouvrir une boite d'alerte avec le message indiqué

© Sidi Mahmoud Kaber

onclick

Evénement classique en informatique, le clic de la souris.

Le code de ceci est :

```
<FORM>
<INPUT TYPE="button" VALUE="Cliquez ici"
   onClick="alert('Vous avez bien cliqué ici')">
</FORM>
```

© Sidi Mahmoud Kaber

onLoad et onUnload

- L'événement **Load** survient lorsque la page a fini de se charger. A l'inverse, **Unload** survient lorsque l'utilisateur quitte la page.

- Les événements **onLoad** et **onUnload** sont utilisés sous forme d'attributs de la balise **<BODY>** ou **<FRAMESET>**

© Sidi Mahmoud Kaber

onLoad et onUnload

pour souhaiter la bienvenue à l'ouverture d'une page et un petit mot d'au revoir au moment de quitter celle-ci.

```
<HTML>
<HEAD>
<SCRIPT LANGUAGE='Javascript'>
function bienvenue() {
alert("Bienvenue à cette page");}
function au_revoir() {
alert("Au revoir");}
</SCRIPT>
</HEAD>
<BODY onLoad='bienvenue()' onUnload='au_revoir()'>
Html normal
</BODY>
</HTML>
```

© Sidi Mahmoud Kaber

onmouseOver et onmouseOut

- L'événement **onmouseOver** se produit lorsque le pointeur de la souris passe au dessus (sans cliquer) d'un lien ou d'une image.

- Cet événement est fort pratique pour, par exemple, afficher des explications soit dans la barre de statut soit avec une petite fenêtre genre infobulle

© Sidi Mahmoud Kaber

onmouseOver et onmouseOut

- L'événement **onmouseOut**, généralement associé à un **onmouseOver**, se produit lorsque le pointeur quitte la zone sensible (lien ou image).

- Notons qui si **onmouseOver** est du Javascript 1.0, **onmouseOut** est du Javascript 1.1.

- En clair, **onmouseOut** ne fonctionne pas avec Netscape 2.0 et Explorer 3.0.

© Sidi Mahmoud Kaber

onmouseOver et onmouseOut

- Le code du gestionnaire d'événement onmouseOver s'ajoute aux balises de lien :
- <A HREF="" onmouseOver="action()">lien</A>

- lorsque l'utilisateur passe avec la souris sur le lien, la fonction action() est appelée.

- L'attribut HREF est indispensable. Il peut contenir l'adresse d'une page Web si vous souhaitez que le lien soit actif ou simplement des guillemets si aucun lien actif n'est prévu.

© Sidi Mahmoud Kaber

Exemple de onmoueOver

Voici un exemple. Par le survol du lien "message important", une fenêtre d'alerte s'ouvre.

Le code est :

```
<BODY>
...
<A HREF=""
   onmouseOver="alert('Coucou')">message
   important</A>
...
<BODY>
```

© Sidi Mahmoud Kaber

Exemple de onmoueOver

ou si vous préférez utiliser les balises <HEAD>

```
<HTML>
<HEAD>
<SCRIPT language="Javascript">
function message(){
alert("Coucou")
}
</SCRIPT>
</HEAD>
<BODY>
<A HREF="" onmouseOver="message()">message important</A>
</BODY>
</HTML>
```

© Sidi Mahmoud Kaber

La syntaxe de onmouseOut

- L'événement se produit lorsque le pointeur de la souris quitte le lien ou la zone sensible.
- onmouseOver est du Javascript 1.0 et sera donc reconnu par tous les browsers,
- onmouseOut est du Javascript 1.1 et ne sera reconnu que par Netscape 3.0 et plus et Explorer 4.0 et plus
- Le code est le suivant :
- <A HREF="" onmouseOver="alert('Coucou')" onmouseOut="alert('Au revoir')">message important</A>

© Sidi Mahmoud Kaber

onFocus

- L'événement onFocus survient lorsqu'un champ de saisie a le focus c.-à-d. quand son emplacement est prêt à recevoir ce que l'utilisateur à l'intention de taper au clavier.

- C'est souvent la conséquence d'un clic de souris ou de l'usage de la touche "Tab

© Sidi Mahmoud Kaber

```
<HTML>
<HEAD>
<script language="Javascript">
function sur_image()
{
document.images[" Enis"].src= " 2eme nom de fichier.gif"
}
</script>
</HEAD>
<BODY>
<A HREF="http://www.yahoo.fr" onmouseOver="sur_image() "
onmouseOut=" out_image()">
<IMG SRC="button_dim.gif" name=" Enis"> </A>
</BODY>
</HTML>
```

© Sidi Mahmoud Kaber

```
<HTML>
<HEAD>
<script language="Javascript">
function sur_image()
{
document.images["homeButton"].src="button_hot.gif"
}
function dimDown() {
document.images["homeButton"].src="button_dim.gif"
}
</script>
</HEAD>
<BODY>
<A HREF="http://www.yahoo.fr" onmouseOver="sur_image()"
onmouseOut="dimDown();">
<IMG SRC="button_dim.gif" name="homeButton" width=100 height=50
border=0> </A>
</BODY>
</HTML>
```

© Sidi Mahmoud Kaber

onBlur

- L'événement **onBlur** a lieu lorsqu'un champ de formulaire perd le focus.
- Cela se produit quand l'utilisateur ayant terminé la saisie qu'il effectuait dans une case, clique en dehors du champ ou utilise la touche "Tab" pour passer à un champ.
- Cet événement sera souvent utilisé pour vérifier la saisie d'un formulaire.
- **Le code est :**

```
<FORM>
<INPUT TYPE=text onBlur="alert('Ceci est un Blur')">
</FORM>
```

© Sidi Mahmoud Kaber

onchange

- Cet événement s'apparente à l'événement onBlur mais avec une petite différence.
- Non seulement la case du formulaire doit avoir perdu le focus mais aussi son contenu doit avoir été modifié par l'utilisateur

