Technologie Web – Partie 3

JavaScript

Pr. Youness IDRISSI KHAMLICHI
E-mail : **youness.khamlichi@usmba.ac.ma**

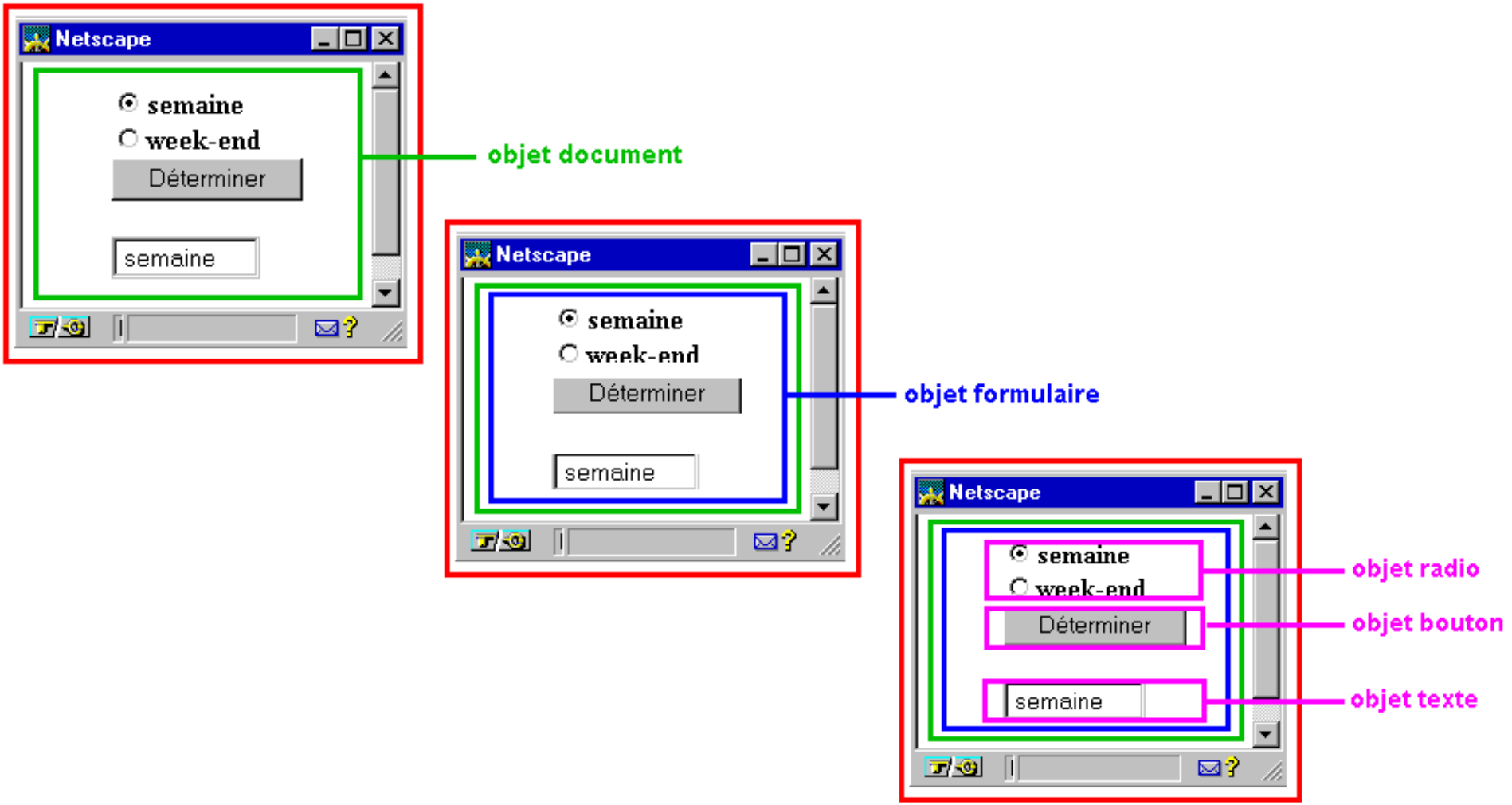
Introduction

- Javascript est un langage de scripts **incorporé aux balises Html qui permet d'améliorer la présentation et l'interactivité des pages Web**
- Javascript est une extension du code Html des pages Web.
- Les scripts vont être gérés et exécutés par le browser lui-même sans devoir faire appel aux ressources du serveur
- Les scripts peuvent être écrits dans la page HTML ou dans un autre fichier à part avec extension .js

