le cnam

JavaScript

Octobre 2013

Sommaire

le cnam

Introduction

JavaScript = technologie «côté client»

De manière générale, le code JavaScript est écrit dans le même fichier que le document HTML.

A défaut de pouvoir placer tout le code JavaScript dans l'en-tête de la page HTML, il faut y mettre au moins les définitions de fonctions, ce qui permet de les charger en premier.

Charger en premier le code des fonctions, permet d'éviter des messages d'erreur en cas de tentative d'accès à des fonctionnalités non encore chargées.

Appeler le code JavaScript (1)

1 - Appel par activation des attributs (liés à la réception d'événements) de certaines balises HTML

```
<html>
  <head><title> Mon premier Script </title></>
  <body  OnLoad = "alert ('Hello world !!!!')">
    ….
  </body>
</html>
```

Appeler le code JavaScript (2)

2 - Appel par un lien hypertexte

Il est possible d'intégrer la fonction dans un lien hypertexte en utilisant le pseudo protocole **JavaScript**.

```
Cliquez <a   href =''javascript : alert ('Hello world !!!!') "> ici </a>
```

Appeler le code JavaScript (3)

3 - Appel au sein du corps d'un document HTML

```
<html>
  <head><title> Mon premier Script </title><head>
  <body>
        <script type="text/javascript">


              alert ("Hello world !!!!") ;



        </script>
  </body>
</html>
```

le cnam

Eléments du langage

Octobre 2013

Variable

Une variable est un espace réservé en mémoire dont le contenu peut changer pendant la durée d'exécution du script. Une variable permet de stocker et de manipuler des données (objets ou classiques).

Pour déclarer une variable, on utilise le mot-clé **var** suivi du nom de la variable

Ex :

```
var pi = 3.14;

var message = "Bonjour";
```

Fonction

Une fonction permet de regrouper un ensemble d'instructions afin d'en faire une unité cohérente à laquelle on donne un nom, et à laquelle on peut faire appel à tout moment comme à une nouvelle instruction du langage.

```
function Nom_Function(arg1, Arg2, …,Argn) {
  instruction 1
   …
  instruction M
  return valeur
}
```

Fonction

```
<head>

<script type="text/javascript">

 function perimetreCercle(rayon){
    return 2*3.14*rayon;
 }

  var rayon, peri, mess ;
  rayon = prompt("merci de saisir le rayon du cercle");
  peri = perimetreCercle(rayon);
  mess = "Le périmètre du cercle de rayon "+ rayon + "est :" + peri;
  alert (mess) ;


</script>

</head>
```

Instructions conditionnelles

if … else

```
if (condition) {
   instruction 1
     …
   instruction N
 }
else {
   instruction 1
     …
   instruction M
}
```

Instructions conditionnelles

Exemple

```
<script type="text/javascript">
  var pswd;
  pswd = prompt("Veuillez saisir votre mot de passe");
  if (pswd == "Java") {
  window.location ="pagejava.htm";
  }
  else {
    alert("Le mot de passe saisi est incorrect !");
  }
</Script>
```

Instructions conditionnelles

switch

```
switch (expression) {
   case valeur1 :
      instruction 1;
      break;
     …
   case valeurN :
      instruction N;
      break;

   default :
      instruction;
}
```

Instructions conditionnelles

```
switch (jour){
   case "lundi" :
       document.write("Nous somme lundi aujourd'hui");
        break;
    case "mardi" :
       document.write("Nous somme mardi aujourd'hui");
        break;
……..
      default :
        document.write ("Désolé, je ne connais pas");
}
```

Boucles

while

```
while (condition) {
   instruction 1
     …
   instruction N
 }
```

do …while

```
do {
   instruction 1
     …
   instruction N
 } while (condition)
```

Boucles

Exemple

```
<script type="text/javascript">
  var pswd;
  while (pswd != "Java"){
      pswd = prompt("Veuillez saisir votre mot de passe ! ");
   }
  window.location ="pagejava.htm";
</script>
```

Boucles

for

```
for (expr_initiale; condition; expr_repetee) {
   instruction 1
     …
   instruction N
 }
```

```
for(i=0;i<=20;i++){
   document.write(i + "<br>");
   document.write(tab[i]);
}
```

le cnam

Les Objets de Javascript

Octobre 2013

JavaScript et la notion d'objet

Bien que JavaScript ne soit pas un véritable langage orienté objet, il utilise essentiellement la notion d'objet. Il propose un grand nombre d'objets prédéfinis, mais le développeur peut en créer d'autres.