© Sidi Mahmoud Kaber

onselect

- Cet événement se produit lorsque l'utilisateur a sélectionné (mis en surbrillance ou en vidéo inverse) tout ou partie d'une zone de texte dans une zone de type text ou textarea

© Sidi Mahmoud Kaber

Gestionnaires d'événement disponibles en Javascript

Objets	Gestionnaires d'événement disponibles
Fenêtre	onLoad, onUnload
Lien hypertexte	onClick, onmouseOver, on mouseOut
Elément de texte	onBlur, onChange, onFocus, onSelect
Elément de zone de texte	onBlur, onChange, onFocus, onSelect
Elément bouton	onClick
Case à cocher	onClick
Bouton Radio	onClick
Liste de sélectionon	Blur, onChange, onFocus
Bouton Submit	onClick
Bouton Reset	onClick

© Sidi Mahmoud Kaber

Changement d'images

```
<HTML>
<HEAD>
<SCRIPT LANGUAGE="Javascript">
function lightUp() {
document.images["homeButton"].src="button_hot.gif"
}
function dimDown() {
document.images["homeButton"].src="button_dim.gif"
}
</SCRIPT>
</HEAD>
<BODY>
<A HREF="#" onmouseOver="lightUp();" onmouseOut="dimDown();">
<IMG SRC="button_dim.gif" name="homeButton" width=100 height=50
    border=0> </A>
</BODY>
</HTML>
```

© Sidi Mahmoud Kaber

L'image invisible

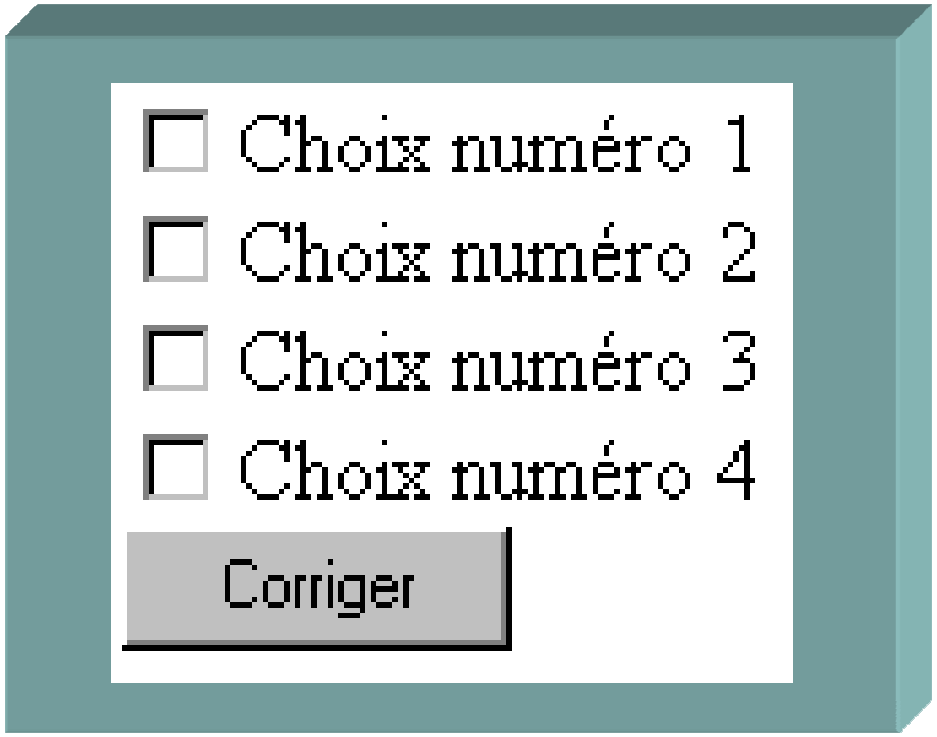
- Comment rendre l’image invisible?
- On peut prévoir une image invisible de la même couleur que l'arrière plan (même transparente).
- On la place sur le chemin de la souris de l'utilisateur et son survol peut ,masquer l’image initialement existante.

© Sidi Mahmoud Kaber

JAVASCRIPT

Les Formulaires ... Suite

Exercice



Ecrire un programme qui permet d'afficher quatre cases a cocher et de tester le bon choix !!!

© Sidi Mahmoud Kaber

Correction

```
<HTML>
<HEAD>
<script language="javascript">
function reponse(form4) {
if ( (form4.check1.checked) == true && (form4.check2.checked) == true && (form4.check3.checked)
      == false && (form4.check4.checked) == true)
{ alert("C'est la bonne réponse! ") }
else
{alert("Désolé, continuez à chercher.")}
}
</SCRIPT>
</HEAD>
<BODY>
Entrez votre choix :
<FORM NAME="form4">
<INPUT TYPE="CHECKBOX" NAME="check1" VALUE="1">Choix numéro 1<BR>
<INPUT TYPE="CHECKBOX" NAME="check2" VALUE="2">Choix numéro 2<BR>
<INPUT TYPE="CHECKBOX" NAME="check3" VALUE="3">Choix numéro 3<BR>
<INPUT TYPE="CHECKBOX" NAME="check4" VALUE="4">Choix numéro 4<BR>
<INPUT TYPE="button"NAME="but" VALUE="Corriger" onClick="reponse(form4)">
</FORM>
</BODY>
</HTML>
```

© Sidi Mahmoud Kaber

```
Entrez votre choix :
<FORM NAME="form4">
<INPUT TYPE="CHECKBOX" NAME="check1" VALUE="1">Choix
   numéro 1<BR>
<INPUT TYPE="CHECKBOX" NAME="check2" VALUE="2">Choix
   numéro 2<BR>
<INPUT TYPE="CHECKBOX" NAME="check3" VALUE="3">Choix
   numéro 3<BR>
<INPUT TYPE="CHECKBOX" NAME="check4" VALUE="4">Choix
   numéro 4<BR>
<INPUT TYPE="button"NAME="but" VALUE="Corriger"
   onClick="reponse()">
</FORM>
```

