

Chapitre II : Langage JavaScript

1

Pré-requis : Bases de données, Programmation orientée objet

Objectifs : Créer des sites dynamiques avec PHP, JavaScript et MySQL

Structure de base du HTML

2

I. Introduction au langage Javascript

II. Implantation du code Javascript

III. Boîtes de dialogue

IV. Les variables

V. Les opérateurs

VI. Les structures de contrôle

VII. Les fonctions

VIII. Notion d'objet

IX. Les formulaires

X. Les événements

XI.

Introduction au langage Javascript

3

1. Motivations

- Afficher un message d'erreur lors d'une saisie erronée dans un champ d'un formulaire.
- Afficher la date de jour
- Connaître la nombre de visiteurs de votre site.
- Faire de l'animation publicitaire
- ...

HTML ne peut pas faire ca, (affiche des données statiques)

Définition

4



- Le **Javascript** est un langage de script **incorporé** dans un document HTML.
- Un langage de programmation qui permet d'apporter des améliorations au langage HTML en permettant d'exécuter des commandes du côté client.
- Javascript n'a aucune relation avec le langage Java.
-

Notions de base

5

- La casse : Javascript est sensible à la casse, il fait la différence entre les minuscules et le majuscules.
- Le point virgule: chaque instruction se termine par un point virgule (on peut omettre lorsqu'il y a retour à la ligne)
- Les commentaires:
 - `//` Tous les caractères derrière le sont ignorés
 - `/*` Toutes les lignes comprises entre ces repères sont ignorées par l'interpréteur de code `*/`

Implantation des scripts

6

On peut placer du code JS dans une page HTML à 3 endroits et sous des formes bien différentes.

1. Entre les balises

- **Dans la section `<HEAD>`**, il n'est pas exécuté tout de suite.
- **Dans le corps du document** (entre `<body>` et `</body>`), il est immédiatement exécuté en même temps que le code HTML est interprété.

Exemple

7

Exemple

```
<script language="javascript">  
//Fenêtre pour la saisie d'une variable  
Var=prompt("insérer votre Prénom ");  
/* la fenêtre pour  
l'affichage*/  
alert("Bonjour "+Var);  
</script>
```

Implantation des scripts

8 2. Associé à une balise HTML qui gère un événement

- ❑ Le code JS est généralement inséré sous forme d'un appel de fonction, affectée à un gestionnaire d'événement.
- ❑ Un événement survient à l'initiative de l'utilisateur, (clique sur un bouton, après la saisie du texte dans un champ de formulaire, ...)

<balise ... onEvenement="code JS" | "fonction JS">

Exemple

```
<a onClick="Var=prompt('donnez votre prénom');  
alert('Bonjour '+Var);">
```

Cliquez ici!

```
</a>
```


Implantation des scripts

9 3. Dans un fichier externe

- On peut placer le code dans un fichier spécifique d'extension .js et appeler ce fichier au besoin.
- **Exemple**
- Avec : le fichier `essai.js` contient le code :
`document.write("Bonsoir")`

```
<script language="javascript" src="source.js">
```

```
.....
```

```
</script>
```

Exemple

```
<HTML><HEAD><TITLE>Page d'exemples</TITLE></HEAD>
```

```
<BODY>
```

```
<H4>Exemple </H4>
```

```
<SCRIPT language="JavaScript" src="essai.js" >    </SCRIPT>
```

```
</BODY></HTML>
```

Avec : le fichier `essai.js` contient le code :

```
document.write("Bonsoir");
```

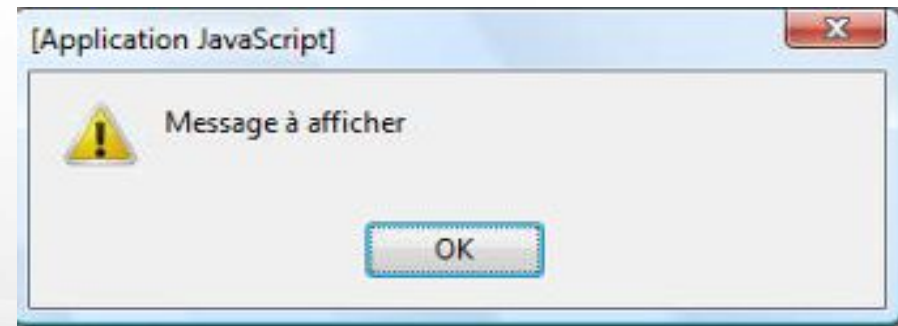
Les Boîtes de dialogue Javascripts

10

On considère 3 types de boîtes de dialogue:

