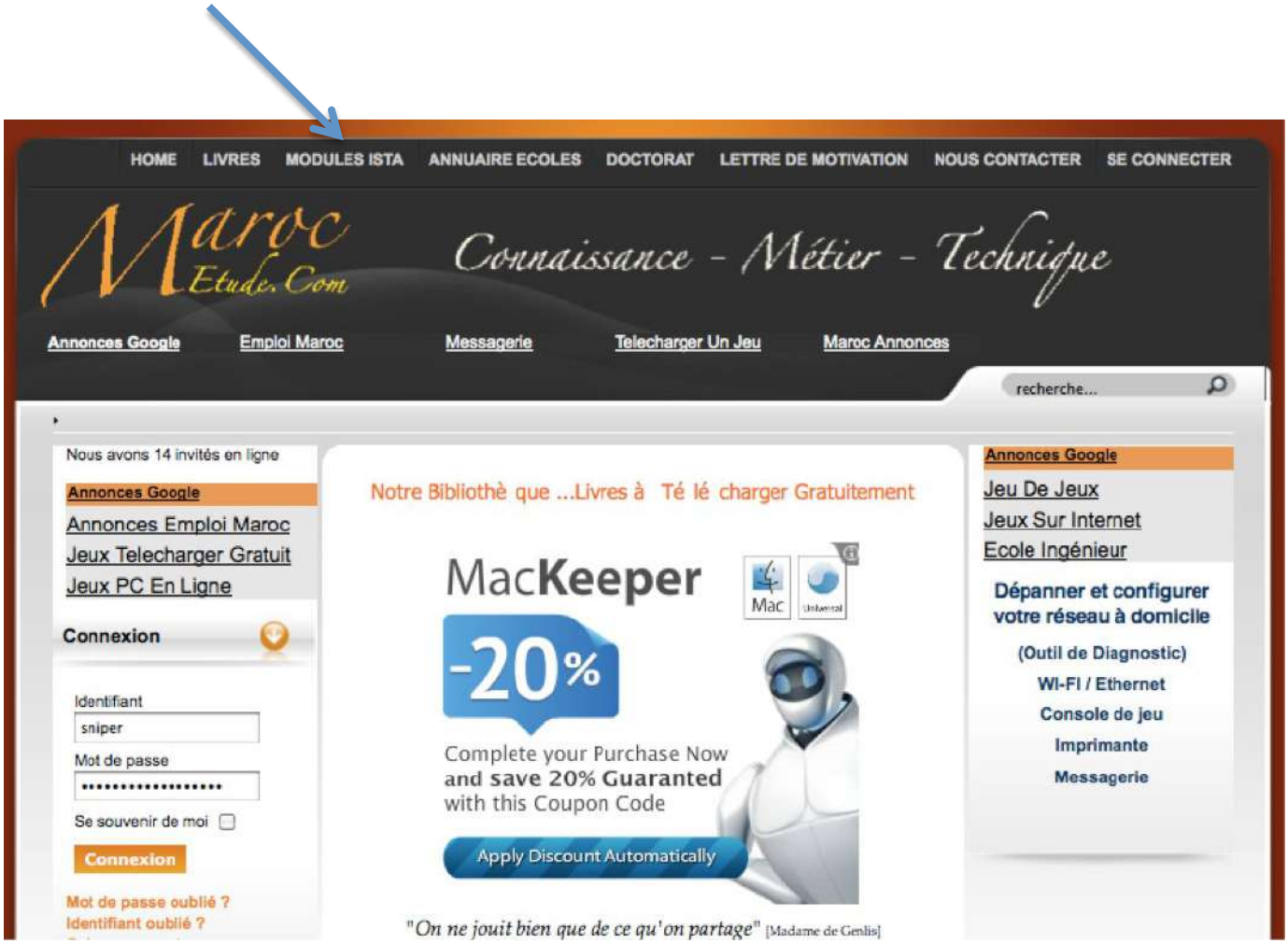
PORTAIL DE LA FORMATION PROFESSIONNELLE AU MAROC

Télécharger tous les modules de toutes les filières de l'OFPPT sur le site dédié à la formation professionnelle au Maroc : **www.marocetude.com**

Pour cela visiter notre site www.marocetude.com et choisissez la rubrique :

MODULES ISTA



PRATIQUE DE JAVASCRIPT

Pour étendre les possibilités des pages HTML, pour les rendre plus interactives et pour effectuer certains traitements et contrôles au niveau du client avant de soumettre la page web à un serveur, le développeur utilise des scripts. Deux langages de scripts sont très utilisés actuellement sur le marché: JavaScript(de Netscape) et VBScript(de Microsoft).
Par rapport à VBScript, JavaScript présente l'avantage d'être supporté par les deux navigateurs les plus connus au monde Netscape Navigator et Internet Explorer alors que VBScript n'est pas supporté par Netscape Navigator. En plus il est léger et orienté objet.
Remarque : Malgré quelques ressemblances apparente JavaScript est très différent de Java.