© Sidi Mahmoud Kaber

```
<script language="javascript">
function reponse()
{
if ( (form4.check1.checked) == true &&
    (form4.check2.checked) == true &&
    (form4.check3.checked) == false &&
    (form4.check4.checked) == true)
{ alert("C'est la bonne réponse! ") }
else
{alert("Désolé, continuez à chercher.")}
}
</SCRIPT>
```

© Sidi Mahmoud Kaber

Propriétés

Propriété	Description
name	indique le nom du contrôle. Toutes les cases à cocher portent un nom différent.
checked	indique l'état en cours de l'élément case à cocher.
defaultchecked	indique l'état du bouton sélectionné par défaut.
value	indique la valeur de l'élément case à cocher.

© Sidi Mahmoud Kaber

Exercice

© Sidi Mahmoud Kaber

Correction

```
<HTML>
<HEAD>
<script language="javascript"> function liste(form5) {
alert("L\'élément " + (form5.list.selectedIndex + 1)); }
</SCRIPT>
</HEAD>
<BODY>
Entrez votre choix :
<FORM NAME="form5">
<SELECT NAME="list">
<OPTION VALUE="1">Elément 1
<OPTION VALUE="2">Elément 2
<OPTION VALUE="3">Elément 3
</SELECT>
<INPUT TYPE="button"NAME="b" VALUE="Quel est l'élément retenu?"
    onClick="liste(form5)"> </FORM>
</BODY>
</HTML>
```

© Sidi Mahmoud Kaber

Entrez votre choix :

```
<FORM NAME="form5">
<SELECT NAME="list">
<OPTION VALUE="1">Elément 1
<OPTION VALUE="2">Elément 2
<OPTION VALUE="3">Elément 3
</SELECT>
<INPUT TYPE="button"NAME="b" VALUE="Quel est
   l'élément retenu?" onClick=" reponse2()">
</FORM>
```

© Sidi Mahmoud Kaber

```
<script language="javascript">
 function reponse2()
 {
AA= form5.list.selectedIndex + 1
alert("L\'élément " + AA);
}
</SCRIPT>
```

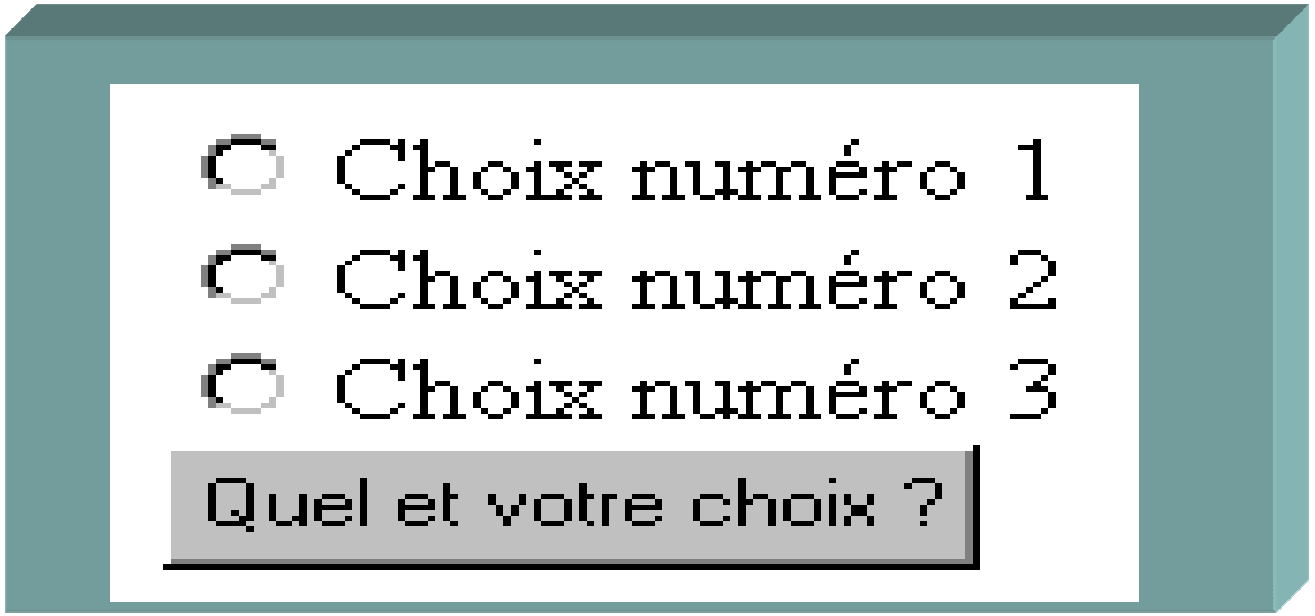
© Sidi Mahmoud Kaber

Propriétés

Propriété	Description
name	indique le nom de la liste déroulante.
length	indique le nombre d'éléments de la liste. S'il est indiqué dans le tag <SELECT>, tous les éléments de la liste seront affichés. Si vous ne l'indiquez pas un seul apparaîtra dans la boite de la liste déroulante.
selectedIndex	indique le rang à partir de 0 de l'élément de la liste qui a été sélectionné par l'utilisateur.
defaultselected	indique l'élément de la liste sélectionné par défaut. C'est lui qui apparaît alors dans la petite boite.

