1. **Définition d’une fonction  
   I.1 - Définition**  
   Une **fonction** est un sous-programme qui fournit un service à celui qui l’utilise (l’utilisateur de la fonction) et retourne une valeur de retour.

"Bonjour"

Pour fonctionner, la fonction a besoin d’informations :

ce sont les **arguments**, ou **paramètres** : ils sont fournis en entrée à la fonction.

**résultat** : valeur de retour

"Il fait beau"

. .12 .

*fonction* donnerLongueur

**I.2 - Exercice 1 : service rendu par une fonction**

**Travail à faire** : Compléter les espaces vides laissés ci-dessous.

a)

BONJOUR

mettreEnMajuscules

"Bonjour"

b)

3

8

calculerSomme

11

c)

10

20

calculerMoyenne

15

d)

"1"

"172.16.0.1"

nbrCarDansMot

3

e)

(format)

obtenirDateDuJour

23/09/2021. . . .

"jj/mm/aaaa"

f)

true (booléen)

"24,51"

estNumérique

g)

"70 ~~av. J. Jaurès~~"

false (booléen)

estNumérique

1. **Appel d’une fonction en Java**

**Travail à faire** : Compléter les espaces vides laissés ci-dessous.

**Exemple 1** :

texte longueur

12

50

5

5

Il fait beau

String texte = "Il fait beau";

Il fait beau

int longueur = **getLongueur**(texte);

System.out.println("longeur du texte : " + longueur);

Qu’est-ce qui s’affiche ? longeur du texte : 12 . . . . . . . . . . . . .

**Exemple 2** :

nb1 nb2 lemax

int nb1, nb2, lemax ;

nb1 = 5;

nb2 = 50;

50

50

5

lemax = **max**(nb1, nb2);

System.out.println("le plus grand est : " + lemax);

Qu’est-ce qui s’affiche ? le plus grand est : 50 . . . . . . . . . . . . .

**Pour résumer :**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Variable dans laquelle sera affectée la **valeur retournée** par la fonction |  | **Nom** de la fonction |  | **Paramètre** (ou **argument**) de la fonction (mis entre parenthèses) |

longueur = **getLongueur** (texte) ;

1. **Signature d’une fonction**

La **signature d’une fonction** exprime en une ligne quel est son **contrat,** c’est-à-dire ses **conditions d’utilisation**.

Par exemple, la documentation de Java donne la signature de la fonction **max** de la classe **Math** :

public static int **max**​ (int a, int b)

**nom de la fonction . . .**

**1er paramètre de type int . . .**

**2è paramètre int . . .**

**Type de la valeur retournée**

**. . .**

**Modificateur d’accès : public = fonction accessible de partout. . .**

**Description** : Retourne la plus grande des deux valeurs entières (int) a et b.

**Exercice 2 :** compléter le programme suivant qui demande de saisir deux entiers et qui affiche le plus grand, obtenu avec la fonction **max**.

Scanner sc = new Scanner(System.in);

System.out.println(“veuillez saisir deux nombres”);

double nb1 = sc.nextDouble();

double nb2 = sc.nextDouble();

double max; // le max

// obtenir le plus grand en utilisant **Math.max fonctionne aussi pour des double**

max = Math.max( nb1, nb2) ;

System…

. . .

**Exercice 3 :** compléter le programme suivant qui permet d’obtenir le nombre le plus petit de deux nombres saisis. Utilisez la fonction **Math.min** dont voici la signature :

double double a, double b

public static ~~int~~ **min**​ (~~int~~ a, ~~int~~ b)

**Description** : Retourne la plus petite des deux valeurs entières (int) a et b.

// programme

Scanner sc = new Scanner(System.in);

System.out.println(“veuillez saisir deux nombres”);

double nb1 = sc.nextDouble();

double nb2 = sc.nextDouble();

double leMin ;

leMin = Math.min( nb1, nb2) ; // etc.

**Exercice 4 : Simuler un dé à 6 faces**

**Objectif :** créer un programme qui génère un nombre aléatoire compris entre 1 et 6.

// obtenir un nombre au hasard compris entre 0 et 1

// à mettre dans variable nbHasard

double nbHasard ;

nbHasard = Math.random() ;

// multiplier ce nombre par 6

nbHasard = nbHasard \* 6;

// arrondir ce nombre à l'entier inférieur

// avec la fonction floor

// et ajouter +1

nbHasard = Math.floor(nbHasard);

nbHasard++;

// ou nbHasard = nbHasard  + 1;

// ou nbHasard += 1;

// affichage

System.out.println( nbHasard );

**Documentation** **des fonctions** :

public static double random()

Retourne une valeur (double) positive, comprise entre 0.0 (inclus) et 1.0 (exclu). La valeur est générée aléatoirement.

public static double floor (double a)

Retourne le nombre entier le plus grand, qui est inférieur ou égal à l’argument.

Paramètre : a, une valeur

1. **Ecriture d’une fonction**

Le développeur peut aussi écrire ses propres fonctions qui seront susceptibles d’être réutilisées.

Intérêts : gain de temps et meilleure lisibilité du programme principal.

// programme principal note1 note2 moy

Scanner sc = new Scanner(System.in);

System.out.println(“Saisissez deux notes”);

10

double note1 = sc.nextDouble();

20

double note2 = sc.nextDouble();

double moy ;

15

moy = ***calculerMoyenne***(note1, note2);

System.out.println("la moyenne est " + moy);

// fonction écrite par le développeur nb1 nb2 moyenne

10

20

**public double** ***calculerMoyenne***(double nb1, double nb2)

{

double moyenne ;

15

moyenne = (nb1 + nb2) / 2;

**return** moyenne;

}

Qu’est-ce qui s’affiche ? . . . . . . . . . . . . .

La moyenne est 15

Remarque sur la portée des variables :

note1 et note2 ont une **portée** limitée au programme principal (main)

Les paramètres nb1 et nb2 ont une portée limitée à la fonction calculerMoyenne.

. . . . . . . . . . . . .