UTILISATION

Les Scripts JavaScript sont intégrés dans une page web de plusieurs manières :

- En écrivant le script entre une balise <Script> et </Script> :

```
<Script Language="JavaScript">
    <!--
        Ecriture Script
    -->
    <NoScript>
        Traitement si le navigateur n'accepte pas les scripts
    </NoScript>
</Script>
```

- En associant le script à un événement :

```
<NomBalise…..onEvénement="Ecriture Script">
```

- En associant le script à un lien :

```
<a href="JavaScript:Ecriture Script"…>…</a>
```

Remarque : Pour empêcher que le script dans href ne remplace le document courant, on applique l'opérateur void, qui neutralise toute valeur ou tout effet de retour : …

- En écrivant le script dans un fichier ayant l'extension .js et en l'appelant dans la page web à l'aide de l'instruction :

```
<Script Language="JavaScript" src="FichierScript.js></Script>
```

Remarque:

- Les Scripts peuvent être placés dans le Head ou dans le Body. Dans le Head ils seront reconnus dés le chargement de la page.
- Un gestionnaire d'événement est la procédure particulière attachée à un événement dans une balise HTML
- Il est possible d'utiliser plusieurs gestionnaires d'événements dans une même balise

CONCEPTS DE PROGRAMMATION

Insertion de commentaires

- Sur une ligne :

```
//Commentaire
```

- Sur plusieurs lignes :

```
/*Commentaire
Commentaire*/
```

utilisation des variables

Pour déclarer une variable :

```
var Variable1 [=Valeur_initiale1], Variable2 [=Valeur_initiale2], …
```

Remarque : La déclaration se fait de manière implicite si une variable est utilisée dans le script mais qu'elle n'a pas été déclarée à l'aide de var.

utilisation des tableaux (objet Array)

- Déclaration:
  - Tableaux à une dimension :
    - Le nombre d'éléments du tableau est inconnu :

var tableau=new Array()

· Le nombre d'éléments du tableau est connu mais le contenu est non connu :
var tableau=new Array(NombreEléments)

· Le nombre d'éléments du tableau est connu et le contenu est connu :
var Tableaux=new Array(ElémentTableau1, ElémentTableau2,…)

ou

var Tableaux=[ElémentTableau1, ElémentTableau2,…]

- Tableaux à plusieurs dimensions : Ils sont gérés comme des tableaux de tableaux à une dimension
var Tableau =new Array(Vide|Nombre_Elément|ListeEléments)
Tableau[0]=new array(Vide|Nombre_Eléments|ListeEléments)
Tableau[1]=new array(Vide|Nombre_Eléments|ListeEléments)
…

❋ Remplissage
- Tableaux à une dimension :
Tableau[position]=valeur
ou
Tableau[position]= new array(Vide|Nombre_Eléments|ListeEléments)
- Tableaux à plusieurs dimensions :
Tableau[positionLigne,PositionColonne]=valeur
ou
Tableau[positionLigne,PositionColonne]= new array(Vide|Nombre_Eléments|ListeEléments)

Remarque : Le premier élément du tableau commence à l'indice 0

❋ Quelques propriétés
- length : Retourne le nombre d'éléments du tableau (Tableau.length)

❋ Quelques méthodes
- join() : Regroupe tous les éléments avec un caractère de séparation aléatoire (Tableau.join())
- join(car) : Regroupe tous les éléments avec un caractère de séparation car (Tableau.join(car))
- reverse() : Inverse l'ordre des éléments contenus dans le tableau (Tableau.reverse())
- sort() : Trie le suivant l'ordre croissant (Tableau.sort())

Structures conditionnelles

❋ if

```
if (condition)
{Instructions}
else
{Instructions}
```

❋ switch

```
switch(expression)
{
        case const1:
                instructions
                break
        case const2:
                Instructions
                break
        ...
        default:
                Instructions
}
```

❋ ?