Javascript n'est pas Java

Javascript	Java
Code intégré dans la page Html	Module (applet) distinct de la page Html
Code interprété par le browser au moment de l'exécution	Code source compilé avant son exécution
Codes de programmation simples mais pour des applications limitées	Langage de programmation beaucoup plus complexe mais plus performant
Permet d'accéder aux objets du navigateur	N'accède pas aux objets du navigateur
Confidentialité des codes nulle (code source visible)	Sécurité (code source compilé)

Notion d'objet



Notion d'objet

fenêtre	document	formulaire	radio bouton texte

(window).document.form.radio[0]

Syntaxe

```
<SCRIPT LANGUAGE="Javascript">.
…..
…..code javascript

 </SCRIPT>.
```

Commentaire

// commentaire
Tout ce qui est écrit entre le // et la fin de la ligne sera ignoré.

Il sera aussi possible d'inclure des commentaires sur plusieurs lignes avec le code

/* commentaire */

Exemple

<HTML>	Html normal
<HEAD>	...
<TITLE>Mon premier Javascript</TITLE>	...
</HEAD>	...
<BODY>	...
Bla-bla en Html	...
<SCRIPT LANGUAGE="Javascript">	Début du script
// affichage du texte: votre texte	
--alert("votre texte");	
	Fin du script
</SCRIPT>	Html normal
Suite en Html	...
</BODY>	...
</HTML>	

Extension .js pour scripts externes

```
<SCRIPT LANGUAGE='javascript'
    SRC='http://site.com/javascript.js'></SCRIPT>
```

```
<SCRIPT LANGUAGE='javascript'
    SRC='http://site.com/javascript.js />
```

Notion de variable

Les variables peuvent se déclarer de deux façons :
soit de façon implicite ou explicite.

La commande qui permet de déclarer une variable est le mot var.

Par exemple :

- var Numero = 1

- var Prenom = " Adil"

Variable

Exemple:

```
<HTML>
<BODY>
    <SCRIPT LANGUAGE="Javascript">

        var texte = "Mon chiffre préféré est le ";
        var variable = 7;
        document.write(texte + variable);

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

Types de données

Type	Description
Des nombres	Tout nombre entier ou avec virgule tel que 22 ou 3.1416
Des chaînes de caractères	Toute suite de caractères comprise entre guillemets telle que "suite de caractères"
Des booléens	Les mots true pour vrai et false pour faux
Le mot null	Mot spécial qui représente pas de valeur

Affichage du texte

La méthode write(): méthode de l'objet document, pour écrire dans le document

- document.write("votre texte");
- document.write(resultat); // soit la variable resultat
- document.write("Le résultat est " + resultat);

- document.write("<B>Le résultat est</B>" + resultat);

ou

- document.write ("<B>" + "Le résultat est " + "</B>" + resultat);

Affichage du texte

Exemple :

```
<HTML>
<BODY>
        <H1>Ceci est du Html</H1>
        <SCRIPT LANGUAGE="Javascript">

            document.write("<H1>Et ceci du Javascript</H1>");

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

Lecture

Javascript vous propose une autre boite de dialogue, dans le cas présent appelée boite d'invite, qui est composée d'un champ comportant une entrée à compléter par l'utilisateur.

Cette entrée possède aussi une valeur par défaut.

Syntaxe :

```
prompt("texte de la boite d'invite", "valeur par défaut");
```

Les écriture en Javascript...

variable.big();
variable.small();
variable.bold();
variable.italics();
variable.fontcolor(color); //ex. Var.fontcolor(“red”);
variable.fontsize(x);
variable.strike();// texte barré
variable.sub(); //indice
variable.sup();// exposant
Exemple :

```
document.write(nom.fontcolor("green").italics().bold().strike());
```

Les écriture en Javascript...

```
str="Ensa Fes";
x=3;
document.write("<FONT SIZE=3>" +str+"</FONT>");
document.write("<FONT SIZE=3>" +" Ensa Fes
</FONT>");
document.write(str. fontcolor(“red”) );
document.write(str.fontsize(x));
document.write(str.italics());
document.write(str.bold());
```

