

A dark blue vertical bar runs down the left side of the page. A blue arrow points to the right from this bar, containing the text '2020/2021'.

2020/2021

# Fonction et procédure

INFO 0101

Several thin, curved lines in dark blue and light grey originate from the bottom left corner and sweep upwards and to the right.

quentin juilliard  
URCA

## Table des matières

Fonction .....	2
Les données pouvant être utilisées dans un fonction .....	2
Variables locales .....	2
Paramètre.....	2
Appel de fonction .....	0
Procédure : .....	1
Fonction et procédure en JAVA.....	1
Fonctionnement .....	1
Exemple de code avec la fonction procédure : .....	2

Exemple 1 :

Calculer le maximum de a et b, puis de c et d

Remarque Exemple 1 :

Même suite d'actions pour le calcul des 2 maxima.

Seules les variables utilisées sont différentes.

Dans les deux cas, le résultat est une valeur de type entier.

Principe « Diviser pour régner ».

Chaque fonction résout un sous-problème.

L'algorithme appelle ces fonctions.

Mise en commun d'actions : facilité de conception

### Fonction

Bloc (suite) d'action (instruction), nommé et éventuellement paramétré, retournant un résultat.

Lorsqu'une fonction est déclarée, on peut l'appeler par son nom. L'appeler (utiliser) autant de fois que nécessaire

Renvoie toujours un résultat ( une valeur), en convention, il retourne une valeur et on associe un type à une fonction : type de résultat.

Exemple 2 :

### Les données pouvant être utilisées dans un fonction

#### Variables locales

Les variables déclarées dans un fonction sont dites « locales » ( à cette fonction).

Elles sont utilisables uniquement dans le corps de la fonction ( la suite d'actions composant la fonction).

Elles n'existent que pour cette fonction.

#### Paramètre

Les paramètres permettent de transmettre des valeurs à une fonction. L'action réalisée dépend des valeurs transmises

x et y sont appelés les « paramètres formels » de la fonction max. Ils sont nommés (x et y). Ils sont également typés (x et y sont de type entier)

a et b sont appelés les « paramètres effectifs » du 1er appel. À chaque paramètre formel, on associe un paramètre effectif (du même type). De même, c et d sont les paramètres effectifs du second appel

## Appel de fonction

Déclaration -> paramètre formels

La signature d'une fonction est définie par son nom, le type de résultat, la liste des paramètres formels avec leurs types associés.

```
Fonction f (i : entier, j : entier, c : caractère) : réel
```

Utilisation (appel) -> paramètres effectifs

Lors d'un appel de fonction le déroulement est le suivant :

- 1/ Création d'autant de variables qu'il y a de paramètres formels, en respectant le type de chaque paramètre
- 2/ Initialisation des variables créées en recopiant la valeur des paramètres effectifs
- 3/ Les variables créées seront accessibles par le nom du paramètre formel associé

Paramètres et variables locales :

- 1/ Seules les actions du corps de la fonction peuvent les utiliser
- 2/ Existent uniquement pendant la durée d'exécution de la fonction
- 3/ Leur valeur sera perdue après le retour de l'appel

Exemple 3 :

### Procédure :

Bloc (suite) d'actions (instructions), nommé et éventuellement paramétré, ne renvoyant pas de résultat.

Les paramètres formels et paramètres effectifs (similaire aux fonctions)

Les fonctions et les procédures structurent les algorithmes !

<pre>Procédure afficherMultiples (x, y, m : entier) Déclarations   Variables locales     z : entier Début   //Pseudo-code de afficherMultiples... Fin</pre>	<pre>Algorithme Multiples Déclarations   Variables     n1, n2, multiple : entier Début   n1 ← lire ()   n2 ← lire ()   multiple ← lire ()   afficherMultiples(n1, n2, multiple) Fin</pre>
---	---

## Fonction et procédure en JAVA

### Fonctionnement

Avec java, en info0101, un programme est défini par une classe.

Les fonctions et procédures sont définies au niveau de cette classe.

Liste des paramètres, chaque paramètre est précédé par son type. Les paramètres sont séparés par une virgule.

Mot clé RETURN, il permet de définir la valeur du résultat de la fonction. Si des instruction sont placées après le mot clés return, elle ne sont jamais exécutées.

La suite d'instructions est définie dans un bloc entre {}. Elle commence par la déclaration des variables locales.

La déclaration de procédure :

```
public static void nomProcédure(liste_param) {
    suite_instructions ;
    //y compris les déclarations de variables locales
}
```

La déclaration de fonction :

```
public static type_de_retour nomFonction(liste_param) {
    suite_instructions ;
    //y compris les déclarations de variables locales
    return valeur ; //finit en renvoyant une valeur
}
```

Exemple de code avec la fonction procédure :

```
class Appels {  
    public static void main (String [] args) {  
        int i = 1;  
        while (i < 2)  
            i = f(i+1, i-3);  
        System.out.println("Dans le main : i = " + i);  
    } //fin du main (programme principal)  
  
    public static int f (int i, int j) {  
        p (i + j);  
        System.out.println("Dans f : i = " + i + " ; j = " + j);  
        return (2 * i - j + 3);  
    } //fin de la fonction f  
  
    public static void p (int j) {  
        int i = j + 3;  
        System.out.println("Dans p : i = " + i + " ; j = " + j);  
    } //fin de p  
}
```