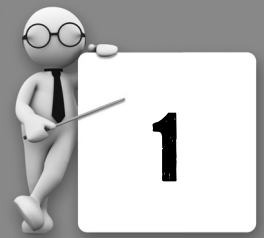


# Utiliser la console et Scinotes



## Quand on ne sait pas !

Scilab est un logiciel mathématique qui peut être utilisé comme un simple calculateur ou comme un langage de programmation.

- Lorsqu'on ouvre Scilab, on se trouve sur ce qu'on appelle la *console*. On peut s'en servir comme d'une simple calculatrice mais on peut aussi écrire des *instructions* ou *commandes* permettant de réaliser une certaine tâche, à la suite du symbole suivant :

-->

On valide la frappe par la touche « entrée » (retour à la ligne) du clavier et Scilab affiche le ou les résultats demandés s'il y en a.

- On accède à SciNotes en cliquant sur la première icône en haut à gauche :



Dans la fenêtre qui s'ouvre, on écrit des instructions. L'ensemble de ces instructions est appelé *script* ou *programme*. On enregistre ce script dans un fichier et on peut le modifier, le sauvegarder et l'exécuter, ce que l'on ne peut pas faire dans la console.

- Scilab possède deux constantes réelles :  $\pi$  qui est noté %pi et  $e$  qui est noté %e.
- Scilab utilise des variables qui jouent le rôle de « boîtes mémoires » dans lesquelles on peut stocker un contenu (numérique, matriciel...). On nomme ces variables par une lettre ou une chaîne de caractères.

## Que faire !

Voici trois instructions essentielles pour bien démarrer.

- L'instruction d'affectation.

Elle permet d'affecter une valeur à une variable. Elle est notée = et spécifie que la valeur placée à droite de ce symbole est affectée à la variable dont le nom est écrit à gauche.

Si une variable se situe à droite du symbole d'affectation, c'est le contenu de cette variable qui est affectée à celle dont le nom est indiqué à gauche du symbole =.

### EXEMPLE 1

L'instruction `x=3.2` consiste à placer la valeur 3,2 dans une variable notée ici `x` et créée automatiquement par `Scilab`.

- ▶ Noter que c'est le point, et non la virgule, qui fait office de séparateur décimal.
- ▶ Noter que `Scilab` distingue minuscule et majuscule. Ainsi, `x` et `X` désignent deux variables distinctes.

À l'issue de l'enchaînement `x=3; x=4`, la variable `x` contient la valeur 4. C'est toujours la dernière instruction d'affectation qui fait foi. Les contenus précédents sont perdus.

L'enchaînement `x=5; y=x` stocke la valeur 5 dans une variable `x` puis crée une variable `y` dont le contenu est le nombre 5 (correspondant ici au contenu de la variable `x`).

### ■ L'instruction `input`.

L'instruction `a=input('...')` affiche le texte écrit entre les apostrophes. `Scilab` attend une réponse. Si la syntaxe de cette réponse est correcte, `Scilab` l'affecte à la variable dont le nom est indiqué au début de l'instruction (ici `a`).

### EXEMPLE 2

Dans la console, voici ce qui se passe si on tape l'instruction

```
n=input('entrez un nombre entier : ')

```

et que l'utilisateur entre la valeur 3 :

```
entrez un entier : 3
n =
  3.

```

### ■ L'instruction `disp`.

L'instruction `disp('...')` affiche le texte écrit entre les apostrophes. L'instruction `disp(a)` affiche le contenu de la variable indiquée entre les apostrophes (ici, `a`).

### EXEMPLE 3

L'instruction `disp('bonjour')` renvoie :

```
bonjour

```

Si la variable `u` contient la valeur 5, l'instruction `disp(u)` renvoie :

```
5

```

Dans la console, on tape l'instruction : `disp(sol, 'la solution est :')`

En supposant que la variable `sol` contienne la valeur `-3`, `Scilab` renvoie

```
la solution est :
- 3.