Les instructions de formatage de document

- **document.bgColor**

Cette instruction permet de spécifier la couleur d'arrière-plan d'un objet document. On peut employer le nom ou la valeur RGB de la couleur.

```
document.bgColor="white";
document.bgColor="#FFFFFF";
```

- **document.fgColor**

Cette instruction permet de spécifier la couleur du texte d'un objet document.

```
document.fgColor="black";
document.fgColor="#000000";
```

Les instructions de formatage de document

- **document.title**

Cette instruction permet d'afficher le titre du document:

```
document.write("le titre du document est " + document.title
+ "<br>");
```

- **document.lastModified**

La propriété javascript **lastModified** de l'objet **document** permet de savoir quand la page HTML a été modifiée.

```
document.write("la dérniere modification : " +
document.lastModified + "<br>");
```

Les opérateurs

Les opérateur de calcul: x=11

Signe	Nom	Signification	Exemple	Résultat
+	plus	Addition	x + 3 11	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

Les opérateurs

Les opérateurs de comparaison:

Signe	Nom	Exemple	Résultat
==	égal	x==11	true
<	inférieur	x<11	false
<=	inférieur ou égal	x<=11	true
>	supérieur	x>11	false
=<	supérieur ou égal	x>=11	true
!=	différent	x!=11	false

Les opérateurs

Les opérateurs associatifs:

Signe	Description	Exemple	Signification	Résultat
+=	plus égal	x += y	x = x + y 11 , 5	16
-=	moins égal	x -= y	x = x - y	6
*=	multiplié égal	x *= y	x = x * y	55
/=	divisé égal	x /= y	x = x / y	2.2

Les opérateurs

Les opérateurs d'incrémentation :

Signe	Description	Exemple	Signification	Résultat
x++	incrémentation (x++ est le même que x=x+1)	y = x++	3 puis plus 1	4
x--	décrémentation (x-- est le même que x=x-1)	y= x--	3 puis moins 1	2

Les fonctions

Une fonction est un groupe de ligne(s) de code de programmation destiné à exécuter une tâche bien spécifique

- Déclaration

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

- Appel

```
nom_de_la_fonction();
```

Les fonctions

- Passer une valeur à une fonction

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

- Appel

```
Exemple(‘’ mon texte ‘’);
```

Les fonctions

➢ Variables locales et variables globales

Variable locale

Une variable déclarée dans une fonction par le mot clé *var* aura une portée limitée à cette seule fonction, donc c'est *variable locale*

Variable globale

Si la variable est déclarée contextuellement (sans utiliser le mot var), sa portée sera globale

Les fonctions

➢ Variables locales et variables globales

Exemple : Variable locale

```
function produit(nombre1,nombre2)
{
     var  produit = nombre1*nombre2
}
```

Exemple : Variable globale

```
function produit(nombre1,nombre2)
{
       produit = nombre1*nombre2
}
```

Les fonctions

➢ Variables globales

Les variables déclarées tout au début du script, en dehors et avant toutes fonctions, seront toujours globales, qu'elles soient déclarées avec var ou de façon contextuelle.

Exemple : Variable globale

```
<SCRIPT LANGUAGE="javascript">
    var produit=1;
    function produit(nombre)
    {
        var  produit = nombre*nombre*nombre;
    }
</SCRIPT>
```

Les événements

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

onévénement="fonction() "
Ou
onévénement= " instruction javascript"

Les fonctions prédéfinies

- **eval**
  - Cette fonction **exécute** un code Javascript à partir d'une *chaîne de caractères*.

```
...
<SCRIPT LANGUAGE="JavaScript">
function evaluation()
{
        Var x= eval(‘’ 20+ 40’’);
}
</SCRIPT>
```

Les fonctions prédéfinies

- **isFinite**
  - Détermine si le paramètre est un nombre fini. Renvoie *false* si ce n'est pas un nombre ou l'infini positif ou infini négatif.

```
isFinite(240) //retourne true
isFinite("Un nombre") //retourne false
```

- **isNaN**
  - détermine si le parametre n'est pas un nombre (NaN : Not a Number).

```
isNaN("un nombre") //retourne true
isNaN(20) //retourne false
```

Les fonctions prédéfinies

- **parseFloat**
  - analyse une chaîne de caractères et retourne un nombre décimal.

```
var numero="125";
var nombre=parseFloat(numero); //retourne le nombre 125
```

