

MODULE : ATELIER WEB2

PUBLIC : LIS2 TECHNOLOGIES DE
L'INFORMATIQUE

1

Aymen BOUREGBA

Javascript

- Introduction au langage Javascript
- Implantation du code Javascript
- Boites de dialogue
- Les variables
- Les opérateurs
- Les structures de contrôle
- Les fonctions
- Notion d'objet
- Les formulaires
- Les événements
- Les objets de noyau

Introduction au langage Javascript

3

Introduction au langage Javascript

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 ça, (affiche des données statiques)

☺ Langages de scripts

4

Introduction au langage Javascript

2. Définition

- 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.

Travail de recherche : Javascript n'a aucune relation avec Java: Comparez les deux langages.

AU : 2010/2011

5

Introduction au langage Javascript

3. Notions de base

- **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 */

AU : 2010/2011

6

Implantation des scripts

7

Implantation des scripts

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

1. Entre les balises `<SCRIPT>...</SCRIPT>`

```
<script language="Javascript" type="text/javascript">
```

Placez ici le code de votre script

```
</script>
```

- 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.

8

Implantation des scripts

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>
```

AU : 2010/2011

9

Implantation des scripts

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>
```

AU : 2010/2011

10

Implantation des scripts

3. Dans un fichier externe

On peut placer le code dans un fichier spécifique d'extension **.js** et appeler ce fichier au besoin.

```
<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");
```

AU : 2010/2011

11

Les Boites de dialogue Javascrpts

AU : 2010/2011

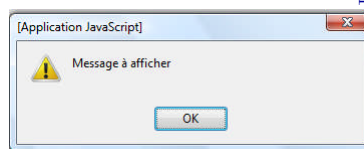
12

Les boites de dialogue(1/3)

On considère 3 types de boites de dialogue:

alert(), prompt(), confirm()

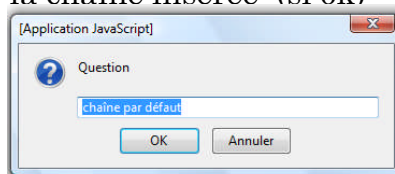
✓ **alert('Message à afficher')**



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

➤ Var ← la valeur de la chaîne insérée (si ok)

➤ si non Var ← Null



AU : 2010/2011

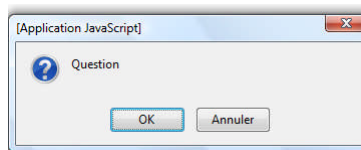
13

Les boites de dialogue(3/3)

✓ **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);}
```

AU : 2010/2011

14

Les variables

15

Les variables

1. Le concept de variables

- 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.

16

Les variables

2. Déclaration des variables

- 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

AU : 2010/2011

17

Les variables

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....)

AU : 2010/2011

18

Les variables

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	

AU : 2010/2011

19

Les variables

3. La portée des variables

- 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.

AU : 2010/2011

20

Les variables

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>
```

AU : 2010/2011

21

Les variables

3. Les types de données dans les variables

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

Nom du type	Description	Exemples
undefined	Variable indéfinie	var mavariabe;
boolean	Variable logique. 2 états possibles true ou false	var mavariabe=true; var mavariabebis=false;
number	Variable numérique	var variable=666;
string	Variable chaîne de caractères	var variable="Marco Pantani";

AU : 2010/2011

22

Les variables

4. 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>
```

AU : 2010/2011

23

Les variables

5. 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 :

o **parseInt(chaine,[base])**: conversion d'une chaîne en entier dans une base désignée (10, 2, 8,16...).

o **parseFloat()** permet de convertir une variable en nombre décimal

RQ: en ca d'erreur elles retournent *NaN* (Not a Number)

o **toString()** permet de convertir une variable en une chaîne de caractères

AU : 2010/2011

24

Les opérateurs

25

Les opérateurs

1. Les opérateurs arithmétiques

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

26

Les opérateurs

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

AU : 2010/2011

27

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

AU : 2010/2011

28

Les opérateurs

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

AU : 2010/2011

29

Les opérateurs

5. Les opérateurs d'incrémentatation

Opérateur	Dénomination	Effet	Syntaxe
++	Incrémentatation	Augmente d'une unité la variable	x++
--	Décrémentatation	Diminue d'une unité la variable	x--

AU : 2010/2011

30

Les opérateurs

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'
```