Le concept d'objet fait référence à trois notions fondamentales :

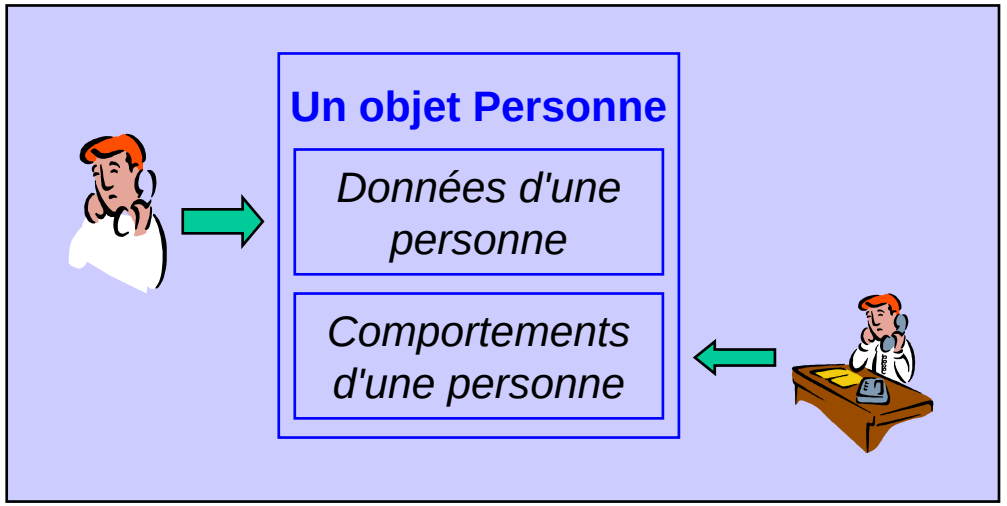
- propriété,
- méthode,
- événement.

JavaScript et la notion d'objet

Un objet est une entité à part entière qui *encapsule* des informations (*propriétés*) et des comportements (*méthodes*).

Les *propriétés* sont des données portées par l'objet, elles définissent les caractéristiques de l'objet.

Les *méthodes* représentent des traitements applicables à l'objet ou à une de ses propriétés.