- **parseInt**
  - analyse une chaîne de caractères et retourne un nombre **entier** de la base spécifiée.
  - La base peut prendre les valeurs :
  - *16* (hexadécimal) *10* (décimal), *8* (octal), *2* (binaire).

```
var prix=30.75;
var arrondi = parseInt(prix, 10); //retourne 30
```

Les fonctions prédéfinies

- **Number**
  - convertit l'objet spécifié en valeur numérique

```
var jour = new Date(“Octobre 09, 2014 03:24:00");//converit la date en millisecondes
alert (Number(jour));
```

- **String**
  - convertit l'objet spécifié en chaîne de caractères

```
jour = new Date(430054663215);//Convertit le nombre en date Mois jour, Année etc.
alert (String(jour));
```

Les événements

➢ 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>
```

Les événements

- 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.

Exemple

```
<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>
```

Les événements

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

➢ **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.

Les événements

➢ 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

➢ onchange

Non seulement la case du formulaire doit avoir perdu le focus mais aussi son contenu doit avoir été modifié par l'utilisateur.

➢ onselect

Cet événement se produit lorsque l'utilisateur a sélectionné tout ou partie d'une zone de texte.

Les événements

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	onBlur, onChange, onFocus
Bouton Submit	onClick
Bouton Reset	onClick

Les conditions

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

OU

```
(expression) ? instruction a : instruction b
```

L'expression for

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

Exemple :

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

L'expression While

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

Exemple :

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

Formulaires

- JavaScript associé aux formulaires permet
  - d'assister et de guider le visiteur,
  - de contrôler la saisie,
  - de faire des traitements
    (calcul, manipulation de chaînes de caractères)

Propriétés et méthodes

name	Nom du formulaire
action	Adresse du script de serveur à exécuter
method	Méthode d'appel du script (get ou post)
enctype	Type d'encodage du formulaire

submit	Déclenche l'action du formulaire
reset	Réinitialise avec les valeurs par défaut

L'élément **input**

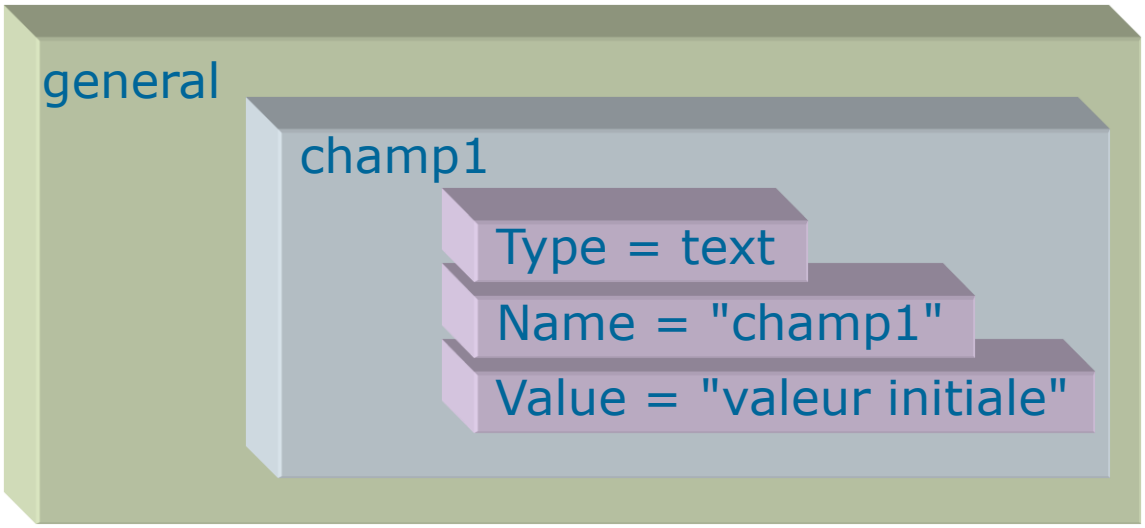
- L'objet **input** est le plus utilisé de tous.
- Il permet d'afficher
  - des champs texte,
  - des boutons,
  - des radio-boutons,
  - des cases à cocher,
  - le champ caché,
  - les boutons spéciaux **reset** et **submit**.

L'accès aux éléments

- Les éléments de formulaire sont des objets JavaScript
- Voyons comment accéder via JavaScript aux objets d'un formulaire
- Supposons le formulaire suivant