© Sidi Mahmoud Kaber

Exercice



<INPUT TYPE="radio" NAME="choix" VALUE="1">Choix numéro 1

© Sidi Mahmoud Kaber

Correction

```
<HTML>
<HEAD>
<SCRIPT language="javascript">
function Choisir(form3) {
if (form3.choix[0].checked) { alert("Vous avez choisi la proposition " + form3.choix[0].value) };
if (form3.choix[1].checked) { alert("Vous avez choisi la proposition " + form3.choix[1].value) };
if (form3.choix[2].checked) { alert("Vous avez choisi la proposition " + form3.choix[2].value) };
}
</SCRIPT>
</HEAD>
<BODY>
Entrez votre choix :
<FORM NAME="form3">
<INPUT TYPE="radio" NAME="choix" VALUE="1">Choix numéro 1<BR>
<INPUT TYPE="radio" NAME="choix" VALUE="2">Choix numéro 2<BR>
<INPUT TYPE="radio" NAME="choix" VALUE="3">Choix numéro 3<BR>
<INPUT TYPE="button"NAME="but" VALUE="Quel et votre choix ?" onClick=«
     Choisir(form3)">
</FORM>
</BODY>
</HTML>
```

© Sidi Mahmoud Kaber

```
<FORM NAME="form3">
<INPUT TYPE="radio" NAME="choix" VALUE="1">Choix
   numéro 1<BR>
<INPUT TYPE="radio" NAME="choix" VALUE="2">Choix
   numéro 2<BR>
<INPUT TYPE="radio" NAME="choix" VALUE="3">Choix
   numéro 3<BR>
<INPUT TYPE="button"NAME="but" VALUE="Quel et votre
   choix ?" onClick=« Choisir(form3)">
</FORM>
```

© Sidi Mahmoud Kaber

```
<SCRIPT language="javascript">
function Choisir()
{
if (form3.choix[0].checked) { alert("La reponse est
   correcte ") };
if (form3.choix[1].checked) { alert("La reponse est
   fausse ") };
if (form3.choix[2].checked) { alert("La reponse est
   fausse ") };
}
</SCRIPT>
```

© Sidi Mahmoud Kaber

Propriétés

Propriété	Description
name	indique le nom du contrôle. Tous les boutons portent le même nom.
index	l'index ou le rang du bouton radio en commençant par 0.
checked	indique l'état en cours de l'élément radio
defaultchecked	indique l'état du bouton sélectionné par défaut.
value	indique la valeur de l'élément radio.

© Sidi Mahmoud Kaber

Une minuterie

- Javascript met à votre disposition une minuterie (ou plus précisément un compteur à rebours) qui permettra de déclencher une fonction après un laps de temps déterminé.
- La syntaxe de mise en route du temporisateur est :
- nom_du_compteur = setTimeout("fonction_appelée()", temps en milliseconde)
- Ainsi, setTimeout("demarrer()",5000) va lancer la fonction demarer() après 5 secondes.
- Pour arrêter le temporisateur avant l'expiration du délai fixé, il y a :
- clearTimeout(nom_du_compteur)

© Sidi Mahmoud Kaber

```
<HTML>
<BODY onLoad='compt=setTimeout("self.close();",4000)'>
<H1>Ceci est un test</H1>
<FORM>
<INPUT TYPE="button" value=" Continuer "
   onClick="clearTimeout(compt);self.close();">
<INPUT TYPE="button" value=" initialiser " onClick='
   clearTimeout(compt);
   compt=setTimeout("self.close();",4000)'>
</FORM>
</BODY>
</HTML>
```

© Sidi Mahmoud Kaber

Les messages d'erreur

- **Les types d'erreurs.**
- Il y a 3 grandes catégories d'erreurs dans l'utilisation d'un programme Javascript :
  - les erreurs au chargement.
  - les erreurs à l'exécution.
  - les erreurs de logique

© Sidi Mahmoud Kaber

Les erreurs au chargement

- Au chargement du script par le browser, Javascript passe en revue les différentes erreurs qui peuvent empêcher le bon déroulement de celui-ci.

- Les erreurs au chargement, nombreuses lors de l'apprentissage de Javascript, sont souvent dues à des fautes de frappe et/ou des erreurs de syntaxe.

- Pour vous aider à déterminer l'erreur, Javascript affiche sa fameuse boite de message d'erreur, vous indique le problème et le texte de l'erreur. Ne perdez pas de vue que Javascript ne vous indique pas toujours l'erreur véritable et que selon l'erreur, celle-ci peut se situer bien plus avant dans le script. Des exemples classiques d'erreurs au chargement sont des parenthèses ou des accolades non fermées, des guillemets manquants, etc.

© Sidi Mahmoud Kaber

Les erreurs à l'exécution

- Ici votre script se charge sans problème, mais cette satanée boite de message d'erreurs apparaît lorsque l'exécution du script est demandée.

- Alors que les erreurs au chargement étaient surtout dues au mauvais usage de la syntaxe, les erreurs à l'exécution proviennent d'un mauvais usage des commandes ou des objets Javascript.

- Un exemple d'erreur à l'exécution est un appel erroné à une variable ou une fonction inexistante (car il y a, par exemple, une erreur dans le nom de la variable ou de la fonction).

© Sidi Mahmoud Kaber

Les erreurs de logique

- Ce sont les plus vicieuses car le "débuggeur" de Javascript ne signale bien entendu aucune erreur et votre script se déroule correctement. Hélas, à l'arrivée, le résultat ne correspond pas à celui espéré.
- Il n'y a plus qu'à revoir la construction logique de votre script.
- De nombreuses erreurs de logique sont dues à des valeurs de variables incorrectes.
- Voici quelques conseils :
- Dans le cas où l'utilisateur doit entrer une valeur, celle-ci était-t-elle au bon format?
- N'est-il pas utile de prévoir un petit script pour vérifier le format d'entrée ?
- On peut ajouter des points de contrôle de valeur de variable ou de passage avec l'instruction alert(variable) ou alerte("Point de passage1").

© Sidi Mahmoud Kaber

Les grands classiques des erreurs: Conseils

- Soyez vigilant au nom des variables (case sensitive). Mavariable et mavariable sont deux variables distinctes. Eviter d'utiliser des noms de variables trop rapprochants.