JavaScript et la notion d'objet

Un objet Personne

Informations

nom = Dulac
age = 30
profession = Infirmière

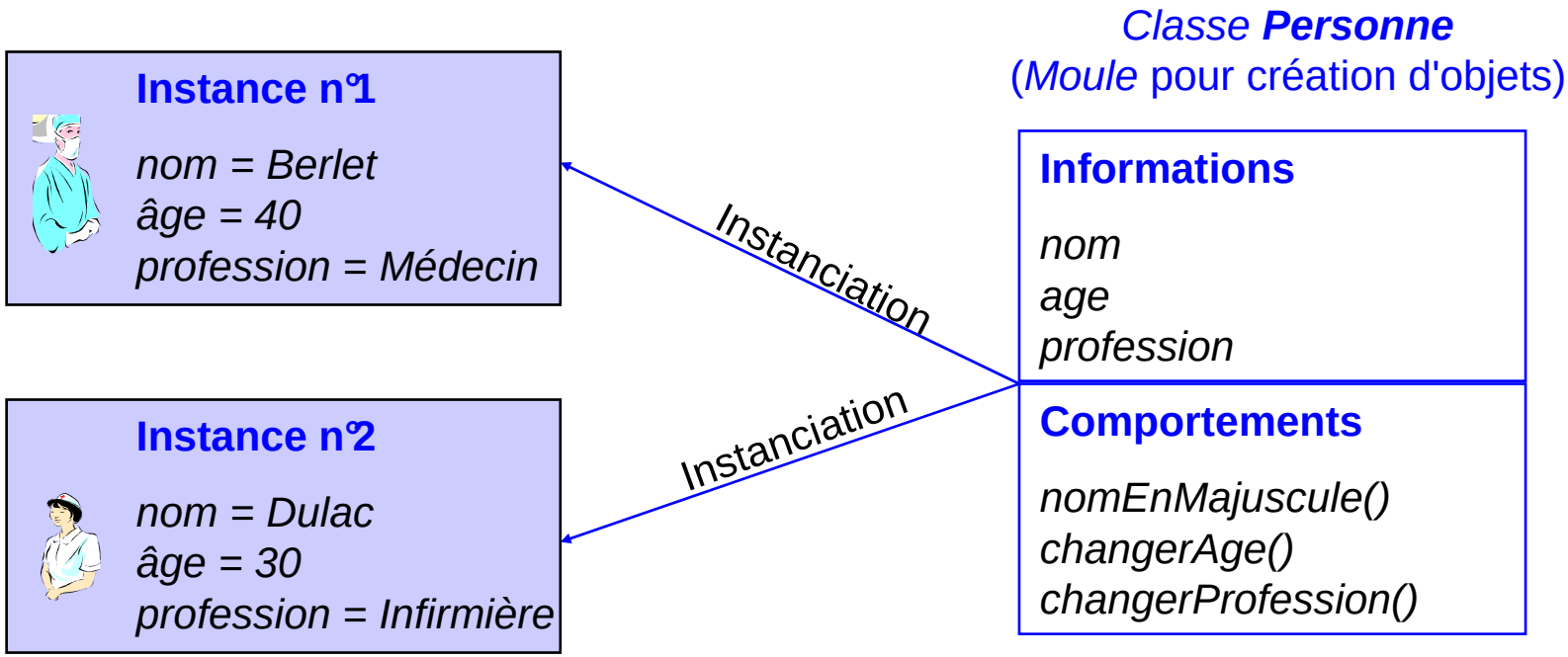
Comportements

nomEnMajuscule()
changerAge()
changerProfession()

JavaScript et la notion d'objet

Tous les objets ayant les mêmes propriétés et les mêmes méthodes sont rassemblés dans une famille.

Une telle famille est appelée **classe**, les objets qu'elle rassemble sont des **instances.**



JavaScript et la notion d'objet

Accéder aux propriétés et aux méthodes d'un objet

La syntaxe suivante permet d'accéder aux propriétés et aux méthodes d'un objet :

nomObjet.unePropriete

nomObjet.uneMethode()

Exemple :

l'objet *document* possède une couleur de fond, propose la méthode write(), ...

```
document.bgColor = "green"
document.write ("<h1>Introduction à JavaScript</h1>")
```

Les objets prédéfinis

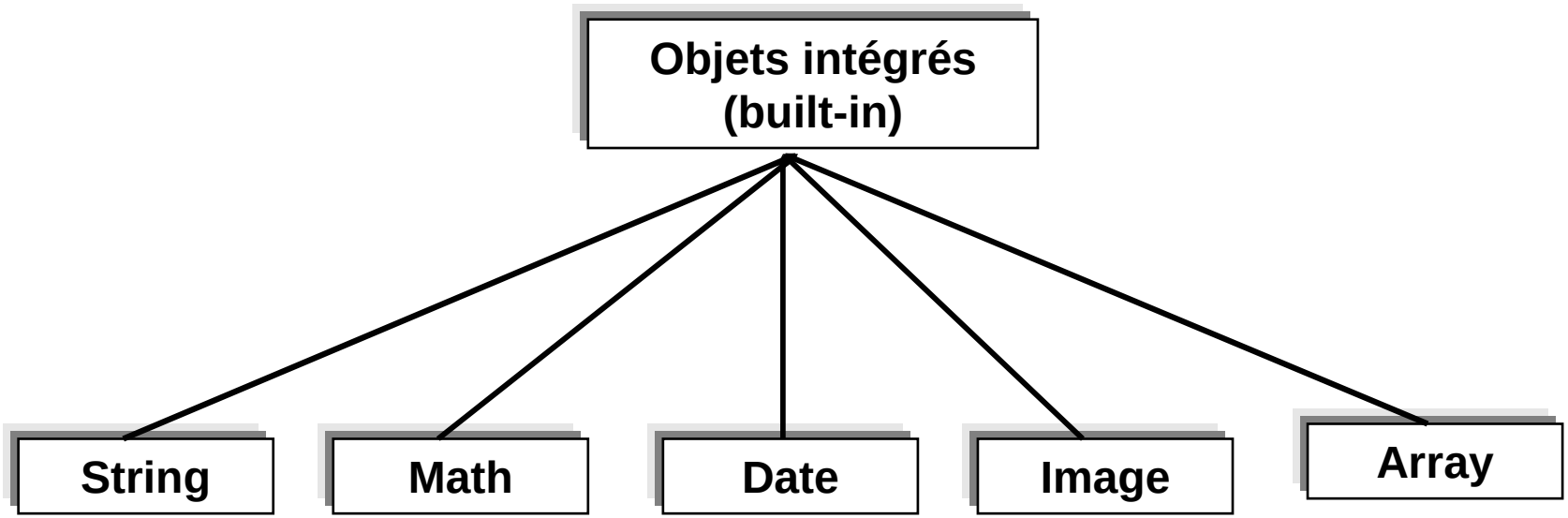
Il existe deux types d'objets prédéfinis dans JavaScript :

- les objets intégrés (built-in),
- les objets du navigateur.

