

2.1.5 Les fonctions et les procédures

Une fonction peut d'une part effectuer une action, par exemple afficher quelque chose ou modifier la mémoire, et d'autre part renvoyer une valeur. Une fonction qui ne renvoie pas de valeur s'appelle une *procédure*.

Dans certains langages, comme Pascal, les procédures sont distinguées et déclarées par un mot clé spécial. En Caml, à l'inverse, une procédure est simplement une fonction qui renvoie une valeur de type `unit`. Comme son nom l'indique `unit` est un type singleton qui ne contient qu'une valeur, écrite `()`. En Caml, une procédure renvoie donc invariablement la valeur `()`, ce qui ne transmet aucune information.

Le cas de Java et de C est intermédiaire puisque l'on déclare une procédure dans ces langages en remplaçant le type de la valeur retournée par le mot clé `void`. Contrairement au type `unit` de Caml, il n'y a pas réellement de type `void` en Java et en C. Par exemple, on ne peut pas déclarer une variable de type `void`.

L'appel d'une fonction, comme `hypothenuse(a,b)`, est une expression et celui d'une procédure, comme `annoncerRer("Massy-Palaiseau", "8h55")` ;, est une instruction.

Il y a cependant quelques nuances à apporter, car l'appel d'une fonction peut également être une instruction. On peut, par exemple, écrire l'instruction `hypothenuse(a,b)` ;. La valeur retournée par la fonction est simplement perdue. Cependant, même si les langages l'autorisent, une telle utilisation d'une fonction est jugée peu élégante. Les compilateurs de Caml, par exemple, donnent un avertissement au programmeur.

En Java et en C, une procédure, c'est-à-dire une fonction dont le type de retour est le type fictif `void` ne peut pas être utilisée comme une expression. En effet, pour écrire

```
x = annoncerRer("Massy-Palaiseau", "8h55");
```

il faudrait que la variable `x` soit de type `void` et nous avons vu qu'il n'y a pas de telle variable. En Caml, en revanche, une procédure n'est autre qu'une fonction dont le type de retour est `unit` et rien n'empêche d'écrire

```
x := annoncerRer("Massy-Palaiseau", "8h55")
```

pourvu que la variable `x` ait le type `unit ref`. Cependant, cela ne sert pas à grand chose.

De manière générale, quel que soit le langage, il est jugé élégant d'imposer une séparation entre les fonctions et les procédures. Les fonctions renvoient une valeur, n'effectuent pas d'action comme afficher quelque chose ou modifier la mémoire et sont utilisées comme expressions. Les procédures ne renvoient pas de valeur, peuvent effectuer une action, et sont utilisées comme instructions.