- Le nom de la fonction a-t-il bien la même orthographe dans la déclaration et dans l'appel. Le nom des fonctions est-il bien unique dans le script?

- N'oubliez pas les guillemets ou apostrophes avant et après les chaînes de caractères. •Avez-vous bien mis des virgules entre vos différents paramètres ou arguments?

© Sidi Mahmoud Kaber

Les grands classiques des erreurs: Conseils

- Avez-vous placé vos accolades au bon endroit sans avoir oublié de les fermer (surtout dans le cas de blocs de commandes imbriquées).

- Assurez-vous que les noms des objets Javascript sont corrects. Le piège est que les objets Javascript commencent par une majuscule (Date, Math, Array...) mais que les propriétés commencent par une minuscule (alert).

- La confusion entre = opérateur d'affectation et == opérateur de comparaison

© Sidi Mahmoud Kaber

Problème: le script et le tableau

- On recommande dans la littérature de ne pas placer de tag <SCRIPT> dans des balises <TD> .
- Mais plutôt de commencer le tag <SCRIPT> avant le tag <TD> et d'écrire le tag <TD> jusqu'au tag </TD> par une écriture document.write. Et même, pour être sur de son coup, d'écrire tout la tableau avec le document.write

© Sidi Mahmoud Kaber

Problème: le script et le tableau

```
<SCRIPT LANGUAGE="Javascript">
<!--
document.write("<TABLE BORDER=1>");
document.write("<TR>");
document.write("<TD>");
document.write("votre texte");
document.write("</TD>");
document.write("<TD>");
document.write("votre texte");
document.write("</TD>");
document.write("</TR>");
document.write("</TABLE>");
//-->
</SCRIPT>
```

© Sidi Mahmoud Kaber

Adapter le script selon le browser du lecteur

- Avec les méthodes et propriétés de l'objet navigator (voir ce chapitre), il y aura moyen de détecter le type et la version du browser. Ce qui sera très utile pour adapter vos scripts au browser et à la version de celui-ci.

© Sidi Mahmoud Kaber

Adapter le script selon le browser du lecteur

```
<SCRIPT LANGUAGE = "JavaScript">
<!--
var name = navigator.appName ;
if (name == 'Microsoft Internet Explorer') {
document.write('Attention! Vous utilisez Microsoft Explorer
    3.0.') <BR>');
document.write('Avec ce browser, certains scripts peuvent
    ne pas fonctionner correctement');
}
else { null }
//-->
</SCRIPT>
```

© Sidi Mahmoud Kaber

JAVASCRIPT

L'objet window

Utilisation de la barre d'état

- Avec Javascript, la barre d'état peut être utilisée pour afficher des messages de votre choix.

Propriété	Description
status	valeur du texte affiché dans la barre d'état de la fenêtre.
DefaultStatus	valeur par défaut qui s'affiche dans la barre d'état.

© Sidi Mahmoud Kaber

Exemple ...

- Généralement, cet événement est mis en oeuvre par un onmouseOver() sur un lien hypertexte.
- En voici un exemple :

```
<HTML>
<BODY>
<A HREF="#" onmouseover="self.status='Votre texte'; return
    true;"> A voir ici </A>
</BODY>
</HTML>
```

Il est indispensable d'ajouter return true;

© Sidi Mahmoud Kaber

Ouverture et fermeture de fenêtres

Méthodes	Description
open()	ouvre une nouvelle fenêtre.
close()	ferme la fenêtre en cours.

La syntaxe est :

[window.]open("URL","nom_de_la_fenêtre","caractéristiques_de_la fenêtre")

© Sidi Mahmoud Kaber

Caractéristiques de la fenêtre

Caractéristique	Description
toolbar=yes ou no	Affichage de la barre d'outils
location=yes ou non	Affichage de champ d'adresse (ou de localisation)
directories=yes ou no	Affichage des boutons d'accès rapide
status=yes ou no	Affichage de la barre d'état
menubar=yes ou no	Affichage de la barre de menus
scrollbars=yes ou no	Affichage des barres de défilement. (scrollbars=no fonctionne mal sous Explorer 3.0)
resizable=yes ou no	Dimensions de la fenêtre modifiables
width=x en pixels	Largeur de la fenêtre en pixels
height=y en pixels	Hauteur de la fenêtre en pixels

© Sidi Mahmoud Kaber

Exemple ... fermer une fenêtre

Fichier test.htm :

```
<HTML>
<BODY>
<H1>Ceci est un test<HI>
<FORM>
<INPUT TYPE="button" value= " Continuer " onClick="self.close()">
</FORM>
</BODY>
</HTML>
```

où self.close() fermera la fenêtre courante, c.-à-d. la nouvelle fenêtre.

© Sidi Mahmoud Kaber

```
<FORM>
<INPUT TYPE ="button" value="Ouvrir une nouvelle
   fenêtre"
onClick="open('test.htm', 'new', 'width=300,
height=150,
toolbar=no,
location=no,
directories=no,
status=no,
menubar=no,
scrollbars=no,
resizable=no')">
</FORM>
```

© Sidi Mahmoud Kaber

Exemple: Ouverture de fenêtre

Dans la page de départ :

```
<SCRIPT LANGUAGE="javascript">
<!--
function new_window() {
xyz="open('test.htm', 'new',
     'width=300,height=150,toolbar=no,location=no,
directories=no,status=no,menubar=no,scrollbars=no,resizable=no')">
// sans espaces ni passage à la ligne
}
// -->
</SCRIPT>
<FORM>
<INPUT TYPE ="button" value="Ouvrir une nouvelle fenêtre"
onClick="new_window()">
</FORM>
```

© Sidi Mahmoud Kaber

Exemple :Fermeture automatique après x secondes

La page test.htm devient testc.htm

```
<HTML>
<BODY onLoad='compt=setTimeout("self.close();",4000)'>
<H1>Ceci est un test</H1>
<FORM>
<INPUT TYPE="button" value=" Continuer "
     onClick="clearTimeout(compt);self.close();">
</FORM>
</BODY>
</HTML>
```

Dans la page de départ :

```
<FORM>
<INPUT TYPE ="button" value="Ouvrir une nouvelle fenêtre"
onClick="open('testc.htm', 'new', 'width=300,height=150,toolbar=no,location=no,
directories=no,status=no,menubar=no,scrollbars=no,resizable=no')">
(sans espaces ni passage à la ligne)
</FORM>
```

© Sidi Mahmoud Kaber

JAVASCRIPT

L'objet String

L'objet String

- la manipulation des caractères si utile pour l'aspect programmation de Javascript.