Les objets intégrés sont basiques et indépendants du navigateur du client.

Les objets du navigateur permettent d'agir sur l'état du navigateur et des documents HTML affichés. Ils sont fortement hiérarchisés dans le sens où les descendants d'un objet correspondent à ses propriétés.

Les objets intégrés



Les objets intégrés : la classe String

Lorsque nous définissons une constante ou une variable chaîne de caractères, JavaScript crée de manière transparente une instance de la classe String. Cette classe propose une propriété (length) et un ensemble de méthodes.

On distingue :

- les méthodes de traitement des chaînes de caractères,
- les méthodes pour appliquer des commandes HTML

Les objets intégrés : la classe String

Méthodes de traitement des chaînes de caractères

```
chaine = new String ("Solution Informatique");


var chainemaj = chaine.toUpperCase(); // transforme la chaîne en capitales

var chaineminj = chaine.toLowerCase(); // transforme la chaîne en minuscules

var car = chaine.charAt(i) ; // retourne le caractère placé à la position i

var souschaine = chaine.substr(debut, longueur); // Retourne une certaine partie de
                                                 //la chaine.

var souschaine = chaine.substring(debut, fin); // Retourne une certaine partie de la
                                               // chaine.

var pos = chaine.indexOf(ch1); // Retourne la position de la première occurrence
                               // de ch1 dans chaine.
```

Les objets intégrés : la classe String

Méthodes pour appliquer des commandes HTML

```
chaine = "CNAM";
chaine.link("www.cnam.fr");        ➔  <a href = "www.cnam.fr">CNAM</a>
chaine.bold();                     ➔  <b>CNAM</b>
chaine.italics();                  ➔  <i>CNAM</i>
chaine.fontcolor("#00FF00");       ➔  <font color = "#00FF00">CNAM</font>
chaine.fontsize(5);                ➔  <font size = "5">CNAM</font>
```

Les objets intégrés : la classe Math

La classe Math possède des propriétés (qui renvoient les valeurs de quelques constantes mathématiques telles que Pi, e, …) et des méthodes qui correspondent aux fonctions mathématiques usuelles

```
var pericercle = 2* Math.PI *rayon;
```

quelques méthodes

```
Math.sin(x)
Math.cos(x)
Math.sqrt(x);
Math.round(x);  // Arrondit à l'entier le plus proche
Math.ceil(x);       // entier le plus proche supérieur ou égal à x
Math.floor(x);      // entier le plus proche inférieur ou égal à x
```

Les objets intégrés : la classe Date

La classe Date n'a pas de propriété, elle propose un ensemble de méthodes permettant de manipuler les informations de dates.

En interne, JavaScript calcule les dates en millisecondes à partir du 1 janvier 1970 à 1 heure du matin.

Contrairement aux autres objets intégrés du langage, la création d'une instance de Date nécessite l'appel au mot clé **new**.

Les objets intégrés : la classe Date

```
var maintenant = new Date();  // représente la date et l'heure actuelles


var uneDate = new Date(chaine);
```

chaine représente une chaîne de caractères sous l'une des deux formes suivantes :

- ***"Mois Jour, Année Heure:Minute:Seconde"***
- ***"Mois Jour, Année"***

***Mois* est en anglais et en toutes lettres et le reste des caractères est formé de chiffres.**

```
var uneDate = new Date("December 10, 2004 12:05:00");
var uneDate = new Date("December 10, 2004");
```

Les objets intégrés : la classe Array

Les Tableaux

Un tableau est une variable capable de stocker plusieurs valeurs.

A chaque valeur stockée est associé un index numérique qui permet ensuite d'accéder simplement à la valeur souhaitée

Il existe deux méthodes pour créer un tableau :

- comme objet de la classe Array
- sous forme littérale (depuis version JavaScript 1.2)

Les deux lignes suivantes sont équivalentes :

```
var jours = new Array("Lundi","Mardi","Mercredi","Jeudi","Vendredi")
var jours = ["Lundi","Mardi","Mercredi","Jeudi","Vendredi"];
```

Les objets intégrés : la classe Array

On peut aussi déclarer un tableau de la manière suivante :

var nom_tableau = new Array();

var nom_tableau = new Array(n);

```
var jours = new Array(7);
jours[0] = "Dimanche";
jours[1] = "Lundi";
jours[2] = "Mardi";
jours[3] = "Mercredi";
jours[4] = "Jeudi";
jours[5] = "Vendredi";
jours[6] = "Samedi";
```