(Condition)?Instructions si condition réalisée : Instructions si condition non réalisée

Structures répétitives

❋ for
- for (Initialisation de la boucle, Condition de répétition, Variation des compteurs de la boucle)
{Instructions}
- for (variable in tableau)

{Instructions}

- for (variable in objet)
  {Instructions}

❋ while

while (condition)
{Instructions}

❋ do

{Instructions}
while (condition)

fonctions

❋ Déclaration

```
function nom_Fonction(argument1, argument2…)
{
        instructions
        [return valeur]
}
```

❋ Appel

L'appel se fait :

A partir d'une autre fonction : NomFonction(…)

ou

Suite à un événement : <NomBalise …..onEvénement="NomFonction(….)">

ou

A partir d'un lien : …

Remarque : Pour empêcher que le script dans href ne remplace le document courant, on applique l'opérateur void, qui neutralise toute valeur ou tout effet de retour : …

ou

Si la fonction retourne une valeur, à partir d'une variable : Variable=NomFonction(….)

Remarque :

- Les variables déclarées au niveau du Script sont considérées comme globales.
- Les variables déclarées dans des fonctions sont considérées comme locales si l'instruction **var** est utilisée dans leur déclaration (déclaration explicite) sinon elles sont considérées comme globales.

Manipulation des chaînes de caractères(objet String)

Une chaîne de caractères est composée d'une suite de caractères quelconques, y compris des balises HTML. Des caractères spéciaux peuvent aussi être insérés dans les chaînes :\n (nouvelle ligne), \r (Entrée), \t (tabulation), \f (saut de page), \b (retour arrière), \' pour une apostrophe…

Pour concaténer deux chaînes de caractères on utilise le symbole +

❋ Quelques propriétés

- length : Retourne la longueur d'une chaîne de caractère (chaîne.length)

❋ Quelques méthodes

- toLowerCase() : Convertit une chaîne de caractères en minuscules(chaîne.toLowerCase())
- toUpperCase() : Convertit une chaîne de caractères en majuscules(chaîne.toUpperCase())
- charAt(p) : Retourne le caractère à la position p (chaîne.charAt(p)
- indexOf(Ch) : Retourne la position de ch dans une chaîne de caractères (Chaine1.indexOf(Chaîne2))
- indexOf(Ch, p) : Retourne la position de ch dans une chaîne de caractères en commençant la recherche par la position p (Chaine1.indexOf(Chaîne2, p))
- lastIndexOf(Ch) : Retourne la position de ch dans une chaîne de caractères en commençant la recherche par la fin(Chaine1.lastIndexOf(Chaîne2))
- lastIndexOf(Ch, p) : Retourne la position de ch dans une chaîne de caractères en commençant la recherche par la fin et en s'arrêtant à la position p (Chaine1.lastIndexOf(Chaîne2, p))
- substring(p1,p2) : Retourne une sous-chaîne allant de la position p1 à la position p2-1 (chaîne.substring(p1,p2))
- split(séparateur) : Fractionne une chaîne de caractères en se basant sur le séparateur et retourne les différentes parties dans un tableau de chaînes de caractères (Tabelau=Chaîne.split(séparateur))
- isNaN(élément) : Revoie true si élément est un nombre et false sinon (variable=isNaN(élément))

…

Utilisation des fonctions date et heure (objet Date)

- Pour manipuler la date en cours

  Variable =new Date()

Remarque : La date qui sera stockée dans variable est sous la forme suivante :

"NomJour Mois jourMois Heure:Minutes:Secondes Année"

- Pour manipuler une autre date

  Variable =new Date(date)

date peut être donnée sous les formes :

- "année, mois, jour"
- "année, mois, jour, heures, minutes , secondes"
- "NomMois JourMois , Année Heures:Minutes:Secondes"

- Méthodes
  - getYear() : Retourne l'année d'une date (>1900) (UneDate.getYear())
  - setYear(année) : Modifie l'année d'une date (UneDate.setYear(année))
  - getMonth : Retourne le mois (entre 0 et 11) d'une date (UneDate.getMonth())
  - setMonth(mois) : Modifie le mois d'une date (UneDate.setMonth(mois))
  - getDate() : Retourne le numéro du jour du mois (entre 1 et 31) d'une date (UneDate.getDate())
  - setDate(jourMois) : Modifie le jour du mois d'une date (UneDate.setDate(jourMois))
  - getDay() : Retourne le jour de la semaine (entre 0 et 6) d'une date (UneDate.getDay())
  - setDay(jourSemaine) : Modifie le jour de semaine d'une date (UneDate.setDay(jourSemaine))
  - getHours() : Retourne l'heure (entre 0 et 23) d'une date (UneDate.getHours())
  - setHours(heure) : Modifie l'heure d'une date (UneDate.setHours(heure))
  - getMinutes() : Retourne les minutes(entre 0 et 59) d'une date (UneDate.getMinutes())
  - setMinutes(minutes) : Modifie les minutes d'une date (UneDate.setMinutes(minutes))
  - getSeconds() : Retourne les secondes (entre 0 et 59) d'une date (UneDate.getSeconds())
  - setSeconds(secondes) : Modifie les secondes d'une date (UneDate.setSeconds(secondes))
  - getTime() : Retourne le nombre de Millisecondes écoulées depuis le 1 janvier 1970 00:00:00 (UneDate.getTime())
  - setTime (nombreMillisecondes) : Modifie le nombre de Millisecondes écoulées depuis le 1 janvier 1970 00:00:00 (UneDate.setTime(nombreMillisecondes))

