

## CHAPITRE : ECRITURE D'UN TEXTE

A chaque objet Javascript, le concepteur du langage a prévu un ensemble de méthodes (ou fonctions dédiées à cet objet) qui lui sont propres. A l'objet document, Javascript a dédié la méthode write(). On notera

```
document.write();
```

### 1. LA METHODE WRITE()

La syntaxe est assez simple soit

```
write("votre texte");
resultat= "text2" ;
write(resultat);
write("Le résultat est " + resultat);
write("<B>Le résultat est</B>" + resultat); ou
write ("<B>" + "Le résultat est " + "</B>" + resultat),
```

### 2. L'ECRITURE EN JAVASCRIPT...

**variable.bold();** L'emploi de .bold() affichera la variable comme si elle était comprise entre les balises Html <B></B>.

#### Exemple :

Les quatre instructions Javascript suivantes sont équivalentes :

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

**variable.italics();** L'emploi de .italics() affichera la variable comme si elle était comprise entre les balises Html <I></I>.

**variable.big();** L'emploi de .big() affichera la variable comme si elle était comprise entre les balises Html <BIG></BIG>.

**variable.small();** L'emploi de .small() affichera la variable comme si elle était comprise entre les balises Html <SMALL> </SMALL>.

**variable.fixed();** L'emploi de .fixed() affichera la variable comme si elle était comprise entre les balises Html <TT></TT>.

**variable.fontcolor(color);** L'emploi de .fontcolor(color) affichera la variable comme si elle était comprise entre les balises Html <FONT COLOR="color"> </FONT>.

#### Exemple :

Les quatre instructions Javascript suivantes sont équivalentes :

```
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"));
```

**variable.fontSize(x);** L'emploi de .fontSize(x) affichera la variable comme si elle était comprise entre les balises Html <FONT SIZE="x"></FONT> où x est un nombre de 1 à 7 ou exprimé en plus ou en moins par rapport à 0 par exemple -2, -1, +1, +2.

**variable.strike();** L'emploi de .strike() affichera la variable comme si elle était comprise entre les balises Html<STRIKE></STRIKE>.

**variable.sub();** L'emploi de .sub() affichera la variable comme si elle était comprise entre les balises Html <SUB> </SUB>.

**variable.sup();** L'emploi de .sup() affichera la variable comme si elle était comprise entre les balises Html <SUP></SUP>.

## **CHAPITRE : LES OPERATEURS (COMME DANS LANGAGE C)**

## LES EVENEMENTS

Les événements Javascript, associés aux fonctions, aux méthodes et aux formulaires, ouvrent grand la porte pour une réelle *interactivité* de vos pages.

Passons en revue différents événements implémentés en Javascript.

Description	Événement
Lorsque l'utilisateur clique sur un bouton, un lien ou tout autre élément.	Click
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.	MouseOut
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

### GESTIONNAIRES D'EVENEMENT DISPONIBLES EN JAVASCRIPT

Il nous semble utile dans cette partie "avancée" de présenter la liste des objets auxquels correspondent des gestionnaires d'événement bien déterminés.

Objets	Gestionnaires d'événement disponibles
Fenêtre	onLoad, onUnload
Lien hypertexte	onClick, onMouseOver, onMouseOut
Élément de texte	onBlur, onChange, onFocus, onSelect
Élément de zone de texte	onBlur, onChange, onFocus, onSelect
Élément bouton	onClick
Case à cocher	onClick
Bouton Radio	onClick
Liste de sélection	Blur, onChange, onFocus
Bouton Submit	onClick
Bouton Reset	onClick

### EXEMPLE : CHANGEMENT D'IMAGES

Avec le gestionnaire d'événement `onmouseover`, on peut prévoir qu'après le survol d'un image par l'utilisateur, une autre image apparaisse (pour autant qu'elle soit de la même taille).

Le code est relativement simple.

```
<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>
```

Compléter toujours en Javascript les attributs `width=x height=y` de vos images.

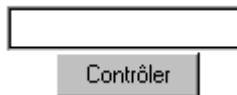
## CHAPITRE: LES FORMULAIRES

Propriété	Description
Name	indique le nom du contrôle par lequel on pourra accéder.
Defaultvalue	indique la valeur par défaut qui sera affichée dans la zone de texte.
Value	indique la valeur en cours de la zone de texte. Soit celle tapée par l'utilisateur ou si celui-ci n'a rien tapé, la valeur par défaut.