Instruction	Description
length	C'est un entier qui indique la longueur de la chaîne de caractères.
charAt()	Méthode qui permet d'accéder à un caractère isolé d'une chaîne.
indexOf()	Méthode qui renvoie la position d'une chaîne partielle à partir d'une position déterminée (en commençant au début de la chaîne pricipale soit en position 0).

© Sidi Mahmoud Kaber

L'objet String

Instruction	Description
LastIndexOf()	Méthode qui renvoie la position d'une chaîne partielle à partir d'une position déterminée (en commençant à la fin soit en position length moins 1).
substring(x,y)	Méthode qui renvoie un string partiel situé entre l position x et la position y-1.
toLowerCase()	Transforme toutes les lettres en minuscules.
toUpperCase()	Transforme toutes les lettres en Majuscules.

© Sidi Mahmoud Kaber

La propriété length

- un entier qui indique le nombre d'éléments dans une chaîne de caractères
- La syntaxe est simple :

x=variable.length;
x=("chaîne de caractères").length;

- La propriété length ne sert pas que pour les Strings, mais aussi pour connaître la longueur ou le nombre d'éléments :
  - de formulaires . Combien a-t-il de formulaires différents ?
  - de boutons radio. Combien a-t-il de boutons radio dans un groupe ?
  - de cases à cocher. Combien a-t-il de cases à cocher dans un groupe ?
  - d'options. Combien a-t-il d'options dans un Select ?
  - de frames. Combien a-t-il de frames "enfants" ?
  - d'ancres, de liens, etc.

© Sidi Mahmoud Kaber

La méthode CharAt()

var str="Javascript";
var chr=str.charAt(0);
var chr="Javascript".charAt(0);
ou var chr=charAt(str,0);
ou var chr=charAt("Javascript",0);
	La réponse est "J".

var str="Javascript";
var chr=str.charAt(9);
var chr=charAt(str,9);
	La réponse est "t".

var str="Javascript"; var chr=charAt(str,13);	La réponse est "" soit vide.

© Sidi Mahmoud Kaber

La méthode indexOf()

- Cette méthode renvoie la position, soit x, d'un string partiel (lettre unique, groupe de lettres ou mot) dans une chaîne de caractères en commençant à la position indiquée par y. Cela vous permet, par exemple, de voir si une lettre, un groupe de lettres ou un mot existe dans une phrase.

```
variable="chaîne_de_caractères";
var="string_partiel";
x=variable.indexOf(var,y);
```

- Si le string partiel n'est pas trouvé dans la chaîne de caractères à analyser, la valeur retournée sera égale à -1.

© Sidi Mahmoud Kaber

La méthode indexOf()

- variable="Javascript"
- var="script"
- x=variable.indexOf(var,0); x vaut 4

- variable="VanlanckerLuc&ccim.be"
- var="@"
- x=variable.indexOf(var); x vaut -1

© Sidi Mahmoud Kaber

La méthode lastIndexOf()

- Méthode fort semblable à indexOf() sauf que la recherche va cette fois de droite à gauche (en commençant donc par la fin).

x=variable.lastIndexOf(var,y);

variable="Javascript"

var="a"

x=variable.indexOf(var,0); ici x vaut 1 soit la position du premier a.

x=variable.lastIndexOf(var,9); ici x vaut 3 soit la position du second a.

© Sidi Mahmoud Kaber

La méthode substring()

- Elle sera particulièrement utile, par exemple, pour prendre différentes données dans une longue chaîne de caractères.

variable = "chaîne de caractères"

resultat=variable.substring(x,y)

- où resultat est un sous ensemble de la chaîne de caractère (ou de la variable).
- str="Javascript";
- str1=str.substring(0,4);
- Resultat :str1="Java"; soit les positions 0,1,2 et 3.

© Sidi Mahmoud Kaber

La méthode toLowerCase()

- Cette méthode affiche toutes les majuscules d'une chaîne de caractères variable2 en minuscules.
- variable2="chaîne de caractères";
- variable1=variable2.toLowerCase();
- Exemple :
- str="JavaScript";
- str1=str.toLowerCase();
- Le résultat sera :str1="javascript";

© Sidi Mahmoud Kaber

La méthode toUpperCase()

- Cette méthode affiche toutes les minuscules d'une chaîne de caractères variable2 en majuscules.
- variable2="chaîne de caractères";
- variable1=variable2.toUpperCase();
- Exemple :
- str="JavaScript";
- str3=str.toUpperCase();
- Le résultat sera : str3="JAVASCRIPT";

© Sidi Mahmoud Kaber

Utilité de toLowerCase() et de toUppercase()

- L'utilité de ces 2 méthodes ne vous saute peut être pas aux yeux. Et pourtant, il faut se rappeler que Javascript est case sensitive. Ainsi une recherche sur Euro ne donnera pas le même résultat que sur euro ou EUro.
- Ainsi, pour les bases de données, il est prudent de tout convertir en minuscules (ou en majuscules). A fortiori, pour certaines informations des utilisateurs introduites par le biais d'un formulaire

© Sidi Mahmoud Kaber

JAVASCRIPT

L'objet Math

Pour manipuler les nombres

abs()

x=Math.abs(y);

La méthode abs() renvoie la valeur absolue (valeur positive) de y. Il supprime en quelque sorte le signe négatif d'un nombre.

y = 4;

x = math.abs(y);

x = Math.abs(4);

x = math.abs(-4);

ont comme résultat x = 4

© Sidi Mahmoud Kaber

ceil()

- x=Math.ceil(y);
- La méthode ceil() renvoie l'entier supérieur ou égal à y.
- Attention ! Cette fonction n'arrondit pas le nombre.
- Comme montré dans l'exemple, si y = 1.01, la valeur de x sera mise à 2.
- y=1.01;
- x=Math.ceil(y);
- a comme résultat 2.