  …

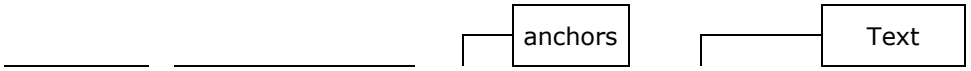
Utilisation des fonctions mathématiques (objet Math)

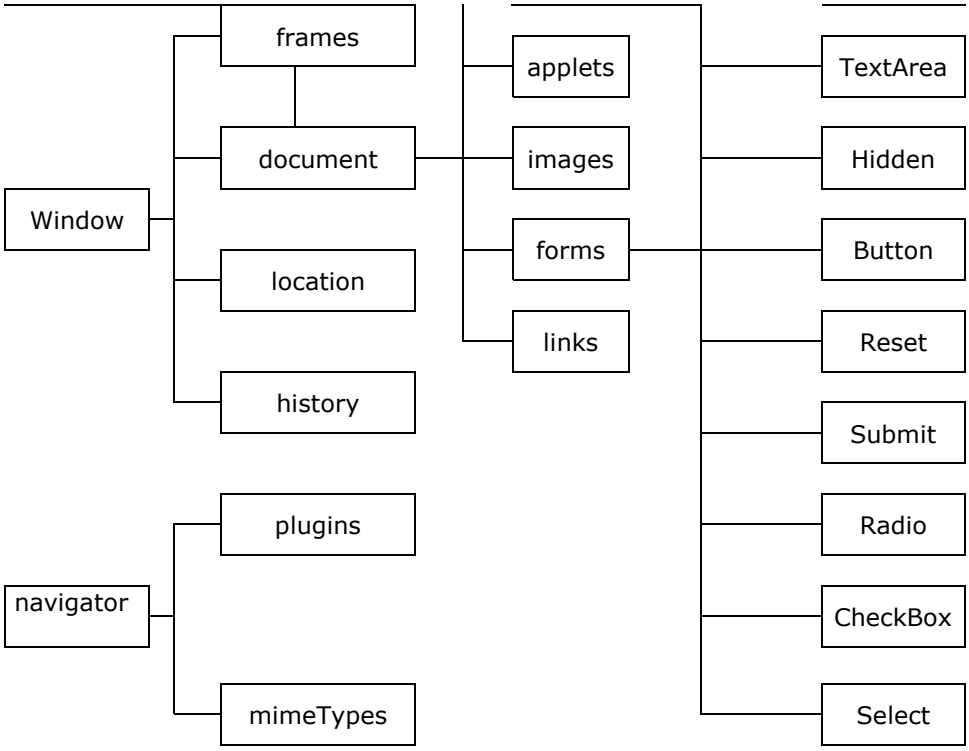
- Quelques propriétés
  - PI : constante PI (Math.PI)
- Quelques méthodes
  - abs(nombre) : Retourne la valeur absolue d'un nombre (Math.abs(nombre))
  - round(nombre) : Arrondit le nombre à l'entier le plus proche (Math.round(nombre))
  - max(nombre1,nombre2) : Renvoie le plus grand des deux nombres (Math.max(nombre1, nombre2))
  - min(nombre1,nombre2) : Renvoie le plus petit des deux nombres (Math.min(nombre1, nombre2))
  - pow(nombre1,nombre2) : Renvoie le résultat de nombre1 puissance nombre2 (Math.pow(nombre1, nombre2))
  - sqrt(nombre) : Retourne la racine carrée de nombre (Math.sqrt(nombre))
  - parseInt(chaîne) : Evalue une chaîne de caractère en un nombre entier (parseInt(chaîne))
  - parseFloat(chaîne) : Evalue une chaîne de caractère en un nombre à virgule flottante (parseFloat(Chaîne))
  - eval(chaîne) : Evalue une chaîne de caractère en un nombre (eval(Chaîne))
  - random() : Retourne un nombre aléatoire entre 0 et 1 (1 non compris) (variable=Math.random())

