

Le langage Scilab

I. Les objets du langage(constantes ou variables)

1) Les types élémentaires

- Les nombres entiers (signés ou non, codés sur 1 octet, 2 octets ou 4 octets).

Par exemple, un entier codé sur 2 octets et signé est un entier appartenant à l'intervalle des entiers [-32768;32767]. Lorsqu'il n'est pas signé, il sera dans l'intervalle [0;65535].

- Les nombres réels et complexes
- Les booléens

Scilab possède un type logique de booléens %t et %T pour vrai (True) et %f et %F pour faux (False).

- Les polynômes
- Les chaînes de caractères

2) Les types structurés

- Les listes
- Les tableaux (ou matrices)

II. Les actions élémentaires

1) L'instruction

L'instruction la plus simple est celle qui comporte une expression unique. Cette expression doit se terminer par un passage à la ligne, une virgule (séparateur d'instructions), trois points suivis d'un passage à la ligne (permettant de continuer l'expression sur la ligne suivante) ou un point virgule (dans ce dernier cas, le résultat de l'expression n'est pas affiché sur l'écran).

1) L'affectation

L'affectation est l'instruction qui permet de donner à une variable une valeur qui est le résultat d'un calcul. L'opération d'affectation est représentée par le symbole =. La forme la plus simple est *nom = expression*.

2) L'appel d'une fonction définie par l'utilisateur.

III. Les opérateurs

1) Les opérateurs numériques

Opérateurs unaires : identité (+), opposé (-)

Opérateurs arithmétiques : addition (+), soustraction (-), multiplication (*), division (/), division inversée (\)

2) Les opérateurs de comparaison

Egalité ==, non égalité <> ou ~ =, inégalités strictes < et >, inégalités larges <= et >=.

Le résultat d'une comparaison est un booléen.

3) Opérateurs logiques

La négation ~, la conjonction &, la disjonction |

P	~ P
T	F
F	T

&	T	F
T	T	F
F	F	F

	T	F
T	T	T
F	T	F

où T : True et F : False

4) Ordre de priorité

Une expression peut parfois devenir relativement complexe, du fait de la présence de multiples opérateurs au sein de l'expression. Le langage Scilab a déterminé un ordre de priorité par défaut :

Priorité	Opérateurs
1. (La plus haute)	~(non) ^
2.	&(et) * /
3.	(ou) + -
4. (la plus basse)	== ~> < <= > >=

IV. Les actions composées

1) Le bloc d'instructions (ou séquence d'instructions)

Un bloc d'instructions se compose d'une ou plusieurs instructions placées les unes après les autres (et séparées par des virgules ou des points-virgules) suivant que l'on souhaite un affichage ou non.

Dans la suite une instruction pourra représenter une instruction simple ou un bloc d'instructions.

2) Les conditionnelles

L'instruction *if* permet l'exécution conditionnelle de séquences d'instructions. Sa syntaxe la plus simple est : *if expression then instructions end*.

Lorsque l'expression *expression* est vraie, la séquence d'instructions *instructions* sera exécutée.

```
-->a=2 ; if a= =2 then 3+4, end  
ans = 7.  
-->a=5 ; if a= =2 then 3+4, end  
-->
```

Dans le deuxième exemple, le résultat du test est faux ; l'addition n'est donc pas exécutée et il n'y a pas d'impression du résultat.

Il est possible de préciser une instruction à exécuter lorsque le test est faux en utilisant le mot-clef *else*. La syntaxe de l'expression est alors la suivante :

if expression then instructions1 else instruction2 end

Lorsque l'expression *expression* est vraie *instruction1* est réalisée sinon c'est *instruction2*.

La conditionnelle multiple

La conditionnelle multiple ou instruction de sélection permet d'envisager plusieurs résultats à une expression :

```
select expression0  
case expression1 then instruction1  
case expression2 then instruction2  
case expression3 then instruction3  
else instruction4  
end
```

La valeur d'*expression0* est calculée puis comparée successivement aux valeurs d'*expression1*, *expression2*, *expression3*, ... Dès que l'une de ces comparaisons rend vrai, l'action correspondante s'effectue et les autres expressions ne sont plus testées. Si aucune de ces comparaisons ne retourne vrai, l'action introduite par le mot-clef *else* est exécutée.

On peut toujours traduire les conditionnelles multiples par des imbrications de conditionnelles :

```
if expression0= =expression1 then instruction1
  elseif expression0= =expression2 then instruction2
  elseif expression0= =expression3 then instruction3
  else expression4
end
```

3) Les boucles (ou instructions répétitives)

Elles ont pour but d'exécutée plusieurs fois une séquence d'instructions.