```
<form name="general">
<input type="text" name="champ1"
        value="Valeur initiale">
</form>
```

L'accès aux éléments



Pour accéder à un objet il faut suivre sa hiérarchie.

Accéder au formulaire

- Le formulaire est un élément de l'objet document
- Pour accéder au formulaire général, il faut écrire

document.forms[0] ou
document.general

- **forms** est le tableau des formulaires de **document**
- On peut accéder à un formulaire par
  - son nom comme indice dans **forms** ou
  - son entier comme indice dans **forms** où
    - Les indices des tableaux commence à 0
  - son nom tout simplement

Accéder à un élément

- Pour accéder maintenant à la zone de texte, on écrit

  document.general.**elements[0]**

  ou

  document.general.**champ1**

- **elements** est le tableau de tous les éléments du formulaire
- On peut accéder à un élément par
  - son nom comme indice dans **elements** ou
  - son entier comme indice dans **elements** ou
  - son nom tout simplement

Manipuler les propriétés d'un élément

- Par exemple, pour placer dans la zone de texte le mot "NOUVEAU", il faut écrire

```
document.general.champ1.value="NOUVEAU";
```

Appeler une méthode d'un élément

- Pour donner le **focus** au champ texte du haut de cette page, il faut appeler la méthode focus() sur cet élément

document.general. champ1.**focus()**

Intégrer du JavaScript dans un événement

- Reprenons l'exemple précédent, et plaçons "NOUVEAU" dans la zone de texte du formulaire à l'aide d'un bouton

```
<form name=“CHG">

<input type="text" name=“nom“ value="Valeur initiale">

<input type="button" value="Changer la zone de texte“
 onClick='document. CHG.nom.value = “nouveau" '>

</form>
```

Accéder aux radio-boutons

```
<form>
 <input type="radio" name="os" value="WXP" checked> WindXP
 < input type="radio" name="os" value="Linux"> Linux
 < input type="radio" name="os" value="Autre"> Autre
 // le bouton défini précédemment
 < input type="button" value="Tester“
   onClick="testerRadio(this.form.os)">

 // une fct de consultation du groupe 'radio' de radio-boutons
 <script language="javascript">
  function testerRadio(radio) {
    for (var i=0; i<radio.length;i++) {
      if (radio[i].checked) {
        alert("Système = "+radio[i].value)
      }   }   } </script>
</form>
```

Accéder aux menus de sélection

```
<select name="liste" size=1>
  <option value="valeur ligne 1">Libellé ligne 1 </option>
  <option value="valeur ligne 2">Libellé ligne 2 </option>
  <option value="valeur ligne 3">Libellé ligne 3 </option>
  <option value="valeur ligne 4">Libellé ligne 4 </option>
</select>
```

name	Nom de la liste
selectedIndex	Indice de la ligne sélectionnée (ligne 1 : indice=0)
length	Nombre de lignes
value	Valeur d'une ligne

Accéder aux menus de sélection

- Pour récupérer l'indice la ligne sélectionnée

  this.form.liste.selectedIndex

- Pour récupérer le nombre de lignes

  this.form.liste.options.length

- Pour récupérer la valeur de la ligne sélectionnée

this.form.liste.options[this.form.liste.selectedIndex].value

Accès en utilisant *getElementsByName*

- document.getElementsByName(String nom)

Retourne un tableau (Array) d'objets HTML ayant nom défini dans la propriété name de la balise de l'objet.

```
<FORM name="form_fruit">
  <INPUT type="checkbox" name="fruit" value="Fraise"> Fraise <BR>
  <INPUT type="checkbox" name="fruit" value="Banane"> Banane <BR>
  <INPUT type="checkbox" name="fruit" value="Pomme"> Pomme <BR>
</FORM>
<SCRIPT language=javascript>
  document.getElementsByName("fruit")[0].checked = true;
  document.getElementsByName("fruit")[1].checked = true;
  document.getElementsByName("fruit")[2].checked = true;
</script>
```

Accès en utilisant *getElementById*

- Object document.getElementById(String id)

Retourne un objet HTML à partir de son id, défini dans la propriété id de la balise de l'objet.