AU : 2010/2011

31

Les structures de contrôle

AU : 2010/2011

32

Les structures de contrôle

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

AU : 2010/2011

33

Les structures de contrôle

- 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> ]
  }
```

AU : 2010/2011

34

Les structures de contrôle

2. Les structures de contrôle itératives

- L'INSTRUCTION ITERATIVE « FOR »

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

- L'INSTRUCTION ITÉRATIVE « WHILE »

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

AU : 2010/2011

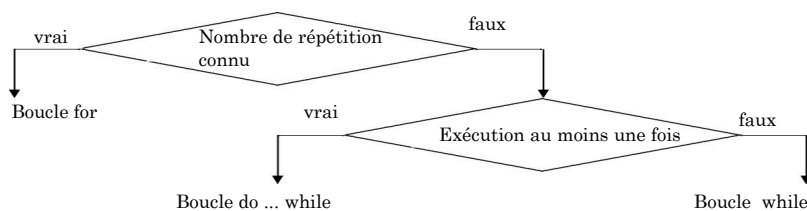
35

Les structures de contrôle

- L'INSTRUCTION ITÉRATIVE « DO.. WHILE »

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

➤ Choix de structure itérative



AU : 2010/2011

36

Les Fonctions

37

Les Fonctions

- 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**.

38

Les Fonctions

Exemple1:

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

Exemple2:

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

AU : 2010/2011

39

Les Fonctions

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);
```

AU : 2010/2011

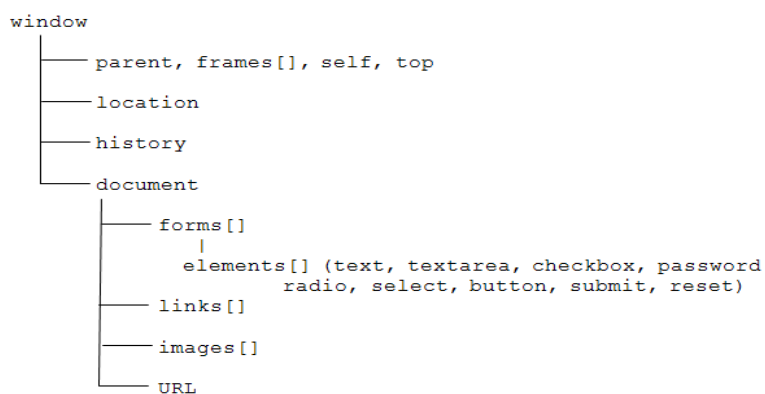
40

Notion d'objet

41

Notion d'objet

- 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.



42

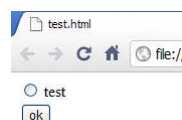
Notion d'objet

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.

AU : 2010/2011

43

Notion d'objet

- 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()

AU : 2010/2011

44

Les formulaires

45

Les formulaires

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.

46

Les formulaires

•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.

AU : 2010/2011

47

Les formulaires

•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.

AU : 2010/2011

48

Les formulaires

•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.

AU : 2010/2011

49

Les événements

AU : 2010/2011

50

Les événements

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

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

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.

Événements	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

AU : 2010/2011

Les événements

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

AU : 2010/2011

54

Les objets de noyau

55

Les objets de noyau

1. L'objet Date

Permet de travailler avec toutes les variables qui concernent les dates et la gestion du temps.

Syntaxe :

- Nom_objet = new Date ();
- Nom_objet = new Date ("année, mois, jour");
- Nom_objet=new Date (année,mois,jour,heures,minutes,secondes);

56

Les objets de noyau

✓ Méthodes de l'objet Date

Méthode	Description	Type de valeur retournée
getDate()	Permet de récupérer la valeur du jour du mois	Entier entre 1 et 31
getDay()	Permet de récupérer la valeur du jour de la semaine pour la date spécifiée	Entier entre 0 et 6
getMonth()	Permet de récupérer le numéro du mois	Entier entre 0 et 11
getFullYear()	Permet de récupérer la valeur de l'année sur 4 chiffres pour la date passée en paramètre	Entier (XXXX)
getYear()	Permet de récupérer la valeur de l'année sur 2 chiffres pour l'objet Date.	Entier (XX)
getHours()	Permet de récupérer la valeur de l'heure	Entier entre 0 et 23
getMinutes()	Permet de récupérer la valeur des minutes	Entier entre 0 et 59
getSeconds()	Permet de récupérer le nombre de secondes	Entier entre 0 et 59
getTime()	Permet de récupérer le nombre de millisecondes depuis le 1 ^{er} janvier 1970	Entier.
getTimezoneOffset()	Retourne la différence entre l'heure locale et l'heure GMT	Entier.

AU : 2010/2011

Les objets de noyau

Méthode	Description	Type de valeur retournée
toGMTString()	Permet de convertir une date en une chaîne de caractères au format GMT	Chaîne de caractère du style : Wed, 28 Jul 1999 15:15:20 GMT
toLocaleString()	Permet de convertir une date en une chaîne de caractères au format local	Chaîne de caractère dont la syntaxe dépend du système, par exemple : 28/07/99 15:15:20
setDate(X)	Permet de fixer la valeur du jour du mois (entre 1 et 31)	
setYear(X)	Permet de fixer l'année	
setMonth(X)	Permet de fixer le numéro du mois (entier entre 0 et 11)	
setHours(X)	Permet de fixer la valeur de l'heure (entier entre 0 et 23)	
setMinutes(X)	Permet de fixer la valeur des minutes (entier entre 0 et 59)	
setSeconds(X)	Permet de fixer la valeur des secondes (entier entre 0 et 59)	

AU : 2010/2011

58

Les objets de noyau

2. L'objet Math

Méthode	Description
abs(valeur)	valeur absolue de valeur Math.
round(valeur)	entier arrondi le plus proche
floor(valeur)	entier immédiatement inférieur (ou égal)
ceil()	entier immédiatement supérieur (ou égal)
sqrt(valeur)	racine carrée
pow(x,y)	x puissance y
min(a,b)	minimum entre a et b
max(a,b)	Maximum entre a et b
random()	renvoie la valeur d'un nombre aléatoire choisi entre 0 et 1.
eval("chane")	évalue une chaîne de caractère sous forme de valeur numérique.