Les objets intégrés : la classe Array

Les méthodes de la classe Array

join(*separateur*) : transforme la table en chaîne de caractères, les éléments sont séparés par la chaîne *separateur* (qui est par défaut une simple virgule).

```
var jours = new Array("Lundi","Mardi","Mercredi")
document.write(jours.join());    // écrit : "Lundi,Mardi,Mercredi"
```

concat() : pour concaténer deux tableaux

```
var joursSuiv = new Array("Jeudi","Vendredi");
var joursRes = jours.concat(joursSuiv );
// joursRes vaut : ["Lundi","Mardi","Mercredi","Jeudi","Vendredi"]
```

Les objets intégrés : la classe Array

Les méthodes de la classe Array

push() : pour ajouter un élément à la fin d'un tableau

```
joursRes.push("Samedi");
alert(joursRes.join());  // affiche "Lundi,Mardi,Mercredi,Jeudi,Vendredi, Samedi"
```

pop() : le dernier élément du tableau est supprimé et livré en retour (*il est dépilé*)

```
alert( joursRes.pop());  // affiche samedi
alert(joursRes.join());   // affiche "Lundi,Mardi,Mercredi,Jeudi,Vendredi"
```

shift() : comme pop mais il s'agit du premier élément

unshift() : comme push mais ajoute l'élément au début du tableau

Les objets intégrés : la classe Array

Un Tableau multidimentionnel

```
var  table1 = new Array (10);
for (var i = 0; i< table1.length; i++){
    table1[i] = new Array(5);
}
```

On définit ainsi un tableau à deux dimensions 10 X 5

Les objets intégrés : la classe Image

La classe Image propose un ensemble de propriétés intéressantes

src	: Adresse de l'image (URL)
width	: Largeur en pixels
height	: Hauteur en pixels
name	: Nom de l'image
border	: La valeur de l'attribut border

Les deux constructeurs suivants permettent de créer une image

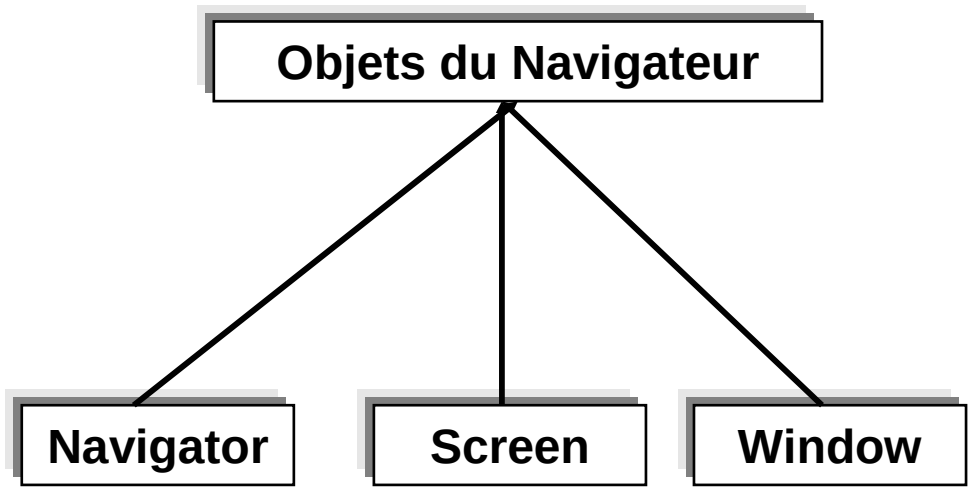
```
var myimage = new Image()
var myimage = new Image(width,height)
```

Les objets du Navigateur

Lors du démarrage d'un navigateur, JavaScript crée automatiquement un certain nombre d'objets. Ces objets permettent d'accéder à des informations concernant le navigateur, les documents HTML affichés, l'écran de la machine, …

Les objets du navigateur appartiennent principalement à trois classes :

- Navigator
- Screen
- Window



Les objets du Navigateur : navigator

L'objet **navigator** offre un ensemble de propriétés, permettant d'obtenir diverses informations de la part du navigateur.

appVersion : numéro de version du navigateur.

appCodeName : le nom de code du navigateur.

language (Netscape): le code langue du navigateur (fr, it, …).

userLanguage (IE) : le code langue du navigateur (fr, it, …).

systemLanguage (IE) : le code langue du SE (fr, it, …).

platform : informations sur le système d'exploitation du navigateur.

userAgent : donne appCodeName, appVersion, et platform en un seul résultat.

cookieEnabled : indicateur booléen sur l'acceptation des cookies.

plugins.length : nombre de plug-ins installés.

plugins[n].name : le nom du plug-in numéro n.

