

Chapitre 5

LES FONCTIONS

Motivations

Les fonctions jouent un rôle central en programmation. Voyons ici un exemple d'utilisation.

On souhaite calculer $3x^2 + 5y$ pour $(x,y) = (1,1)$ puis pour $(x,y) = (2,3)$ et éventuellement d'autres valeurs.

```
// Méthode 1 - dans la console
--> x = 1
--> y = 1
--> 3 * x^2 + 5 * y
8.
--> x = 2
--> y = 3
--> 3 * x^2 + 5 * y
27.
```

```
// Méthode 2 - avec input
x = input("x = ")
y = input("y = ")
disp(3 * x^2 + 5 * y)

// F5
--> exec('...')
x = 1
y = 1
8.

// F5 une 2e fois
--> exec('...')
x = 2
y = 3
27.
```

Une **fonction** permet d'être plus efficace sur ce problème.

```
// Méthode 3 - avec une fonction
function z = f(x,y)
    z = 3 * x^2 + 5 * y
endfunction

// F5 - une seule fois en tout
--> exec('...')
--> f(1,1)
ans =
    8.
--> f(2,3)
ans =
    27.
```

I Définir et utiliser une fonction

I.1 Syntaxe scilab

```
function variable_sortie = nom_fonction(x1, x2, ..., xn)
    instructions
    permettant de définir
    la variable_sortie
endfunction
```

- `nom_fonction` est le nom de la fonction qui sera utilisé ensuite. Pas d'espace, pas de caractère spécial. Préférez les noms de fonctions explicites.
- `x1, x2, ..., xn` sont les paramètres d'entrée. Ce sont les variables de la fonctions. Ne pas mettre de valeur ici, on garde des noms génériques.
- `variable_sortie` est le nom de la variable de sortie, c'est-à-dire du résultat.
- **Pas de disp, pas de input!**

I.2 Faire appel à la fonction

1. On définit notre fonction dans l'éditeur.

```
function z = f(x,y)
    z = 3 * x^2 + 5 * y
endfunction
```

2. On informe scilab de la création de cette fonction en **exécutant** (F5)

```
--> exec('...')
```

Rien ne s'affiche dans la console, c'est normal. La fonction a juste été répertoriée.

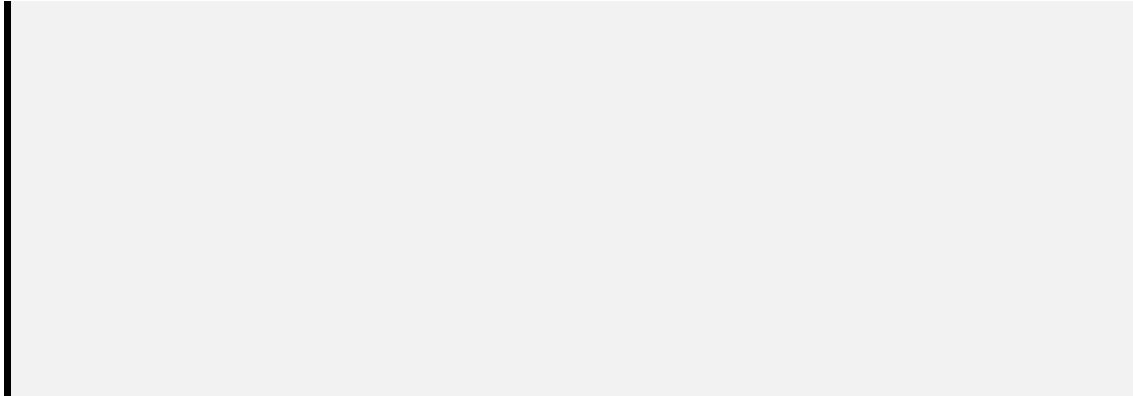
3. Pour l'utiliser, il faut simplement donner le nom de la fonction suivi, entre parenthèses, des valeurs des variables d'entrée.

```
--> f(2,3)
ans =
    27.
```

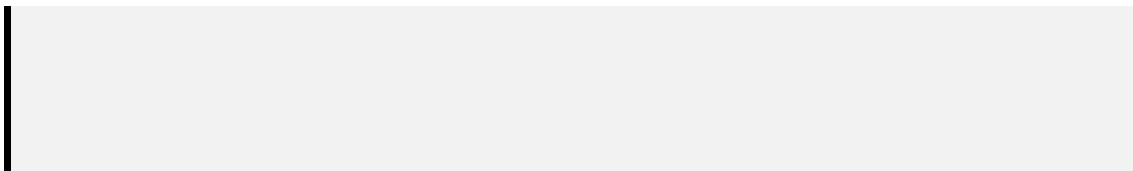
Le résultat est automatique affiché, sans avoir à écrire `disp` dans la fonction.

