

*En C, une définition de fonction s'écrit comme en Java, en omettant le mot clé `static`.*

### 2.1.3 Le retour de valeur

```
a = 3;
b = 4;
c = 5;
d = 12;
u = Math.sqrt(a * a + b * b);
v = Math.sqrt(c * c + d * d);
```

Dans ce programme, on veut isoler le calcul `Math.sqrt(x * x + y * y)` dans une fonction `hypothenuse`. Mais à la différence de la fonction `annoncerRer` qui effectue une action d'affichage, la fonction `hypothenuse` doit calculer une valeur et la renvoyer au programme principal. Ce retour de valeur est le processus inverse du passage d'arguments qui transmet des informations du programme principal vers le corps de la fonction. Le type de la valeur retournée par la fonction s'écrit avant le nom de la fonction. La fonction `hypothenuse`, par exemple, se déclare de la manière suivante.

```
static double hypothenuse (final double x, final double y) {
    return Math.sqrt(x * x + y * y);}

```

Et le programme principal s'écrit alors ainsi.

```
a = 3;
b = 4;
c = 5;
d = 12;
u = hypothenuse(a,b);
v = hypothenuse(c,d);
```

*En Caml, la fonction `hypothenuse` s'écrit*

```
let hypothenuse x y = sqrt(x *. x +. y *. y)
```

*En C, la fonction `hypothenuse` s'écrit comme en Java, en omettant le mot clé `static` et en utilisant la fonction racine carrée de C qui s'écrit `sqrt` et non `Math.sqrt`.*

### 2.1.4 La construction `return`

Comme nous l'avons vu, en Caml, la fonction `hypothenuse` s'écrit