  …

OBJETS DU NAVIGATEUR

Hiérarchie





Pour accéder à un objet, il faut se référer à sa position dans la hiérarchie. Depuis le sommet, il faut suivre l'arborescence jusqu'à atteindre l'objet souhaité.
Pour lire ou modifier une propriété : hièrarchie.objet.propriété
Pour appeler une méthode : hièrarchie.objet.méthode(…)
Exemples :

- Soit une zone de texte txt1 qui se trouve dans le formulaire form1 qui se trouve dans un document page1.htm ne contenant aucun frame. Cette zone de texte est identifiée par : window.document.form1.txt1 qu'on peut également noter form1.txt1
- Soit une zone de texte txt1 qui se trouve dans le formulaire form1 qui se trouve dans un cadre (frame) frame1 dans un document page1.htm. Cette zone de texte est identifiée à partir d'un autre cadre du même document par : parent.frame1.form1.txt1
- Pour modifier le contenu d'une zone de texte, on utilise sa propriété value : form1.txt1.value=valeur
- Soit une image d'un document HTML page1.htm dont l'attribut name est img1. Pour faire référence à cette image : window.document.images["img1"] ou document.images["img1"]
- Pour modifier la propriété src de cette image : document.images["img1"].src="cheminImage"

Remarque : L'objet window représente la fenêtre du navigateur et regroupe les objets contenus dans cette fenêtre. Il est particulièrement utilisé pour la création des fenêtres volantes (popup). Pour des raisons pédagogiques il sera traité en détail après l'objet document.

Objet document

Représente le document en cours. Il offre un ensemble de propriétés, de méthodes et d'objets permettant la manipulation du document et de son contenu.

- Quelques propriétés
  - bgColor : Retourne ou affecte une couleur de fond au document
  - fgColor : Retourne ou affecte une couleur de texte au document
  - linkColor : Retourne ou affecte une couleur pour les liens hypertextes
  - vlinkColor : Retourne ou affecte une couleur pour les liens hypertextes visités
  - alinkColor : Retourne ou affecte une couleur pour les liens hypertextes actifs
  - title : Retourne ou affecte un titre au document
  - location : Retourne ou affecte l'URL du document
  - cookie : Chaîne de caractères reflétant le contenu du fichier cookie.txt
  - lastModified : Retourne la date de dernière modification du document
  - images[indice] ou images["nomImage"] : Représente un tableau contenant toutes les images du document. Il est possible ainsi d'accèder à n'importe quelle image contenue dans le document, de consulter ou de modifier ses caractéristiques.

- anchors[indice] ou anchors["nomAncre"] : Représente un tableau contenant toutes les ancres du document. Il est possible ainsi d'accèder à n'importe quelle ancre contenue dans le document, de consulter ou de modifier ses caractéristiques.
- links[indice] ou links["nomLien"] : Représente un tableau contenant tous les liens (link) du document. Il est possible ainsi d'accèder à n'importe quelle lien contenu dans le document, de consulter ou de modifier ses caractéristiques.
- forms[indice] ou forms["nomFormulaire"] : Représente un tableau contenant tous les formulaires du document. Il est possible ainsi d'accèder à n'importe quel formulaire contenu dans le document, de consulter ou de modifier ses caractéristiques.
- applets[indice] ou applets["nomApplet"] : Représente un tableau contenant toutes les applets du document. Il est possible ainsi d'accèder à n'importe quelle applet contenu dans le document, de consulter ou de modifier ses caractéristiques.

Remarque : Une applet est un programme écrit et compilé par un langage de programmation qu'on a integré au niveau du document.

❋ Quelques méthodes
- write("Texte") : Ecrit Texte sur le document. Texte peut être un simple message, des balises HTML pour la construction des éléments du document, des variables…ou une concaténation de l'ensemble de ces éléments (écrire dans le flux d'un document)
- writeln("Texte") : fait le même travail que write mais retourne à la ligne à la fin.
- …

Objet form

Représente un formulaire contenu dans le document. Pour faire référence à un formulaire dans un document : document.forms[index] ou document.forms["NomForm"] ou NomForm

❋ Quelques propriétés :
- action : Spécifie l'action à exécuter en cas d'appel de submit ou d'activation d'un bouton submit associé au formulaire
- name : Définit le nom du formulaire
- target : Définit l'emplacement où sera affiché l'élément qui sera ouvert suite à l'exécution de l'action du formulaire
- method : Spécifie la méthode utilisée lors du transfert d'information depuis un client vers un serveur

❋ Quelques méthodes
- reset() : réinitialise le formulaire
- submit() : exécute l'action associée au formulaire (propriété action)

