

## Fiche de cours résumé N° 1 : Présentation de l'environnement Scilab – Variables et types de données

Logiciel **Scilab** : Logiciel libre développé par INRIA (*Institut national de recherche en informatique et en automatique- France*) en 1990, de calcul numérique et programmation, de haut niveau, similaire à son concurrent 'payant' *Matlab* (Maths Works).

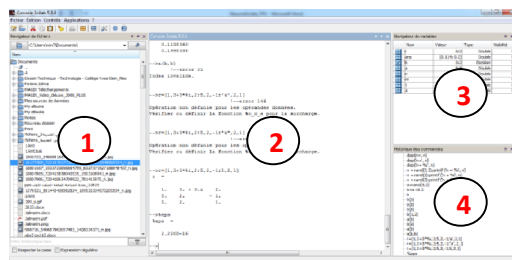
**Interface** : Fenêtres ouvertes par défaut (voir fig.).

**Navigateur de fichiers** (1) : Chercher, ouvrir les fichiers.

**Console** (2) : Espace d'exécution des commandes Scilab.

**Navigateur de variables** (3) : Table des variables déclarées en cours

**Historique** (4) : Historique des commandes exécutées sur console.



### Syntaxes et manipulations de base :

<b>clc</b> : Effacer l'espace du console	<b>whos -type Boolean</b> : Variables prédéfinies de type booléen.
<b>clear</b> : Effacer (écraser) les variables déclarées en cours	<b>help</b> : Chercher de l'aide sous Scilab.
<b>who</b> : Afficher les variables prédéfinies dans Scilab	<b>isdef('A')</b> : Chercher si ('A') est définie/prédéfinie ou pas.
<b>whos -name %</b> : Variables prédéfinies commençant par %	<b>exists('A')</b> : Chercher si ('A') existe ou pas.

### Variables et types de données :

#### Constantes prédéfinies de Scilab :

Constante	Signification	Constante	Signification
%pi	$\pi = 3.14\dots$	%eps	epsilon machine = 2.220D-16 ( $2,220.10^{16}$ )
%e	$e = 2.73\dots$ (nombre d'Euler)	%inf	$+\infty$ (infini)
%i	nombre complexe $i$ tel que $i^2 = -1$	%Nan	"Not A Number" (indéterminé)

#### Déclaration et type de variables :

Scilab reconnaît automatiquement le type de la variable lors de l'affectation selon la syntaxe. Il ne fait pas la distinction entre les réels et les entiers. Il déclare la variable lors de l'affectation, donc, une donnée écrasera une autre en valeur et en type.

Scilab est sensible à la casse. Il différencie entre majuscules et minuscules : **A** et différent de **a**.

#### Exemples :

Variable & Affectation	Type	Remarques
A = 12	Entier	
a = 10.5	Réel	<b>a</b> différent de <b>A</b>
A = 5.5	Réel	Contenu de A écrasé
Mot = 'bien'	Chaîne de caractères	
Choix = %t	Booléen	%t signifie <b>true</b> : vraie
Z = 5+3*%i	Complexe	%i = $i$ tel que ( $i^2 = -1$ )
A=[5.1,3,1 ; 2+3*%i,2,3;2.5,-1,5]	Matrice (différents types de données : entiers, réels, complexes)	$A = \begin{bmatrix} 5,1 & 3 & 1 \\ 2 + 3i & 2 & 3 \\ 2,5 & -1 & 5 \end{bmatrix}$

### Opérations sur les variables :

**Opérateurs arithmétiques** : + plus, - moins, / division, \* multiplication, ^ puissance.

**Exemples** : 2+2, (1-%i)^2, 1+%eps, 3+sqrt(sin(%pi), 'bon'+ 'jour'.

### Opérations sur les variables 'chaînes de caractères' ou string :

Opération	Signification	Opération	Signification
+	Concaténation	<b>Strsubst</b>	Substitution de caractères
<b>strcat</b>	Concaténation	<b>Length</b>	Nombre de caractères dans une chaîne
<b>strindex</b>	Recherche de caractères		

**Formats des nombres** : On distingue deux types de formats ('v' : normal et 'e' à exposant décimal) : Ne dépassant pas après la virgule 17 chiffres pour 'v' et 18 chiffres pour 'e'.