```
<input type="text" id="champ_input"><br>
<input type="button" onclick="f()" value="modifier le champ">
<script>
function f()
{
   var obj = document.getElementById("champ_input")
   alert('le champ a pour valeur : "'+obj.value+'"')
   obj.value="autre valeur"
   alert('maintenant il contient : "'+obj.value+'"')
}
</script>
```

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 principale soit en position 0).
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.

Exemple length :

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

Exemple 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);
```

Exemple indexOf() :

```
variable="Javascript"
var1="script"
x=variable.indexOf(var1,0); x vaut 4
```

Exemple lastIndexOf() :

```
variable="Javascript"
var1="a"
x=variable.indexOf(var1,0);
                ici x vaut 1 soit la position du premier a.
x=variable.lastIndexOf(var1,9);
                ici x vaut 3 soit la position du second a.
```

Exemple substring () :

```
str="Javascript";
str1=str.substring(0,4);  //retourne Java
str2="Javascript".substring(0,4);
str3=str.substring(6,9); //retoune rip
```

Exemple toLowerCase() et toUpperCase() :

```
str="JavaScript";
str1=str.toLowerCase();
str2="JavaScript".toLowerCase();
str3=str.toUpperCase();
str4="JavasScript".toUpperCase();
```

Objet math

Méthode	Description
abs(nb)	Retourne la valeur absolue de nb
ceil(nb)	Retourne le plus petit entier plus grand ou égal de nb
cos(nb)	Retourne le cosinus de nb
exp(nb)	Retourne exponentiel nb
floor(nb)	Retourne le plus grand entier plus petit ou égal à nb (partie entière de nb)
max(nb,nb2)	Retourne le plus grand de deux nombres (ici nb ou nb2)
min(nb,nb2)	Retourne le plus petit de deux nombres (ici nb ou nb2)
pow(nb,exposant)	Retourne nb exposant (où nb est la base de la puissance)
random()	Retourne alétoirement un nombre entre 0 et 1
sin(nb)	Retourne le sinus de nb
sqrt(nb)	Retourne la racine carré de nb

Objet math

- Exemples

```
x=Math.sqrt(y);
x=Math.random();
x=Math.min(y,z);
x=Math.ceil(y);
```

Objet Date

Méthode	Description	Type de valeurs retournée
getDate()	Permet de récupérer la valeur du jour du mois	L'objet retourné est un entier (entre 1 et 31) qui correspond au jour du mois
getDay()	Permet de récupérer la valeur du jour de la semaine pour la date spécifiée	•L'objet retourné est un entier qui correspond au jour de la semaine : 0: dimanche •1: lundi ...
getFullYear()	Permet de récupérer la valeur de l'année sur 4 chiffres pour la date passée en paramètre	L'objet retourné est un entier qui correspond à l'année (XXXX) :

Objet Date

Méthode	Description	Type de valeurs retournée
getHours()	Permet de récupérer la valeur de l'heure	L'objet retourné est un entier (entre 0 et 23) qui correspond à l'objet Date.
getMinutes()	Permet de récupérer la valeur des minutes	L'objet retourné est un entier (entre 0 et 59) qui correspond aux minutes de l'objet Date.
getMonth()	Permet de récupérer le numéro du mois	•L'objet retourné est un entier (entre 0 et 11) qui correspond au mois : 0: janvier •1: février ...
getSeconds()	Permet de récupérer le nombre de secondes	L'objet retourné est un entier (entre 0 et 59) qui correspond aux secondes de l'objet passé en paramètre.

Objet Date

Méthode	Description	Type de valeurs retournée
getTimezoneOffset()	Retourne la différence entre l'heure locale et l'heure GMT (Greenwich Mean Time)	L'objet retourné est un entier, il représente le nombre de minutes de décalage
getYear()	Permet de récupérer la valeur de l'année sur 2 chiffres pour l'objet Date.	L'objet retourné est un entier qui correspond à l'année (XX) : <script language="Javascript"> </script>

Objet Date

- Exemples :

```
variable_date= new Date("Wed, 28 Jul 1999 15:15:20 ");

an=variable_date.getYear();
mois=variable_date.getMonth();
jourm=variable_date.getDate();
```

Objet Date

- Assigner des valeurs aux propriétés date:

variable.setYear(x);: Assigne une année à l'actuelle valeur de variable sous forme d'un entier supérieur à 1900.

variable.setMonth(x); : Assigne un mois à l'actuelle valeur de variable sous forme d'un entier compris entre 0 et 11.

variable.setDate(x); : Assigne un jour du mois à l'actuelle valeur de variable_date sous forme d'un entier compris entre 1 et 31.

