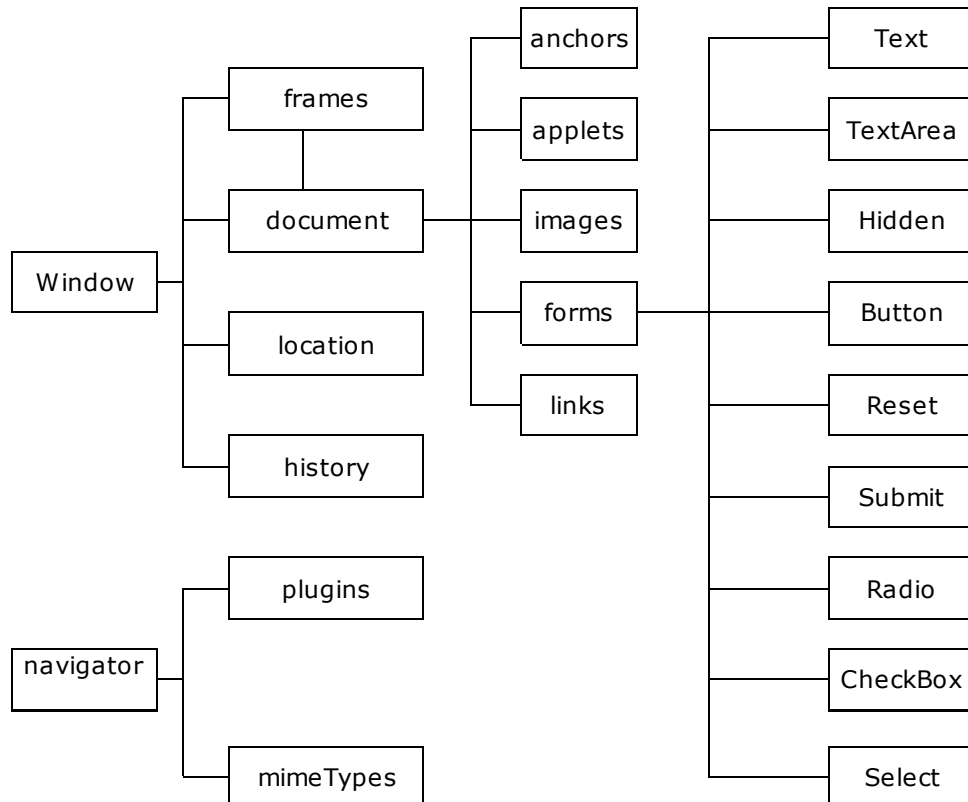


## OBJETS DU NAVIGATEUR

### *Hierarchie*



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 : `hiérarchie.objet.propriété`

Pour appeler une méthode : `hiérarchie.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'accéder 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 ancrs du document. Il est possible ainsi d'accéder 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'accéder 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'accéder 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'accéder 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 intégré 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 propriétés 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 boîte 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 boîte de dialogue avec deux boutons pour "OK" et "Annuler". Si l'utilisateur a appuyé sur Ok, la méthode retourne true sinon elle retourne false. Judicieux 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)",["CaractéristiqueFenêtre1=Valeur, CaractéristiqueFenêtre2=Valeur,..."]) : Ouvre une nouvelle fenêtre, en lui affectant et éventuellement en définissant ses caractéristiques 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 lui attribuer une variable de référence :  
Variable=window.open("fichier","NomFenêtre","caractéristiquesFenêtre")
  - Les propriétés et méthodes de cette nouvelle fenêtre pourront être appelés de la manière suivante :  
Variable.propriété ou Variable.méthode(...)
  - Pour accéder de la nouvelle fenêtre à la fenêtre appelante, aux objets, aux propriétés et aux méthodes de cette fenêtre appelante, 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 position 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) : redimensionne la fenêtre 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 cliquage sur le bouton "Stop" du navigateur. Le chargement d'une page en est interrompu.
  - home() : affiche la page d'accueil du navigateur (correspond à un click sur le bouton Home de la barre d'outils du navigateur)
  - forward() : affiche la page suivante (correspond à un click sur le bouton Suivant de la barre d'outils du navigateur)
  - back() : affiche la page précédente (correspond à un click sur 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'exécution 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'exécution 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 Netscape ont le même nom de code
- navigator.appName : nom du navigateur (Netscape ou Microsoft Internet Explorer)

- navigator.appVersion : sous la forme (NuméroVersion(Système 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 créées, 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 réaffectation 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 accéder à la valeur d'un cookie, il faut la rechercher dans la chaîne document.cookie sachant que deux cookies 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 dans la base de registre pour être chargée à chaque démarrage de la machine.