

Outils de Programmation 2

Chapitre 1 : Prise en main

6 Les variables en Scilab

- Une variable en Scilab est désignée par un identifiant qui est une séquence de lettres et de chiffres.
- Les variables n'ont pas besoin de déclaration de type ou d'allocation de mémoire pour être utilisées, en revanche elles doivent avoir une valeur.
- Les variable en Scilab sont créées par affectation. la création de variable se fait par la syntaxe suivante :

`nom = expression`

- L'identifiant (nom d'une variable) doit respecter les règles suivantes :
 - ▷ Un identifiant doit commencer par une lettre.
 - ▷ Un identifiant ne doit jamais contenir des caractères accentués (é,è,ê).
 - ▷ Un identifiant ne doit jamais contenir des caractères spéciaux (@,≠),seul le caractère "_" underscore est autorisé. Un identifiant ne doit jamais contenir un espace.
 - ▷ On peut utiliser n'importe quel nom (dans la limite de 24 caractères) pour un identifiant à condition que ce ne soit pas une variable prédéfinie en Scilab.

Remarque. Scilab fait la distinction entre les lettres majuscules et les lettres minuscules par exemple `a` et `A` sont deux identifiants différents.

- Il peut être intéressant de savoir si un nom de variable est déjà utilisé, pour cela il y a une commande Scilab notée `exists`.

Exemple.

```
--> a=2;
exists("a")
ans=
1
exists("t")
ans=
0
```

- Pour examiner les variables définies dans une session de travail, on peut utiliser les commandes `who`, `whos`. La commande `whos` donne une description détaillée (nom de la variable, taille, type) par contre la commande `who` donne juste les noms des variables.

7 Types de données

Les quatre principaux types de variables utilisés par Scilab sont les types réels, complexe, chaîne de caractères et logique. Il n'y a pas de type entier à proprement parler.

7.1 Chaîne de caractères

- Pour définir une chaîne de caractères, on utilise des apostrophes « ' » ou des guillemets anglo-saxons « " ».

Exemple.

```
--> "Scilab = Scientific Laboratory"
ans=
Scilab = Scientific Laboratory
```

- Dans le cas où la chaîne contient déjà des apostrophes ou des guillemets il faut les doubler.

Exemple.

```
--> "Je m'appelle Scilab"
ans=
Je m'appelle Scilab
```

- Les opérations de base sur les chaînes de caractères sont **la concat énation** par l'opérateur « + » et l'extraction de sous-chaîne par la commande **part**.

Exemple.

```
--> chaine = "Je m'appelle Scilab " + " et je fais plein de choses"
chaine=
Je m'appelle Scilab et je fais plein de choses
--> part(chaine,14:20)
ans=
Scilab
```

Remarque. Dans l'exemple précédent la fonction **part** permet d'extraire de la chaîne « **chaine** » la sous-chaîne qui se trouve entre les position 14 et 20. Le symbole « : » permet de définir un vecteur d'indices.

- Les chaînes de caractères peuvent aussi utilisées pour définir des expressions ou des instructions de langage. la fonction **evstr** évalue l'expression définie par la chaîne de caractères et la fonction **execstr** exécute une instruction donnée par une chaîne de caractères.

Exemple.

```
--> expression = "sin(3)+1"
expression=
sin(3)+1
--> evstr(expression)
ans=
1.14112
--> instruction = "z=sin(3)+1"
--> execstr(instruction); z
z=
1.14112
```

7.2 Booléens

- L'objet booléen peut prendre deux valeurs « vrais » T et « faux » F. Scilab définit deux variables booléennes %t et %f.
- Les résultats des opérateurs de comparaisons et les opérateurs logiques appliqués à des objet sont des booléens.

Opérateurs logiques	
&	et
	ou
~	non

Opérateurs de comparaison	
<	inférieur strictement
<=	inférieur ou égal
>	supérieur strictement
>=	supérieur ou égal
==	égal
~=	différent

Exemple.

```
--> i=0; i<10
```

```
ans =
```

```
T
```

```
--> 1>=2
```

```
ans =
```

```
F
```

```
--> ~(1>=2)
```

```
ans =
```

```
T
```

On peut affecter les opérations booléennes classiques :

Exemple.

```
--> %t & %t
```

```
ans =
```

```
T
```

```
--> %t | %t
```

```
ans =
```

```
T
```

7.3 Constantes

Certaines constantes sont prédéfinies en Scilab et leur valeur ne peut pas être modifiée

constante	valeur
%pi	3.1415927
%e	2.7182818
%i	$\sqrt{-1}$
%eps	précision de la machine 2.220D-16
%inf	l'infini

Les deux types réels et complexes seront traités dans le chapitre 2 : "Les nombres en Scilab" .