```

Dans cette dernière séquence, Scilab affiche ce qui est demandé *de droite à gauche*. On affiche donc en premier le texte écrit entre les apostrophes (ici, `la solution est :`). Puis apparaît, à la ligne, le contenu de la variable `sol` (ici, la valeur `-3`).

- ▶ Si `x` et `y` sont des variables, l'instruction **disp** (`x, y`) affiche d'abord le contenu de la variable `y` suivi de celui de `x` avec un retour à la ligne entre chaque valeur.

## Conseils

- Dans la console, une instruction d'affectation suivie de la touche entrée provoque l'affichage du contenu de cette variable. Si l'on ne veut pas qu'il y ait d'affichage, il suffit de faire suivre l'instruction par un point-virgule. Cela n'empêche pas l'ordinateur d'effectuer l'instruction bien sûr! On peut écrire sur une même ligne plusieurs instructions en les séparant par des virgules ou par des points-virgules selon que l'on souhaite un affichage ou pas. Par exemple, l'instruction `a=6, b=3; c=5`, tapée sur une même ligne dans la console suivie de la touche entrée, renvoie :

```
a =  
6.  
c =  
5.
```

Il n'y a aucun affichage concernant la variable `b` mais l'affectation de la valeur 3 dans `b` a bel et bien été réalisée. Pour s'en convaincre, il suffit d'ajouter l'instruction **disp** (`b`) et on s'aperçoit que la console affiche bien la valeur 3.

- Dans SciNotes, on écrit les instructions en les séparant par une virgule ou un point-virgule ou encore en allant à la ligne. Contrairement au point précédent, il n'y a aucune incidence sur l'affichage. Cela permet juste de rendre le programme plus lisible. La seule façon de « communiquer » avec la console est d'utiliser dans le programme les instructions **input** ou **disp**.
- Dans un script (écrit dans SciNotes), on peut insérer des commentaires introduits par le symbole `//` . Ces commentaires sont ignorés lors de l'exécution du programme et permettent simplement d'éclairer le sens des instructions et aider à une meilleure compréhension du programme.

## Exemple traité

Compléter les instructions en tenant compte des commentaires indiqués :

```
a= ...; // invite l'utilisateur à entrer la valeur pi  
disp ... // affiche le message "Une valeur approchée de pi  
//est :" suivie de la valeur contenue dans a
```

## ► SOLUTION

On peut proposer :

```
a=input('entrer la valeur pi : ');  
disp(a, 'Une valeur approchée de pi est :')
```

Après la première instruction, l'utilisateur est censé frapper `%pi` au clavier. Après validation par la touche entrée, Scilab renvoie :

```
entrer la valeur pi : %pi  
Une valeur approchée de pi est :  
3.1415927
```

## Exercices

### EXERCICE 1.1

Dans la console, on exécute les instructions :

```
a=1;b=2;c=3;
```

Décrire pour chaque question, ce que renvoie Scilab après les instructions suivantes :

- 1 `a=b;b=c;c=a;disp(c,b,a)`
- 2 `b=c;c=a;a=b;disp(c,b,a)`
- 3 `c=a;a=b;b=c;disp(c,b,a)`

### EXERCICE 1.2

- 1 On propose le script suivant :

```
a=input('a= ')  
b=input('b= ')  
a=b;b=a  
disp(a, 'a=')  
disp(b, 'b=')
```

On suppose que l'utilisateur entre, dans cet ordre, les valeurs 1 et 10 lors des deux premières instructions. Que renvoient les deux dernières instructions ?

- 2 Compléter le script suivant pour qu'il échange le contenu des variables a et b.

```
a=input('a= ')  
b=input('b= ')  
...
```

## Pour vous aider à démarrer

### EXERCICE 1.1

Attention à bien respecter l'ordre des instructions et à bien réfléchir aux contenus des variables au moment de chaque instruction. Enfin, ne pas oublier le fonctionnement de la commande **disp**(c, b, a).