❋ Quelques propriéts pour la manipulation des objets :
- Objet text :
  value : Définit et retourne la valeur contenue dans un champ de type text
- champ select : propriété selectedIndex, value et text
  selectedIndex : Retourne l'index de l'élément sélectionné dans un champ select
  value : retourne la valeur de l'option sélectionnée dans un champ select
  text : retourne le texte contenu entre <option> et </option> pour l'élément sélectionné
- champ radio et checkbox :
  checked : Détermine ou retourne si l'élément a été coché ou non (true ou false)
  value : retourne la valeur du bouton ou de la case sélectionnée

Remarque : pour identifier les boutons radio ou cases à cocher appartenant au même groupe et donc ayant la même propriété name, on leur fait référence de la manière suivante : …NomGroupe[ordreDansleGroupe].propriété

Objet window

Remarque : le mot clé **self** désigne la fenêtre window (fenêtre principale). window et self désignent le même objet et peuvent donc être utilisés indifféremment.

❋ Quelques propriétés
- name : Attribue et retourne le nom de la fenêtre (window.name="Nom")

Remarque :
- Un nom de fenêtre peut être utilisé pour désigner le fenêtre comme cible pour un lien hypertexte <a href="… " target="NameFenêtre">
- Puisqu'un nom de fenêtre reste conservé aussi longtemps que la fenêtre est ouverte, il peut être utilisé pour transmettre certaines valeurs d'une feuille à une autre.

- defaultstatus : Affecte et retourne le texte par défaut qui sera affiché dans la barre d'état du navigateur
- status : Affecte et retourne le contenu actuel de la barre d'état.

exemple :
```
<html>
<head><script language="javascript">window.defaultStatus="Texte Barre d'état"
</script></head>
<body>
<input type="text" name="txtNom" onmouseover="window.status='Saisir le nom ici'">
</body>
</html>
```
Au démarrage, la barre d'état contient le texte "Texte Barre d'état". En faisant passer la souris sur la zone de texte txtNom, le texte "Saisir le nom ici" s'affiche sur la barre d'état mais une fois le pointeur de la souris hors de la zone de texte, la barre d'état reprend le texte "Texte Barre d'état"

- closed : Renvoie true si une fenêtre qui vient d'être ouverte a été refermée (Fenêtre.closed)
- location : Retourne l'URL de la fenêtre en cours(window.location)
- length : Retourne le nombre de frames dans la fenêtre en cours (window.length)

❋ Quelques méthodes

- alert('message') : Affiche une boîte de dialogue contenant le message (alert('message') ou alert('message' + variable…))
- prompt("Message",Valeur par défaut) : Affiche une boite de dialogue avec un champ de saisie, un bouton "Ok" et un bouton "Annuler". Si l'utilisateur a appuyé sur le bouton "Annuler", cette méthode retourne la valeur null. Si l'utilisateur n'a rien saisi et qu'il a appuyé sur le bouton "Ok", la méthode retourne le vide ("") sinon elle retourne la valeur saisie par l'utilisateur (variable=prompt("Message",Valeur par défaut)
- confirm("Message") : Ouvre une boite de dialogue avec deux boutons pour "OK" et "Annuler". Si l'utilisateur a appuyé sur Ok, la méthode retourne true sinon elle retourne falseJudicieux pour forcer l'utilisateur à prendre une décision qui sera traitée dans la suite du programme. Attend comme paramètre un texte interrogatif pour la décision oui/non. Renvoie comme résultat la décision de l'utilisateur.
- open("Fichier à Charger", "Nom Fenêtre (propriété name)"[,"CaractèristiqueFenêtre1=Valeur, CaractèristiqueFenêtre2=Valeur,…"]) : Ouvre une nouvelle fenêtre, en lui affectant et éventuellement en définisant ses caractèristiques d'affichage :
  - width=valeur en pixel : Largeur de la fenêtre
  - height=Valeur en pixel : Hauteur de la fenêtre
  - top=Valeur en pixel : Position verticale du coin supérieur gauche de la nouvelle fenêtre
  - left=Valeur en pixel : Position horizontale du coin supérieur gauche de la nouvelle fenêtre
  - location=yes|no : Indique si la barre d'adresse s'affichera ou non sur la nouvelle fenêtre (par défaut no)
  - menubar=yes|no : Indique si la barre de menus s'affichera ou non sur la nouvelle fenêtre (par défaut no)
  - toolbar=yes|no : Indique si la barre d'outils s'affichera ou non sur la nouvelle fenêtre (par défaut no)
  - status=yes|no : Indique si la barre d'état s'affichera ou non sur la nouvelle fenêtre (par défaut no)
  - scrollbars=yes|no : Indique si les barres d'adresse s'affichera ou non sur la nouvelle fenêtre (par défaut no)
  - resizable=yes|no : Indique si l'utilisateur a oui ou non le droit de redimensionner la nouvelle fenêtre (par défaut no)
  - dependent= yes|no : Indique si la nouvelle fenêtre sera fermée si sa fenêtre parent a été fermée (celle qui a demandé son ouverture)