Format (exemples)	Remarques	Exemples	
		Tapé sur console	Résultat
Format ('v',10)	Normal par défaut	5.2 -10.5215415621	5.2 -10.521542
Format (16) ou Format ('v',16)	Normal défini	5.2 -10.52154156211584	5.2 -10.521541562158
Format ('e',16)	Exposant décimal	5.2 -10.52154156211584	5.200000000D+00 -1.052154156D+01
Format ('v',10)	Normal par défaut	0.00000000000000325	3.250D-15

### Commandes utiles :

Nombres complexes : **real**, **imag**, **abs**, **conj**, **phasemag**. Exemple :  $z=3+5*i$ , **real(z)**

Nombre aléatoire : **rand()** ; **rand(5,2)** : Respectivement, générer un nombre aléatoire ; une matrice (5,2) de nombres aléatoires, entre 0 et 1

Arrondir : **round(x)**, **floor(x)**, **int(x)**, **ceil(x)** : Respectivement, arrondi à l'entier le plus proche, arrondi vers le bas, arrondi vers zéro, arrondi vers le haut.

Autres : **sqrt** (racine carrée). Exemple : **sqrt(A)**, **sqrt(5)**, **sqrt(3+2\*i)**

### Opérateurs et expressions logiques, Opérations sur les booléennes.

Opérateurs logiques pour booléennes		Opérateurs logiques de comparaison			
Opérateur	Signification	Opérateur	Signification	Opérateur	Signification
~	Négation	==	Egal à	>	Strictement supérieur à
&	'ET' logique	<> ou ~=	Différent de	<=	Inférieur ou égal à
	'OU' logique	<	Strictement inférieur à	>=	Supérieur ou égal à

### Applications intéressantes : (voir le résultat de l'exécution sur le console de Scilab).

**a=x<0.2** : expression d'affectation logique à partir d'un scalaire ou d'une matrice x, dont les valeurs sont des booléennes : 'T' pour 'vrai' et 'F' pour 'faux'. Exemple :  $x=\text{rand}(3,8)$ ,  $a=x<0.3$

**bool2s(a)** : convertir les valeurs booléennes en 0 et 1 (1 pour 'T' et 0 pour 'F').

**and(a)** : Prend la valeur 'T' si toutes les valeurs de la matrice [a] sont 'T', ou 'F' ailleurs.

**and(a,'r')** : résultat d'une comparaison des éléments de la matrice [a], entre ses lignes, colonne par colonne, avec l'opérateur (**and**). Si [a] a deux vecteurs lignes  $b1=(1, :)$  et  $b2=(2, :)$ , le résultat est le même que celui de l'expression  $b1\&b2$ .

**and(a,'c')** : résultat d'une comparaison des éléments de la matrice [a], entre ses colonnes, ligne par ligne, avec l'opérateur (**and**). Le résultat est une colonne de valeurs booléennes. Si [a] a deux lignes vecteurs  $b1=(1, :)$  et  $b2=(2, :)$ , le résultat est le même que celui de l'expression  $b1|b2$ .

**find(a)** : donne les positions des valeurs 'T' rangés dans un vecteur ligne uniquement.

**x=y(a)** : si [y] est un vecteur de nombres réels et [a] un vecteur de valeurs booléennes, de même dimension que [y], le résultat est un vecteur [x] de valeurs de [a], uniquement ceux dont ses positions correspondent aux positions de 'T' dans le vecteur [y]. Ceux qui correspondent à la position de 'F' sont ignorés. Comme si on applique un filtre [a] sur la vecteur [y]. Exemple :  $y=[1,5,8,9,11,15,21,23]$

**C(x,x)** : si [C] est une matrice de nombres réels et [x] un vecteur de valeurs booléennes. Le résultat sera une matrice de valeurs réels de la matrice [C], uniquement ceux qui correspondent au résultat 'T' de l'expression logique  $x\&x$ , élément par élément, correspondants à la position de chaque élément de la matrice [C]. Ceux correspondants au résultat 'F' seront ignorés. Comme si on applique un filtre [x,x] sur la matrice [C].

**T(B)** : si [T] est un vecteur colonne, de nombres réels et [B] une matrice de valeurs booléennes. Le résultat est un vecteur d'éléments de la matrice [T], uniquement ceux qui correspondent aux positions des valeurs 'T' de la matrice [B], pris par ordre de ligne puis colonne. Comme si l'on s'agit d'un filtre [B] appliqué sur [T].

Exemple :  $T=\text{rand}(3,5)$ ,  $B=\text{rand}(3,5)<0.3$ ,  $T(B)$

Exécuter les commandes précédentes, une par une, sur le console de Scilab pour voir les résultats.

Document en ébauche. Prière m'informer de toute erreur ou omission.