### EXERCICE 1.2

Pour la question 2, penser à introduire une troisième variable et utiliser l'exercice précédent.

## Solutions des exercices

### EXERCICE 1.1

- 1 Décrivons pas à pas les instructions :

L'instruction  $a=b$  affecte la valeur de  $b$  (c'est-à-dire 2) à la variable  $a$ .

L'instruction  $b=c$  affecte la valeur de  $c$  (c'est-à-dire 3) à la variable  $b$ .

L'instruction  $c=a$  affecte la valeur de  $a$  (qui est de 2 depuis la première instruction) à la variable  $c$ .

Finalement, les variables  $a$ ,  $b$  et  $c$  contiennent respectivement les valeurs 2, 3 et 2. Comme l'instruction **disp**(c, b, a) affiche les contenus de  $a$ ,  $b$ ,  $c$  dans cet ordre avec un retour à la ligne après chaque valeur, Scilab renvoie :

```
2.  
3.  
2.
```

- 2 À l'issue des instructions proposées, les variables  $a$ ,  $b$  et  $c$  contiennent respectivement les valeurs 3, 3 et 1. L'instruction **disp**(c, b, a) renvoie cette fois :

```
3.  
3.  
1.
```

- 3 Cette fois, les variables  $a$ ,  $b$  et  $c$  contiennent respectivement les valeurs 2, 1 et 1. L'instruction **disp**(c, b, a) renvoie :

```
2.  
1.  
1.
```

## EXERCICE 1.2

- 1 Lorsque l'utilisateur entre, dans cet ordre, les valeurs 1 et 10 lors des deux premières instructions, la variable *a* contient donc la valeur 1 et la variable *b* contient la valeur 10.

L'instruction `a=b` affecte la valeur de *b* (c'est-à-dire 10) à la variable *a*.

L'instruction `b=a` affecte la valeur de *a* (qui est désormais égale à 10 depuis l'instruction précédente) à la variable *b*.

Finalement, les variables *a* et *b* contiennent toutes les deux la valeur 10.

Les deux dernières instructions renvoient alors :

```
a=  
    10.  
b=  
    10.
```

- 2 En s'inspirant de l'exercice 1, on propose le script suivant :

```
a=input('a= ')  
b=input('b= ')  
c=a;a=b;b=c
```

L'instruction `c=a` sauvegarde le contenu de la valeur de *a* dans une variable *c*.

L'instruction `a=b` affecte la valeur de *b* dans la variable *a*. La valeur initiale dans *a* est donc effacée mais elle a été judicieusement et préalablement sauvegardée dans la variable *c*.

L'instruction `b=c` affecte la valeur de *c* (c'est-à-dire la même que celle initialement contenue dans *a*) à la variable *b*.

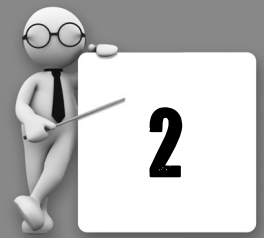
Finalement, les contenus des variables *a* et *b* ont bien été échangés.

**Remarque.** Il ne faudrait pas se laisser abuser par ce script :

```
a=input('a= ')  
b=input('b= ')  
disp(b,'a=')  
disp(a,'b=')
```

S'il semble répondre à la question posée, ce n'est qu'une impression visuelle ! En aucun cas, les **contenus** des variables *a* et *b* n'ont été échangés.

# Utiliser les opérations et les fonctions prédéfinies



## Quand on ne sait pas !

Il faut connaître

- les opérateurs algébriques :  $+$ ,  $-$ ,  $/$ ,  $*$  et  $^$
- les fonctions réelles prédéfinies : **abs**, **exp**, **log**, **sqrt**, **floor**, **cos** et **sin**.
- les opérateurs de test logique : **&** et **|**

**Remarque.** Dans cet ouvrage, tous les mots-clés du langage (comme **abs**, **floor** etc...) apparaîtront en caractère gras. L'utilisation des fonctions **cos** et **sin** est réservée aux élèves de la voie ECS.