Les objets du Navigateur : screen

L'objet **screen** offre un ensemble de propriétés, permettant de connaître la résolution de l'écran de l'utilisateur, ainsi que d'autres informations :

height : hauteur.

width : largeur.

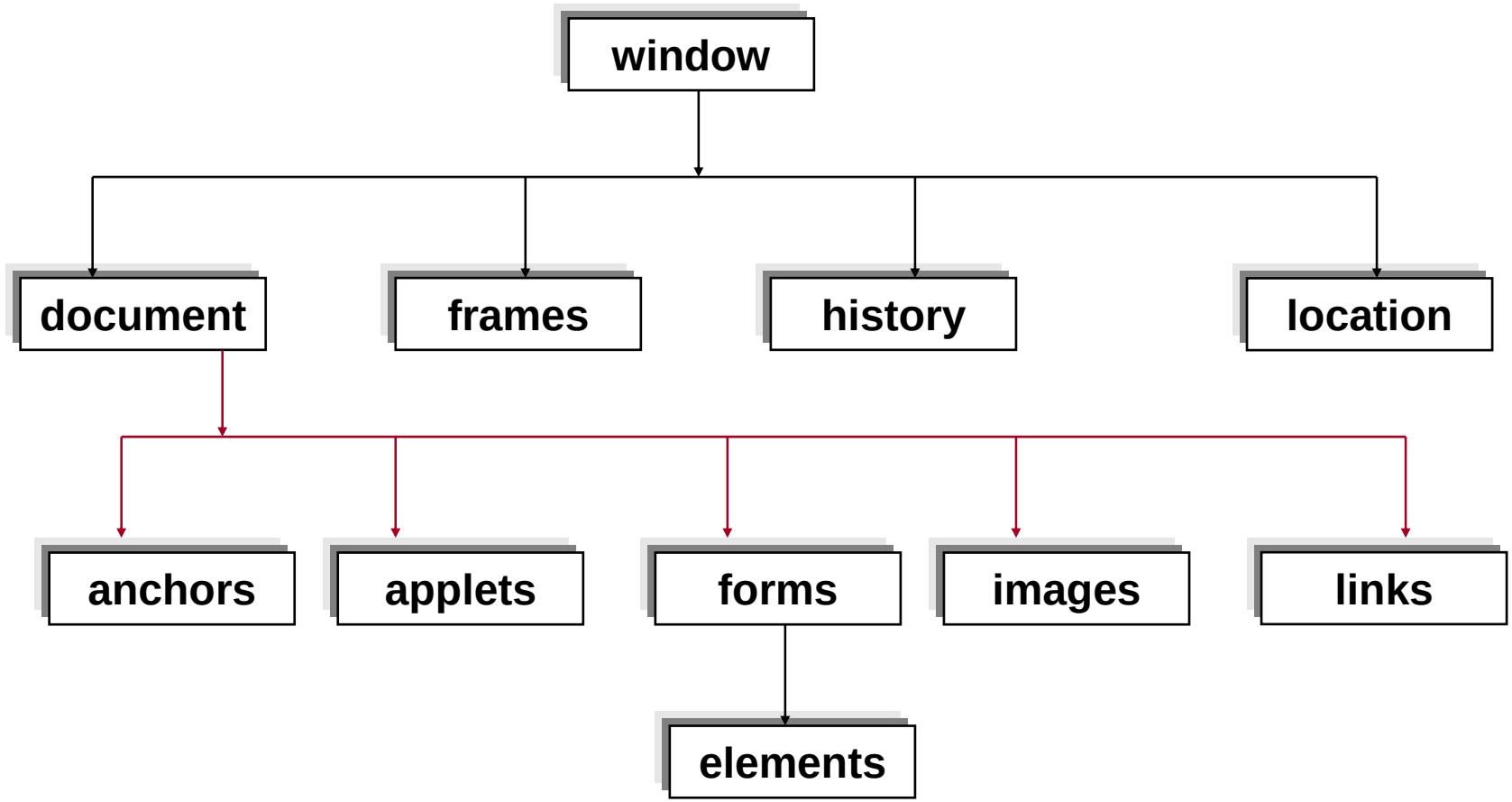
pixelDepth : profondeur de pixels (exprimée en bits/pixels).

colorDepth : profondeur de couleurs (exprimée en bits).

availHeight : hauteur disponible à l'écran.

availWidth : largeur disponible à l'écran.