…

Gestion des tableaux

- L'objet Array (ou tableaux) est une liste d'éléments indexés dans lesquels on pourra ranger (écrire) des données ou aller reprendre ces données (lire).

➢ Décalaration

```
nom_du_tableau = new Array (x);
```

où x est le nombre d'élément du tableau.

➢ Définition des éléments

```
nom_du_tableau[i] = "elément";
```

Gestion des tableaux

- Exemple :

```
Nom= new Array(3);
```

ajout des données :

```
Nom[0]="Said";
Nom[1]="Ali";
Nom[2]="Mohamed";
```

pour accéder un élément, on emploie :

```
document.write(Nom[2]);
```

Gestion des tableaux

➢ Propriétés et méthodes

Elément	Description
length	Retourne le nombre d'éléments du tableau.
join()	Regroupe tous les éléments du tableau dans une seule chaîne. Les différents éléments sont séparés par un caractère séparateur spécifié en argument. Par défaut, ce séparateur est une virgule.
reverse()	Inverse l'ordre des éléments (ne les trie pas).
sort()	Retourne les éléments par ordre alphabétique (à condition qu'ils soient de même nature)

Gestion des tableaux

➢ Propriétés et méthodes – Exemples

- document.write(Nom.join());
  donne comme résultat : Said,Ali,Mohamed.
- document.write(Nom.join("-"));
  a comme résultat : Said-Ali-Mohamed..
- document.write(Nom.reverse().join("-"));
  donne : Mohamed-Ali-Said.

Gestion des tableaux

- **Tableau à deux dimensions**

On peut créer des tableaux à deux dimensions (et plus encore) par un petit artifice de programmation. On déclare d'abord un tableau à 1 dimension de façon classique :

nom_du_tableau = new Array (x);

Ensuite, on déclare chaque élément du tableau comme un tableau à une dimension :

nom_du_tableau[i] = new Array(y);

objet window

Les méthodes mises en œuvre sont :

- **Open**() ouvre une nouvelle fenêtre.
- **Close**() ferme la fenêtre en cours.

La syntaxe est :

[window.]open("URL","nom_de_la_fenêtre","caractéristiques")

- URL: est l'URL de la page que l'on désire afficher dans la nouvelle fenêtre. Si on ne désire pas afficher un fichier htm existant, on mettra simplement "".
- Caractéristiques : est une liste de certaines ou de toutes les caractéristiques de fenêtre suivantes

objet window

➢ Caractéristiques

• **toolbar**=yes ou no	Affichage de la barre d'outils
• **location**=yes ou non	Affichage de champ d'adresse
• **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

objet window

➢ Exemples

Fermeture :

```
<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.

Ouverture :

```
<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')">
    (sans espaces ni passage à la ligne)
</FORM>
```

TP – JavaScript : Exercice 1

- **Le but de cet exercice est de réaliser un questionnaire de type QCM en HTML et JavaScript.**
  - **Un formulaire (ou questionnaire) sous forme de QCM et utilisant les objets du langage JavaScript,**
  - **une procédure de correction dynamique du questionnaire (écrit en JavaScript)**
  - **Un corrigé complet du test dans une page HTML.**
- Le QCM doit contenir 5 questions, 3 à choix multiples et 2 à choix simples. Le formulaire doit contenir aussi 3 boutons :
  - Bouton pour initialiser le QCM
  - Bouton pour afficher, dans une nouvelle fenêtre, toutes les questions et les réponses du QCM. (les réponses justes sont soulignées et colorées par une couleur verte).
  - Bouton qui affiche dans une nouvelle fenêtre le résultat de test
  - Par exemple : - la réponse de la question 1 est correcte
- - la réponse de la question 2 est incorrecte
- Les dimensions des fenêtres sont ajustées selon le contenu

TP – JavaScript : Exercice 2 & 3

Exercice 2

- Écrire un code qui permet de vérifier si l'email tapé par l'utilisateur, dans une zone de saisie, est valide ou pas (par exemple vérifier l'existence de @, et au moins une lettre ou chiffre avant, un point après @).

Exercice 3

- Écrire une fonction appelée **trier()** qui prend une liste (array) de nombres et retourne la liste triée.

TP – JavaScript : Exercice 4

- Ecrire un programme qui détermine la somme de trois mesures de temps données en heures, minutes et secondes et qui donne le résultat en jours, heures, minutes et secondes comme suit: