

À l'ouverture de Scilab, une fenêtre de commandes (la console) s'ouvre et une invite de commande (`-->`) indique que le logiciel attend des instructions.

Il est possible de personnaliser entièrement l'interface en repositionnant chaque sous-fenêtre avec des glisser-déposer.

1 Usage de la console

1.1 Premiers exemples

Tapez :

```
--> x=1
```

Les lignes suivantes s'affichent :

```
x=  
1.  
-->
```

Scilab a automatiquement créé une variable notée x , à laquelle la valeur 1 a été affectée. On peut vérifier ceci en tapant la commande :

```
--> x
```

qui appelle la variable x , et renvoie sa valeur :

```
x=  
1.  
-->
```

Si maintenant vous tapez :

```
-->y=2;
```

rien ne s'affiche. On peut appeler quand même la variable y :

```
--> y
```

On remarque qu'une variable y a bien été créée, et qu'on lui a affecté la valeur 2, mais le résultat ne s'est pas affiché au moment de la déclaration. C'est à cause de la présence du « ; », qui permet de ne pas afficher le résultat des opérations.

Pour afficher la valeur d'une variable, on peut aussi utiliser la commande `disp` :

```
-->disp(y)
```

On peut également utiliser `disp` pour afficher une chaîne de caractère, par exemple :

```
-->disp("bonjour")
```

Tapez maintenant :

```
-->x=x+3
```

Que se passe-t-il ?

Remarque. Si l'on veut **réutiliser une commande déjà tapée** sans avoir à la réécrire, on peut utiliser les flèches du clavier : \uparrow et \downarrow .

Remarque. Scilab dispose d'une **aide en ligne**. Pour y accéder, taper `help`. Pour accéder à la page d'une commande en particulier (par exemple `disp`), on tape `help("disp")`. N'hésitez pas à y recourir si vous oubliez la syntaxe d'une commande : ces pages contiennent notamment de nombreux exemples d'utilisation.

1.2 Opérations usuelles (à mémoriser)

Les opérations usuelles sont +, -, * (pour la multiplication), / (pour la division), ^ (pour une puissance). On dispose également de fonctions usuelles : `abs` (valeur absolue), `sqrt` (racine carrée), `log` (logarithme népérien), `exp`, `cos`, `sin`, `floor` (partie entière).

Enfin, on accède aux constantes e et π en tapant `%e` et `%pi`.

Exercice 1. Tapez successivement les commandes suivantes et vérifiez ce qui se passe. Lorsqu'il y a un message d'erreur ou que le résultat paraît étrange, expliquer pourquoi et donner la commande correcte pour effectuer l'opération voulue.

```
x=5,5
x=7
8x
pi
3*5
x+12
x=x+12
x+12
disp("x")
y=sqrt(x^3)
y=4
disp(y,"y=")
disp(x,y)
z=3;
disp(Z)
```

Exercice 2. On considère la fonction f qui à tout x réel associe $\frac{\cos(1 + 4 \ln(x)^2)}{3 + x^4}$. En utilisant une variable x , calculer les valeurs de f pour les entiers de 1 à 5.

2 Boucle for

2.1 Syntaxe

L'instruction `for` permet d'itérer une commande un nombre déterminé de fois. Par exemple :

```
for i=1:5; disp(i); end
```

donne l'instruction : pour i allant de 1 à 5 (valeurs entières), afficher i .

Remarque. Attention : si vous lancez l'instruction en ayant oublié le `end`, la console restera bloquée en boucle dessus et vos commandes suivantes ne rendront aucun résultat. Pour revenir à un état normal, il faut taper `end` autant de fois que nécessaire dans la console, ou aller chercher l'instruction « Interrompre » dans le menu « Contrôle ».

2.2 Exemples

Pour calculer la somme $\sum_{k=0}^{10} k$, on part d'une somme vide à laquelle on rajoute successivement les différentes valeurs de k . On peut par exemple utiliser le programme suivant (qui stocke le résultat final dans la variable `S`) :

```
S=0;
for i=0:10; S=S+i ;end
```

Exercice 3. Procéder de même pour calculer $\sum_{k=0}^{10} k^2$, $\sum_{k=0}^{10} k2^k$ puis $10!$.

2.3 Utilisation d'un tableau

Si S est un tableau, les commandes `sum(S)` et `prod(S)` permettent d'obtenir la somme et le produit de ses éléments.

Pour obtenir $\sum_{k=1}^{10} k$ (resp. $10!$), on peut donc construire un tableau $S = [1, 2, 3, \dots, 9, 10]$ avec la commande `S=1:10`, puis faire la somme (resp. le produit) de ses coordonnées avec `sum(S)` (resp. `prod(S)`).

Exercice 4. Que renvoie la commande `1:2:10`? En se servant d'une syntaxe similaire, créer un tableau contenant tous les nombres multiples de 3 entre 1 et 20.

Le problème de l'écriture du tableau est par contre plus complexe si l'on veut calculer $\sum_{k=1}^{10} k^2$ ou $\sum_{k=1}^{10} k2^k$.

L'idée est de construire un tableau à dix cases (une ligne et 10 colonnes) initialement rempli de zéros :

```
S=zeros(1,10)
```

Remarque. Attention : le 1 de cette commande signifie « à une ligne ». La numérotation des cases d'un tableau n'est pas modifiable et va toujours de 1 au nombre total de cases.

On remplit ensuite ce tableau en modifiant ses cases :

```
for i=1:10; S(i)=i^2; end
```

Ceci permet enfin de calculer la somme $\sum_{k=1}^{10} k^2$ avec `sum(S)`.

Exercice 5. En utilisant un tableau, calculer $\sum_{k=1}^{10} k2^k$.