Les objets du Navigateur : window



Les objets du Navigateur : window

La méthode prompt() de l'objet *window*

Cette méthode permet d'ouvrir une fenêtre composée d'un message et d'un champ de saisie. Elle retourne la valeur saisie par l'utilisateur.

prompt (message, "valeur par défaut");

La méthode alert() de l'objet *window*

Cette méthode permet d'ouvrir une fenêtre composée d'un message et d'un bouton OK.

alert (message);

La méthode confirm() de l'objet *window*

Cette méthode permet d'ouvrir une fenêtre de confirmation, composée d'un message, d'un bouton OK et d'un bouton Annuler. Elle retourne true ou false.

confirm (message);

Les objets du Navigateur : window

Exercice

```
<html><head>
        <title> Inviter l'Internaute à saisir une valeur </title>
        <script type="text/javascript">

            var nom;
            nom = prompt ("Veuillez saisir votre nom !" , "");

        </script>
 </head>
 <body>
        <script type="text/javascript">

          alert ("Bonjour, "+ nom);

        </script>
        <p>Vous êtes sur un document personnalisé</p>
</body></html>
```

Les objets du Navigateur : window. history

L'objet **history** permet de programmer des fonctionnalités identiques à celles des boutons *Précédent* et *Suivant* du navigateur.

Les méthodes associées à cet objet sont :

back(), forward(), go(entier), go(chaine) avec :

- si entier positif alors avance de entier documents dans l'historique
- si entier négatif alors recule de entier documents dans l'historique
- si entier vaut 0 alors recharge le document courant

- si chaine alors se positionne sur le document de l'historique dont l'URL contient cette chaîne

```
<a href = "javascript:window.history.back()"> Pr&eacute;c&eacute;dente</a>

<a href = "javascript:window.history.forward()"> Suivante</a>

<a href = "javascript:window.history.go(0)"> Actualiser</a>
```

Les objets du Navigateur : window.location

Pour rediriger les Internautes vers une autre page, on utilise la propriété **location** de l'objet **window.** Cette propriété est en fait un objet avec un certain nombre de propriétés et de méthodes, comme :

- la propriété href : pour indiquer l'URL à charger
- la méthode reload() : pour recharger le document courant

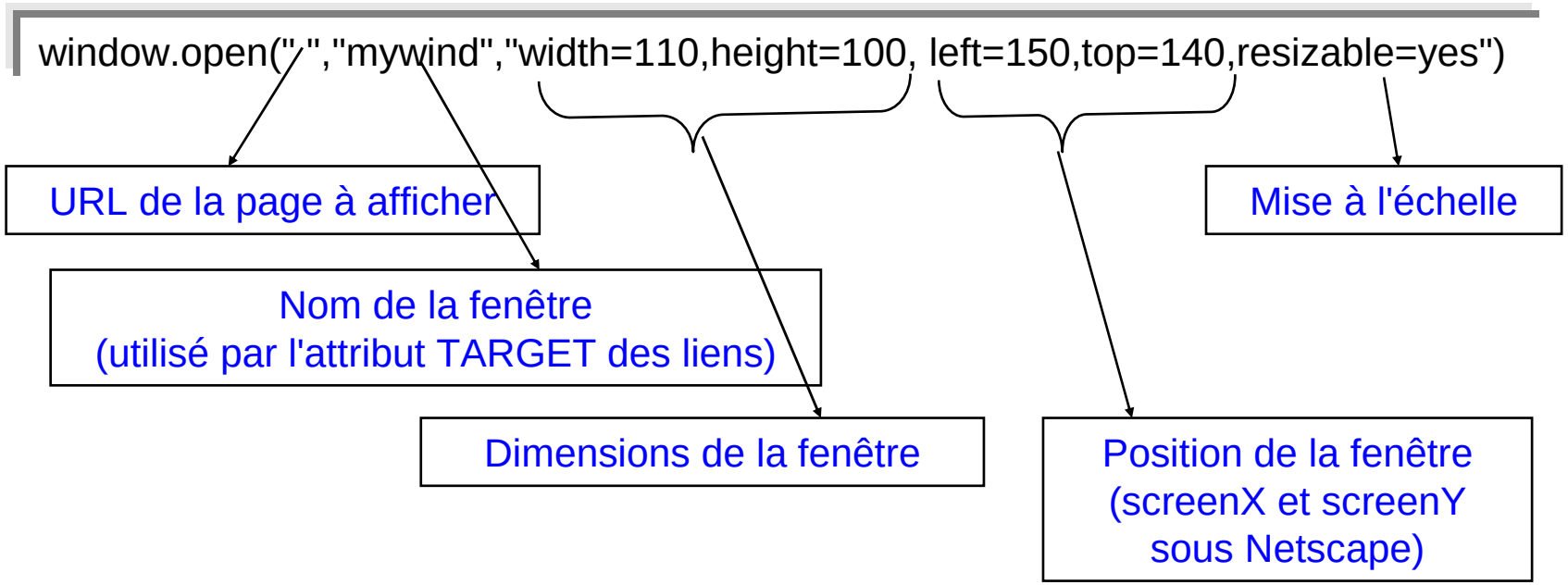
```
window.location.href = url
window.location.reload()
```