II Exemples

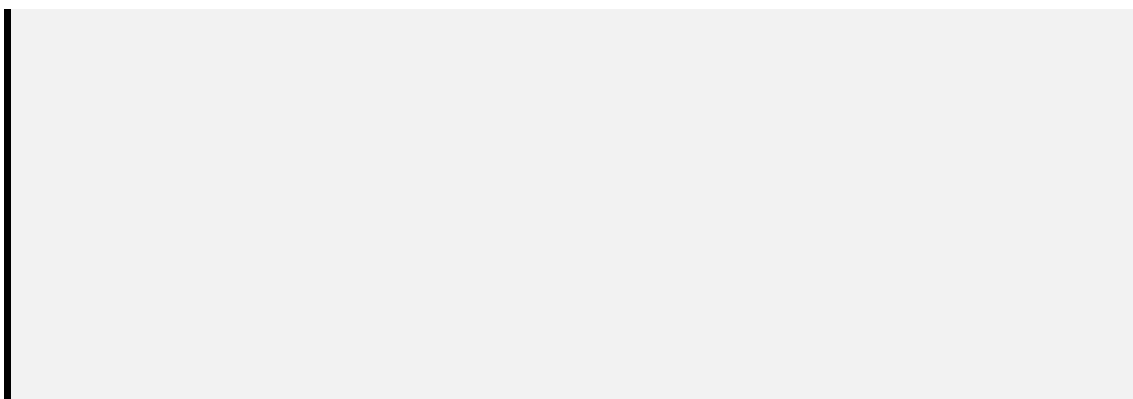
1 – Définir en scilab la fonction $g : x \mapsto \begin{cases} 2x & \text{si } x \leq 0 \\ \ln(x) & \text{si } x > 0 \end{cases}$.



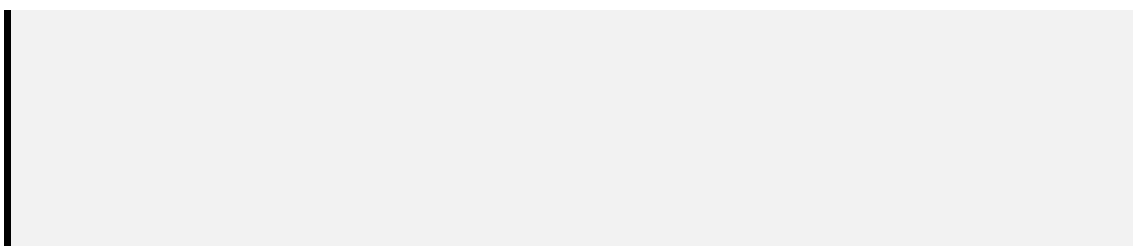
Calculer alors $g(5)$.



2 – Soit $(u_n)_{n \in \mathbb{N}}$ la suite définie par $\begin{cases} u_0 = 0 \\ \forall n \in \mathbb{N}, u_{n+1} = \sqrt{u_n + 2n} \end{cases}$.
Créer une fonction Suite_u(n) prenant en entrée un entier naturel n et renvoyant la valeur de u_n .



On considère maintenant la suite $(v_n)_{n \in \mathbb{N}}$ définie par $v_n = u_n + \frac{1}{n+1}$. Créer une fonction Suite_v(n) prenant en entrée un entier naturel n et renvoyant la valeur de u_n .



III Variables locales

Les **variables locales** d'une fonction sont :

- les variables d'entrée d'une fonction (les paramètres d'entrée);
- la variable de sortie;
- les variables dont la valeur est définie dans les instructions de la fonction.

Cela signifie que leur portée est réduite à la fonction.

```
function z = fonction1(x,y)
    z = 4*x + 5*y
endfunction
```

Variables locales de cette fonction :

```
--> exec(...)
--> fonction1(1,4)
ans =
    24.

--> x
Variable non définie : x

--> y
Variable non définie : y

--> z
Variable non définie : z

--> x = 2

--> fonction1(1,4)
ans =
    24.

--> x
x =
    2.
```

Les lettres x, y, z peuvent donc être utilisées ailleurs, sans aucune incidence sur la fonction1. Mais il ne faut pas non plus considérer que la variable z est connue globalement. **Seul le nom de la fonction est mémorisé.**