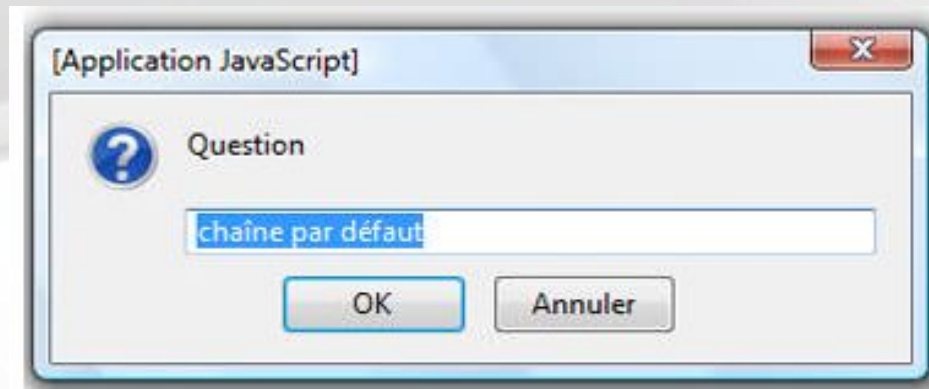
alert(), prompt(), confirm()

✓ **alert('Message à afficher')**



✓ **Var=prompt('Question','chaîne par défaut');**

- Var □ la valeur de la chaîne insérée (si ok)
- si non Var □ Null

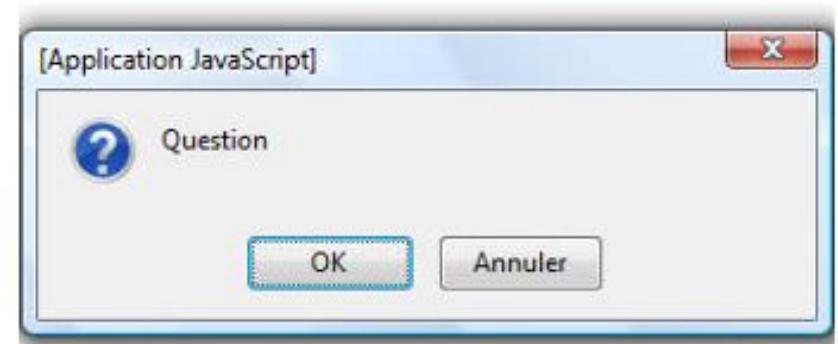


Les Boîtes de dialogue Javascripts

11

✓ **Var=confirm('Question');**

- Var □ true si OK
- Var □ false si Annuler



Exercice : Interprétez les lignes suivantes:

- `Var=prompt("insérer votre Prénom " , " Prénom");
alert('Vous avez inséré '+Var); ok=confirm('Afficher Votre
Prénom? '); if (ok==true) {document.write('BONJOUR
' +Var);}`

Les variables

12

- Une variable est un objet repéré par son nom, pouvant contenir des données, qui pourront être modifiées lors de l'exécution du programme.
- Javascript est faiblement typé: il n'est pas nécessaire de déclarer le type et peut changer le type à tout moment.
- les variables se déclarent de deux façons:

- **Explicitement** par le mot clé var var indice;

- **Implicitement** par l'apparition à gauche du signe égal

Indice=256

Le nom de variable doit respecter la syntaxe suivante:

- La variable doit commencer par une lettre ou un souligné « _ »
- un nom de variables peut comporter des lettres, des chiffres et les caractères _ et \$ (les espaces ne sont pas autorisés!)
- un nom de variable ne peut utiliser des mots dits réservés (else, if, var, case....)

Les variables

13

Exercice:

correct	incorrect	Raison
Variable	Nom de Variable	
Nom_De_Variable	12Nom_De_Variable	
nom_de_variable	toto@mailcity.com	
nom_variable_123	Nom-de-variable	
_707	break	

Les variables

14

- La portée des variables est la visibilité de ces variables dans le code javascript, on distingue :

Les variables locales	Les variables globales
Variable déclarée explicitement: dans une fonction, elle ne sera visible qu'à l'intérieur de cette fonction.	Variable déclarée implicitement
	Variable déclarée explicitement : en dehors des déclarations des fonctions.

Les variables

15

Exemple

```
<SCRIPT language="Javascript">  
var a = 12; var b = 4;  
  
function MultipliePar2(b) { var a = b * 2; return a; }  
  
document.write("Le double de ",b," est  
",MultipliePar2(b), "<br>");  
  
document.write("La valeur de a est ",a);  
</SCRIPT>
```

Les variables

16

le Javascript n'autorise la manipulation que de 4 types de données :

Nom du type	Description	Exemples
undefined	Variable indéfinie	<code>var mavariable;</code>
boolean	Variable logique. 2 états possibles true ou false	<code>var mavariable=true;</code> <code>var mavariablebis=false;</code>
number	Variable numérique	<code>var variable=666;</code>
string	Variable chaîne de caractères	<code>var variable="Marco Pantani";</code>

Les variables

17

Identifiez le type d'une variable

- la fonction **typeof(Nom_du_var)** retourne le type d'une variable

Exemple:

```
<script language="javascript"> Varnumber=666;  
document.write("le type de la variable Varnumber est <h1>  
",typeof(Varnumber),"</h1>");  
</script>
```

Les variables

18

Conversion des types

- Javascript gère la conversion des types d'une façon transparente (implicite)
 - Parfois il est nécessaire de faire la conversion d'une façon explicite , On distingue les fonctions de conversions suivantes :
 - **parseInt(chaine,[base])**: conversion d'une chaîne en entier dans une base désignée (10, 2, 8,16...).
 - **parseFloat()** permet de convertir une variable en nombre décimal
- RQ: en cas d'erreur elles retournent *NaN* (Not a Number)
- **toString()** permet de convertir une variable en une chaîne de caractères

Les opérateurs

19

1. Les opérateurs arithmétiques

Opérateur	Dénomination
+	opérateur d'addition
-	opérateur de soustraction
*	opérateur de multiplication
/	opérateur de division
=	opérateur d'affectation
%	opérateur modulo

Les opérateurs

20

2. Les opérateurs de comparaison

Le résultat renvoyé par une opération de comparaison est une valeur booléenne

Opérateur	Dénomination
== A ne pas confondre avec le signe d'affectation (=)!!	opérateur d'égalité
!=	opérateur de différence
<	opérateur d'infériorité stricte
<=	opérateur d'infériorité
>	opérateur de supériorité stricte
>=	opérateur de supériorité

Les opérateurs

3. Les opérateurs logiques

Opérateur	Effet	Syntaxe
 OU logique	Vérifie qu'une des conditions est réalisée	expression1 expression2
&& ET logique	Vérifie que toutes les conditions sont réalisées	expression1 && expression2
! NON logique	Inverse l'état d'une variable booléenne (retourne la valeur 1 si la variable vaut 0, 0 si elle vaut 1)	!expression

Les opérateurs

22

4. Les opérateurs d'affectation

Opérateur	Exemple	Signification
$+=$	$X+=Y$	$X=X+Y$
$-=$	$X-=Y$	$X=X-Y$
$*=$	$X*=Y$	$X=X*Y$
$/=$	$X/=Y$	$X=X/Y$
$\%=$	$X\%=Y$	$X=X\%Y$

Les opérateurs

23

5. Les opérateurs d'incrémentation

Opérateur	Dénomination	Effet	Syntaxe
++	Incrémentation	Augmente d'une unité la variable	x++
--	Décrémentation	Diminue d'une unité la variable	x--

6. Opérateur de concaténation

L'opérateur '+' lorsqu'il est utilisé avec des chaînes de caractères permet de les concaténer, c'est-à-dire de joindre bout-à-bout les deux chaînes de caractères : →

```
var1='a'
```

```
var2=var1+'b' var2='ab'
```


Les structures de contrôle

25

1. Les structures de contrôle conditionnelles

• L'INSTRUCTION CONDITIONNELLE « IF »

✓ L'instruction conditionnelle « if » simple

```
if (<condition>)  
{<bloc d'instructions >}
```

✓ L'instruction conditionnelle « if ... else »

```
if (<condition>)  
{<bloc d'instructions 1>}  
else  
{<bloc d'instructions 2>}
```

RQ : Il est à noter que les instructions « if » peuvent être imbriquées

• L'INSTRUCTION DE SÉLECTION MULTIPLE « SWITCH »

```
switch ( <variable> )  
{  
case Cte1 :  
<bloc d'instructions 1>  
break;  
case Cte2 :  
<bloc d'instructions 2>  
break;  
....  
case CteN:  
<bloc d'instructions N>  
[ default :  
<bloc d'instructions> ]  
}
```

Les structures de contrôle

2. Les structures de contrôle itératives

- L'INSTRUCTION ITERATIVE « FOR »

```
for ( <initialisation> ; <test_de_boucle> ; <incrémentement> )  
{  
  <bloc d'instructions>  
}
```

- L'INSTRUCTION ITÉRATIVE « WHILE »