Les objets du Navigateur : window

La méthode open de l'objet window

La méthode open de l'objet window permet d'ouvrir une nouvelle fenêtre de navigateur.

```
window.open("","mywind","width=110,height=100, left=150,top=140,resizable=yes")
```



Les objets du Navigateur : window

La méthode open de l'objet window

La méthode open possède des arguments supplémentaires, qui ne peuvent prendre que l'une des deux valeurs : yes ou no.

location	: affichage de la zone d'adresses URL.
directories	: affichage de la barre d'outils personnelle.
menubar	: affichage de la barre des menus.
toolbar	: affichage de la barre d'outils de navigation.
status	: affichage de la barre d'état
scrollbars	: affichage des barres de défilement.

```
window.open ("","mywind","location=yes,directories=yes,menubar=yes,status=yes,
                toolbar=yes,scrollbars=yes")
```

Les objets du Navigateur : window.document

L'objet *document* offre un ensemble de propriétés et de méthodes, permettant de manipuler toutes les parties importantes d'une page Web.

Les propriétés les plus utilisées sont :

bgColor : la couleur d'arrière-plan.

links : contient une liste de tous les liens de la page, sous forme de table. Le premier lien se trouvant à l'adresse document.links[0].

anchors : contient les liens ancrés.

images : contient les adresses des images insérées dans la page.

forms : contient les formulaires de la page. Les éléments d'un formulaire sont répertoriés dans document.forms[].elements[]

Les objets du Navigateur : window.document

La propriété document.forms[].elements[] dispose aussi de plusieurs propriétés comme value, type, name, …

Exemple : document.forms[0].elements[0].value = "Hubert"

On peut utiliser aussi les noms des objets nommés.

Exemple : document.editForm.description.value.length

La méthode write() de l'objet *document*

Cette méthode permet d'écrire dans un document.

```
document.write ("<h1>Introduction au langage Java</h1>");
```

le cnam

Evénement et Gestionnaire d'événements

Octobre 2013

Gestionnaire d'événements

Grâce au concept *d'événement*, JavaScript permet de créer des programmes interactifs qui réagissent aux actions de l'utilisateur.

Un *gestionnaire d'événements* est une fonction (ou tout simplement une instruction JavaScript) associée à un *événement* (clic sur un bouton, passage du pointeur de souris sur un élément, …).

Gestionnaire d'événements

OnClick : Déclenché par un clic sur un élément HTML

OnMouseOver : Déclenché par le survol d'un élément par le pointeur

OnMouseOut : Déclenché à l'abandon d'un élément par le pointeur

OnFocus: Déclenché lorsqu'un objet reçoit le focus

OnBlur : Déclenché lorsqu'un objet perd le focus

OnChange : Déclenché lorsque la valeur d'un champ de saisie est modifiée

OnSelect : Déclenché lorsque l'utilisateur sélectionne du texte

OnReset : Déclenché en cas d'activation du bouton Reset

OnSubmit : Déclenché avant l'envoi d'un formulaire

OnLoad : Déclenché à la fin du chargement d'un document

OnResize : Déclenché lorsqu'un élément est redimensionné

Gestionnaire d'événements : onClick

```
<head>

  <script type="text/javascript">
     function hello() {
        alert ("Hello Word !!! ");
     }
  </script>

</head>


<body>

< form>
  <input   type = "Button"   value = "Cliquez ici"  onClick = "hello()">
</form>

</body>
```

Gestionnaire d'événements : onSubmit

Pour vérifier la présence des champs obligatoires d'un formulaire

```
<head>
  <script type="text/javascript">
    function checkForm() {
      if ( document.form1.age.value.length == 0) return false;
      …
      return true;
    }
  </script>
</head>
<body>
<form name="form1" onSubmit="checkForm()" >
  <input   type = "text"   name = "age" >
</form>
</body>
```

Gestionnaire d'événements : onMouseOver

Pour sensibiliser une image :

```
<p><img src = "@.gif"  onMouseOver = "bonjour()"></p>
```

Pour sensibiliser un titre :

```
<h1  onMouseOver = "bonjour()"> Votre Solution Informatique</h1>
```