  Remarque :
  - Pour faire référence à une fenêtre, il faut qu'au moment de l'ouverture leui attribuer une variable de référence :

    Variable=window.open("fichier","NomFenêtre"[,"caractèristiquesFenêtre"])

- Les propiétés et méthdes de cette nouvelle fenêtre pourront être appelés de la manière suivante :
  Variable.propriété ou Variable.méthode(…)
- Pour accèder de la nouvelle fenêtre à la fenêtre appllante, aux objets, aux propriétés et aux méthodes de cette fenêtre appellante, il faut utiliser le mot clé opener à la place de window.

- close() : Ferme une fenêtre (fenêtre.close())
- moveTo(valeurLeft, ValeurTop) : déplace une fenêtre en positionnant la propriété left à valeurLeft et la propriété top à ValeurTop (fenêtre. moveTo(valeurLeft, ValeurTop))
- moveBy(valeurLeft, ValeurTop) : Déplace une fenêtre d'autant de valeurLeft par rapport à sa posisiton actuelle et de valeurTop par rapport à sa position verticale actuelle (valeurLeft et ValeurTop peuvent être positives ou négatives)
- resizeTo(valeurLargeur,valeurHauteur) : Modifie la taille de la fenêtre en affectant valeurLargeur à la largeur et valeurHauteur à la hauteur
- resizeBy(valeurLargeur,valeurHauteur) : redimentionne la fnêtre en modifiant sa largeur de valeurLargeur sur la droite et de valeurHauteur sur le bas (valeurLargeur et ValeurHauteur peuvent être positives ou négatives)
- print() : Imprime le contenu d'une page (fenêtre.print())
- focus() :Rend une fenêtre active (fenêtre.focus())
- stop() : Correspond à un cliquement sur le bouton "Stop" du navigateur. Le chargement d'une page en est interrompu.
- home() : affiche la page d'acceuil du navigateur (correspond à un click sut le bouton Home de la barre d'outils du navigateur)
- forward() : affiche la page suivante (correspond à un click sut le bouton Suivant de la barre d'outils du navigateur)
- back() : affiche la page précédente (correspond à un click sut le bouton Précédent de la barre d'outils du navigateur)
- setTimeout("fonction()|Instructions",t) : Permet de déclencher une fonction ou des instructions JavaScript après un temps t en millisecondes
  (variableCompteur=setTimeout("fonction()|Instructions",t))
- clearTimeout(variableCompteur) : Arrête l'éxécution de la fonction ou des instructions JavaScript associés à setTimeout avant l'expiration du délai
- setInterval("fonction()|Instructions",t) : Exécute une fonction ou des instructions JavaScript toutes les t millisecondes (variableCompteur=setInterval("fonction()|Instructions",t))
- clearInterval(variableCompteur) : Arrête l'éxécution de la fonction ou des instructions JavaScript associés à setInterval avant l'expiration du délai

Objet history

Offre les propriétés et méthodes nécessaires à l'accès à la liste d'historique stockée par le navigateur.

- Quelques méthodes
  - back() : Charge le document précédent
  - forward() : Charge le document suivant
  - go(p): Charge le document se trouvant à la position : position courante + p (p>0 ou p<0)

Objet navigator

- Quelques propriétés
  - navigator.appCodeName : retourne un code identifiant le navigateur mais IE et Nestscape ont le même nom de code
  - navigator.appName : nom du navigateur (Netscape ou Microsoft Internet Explorer)
  - navigator.appVersion : sous la forme (NuméroVersion(Sytème d'exploitation, codeNationalité de la version)

LES STYLES ET JAVASCRIPT

Les propriétés de styles en JavaScript sont déduites de celles utilisées en CSS en respectant la règle suivante : pour chaque propriété composée de deux mots séparés par un tiret, le tiret va disparaître et la première lettre du deuxième mot sera en majuscule.