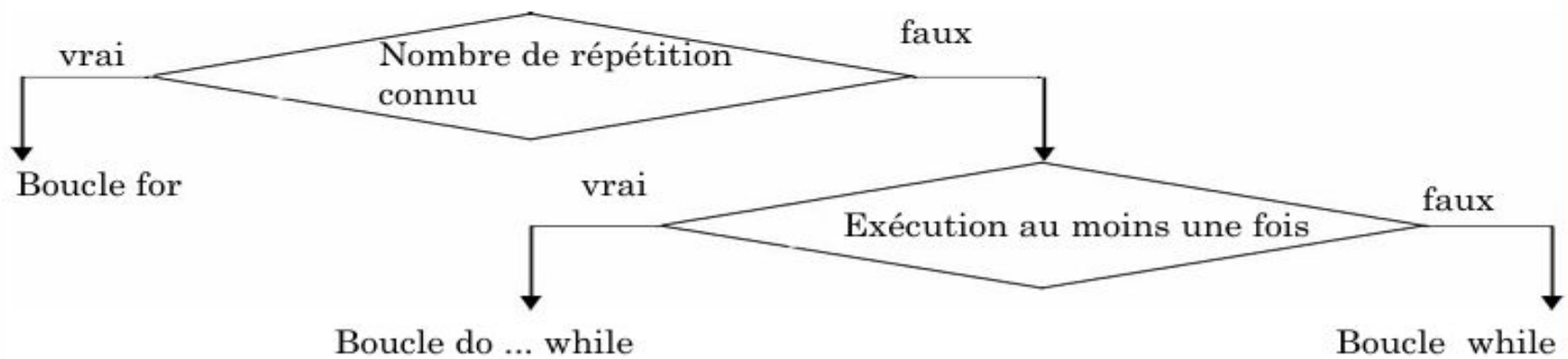
```
while ( <condition> )  
{  
  <bloc d'instructions>  
}
```

Les structures de contrôle

• L'INSTRUCTION ITÉRATIVE « DO.. WHILE »

```
do  
{  
  <bloc d'instructions>  
}  
while (<condition>);
```

➤ Choix de structure itérative



Les Fonctions

29

On appelle *fonction* un **sous-programme** qui permet d'effectuer un ensemble d'instructions par simple appel de la fonction dans le **corps** du programme principal.

Une fonction en javascript est déclarée comme suit:

```
function Nom_De_La_Fonction(argument1,  
argument2, ...)
```

```
{ liste d'instructions }
```

- une fonction peut retourner ou non un résultat, pour retourner un résultat on utilise l'instruction **return**.

Les Fonctions

30

Exemple1:

```
function Affiche(message)  
{document.write(message);}
```

Exemple2:

```
function somme(a,b)  
{ som=a+b;  
  return som;}
```

Appel des fonctions

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

Affiche('hello'); Res=somme(1,3);

Notion d'objet

31

- Quand une page web est chargée, le navigateur crée plusieurs objets pour représenter les informations y figurant.

Ces objets sont classés de manière hiérarchique. L'objet le plus haut de la hiérarchie étant l'objet de la classe window

```
window
├── parent, frames[], self, top
├── location
├── history
├── document
│   ├── forms[]
│   │   └── elements[] (text, textarea, checkbox, password
│   │                       radio, select, button, submit, reset)
│   ├── links[]
│   ├── images[]
│   └── URL
```

Notion d'objet

32

Exemple: soit le formulaire suivant:

Pour accéder au bouton radio :

- ✓ `window.document.form1.r1`
- ✓ `document.forms[0].r1`
- ✓ `document.getElementById('va')`

```
<form name="form1">  
  <input type="radio"  
    name="r1" value="test"  
    id="va"/> test <br />  
  <input type="button"  
    name="" value="ok" />  
</form>
```



RQ : l'objet window est repris par défaut par Javascript et devient donc facultatif.

Notion d'objet

33

Un objet est caractérisé par ses propriétés et ses méthodes.

- Une propriété est un attribut, une caractéristique, une description de l'objet
 - **document.form1.r1.value**
- Les fonctionnalités que les objets peuvent réaliser sont ces méthodes
 - **document.write()**

Les formulaires

34

Avec Javascript, on peut accéder à chaque élément d'un formulaire pour, par exemple, y aller lire ou écrire une valeur, noter un choix auquel on pourra associer un gestionnaire d'événement. → pages interactives

• ligne de texte, textarea

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.

Les formulaires

35

•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.

•Les boutons case à cocher (checkbox)

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.

Les formulaires

37

•Liste de sélection

Propriété	Description
name	indique le nom de la liste déroulante.
length	indique le nombre d'éléments de la liste.
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.

