# Scilab, comment ça marche?

The anonymous M. Maths & FireGrain

18 octobre 2014

Scilab execute une liste d'instruction présentes dans un fichier sci.

Il n'y a pas de type particulier dans scilab, on gère des tableaux(Matrices) ou des nombres.
Du coup, pour déclarer une variable il faut ecrire le nom de la variable et l'initialiser.
Exemple : i=0
*Note :* si vous souhaitez mettre plusieurs instruction dans une même ligne alors il faut les séparer par des points virgules, sinon inutile d'en mettre.
**les Tableaux :** je peux déclarer un tableau de la façon suivante

$$M = [\overbrace{0,1,1}^{1^{er} ligne}\ ;\ \overbrace{0,0,1}^{2^{eme} ligne}\ ;1,0,0]$$

ou encore avec les fonctions ones(nbl, nbc), zeros(nbl, nbc) qui créent respectivement un tableau de 1 et un tableau de zéros de nbl ligne(s) et nbc colonne(s). Avec la fonction eye(nbl,nbC) qui crée une matrice **Identité**.
ou encore avec une composition de Matrice, Exemple :

$$P = [[0,1,1];[1,1,0];ones(2,3)]$$
$$G = [eye(3,3);P]$$

**Les affichages**, se font avec la fonction disp(texte1,texte2,valeur1,...) en sachant que la console va afficher les paramètres dans l'ordres inverse dans laquelle ils ont été passés.

**La congruence**, se fait avec la fonction modulo(val1,val2) qui calcule val1 modulo val2
Exemple : modulo(6,2) va calculer 6 modulo 2 qui donnera 0.
**Les fonctions** s'écrivent de la façon suivante : **function y=f(x)** ***instruction*** **endfunction** (et oui c'est aussi simple que cela) où y est la variable que renvoie la fonction, f le nom de la fonction et x le paramètre passé. On peut renvoyer un tableau et recevoir plusieur paramètre.
Exemple :

$$function\ y = f(M)$$
$$y = modulo(G * M, 2)$$
$$endfunction$$

y est alors une matrice

$$function\ [colonne1, colonne2] = fonctionMatrice()$$
$$colonne1 = ones(5,1)$$
$$colonne2 = zeros(5,1)$$
$$endfunction$$

on renvoie une colonne de 1 et une autre de 0
**les conditionnelles** se structure de la façon suivante : **if (condition) then** ***instruction*** **else** ***instruction*** **end**
**les boucles conditionnelles** se structure de la façon suivante : **while(condition)** ***instruction*** **end**
**les boucles itératifs** se structure de la façon suivante : **for compteur=(valeur_initial,valeur_final)** ***instruction*** **end**
voilà vous connaissez tous ce que je sais.