© Sidi Mahmoud Kaber

floor()

- x=Math.floor(y);
- La méthode floor() renvoie l'entier inférieur ou égal à y.
- Attention ! Cette fonction n'arrondit pas le nombre.
- Comme montré dans l'exemple, si y = 1.99, la valeur de x sera mise à 1.
- y=1.999;
- x=Math.floor(y);
- a comme résultat 1.

© Sidi Mahmoud Kaber

round()

- La méthode round() arrondit le nombre à l'entier le plus proche.
- y=20.355;
- x=Math.round(y);
- a comme résultat
- x=20;
- Attention ! Certains calculs réclament une plus grande précision. Pour avoir deux décimales après la virgule, on utilisera la formule :
- x=(Math.round(y*100))/100;
- et dans ce cas
- x=20.36;

© Sidi Mahmoud Kaber

max()

- x=Math.max(y,z);
- La méthode max(y,z) renvoie le plus grand des 2 nombres y et z.
- y=20; z=10;
- x=Math.max(y,z);
- a comme résultat
- x=20;

© Sidi Mahmoud Kaber

min()

- x=Math.min(y,z);
- La méthode min(y,z) renvoie le plus petit des 2 nombres y et z.
- y=20; z=10;
- x=Math.min(y,z);
- a comme résultat
- x=10;

© Sidi Mahmoud Kaber

pow()

- x=Math.pow(y,z);
- La méthode pow() calcule la valeur d'un nombre y à la puissance z.
- y=2; z=8
- x=Math.pow(y,z);
- a comme résultat
- 28 soit 256

© Sidi Mahmoud Kaber

random()

- x=Math.random();
- La méthode random() renvoie la valeur d'un nombre aléatoire choisi entre 0 et 1.
- Attention ! Cela ne fonctionne que sous Unix.
- A la place, on peut utiliser la fonction suivante qui renvoie un nombre "aléatoire" entre 0 et 9.
- function srand(){ t=new Date(); r=t.getTime(); p="a"+r; p=p.charAt((p.length-4)); x=p; }

© Sidi Mahmoud Kaber

sqrt()

- x=Math.sqrt(y);
- La méthode sqrt() renvoie la racine carrée de y.
- y=25;
- x=Math.sqrt(y);
- a comme résultat
- x=5;

© Sidi Mahmoud Kaber

parseFloat()

- x=parseFloat("variable");
- Cette fonction convertit une chaîne contenant un nombre en une valeur à virgule flottante. Ou si vous préférez, elle retourne les chiffres derrière la virgule d'un nombre.
- Attention ! Le résultat risque d'être surprenant si Javascript rencontre autre chose dans la chaîne que des nombres, les signes + et -, le point décimal ou un exposant. S'il trouve un caractère "étranger", La fonction ne prendra en compte que les caractères avant le caractère "étranger".
- Si le premier caractère n'est pas un caractère admis, x sera égal à 0 sous Windows et à "NaN" sur les autres systèmes.
- str='-.12345';
- str1='$5.50';
- x=parseFloat(str);
- aura comme résultat
- x= -.12345;
- x=0 ou "NaN";

© Sidi Mahmoud Kaber

parseInt()

- x=parseInt(variable);
- Retourne la partie entière d'un nombre avec une virgule.
- str='1.2345';
- x=parseInt(str);
- x=1;

© Sidi Mahmoud Kaber

eval()

- x=eval(variable);
- Cette fonction évalue une chaîne de caractère sous forme de valeur numérique. On peut stocker dans la chaîne des opérations numériques, des opérations de comparaison, des instructions et même des fonctions.
- str='5 + 10';
- x=eval(str);
- a comme résultat
- x=15;

© Sidi Mahmoud Kaber

Les fonctions trigonométriques

- Voici (sans commentaires) ces différentes fonctions :
- x=Math.PI;
- x=Math.sin(y);
- x=Math.asin(y);
- x=Math.cos(y);
- x=Math.acos(y);
- x=Math.tan(y);
- x=Math.atan(y);

© Sidi Mahmoud Kaber

Les fonctions logarithmiques

- Pour les initiés :
- x=Math.exp(y);
- x=Math.log(y);
- x=Math.LN2;
- x=Math.LN10;
- x=Math.E;
- x=Math.LOG2E;
- x=Math.LOG10E;

© Sidi Mahmoud Kaber

JAVASCRIPT

L'objet Date

new Date()

- Cette méthode renvoie toutes les informations "date et heure" de l'ordinateur de l'utilisateur.
- variable=new Date();
- Ces informations sont enregistrées par Javascript sous le format :
- "Fri Dec 17 09:23:30 1998"
- Attention ! La date et l'heure dans Javascript commence au 1e janvier 1970. Toute référence à une date antérieure donnera un résultat aléatoire.
- La méthode new date () sans arguments renvoie la date et l'heure courante.
- Pour introduire une date et une heure déterminée, cela se fera sous la forme suivante :
- variable=new Date("Jan 1, 2000 00:00:00");
- Toutes les méthodes suivantes vous faciliteront la tâche pour accéder à un point précis de cette variable (en fait un string) et pour modifier si besoin en est le format d'affichage.