AU : 2010/2011

59

Les objets de noyau

✓ Les fonctions trigonométriques ✓ Les fonctions logarithmiques

```
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);
```

```
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;
```

AU : 2010/2011

60

Les objets de noyau

3. L'objet tableau

Un tableau est un ensemble de données de même type, distingués par un indice. L'indice est placé entre [].

Syntaxe :

- var nom_tableau = new Array(dimension) ;
- var Les4saisons = new Array("printemps", "été", "automne", "hiver")
- var Les4saisons2 = ["printemps", "été", "automne", "hiver"];

AU : 2010/2011

61

Les objets de noyau

✓ La propriété length

La propriété **length** donne le nombre d'éléments du tableau.

Syntaxe : Nom_Tableau.length.

Exemple

```
<script language="JavaScript">
var mois= new Array(12);
var NbMois = mois.length ;
document.write("Il y a " + NbMois + "mois dans l'année");
document.write(" partagés en " + NbMois / 3 + " trimestres");
</script>
```

AU : 2010/2011

62

Les objets de noyau

✓ Les méthodes

Méthode	Description
reverse()	Elle change l'ordre des éléments d'un tableau
sort()	Elle trie un tableau suivant un ordre croissant ou décroissant
concat()	Elle permet de concaténer plusieurs tableaux, c'est-à-dire de créer un tableau à partir des différents tableaux. La méthode concat admet un seul paramètre: <i>NomTableau1.concat(Nom_Tableau2)</i> ;
push()	Elle ajoute un ou plusieurs éléments au tableau. L'ajout se fait à la fin du tableau : <i>NomTab.push(val1, val2, ..., val n)</i> ;
unshift()	Elle ajoute un ou plusieurs éléments au tableau. L'ajout se fait au début du tableau
pop()	Elle supprime le dernier élément du tableau et retourne sa valeur
shift()	Elle supprime le premier élément du tableau et retourne sa valeur
join()	Elle a pour fonction de transformer un tableau en une chaîne de caractère dans laquelle les éléments sont séparés par un caractère de séparation. Le caractère de séparation par défaut est la virgule.

AU : 2010/2011

63

Les objets de noyau

4. L'objet string

Syntaxe :

- chaîne = new String("Je m'appelle Anis")
- chaîne = "Je m'appelle Anis"

✓ La propriété length

La propriété **length** indique la taille d'une chaîne de caractères (Le nombre de ses caractères). Sa syntaxe est: nomChaine.length.

AU : 2010/2011

64

Les objets de noyau

✓ Les méthodes

Méthode	Fonction	Syntaxe
big	Afficher le texte en une police plus grosse	ch.big()
bold	Afficher du texte en gras	ch.bold()
italics	Afficher du texte en italique	ch.italics()
small	Afficher du texte avec une police plus petite	ch.small()
strike	Afficher du texte barré	ch.strike()
sup	Afficher du texte de style exposant	ch.sup()
sub	Afficher du texte avec un style indice	ch.sub()
fontSize	Définir la taille du texte d'une chaîne de caractères	ch.fontSize(taille)
fontcolor	Définir la couleur d'une chaîne de caractères	ch.fontcolor(couleur)

AU : 2010/2011

35

Les objets de noyau

Méthode	Fonction	Syntaxe
slice	Extraire la sous chaîne débutant à l'indice désigné par le premier paramètre et s'achevant à l'indice désigné par le second paramètre.	nomChaine.slice(indDeb[,indFin])
substr	renvoie un string partiel situé entre la position x et la position y-1	nomChaine.substr(indDeb[,longueur])
charAt	Extraire un caractère dans une chaîne.	nomChaine.charAt(indice)
indexOf	Méthode qui renvoie la position d'une chaîne partielle à partir d'une position déterminée. Elle renvoie -1 si la chaîne n'est pas trouvée.	nomChaine.indexOf(<sous chaîne>[,<indiceDebut>])
lastIndexOf	Même fonction que indexOf sauf que elle retourne l'indice du premier caractère de la dernière occurrence de la sous chaîne recherchée.	nomChaine.lastIndexOf(<sous chaîne>[,<indiceDebut>])
split	Construit un tableau dont les éléments sont les sous chaînes ainsi séparées.	nomChaine.split([séparateur])
toLowerCase	Convertir une chaîne en minuscule	nomChaine.toLowerCase()
toUpperCase	Convertir une chaîne en majuscule	nomChaine.toUpperCase()

AU : 2010/2011