L'instruction *for*

La boucle *for* permet de répéter une instruction alors qu'une variable particulière, la variable de boucle, décrit un ensemble de valeurs.

```
for variable=valeurs do instruction end
```

Valeurs est la liste des valeurs que va décrire la variable de boucle *variable*.

D'une manière générale, les valeurs sont définies par une valeur de début, une valeur de fin et d'un pas de progression (ou d'incrément) entre ces bornes.

Le mot-clef *do* peut être remplacé par un passage à la ligne, une virgule ou un point virgule.

Notons encore que la variable de contrôle est une variable locale et n'a pas d'existence à l'extérieur de la boucle.

L'instruction *while*

La boucle *while* va répéter une instruction tant qu'une condition est réalisée.

```
while expression do instruction end
```

Il faut dans ce cas que *expression* fournisse un booléen (donnant le résultat du test).

L'instruction *while* permet une autre forme dont la syntaxe est la suivante :

```
while expression do instruction1 else instruction2 end
```

Cette autre écriture permet de préciser une instruction (ou bloc d'instructions) qui sera exécutée à la fin de la boucle *while* lorsque la condition testée sera de venue fausse.

L'instruction *break*

Utilisée à l'intérieur de la boucle *for* ou *while*, l'instruction *break* permet d'interrompre la boucle et de sauter immédiatement à l'instruction qui suit cette boucle dans le texte du programme en cours d'exécution.

4) Identificateurs

Scilab permet de définir des variables, qui peuvent être locales à une procédure, résider dans l'espace de travail, ou encore être globales.

Un identificateur est représenté par une suite de caractères, dont le premier est une lettre ou l'un des six caractères % _ # ! \$? ; le reste du nom peut être composé de ces caractères, de lettres ou de chiffres. Scilab ne prend en compte que les 24 premiers caractères d'un identificateur.

5) Variables et constantes prédéfinies

Scilab met à la disposition de l'utilisateur quelques variables et constantes prédéfinies :

- Inf dénote ∞ . Il peut être utilisé dans certaines opérations :

```
--> 1/%inf
```

```
ans =
```

```
0
```

- Nan (Not A Number) est le résultat typiquement renvoyé par ∞/∞ , par exemple.

- Pi (obtenu par %pi) dont la valeur vaut approximativement pi.
- i (%i) le nombre complexe tel que $i^2 = -1$
- %T et %F sont les variables booléennes signifiant respectivement vrai (pour true) et faux pour (false).
- La tolérance de "zéro machine" est fixée par la variable standart eps. C'est-à-dire que pour certaines opérations telles que la recherche de zéros de fonctions, par exemple, les nombres x tels que $-\text{eps} < x < \text{eps}$ seront considérés comme nuls.

En modifiant cette borne, vous pouvez renforcer ou abaisser la sévérité des tests de nullité. La valeur standard vaut :

```
-->%eps
%eps =
  2.220E-16
```

V. Fonctions mathématiques prédéfinies

Opération	description
abs(x)	valeur absolue
atan(x)	arc tangente
ceil(x)	partie entière, arrondie vers $+\infty$
conj(x)	conjugué de x
cos(x)	cosinus
disp(x,y,...)	affichage des valeurs de paramètres x, y, ...
exp(x)	exponentielle
fix(x)	partie entière, arrondie vers 0
floor(x)	partie entière, arrondie vers $-\infty$
format(n)	sélection de n chiffres significatifs pour l'affichage des nombres
imag(x)	partie imaginaire de x
int(x)	partie entière, arrondie vers 0 (comme fix)
log(x)	logarithme népérien
modulo(x,y)	reste de la division de x par y
rand()	nombre aléatoire compris entre 0 et 1
real(x)	partie réelle de x
round(x)	partie entière, arrondie vers l'entier le plus proche
sin(x)	sinus
sqrt(x)	racine carrée
string(x)	chaîne de caractères représentant paramètre
tan(x)	tangente

VI Principales commandes de Scilab

Help	Aide en ligne : sans paramètre, ouvre une fenêtre décrivant la commande help. Suivi du nom d'une fonction du langage, ouvre une fenêtre décrivant cette fonction
apropos	Aide en ligne : suivi d'un mot ou d'une chaîne de caractères, ouvre une fenêtre contenant un bref résumé des aides en ligne contenant ces mots.
what	liste des mots-clefs du langage
who	liste des variables définies
clear	Avec des paramètres, efface les variables correspondantes
global	déclaration de variables globales
clearglobal	effacement de variable globale
edit	Appel d'un éditeur de texte (créer ou modifier des fichiers de script)
quit	Arrêt de Scilab