

1 Utiliser SciNotes

1.1 Présentation

La console montre ses limites dès que l'on souhaite écrire un programme un peu complexe, ou tout simplement sauvegarder son travail. Scilab dispose d'un éditeur de programmes appelé SciNotes, qui peut être lancé via l'icône la plus à gauche de la barre d'outils ou en utilisant le menu « applications ». Comme toutes les autres fenêtres, SciNotes peut être repositionné dans l'espace de travail avec un glisser-déposer.

Il s'agit d'un éditeur multi-ligne : lorsque l'on appuie sur « entrée », cela ne valide pas la commande mais passe simplement à la ligne suivante. On peut ainsi préparer une suite d'instructions que Scilab n'exécutera que quand on lui demandera.

L'interface de SciNotes propose tous les outils usuels (copier, coller, ouvrir un fichier .sce, sauvegarder dans un fichier .sce) ainsi qu'une coloration syntaxique permettant de repérer plus facilement les erreurs de frappe. Des commandes permettent d'exécuter le programme lorsqu'on a fini de le rédiger (attention, on ne peut pas exécuter un fichier qui n'a jamais été enregistré).

Exercice 1. Créez un fichier SciNotes pour sauvegarder votre travail et enregistrez le sur votre ordinateur.

1.2 Exécution du fichier

À l'exécution, Scilab applique l'ensemble des instructions du fichier. La console affiche alors :

```
-->exec("adressedevotrefichier/Fichier.sce", -1)
```

puis renvoie les éventuels résultats.

Attention : lorsqu'on compile un programme SciNotes, les instructions sont muettes par défaut (comme si les lignes se terminaient toutes par des points virgule). Si on veut faire afficher un résultat, il faut utiliser la commande `disp`.

Exercice 2. Dans votre éditeur, écrire des lignes de code qui à la compilation affectent la valeur 3 à la variable x et affichent à l'écran $x = 3$.

1.3 Commenter son code

SciNotes permet d'insérer des commentaires dans un programme, c'est-à-dire du texte qui ne sera pas compilé. Cela permet entre autres d'explicitier des instructions, de mettre des titres ou de neutraliser temporairement une ligne de code que l'on ne souhaite plus exécuter.

Un commentaire doit être précédé de `//`.

L'utilisation de commentaires est recommandée : entre deux séances informatiques, on oublie facilement ce qu'on avait voulu faire la fois précédente.

Exemple 1. On peut écrire :

```
//1 Utiliser SciNotes  
x = 3  
//On attribue la valeur 3 à x
```

2 Fonctions

2.1 Syntaxe

Pour définir une fonction f qui prend un argument x et renvoie une valeur y , on utilise la syntaxe suivante :

```
function y = f(x)
expression de y en fonction de x
endfunction
```

À noter que x et y ne sont pas forcément des réels : ils peuvent notamment être des tableaux. Il n'est pas nécessaire/possible de mentionner les espaces de départ et d'arrivée dans la syntaxe (Scilab se contentera de sortir en erreur si vous tentez d'effectuer des opérations non autorisées).

Exemple 2. Si on veut construire la fonction $g : \mathbb{R} \mapsto \mathbb{R}$ définie par $x \mapsto x^2$, on écrira

```
function y=g(x)
y=x^2
endfunction
```

Remarque. La compilation d'une fonction n'affiche rien à l'écran. Si on veut tester une fonction, il faut l'appeler sur des valeurs particulières. Avec la fonction g précédente, on peut par exemple taper $g(3)$ pour calculer 3^2 et vérifier qu'on retombe bien sur 9.

Remarque. Si vous compilez une nouvelle fois le fichier contenant votre fonction, la console affichera un message du type :

Attention : Redéfinition de la fonction : g . Utilisez funcprot(0) pour ...

C'est tout à fait normal, le message vous avertit simplement que l'ancienne définition de la fonction a été écrasée par la nouvelle.

Exercice 3. On considère la fonction f qui à tout x réel associe $f(x) = \frac{\cos(1 + 4 \ln(x)^2)}{3 + x^4}$.

1. La définir dans Scilab.
2. En utilisant une boucle for, créer un tableau à 5 cases contenant les valeurs de f pour les entiers de 1 à 5.

2.2 Arguments multiples

Il est possible de définir des fonctions qui prennent plusieurs arguments en entrée. La syntaxe est alors :

```
function y = f(x1,x2,...)
expression de y en fonction des xi
endfunction
```

Exercice 4. Définir une fonction S qui prend en argument un entier n et un réel x et renvoie $\sum_{k=1}^n kx^k$.

Remarque. Une fonction ne peut pas avoir plusieurs arguments de sortie. Mais si on souhaite qu'elle renvoie plusieurs valeurs, il est toujours possible de lui faire renvoyer un tableau qui contient toutes les valeurs en question.