### Lire une valeur dans une zone de texte

Voici un exemple que nous détaillerons :

Voici une zone de texte. Entrez une valeur et appuyer sur le bouton pour contrôler celle-ci.

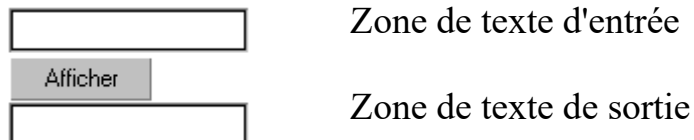


Le script complet est celui-ci :

```
<HTML>
<HEAD>
<SCRIPT LANGUAGE="javascript">
function controle(form1) {
var test = document.form1.input.value;
alert("Vous avez tapé : " + test);
}
</SCRIPT>
</HEAD>
<BODY>
<FORM NAME="form1">
<INPUT TYPE="text" NAME="input" VALUE=""><BR>
<INPUT TYPE="button" NAME="bouton" VALUE="Contrôler" onClick="controle(form1)">
</FORM>
</BODY>
</HTML>
```

### Exercice : Ecrire une valeur dans une zone de texte

Ecrire Le code JavaScript permettant d'entrer une valeur quelconque dans la zone de texte d'entrée. Appuyer sur le bouton pour afficher cette valeur dans la zone de texte de sortie.



### LES BOUTONS RADIO

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.

Prenons un exemple :

```
<HTML>
```

```

<HEAD>
<SCRIPT language="javascript">
function choixprop(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="Mlle">.Mlle<BR>
<INPUT TYPE="radio" NAME="choix" VALUE="Mme">.Mme<BR>
<INPUT TYPE="radio" NAME="choix" VALUE="Mr">.Mr<BR>
<INPUT TYPE="button"NAME="but" VALUE="Quel et votre choix ?"
onClick="choixprop(form3)">
</FORM>
</BODY>
</HTML>

```

### LES BOUTONS CASE A COCHER (CHECKBOX)

Les boutons case à cocher sont utilisés pour noter un ou plusieurs choix (pour rappel avec les boutons radio un seul choix) parmi un ensemble de propositions. A part cela, sa syntaxe et son usage est tout à fait semblable aux boutons radio sauf en ce qui concerne l'attribut name.

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.

Entrez votre choix :

Il faut sélectionner les numéros 1,2 et 4 pour avoir la bonne réponse.

- Choix numéro 1
- Choix numéro 2
- Choix numéro 3
- Choix numéro 4

Corriger

```

<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>
```

## LISTE DE SELECTION

La boîte de la liste est créée par la balise `<SELECT>` et les éléments de la liste par un ou plusieurs tags `<OPTION>`. La balise `</SELECT>` termine la liste.

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 <code>&lt;SELECT&gt;</code> , tous les éléments de la liste seront affichés. Si vous ne l'indiquez pas un seul apparaîtra dans la boîte 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 boîte.

Un petit exemple comme d'habitude :

Entrez votre choix :

```
<HTML>
<HEAD>
<script language="javascript"> fonction 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>
<OPTION VALUE="2">Elément 2 </OPTION>
<OPTION VALUE="3">Elément 3 </OPTION>
</SELECT>
<INPUT TYPE="button"NAME="b" VALUE="Quel est l'élément retenu?" onClick="liste(form5)">
</FORM>
</BODY>
</HTML>
```

## LE CONTROLE TEXTAREA

L'objet textarea est une zone de texte de plusieurs lignes.

La syntaxe Html est :

```
<FORM>
<TEXTAREA NAME="nom" ROWS=x COLS=y>
texte par défaut
</TEXTAREA>
</FORM>
```

où `ROWS=x` représente le nombre de lignes et `COLS=y` représente le nombre de colonnes.

L'objet textarea possède plusieurs propriétés :

<b>Propriété</b>	<b>Description</b>
Name	indique le nom du contrôle par lequel on pourra accéder.
Defaultvalue	indique la valeur par défaut qui sera affichée dans la zone de texte.
Value	indique la valeur en cours de la zone de texte. Soit celle tapée par l'utilisateur ou si celui-ci n'a rien tapé, la valeur par défaut.

A ces propriétés, il faut ajouter onFocus(), onBlur(), onSelect() et onChange().

En Javascript, on utilisera `\r\n` pour passer à la ligne.

Comme par exemple dans l'expression `document.Form.Text.value = 'Check\r\nthis\r\nout'`.