Les événements

38

- Les événements HTML sont des « choses » qui arrivent aux éléments HTML.
- Lorsque JavaScript est utilisé dans des pages HTML, JavaScript peut « réagir » sur ces événements.

Un événement HTML peut être quelque chose que fait le navigateur ou quelque chose qu'un utilisateur fait.

Voici quelques exemples d'événements HTML :

- Le chargement d'une page Web HTML est terminé
- Un champ de saisie HTML a été modifié
- Un bouton HTML a été cliqué

Les événements

39

1. onclick<button>

```
<button  
onclick="document.getElementById('demo').innerHTML  
= Date()">The time is?</button>
```

Les événements

40

Événement	Description
Abort (onAbort)	Cet événement a lieu lorsque l'utilisateur interrompt le chargement de l'image
Blur (onBlur)	Se produit lorsque l'élément perd le focus, c'est-à-dire que l'utilisateur clique hors de cet élément, celui-ci n'est alors plus sélectionné comme étant l'élément actif.
Change (onChange)	Se produit lorsque l'utilisateur modifie le contenu d'un champ de données.
Click (onClick)	Se produit lorsque l'utilisateur clique sur l'élément associé à l'événement.
dblclick (onDblclick)	Se produit lorsque l'utilisateur double-clique sur l'élément associé à l'événement (un lien hypertexte ou un élément de formulaire). Cet événement n'est supporté que par les versions de Javascript 1.2 et supérieures
dragdrop (onDragdrop)	Se produit lorsque l'utilisateur effectue un glisser-déposer sur la fenêtre du navigateur. Cet événement n'est supporté que par les versions de Javascript 1.2 et supérieures
error (onError)	Se déclenche lorsqu'une erreur apparaît durant le chargement de la page. Cet événement fait partie du Javascript 1.1.
Focus (onFocus)	Se produit lorsque l'utilisateur donne le focus à un élément, c'est-à-dire que cet élément est sélectionné comme étant l'élément actif
keydown (onKeyDown)	Se produit lorsque l'utilisateur appuie sur une touche de son clavier. Cet événement n'est supporté que par les versions de Javascript 1.2 et supérieures

Les événements

41

keypress (onKeyPress)	Se produit lorsque l'utilisateur maintient une touche de son clavier enfoncée. Cet événement n'est supporté que par les versions de Javascript 1.2 et supérieures
keyup (onKeyUp)	Se produit lorsque l'utilisateur relâche une touche de son clavier préalablement enfoncée. Cet événement n'est supporté que par les versions de Javascript 1.2 et supérieures
Load (onLoad)	Se produit lorsque le navigateur de l'utilisateur charge la page en cours
MouseOver (onMouseOver)	Se produit lorsque l'utilisateur positionne le curseur de la souris au-dessus d'un élément
MouseOut (onMouseOut)	Se produit lorsque le curseur de la souris quitte un élément. Cet événement fait partie du Javascript 1.1.
Reset (onReset)	Se produit lorsque l'utilisateur efface les données d'un formulaire à l'aide du bouton Reset.
Resize (onResize)	Se produit lorsque l'utilisateur redimensionne la fenêtre du navigateur
Select (onSelect)	Se produit lorsque l'utilisateur sélectionne un texte (ou une partie d'un texte) dans un champ de type "text" ou "textarea"
Submit (onSubmit)	Se produit lorsque l'utilisateur clique sur le bouton de soumission d'un formulaire (le bouton qui permet d'envoyer le formulaire)
Unload (onUnload)	Se produit lorsque le navigateur de l'utilisateur quitte la page en cours

Les événements

42

Association des événements aux objets

Chaque événement ne peut pas être associé à n'importe quel objet. Il est évident par exemple qu'un événement *OnChange* ne pourra pas s'appliquer à un lien hypertexte.

Evénements	Objets concernés
abort	Image
blur	Button, Checkbox, FileUpload, Layer, Password, Radio, Reset, Select, Submit, Text, TextArea, window
change	FileUpload, Select, Submit, Text, TextArea
click	Button, document, Checkbox, Link, Radio, Reset, Select, Submit
dblclick	document, Link
dragdrop	Window

Les événements

43

error	Image, window
focus	Button, Checkbox, FileUpload, Layer, Password, Radio, Reset, Select, Submit, Text, TextArea, window
keydown	document, Image, Link, TextArea
keypress	document, Image, Link, TextArea
keyup	document, Image, Link, TextArea
load	Image, Layer, window
mousedown	Button, document, Link
mousemove	Aucun spécifiquement
mouseout	Layer, Link
mouseover	Area, Layer, Link
mouseup	Button, document, Link
move	Window
reset	Form
resize	Window
select	text, Textarea
submit	Form
unload	window