

1. LES VARIABLES

1.1. LA DÉFINITION DES VARIABLES

Les variables contiennent des données qui peuvent être modifiées lors de l'exécution d'un programme. On y fait référence par le nom de cette variable.

Pour nommer une variable ou une fonction, le développeur définit **des identificateurs**.

Un identificateur doit commencer par une lettre (alphabet ASCII) ou le signe `_` et se composer de lettres, de chiffres et des caractères `_` et `$` (à l'exclusion du blanc). Le nombre de caractères n'est pas précisé. Pour rappel JavaScript est sensible à la casse. Attention donc aux majuscules et minuscules !

1.2. LA DÉCLARATION DE VARIABLES

Les variables peuvent se déclarer de deux façons :

- soit de façon explicite. On dit à JavaScript que ceci est une variable.
La commande qui permet de déclarer une variable est le mot **var**.
Par exemple :

```
var numAdherent = 1 ;  
var prenomAdherent = "Jean" ;
```

- soit de façon implicite. On écrit directement le nom de la variable suivi de la valeur que l'on lui attribue et JavaScript s'en accommode. Par exemple :

```
numAdherent = 2 ;  
prenomAdherent = "Luc" ;
```

Attention! Malgré cette apparente facilité, la façon dont on déclare la variable aura une grande importance pour la "visibilité" de la variable dans le programme JavaScript. Voir à ce sujet, la distinction entre variable locale et variable globale dans le JavaScript avancé de ce chapitre.

*Pour la clarté de votre script et votre facilité, on ne peut que conseiller d'utiliser à chaque fois le mot-clé **var** pour déclarer une variable.*

1.3. LES TYPES DE DONNÉES SOUS JAVASCRIPT

JavaScript utilise 5 types de données :

Type	Description
Des nombres	Tout nombre entier ou avec virgule tel que 22 ou 3.1416. Tout entier en octal ou hexadécimal tel que 0387, 0xFFA8.
Des chaînes de caractères	Toute suite de caractères comprise entre guillemets ou côtes telle que "suite de caractères" ou 'suite de caractères'
Des booléens	Les mots true pour vrai et false pour faux
Des objets	Toute utilisation de variable par référence vers tout objet natif JavaScript (Array, Date, String ...) ou tout objet du DOM.
Les mots-clés null et undefined	null : Mot réservé qui ne représente pas de valeur, utilisé principalement avec les objets. undefined : Mot réservé qui est renvoyé par une variable référencée qui n'a pas encore été définie.

Une variable non objet est une variable dite de type **scalaire**.

Une variable peut changer de type après une affectation. On peut vérifier le type en cours d'une variable avec l'instruction **typeof**.

Exemple :

```
var maVariable ;      // le type de la variable maVariable est undefined  
maVariable = 324 ;    // son type devient number  
maVariable = "Bonjour" ; // son type devient string  
maVariable = true ;   // son type devient boolean  
maVariable = new Array() ; // son type devient object
```

Notons aussi que contrairement au langage C ou C++, Il ne faut pas déclarer le type de données d'une variable. On n'a donc pas besoin de *int*, *float*, *double*, *char* et autres *long* en JavaScript.

1.4. LES NOMS RÉSERVÉS

Les mots de la liste ci-après ne peuvent être utilisés pour des noms de fonctions et de variables .Certains de ces mots sont des mots clés JavaScript, d'autres ont été réservés pour un futur usage éventuel.

Abstract
boolean break byte
case catch char class const continue
default do double
else extends
false final finally float for function
Goto
if implements import in instanceof int interface
Long
native new null
package private protected public
Return
short static super switch synchronized
this throw throws transient true try
var void
while with

1.5. LES VARIABLES GLOBALES ET LES VARIABLES LOCALES

Les variables déclarées tout au début du script, en dehors et avant toutes fonctions (voir plus loin...), fonctionnent comme des variables globales, qu'elles soient déclarées avec **var** ou de façon contextuelle. On pourra donc les exploiter partout dans le script.

Précision : La notion de variable globale n'existe pas en JavaScript. Toute variable non déclarée avec le mot-clé **var** est une instance d'un objet. L'objet par défaut étant l'objet **window**.

Par exemple, la déclaration implicite de la variable "maVariable" sous-entend qu'elle est une propriété de l'objet **window**. On pourrait l'appeler ainsi : "**window.maVariable**".

Dans une fonction, une variable déclarée par le mot clé **var** aura une portée limitée à cette seule fonction. On ne pourra donc pas l'exploiter ailleurs dans le script. D'où son nom de locale. Par contre, toujours dans une fonction, si la variable est déclarée implicitement (sans utiliser le mot **var**), sa portée sera globale !

Nous reviendrons sur tout ceci dans l'étude des fonctions.