## Que faire !

- Les opérateurs algébriques

Soit  $a$  et  $b$  des variables qui contiennent des réels  $a$  et  $b$ . Pour calculer

- ▶ la somme des deux réels  $a$  et  $b$ , on tape : .....  $a+b$
- ▶ la différence de  $a$  et  $b$ , on tape : .....  $a-b$
- ▶ le produit de  $a$  par  $b$ , on tape : .....  $a*b$
- ▶ le quotient de  $a$  par  $b$ , on tape : .....  $a/b$
- ▶  $a$  à la puissance  $b$ , on tape : .....  $a^b$

### EXEMPLE 1

Si  $a$  est une variable qui contient un réel  $a$ , pour calculer et stocker le réel  $\frac{a^3}{a+2}$ , on tape :  
 $a^3/(a+2)$

Si un calcul n'est pas affecté à une variable, Scilab place automatiquement le résultat dans une variable notée **ans** (*answer* en anglais).

Dans l'exemple ci-dessus, si  $a$  contient la valeur 5, Scilab renvoie :

```
ans
 17.857143
```

■ Les fonctions réelles.

Soit  $x$  une variable qui contient un réel  $x$ . Pour calculer

- ▶ la valeur absolue de  $x$ , on tape : ..... **abs** ( $x$ )
- ▶ le cosinus de  $x$ , on tape : ..... **cos** ( $x$ )
- ▶ le sinus de  $x$ , on tape : ..... **sin** ( $x$ )
- ▶ l'exponentielle de  $x$ , on tape : ..... **exp** ( $x$ )
- ▶ le logarithme népérien de  $x$  (avec  $x > 0$ ), on tape : ..... **log** ( $x$ )
- ▶ la racine carrée de  $x$  (avec  $x \geq 0$ ), on tape : ..... **sqrt** ( $x$ )
- ▶ la partie entière de  $x$ , on tape : ..... **floor** ( $x$ )

**EXEMPLE 2**

Si  $a$  est une variable qui contient un réel  $a$ , pour calculer et stocker le réel  $\frac{a - \lfloor a \rfloor}{\sqrt{a^2 + 1}}$  dans une variable  $x$ , on tape :  $x = (a - \mathbf{floor}(a)) / \mathbf{sqrt}(a^2 + 1)$

■ Les opérateurs de test logique.

Scilab possède deux constantes *booléennes* : %t (*true* en anglais) et %f (*false*) qui contiennent respectivement les valeurs T et F.

Soit  $a$  et  $b$  des variables qui contiennent des réels  $a$  et  $b$ . Pour savoir si

- ▶  $a$  est égal à  $b$ , on tape : .....  $a == b$
- ▶  $a$  est strictement supérieur à  $b$ , on tape : .....  $a > b$
- ▶  $a$  est strictement inférieur à  $b$ , on tape : .....  $a < b$
- ▶  $a$  est supérieur ou égal à  $b$ , on tape : .....  $a \geq b$
- ▶  $a$  est inférieur ou égal à  $b$ , on tape : .....  $a \leq b$
- ▶  $a$  est différent de  $b$ , on tape : .....  $a <> b$

Ces instructions renvoient la valeur T si le test est vrai et F si le test est faux.

On peut relier deux tests par les opérateurs & et | selon les règles suivantes :

- ▶ L'instruction ... & ... renvoie la valeur T si les deux tests placés de part et d'autre du symbole & contiennent la valeur T. Sinon, elle renvoie la valeur F.
- ▶ L'instruction ... | ... renvoie la valeur T si l'un au moins des deux tests placés de part et d'autre du symbole | contiennent la valeur T. Sinon, elle renvoie la valeur F.

**EXEMPLE 3**

On suppose que les variables  $a$  et  $b$  contiennent respectivement les valeurs  $-1$  et  $1$ .