© Sidi Mahmoud Kaber

getYear()

- variable_date=new Date();
- an=variable_date.getYear();
- Retourne les deux derniers chiffres de l'année dans variable_date. Soit ici 03.
- Comme vous n'avez que deux chiffres, il faudra mettre 20 en préfixe soit
- an=“20"+variable_date.getYear();

© Sidi Mahmoud Kaber

getMonth()

- variable_date=new Date();
- mois=variable_date.getMonth();
- Retourne le mois dans variable_date sous forme d'un entier compris entre 0 et 11 (0 pour janvier, 1 pour février, 2 pour mars, etc.). Soit ici 11 (le mois moins 1).

© Sidi Mahmoud Kaber

getDate()

- variable_date=new Date();
- jourm=variable_date.getDate();
- Retourne le jour du mois dans variable_date sous forme d'un entier compris entre 1 et 31.
- Eh oui, ici on commence à 1 au lieu de 0 (pourquoi???).
- A ne pas confondre avec getDay() qui retourne le jour de la semaine

© Sidi Mahmoud Kaber

getDay() / getHours()

- **getDays();**
- variable_date=new Date();
- jours=variable_date.getDay();
- Retourne le jour de la semaine dans variable_date sous forme d'un entier compris entre 0 et 6 (0 pour dimanche, 1 pour lundi, 2 pour mardi, etc.).
- **getHours();**
- variable_date=new Date();
- hrs=variable_date.getHours();
- Retourne l'heure dans variable_date sous forme d'un entier compris entre 0 et 23.

© Sidi Mahmoud Kaber

getMinutes() / getSeconds()

getMinutes();

- variable_date=new Date();
- min=variable_date.getMinutes();
- Retourne les minutes dans variable_date sous forme d'un entier compris entre 0 et 59.

getSeconds();

- variable_date=new Date();
- sec=variable_date.getSeconds();
- Retourne les secondes dans variable_date sous forme d'un entier compris entre 0 et 59.

© Sidi Mahmoud Kaber

Exemple : Un script qui donne simplement l'heure

```
<HTML>
<HEAD>
<SCRIPT LANGUAGE="Javascript">
function getDt(){
dt=new Date();
cal=""+ dt.getDate()+"/"
+(dt.getMonth()+1) + "/20" +dt1.getYear();
hrs=dt.getHours();
min=dt.getMinutes();
sec=dt.getSeconds();
tm=" "+((hrs<10)?"0":"") +hrs+":";
tm+=((min<10)?"0":"")+min+":";
tm+=((sec<10)?"0":"")+sec+" ";
document.write(cal+tm);
}
</SCRIPT>
</HEAD>
<BODY >
<SCRIPT LANGUAGE="Javascript">
getDt();
</SCRIPT>
</BODY>
</HTML>
```

© Sidi Mahmoud Kaber

Exemple 2 : Actualiser l'heure

```
<HTML>
<HEAD>
<SCRIPT LANGUAGE="Javascript">
function getDt(){
dt=new Date();
hrs=dt.getHours();
min=dt.getMinutes();
sec=dt.getSeconds();
tm=" "+((hrs<10)?"0":"") +hrs+":";
tm+=((min<10)?"0":"")+min+":";
tm+=((sec<10)?"0":"")+sec+" ";
document.horloge.display.value=tm;
setTimeout("getDt()",1000);
}
</SCRIPT>
```

```
</HEAD>
<BODY onLoad="getDt()">
<FORM name="horloge">
<INPUT TYPE="text" NAME="display"
  SIZE=15 VALUE ="">
</FORM>
</BODY>
</HTML>
```

© Sidi Mahmoud Kaber

JAVASCRIPT

L'objet Navigator

L'objet Navigator

- Avec l'objet Navigator, on aura la possibilité d'identifier le browser (ainsi que la version de celui-ci) utilisé par le lecteur.
- Ceci est très utile sinon indispensable pour assurer la compatibilité des pages.
- Les propriétés sont peu nombreuses mais intéressantes mais parfois un peu obscures.

© Sidi Mahmoud Kaber

navigator.appCodeName

- Retourne le nom de code du navigateur.

- document.write("Le code name de votre browser est " +navigator.appCodeName);

© Sidi Mahmoud Kaber

navigator.appName

- Retourne le nom ou la marque du browser soit "Netscape", soit "Microsoft Internet Explorer"
- Cette propriété sera plus utile pour faire la différence entre la famille Netscape et la famille Microsoft du browser.
- document.write("Le nom ou la marque du browser est " +navigator.appName);

© Sidi Mahmoud Kaber

navigator.appVersion

- Renvoie des informations consernant la version du navigateur, le système d'exploitation de l'utilisateur, un code de nationalité de la version (avec des variantes).

- Cette information prend la forme :
- 2.0 (Win95; I)
- releaseNumber(platform,country)

© Sidi Mahmoud Kaber

navigator.userAgent

- Renvoie également des informations (sur le header envoyé dans le protocole HTTPd du server de votre visiteur).

- document.write("Le browser a comme user-agent name "+navigator.userAgent);

© Sidi Mahmoud Kaber

Exemple

```
<HTML>
<BODY>
<SCRIPT LANGUAGE="javascript">
document.write("Le code name de votre browser est " +navigator.appCodeName);
document.write("Le nom ou la marque du browser est " +navigator.appName);
document.write("Les informations sur la version sont "+navigator.appVersion);
document.write("Le browser a comme user-agent name "+navigator.userAgent);
</SCRIPT>
</BODY>
</HTML>
```

Résultat:

Le code name de votre browser est Mozilla
Le nom ou la marque du browser est Netscape
Les informations sur la version sont 2.0 (Win95; I)
Le browser a comme user-agent name Mozilla/2.0 (Win95; I)

© Sidi Mahmoud Kaber

Projet

Créer un QCM pour un cours JavaScript avec JavaScript.

© Sidi Mahmoud Kaber