Pour modifier le style d'un élément

NomObjet.style.propriétéStyle=valeur

ou

Id.style.propriétéStyle=valeur

Pour modifier une règle de style à partir d'une feuille de style

Une règle de style peut être récupérée à partir d'une feuille de styles :

variable=document.styleSheets[position feuille Style].rules[position règle]

Ensuite il est possible de modifier le style pour cette règle :

variable.style.propriétéstyle=valeur

Remarque : les feuilles de styles (incorporées ou externes) sont classées par ordre. La première a l'indice 0. De même les règles de style définies dans une feuille de style donnée commencent par l'indice 0.

Pour ajouter une règle de style à une feuille de style

document.styleSheets[position feuille Style].addRule("ElementConcerné","Propriété style1: Valeur;Propriété Style2:Valeur…")

Pour supprimer une règle de style d'une feuille de style

document.styleSheets[position feuille Style].removeRule(Position règle)

Pour désactiver/activer une feuille de style

document.styleSheets[position feuille Style].disabled=true|false

Pour modifier la source d'une feuille de style

document.styleSheets[position feuille Style].href="Chemin feuille de style.css"

LES COOKIES

Les cookies sont des fichiers que les serveurs web créent sur les disques durs de leurs clients (visiteurs du site web) pour y mettre certaines informations et variables qu'ils pourront récupérer (les serveurs) à la prochaine visite du même client.

Ces informations peuvent être créée, lues et modifiées dans JavaScript à l'aide de la propriété cookie de l'objet document.

Il est possible d'avoir plusieurs cookies pour un seul site. Chaque cookie se compose d'un ensemble de champs :

Créer un cookie :

document.cookie="NomCookie=ValeurCookie[;expires=DateExpiration;path=Chemin;domain=Domaine Cookie;secure]"

- NomCookie : Nom à attribuer au cookies (obligatoire)
- valeurCookie : Données à stocker dans le cookie
- DateExpiration : Date à partir de laquelle le cookie sera désactivé et ne sera plus chargé par le serveur web. Si aucune date d'expiration n'a été spécifiée, le cookie est supprimé à la fermeture du navigateur et n'est pas enregistré. De même si on spécifie une date d'expiration qui est déjà passée.

Remarque : La date d'expiration est sous le format : *Jour abrégé,JJ-MM-AA HH:MM:SS GMT*

- Chemin : Indique le chemin pour lequel le cookie sera reconnu. Par défaut le cookie est reconnu sur le site à partir du dossier où il a été créé. Si on souhaite rendre un cookie visible sur tous le site il suffit d'attribuer "/" à path
- DomaineCookie : Site web concerné par le cookie, les autres sites devant l'ignorer. Par défaut il s'agit du site qui a écrit le cookie
- secure : si le mot clé **secure** est indiqué dans un cookie, le cookie ne sera transmis que si la connexion vers le serveur est sécurisée (protocole HTTPS).

Remarque :

- Un même "client" peut stocker un maximum de 300 cookies à 4000 octets chacun, dont 20 maximum pour un même serveur.
- L'emplacement des cookies dépend des navigateurs et des systèmes d'exploitation.

Modifier un cookie :

La réafectation de valeurs à un cookie dont le nom a été déjà enregistré le modifie.
document.cookie="NomCookie=ValeurCookie[;expires=DateExpiration;path=Chemin;domain=DomaineCookie; secure]"
Remarque : Si au cours de la modification, la date d'expiration n'a pas été spécifiée, le cookie est supprimé

Lire un cookie :

La propriété document.cookie stocke des informations sur tous les cookies créés. Pour accèder à la valeur d'un cookie, il faut la rechercher dans la chaine document.cookie sachant que deux cookie ne peuvent pas porter le même nom.

Remarque :
Pour ne pas avoir à écrire le programme de création et de lecture de cookies, il est préférable de créer des fonctions à utiliser au besoin :
Exemple : Pour la lecture d'un cookie

```
function lire_cookie(nomCookie)
{
	var ch1=document.cookie
	var lenCh1=ch1.length
	var ch2=nomCookie+"="
	var lenCh2=ch2.length
	var i=0
	while (i<lenCh1)
	{
		var j=i+lenCh2
		if (ch1.substring(i,j)==ch2)
		{
			var k=ch1.indexOf (";",j)
			if (k==-1)
			{
				k=lenCh1
			}
				return ch1.substring(j, k)
		}
		i=ch1.indexOf(" ",i)+1
		if (i==0)
			{ break}
	}
	return null
 }
```

Remarque : A partir du menu Outils/Options/Confidentialité de Internet Explorer, il est possible de déterminer le comportement que le client aura avec les cookies (bloquer, autoriser, demander à l'utilisateur…). La configuration choisie sera stockée dansla base de registre pour être chargé à chaque démarrage de la machine.