

Premiers pas en programmation

Chap. 2,1

Exercice 1. Découverte et utilisation de deux fonctions fondamentales

Les deux fonctions intégrées `dir` et `help` sont fondamentales. Les utiliser *dans la console* pour répondre aux questions ci-dessous.

1. La fonction intégrée `sum` intégrée au langage calcule-t-elle la somme de deux nombres ?
2. Que réalisent les fonctions intégrées `int`, `float`, `str`, `bool` ?
3. Quel traitement réalise la fonction `pow` du module `math` ? Donner un exemple d'utilisation.
4. Comment se nomme la fonction du module `math` qui calcule la racine carrée d'un nombre passé en argument ?

Exercice 2. Utilisation d'une fonction

1. Peut-on appeler la fonction `randint` du module `random` sans argument ?

Rappel. La fonction intégrée `help` permet d'obtenir la documentation de n'importe quelle fonction.

2. Que retourne l'appel de la fonction `randint` du module `random` ?
3. Appeler cette fonction `randint` de façon à obtenir un nombre entier aléatoire compris entre 1 et 10.
4. Est-il possible que l'appel précédent retourne 1 ou 10 ?

Exercice 3. Définition d'une fonction

1. La fonction définie ci-dessous est-elle syntaxiquement correcte ?

```
def retire_un(n):  
    return 1 - n
```

2. La spécification est la suivante :

```
def retire_un(n: float) -> float:  
    """ Retourne la valeur de n - 1 """
```

La fonction telle qu'elle est définie à la question 1 correspond-elle à cette spécification ?

3. Corriger le corps de la fonction `retire_un` de façon à ce que le calcul effectué corresponde à ce qu'annonce la spécification.

Exercice 4. Calcul de fonctions polynomiales

1. Définir et tester une fonction écrite en Python, nommée `polynomiale`, telle que `polynomiale(a, b, c, d, x)` retourne la valeur de la fonction qui à x associe $ax^3 + bx^2 + cx + d$.

La documentation complète de cette fonction est la suivante :

```
def polynomiale(a:int, b:int, c:int, d:int, x: float) -> float:
    """ Retourne la valeur ax^3 + bx^2 + cx + d
    >>> polynomiale(1, 1, 1, 1, 2) 15.0
    >>> polynomiale(1, 1, 1, 1, 3) 40.0
    """
```

2. Définir et tester une fonction écrite en Python, nommée `polynomiale_carre` qui retourne la valeur de la fonction $ax^4 + bx^2 + c$.

La documentation complète de cette fonction est la suivante :

```
def polynomiale_carre(a:int, b:int, c:int, x: float) -> float:
    """ Retourne la valeur ax^4 + bx^2 + c
    >>> polynomiale(1, 1, 1, 1, 2) 15.0
    >>> polynomiale(1, 1, 1, 1, 3) 40.0
    """
```

Exercice 5. Calcul de la moyenne de trois nombres

1. Définir et tester la fonction nommée `somme`, écrite en Python, dont la spécification est :

```
def somme(x:float, y: float, z: float) -> float:
    """ Retourne la somme des trois nombres passés en argument. """
```

2. Définir et tester une fonction écrite en Python, nommée `moyenne`, qui détermine la moyenne arithmétique de trois nombres.

Remarque. La fonction `moyenne` doit utiliser la fonction `somme` définie à la questions précédente pour effectuer son traitement.

La spécification de la fonction est la suivante :

```
def moyenne(a: float, b: float, c: float) -> float:
    """ Retourne la moyenne des trois nombres passés comme arguments. """
```

3. Définir et tester une fonction écrite en Python, nommée `moyenne_ponderee`, qui détermine la moyenne pondérée de trois nombres avec des coefficients variables. Les paramètres devront être écrits dans l'ordre suivant : *d'abord les trois nombres, puis les trois coefficients*.

Remarque. La fonction `moyenne_ponderee` doit effectuer deux appels à la fonction `somme` pour effectuer son traitement.

La documentation complète de la fonction est la suivante :

```
def moyenne_ponderee(x:float, y:float, z:float, a:int, b:int, c:int) -> float:
    """ Retourne la moyenne pondérée des nombres x, y et z par les coefficients a,
        b et c.

    >>> moyenne_ponderee(5, 10, 15, 1, 1, 1)
    10.0

    >>> moyenne_ponderee(5, 10, 15, 1, 1, 0)
    7.5
    """
```

Exercice 6. Calculs de surfaces et de volumes

1. Définir et tester une fonction écrite en Python, nommée `surface_rectangle`, qui détermine la surface d'un rectangle de longueur a et de largeur b .

Préciser la spécification de la fonction.

2. Définir et tester une fonction écrite en Python, nommée `volume_parallelepipede`, qui détermine le volume d'un parallélépipède rectangle de longueur a , de largeur b et de hauteur h .

Préciser la spécification de la fonction.

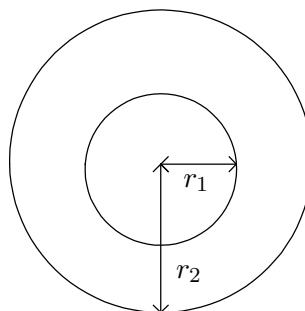
Cette fonction devra utiliser la fonction `surface_rectangle` pour effectuer son traitement son traitement.

3. Définir et tester une fonction écrite en Python, nommée `surface_disque`, qui détermine la surface d'un disque de rayon r .

Rappel. Le module `math` possède une variable `pi`.

Préciser la spécification de la fonction.

4. Définir et tester une fonction écrite en Python, nommée `surface_couronne`, qui détermine la surface d'une couronne de rayon intérieur r_1 et de rayon extérieur r_2 .



Préciser la spécification de la fonction.

Cette fonction devra utiliser la fonction `surface_disque` pour effectuer son traitement.

5. Définir et tester une fonction écrite en Python, nommée `volume_tube`, qui détermine le volume de la partie pleine d'un tube de longueur l , dont la section est une couronne de rayon intérieur r_1 et de rayon extérieur r_2 .

Préciser la spécification de la fonction.

Cette fonction devra utiliser la fonction `surface_couronne` pour effectuer son traitement.

Exercice 7. Calcul d'une distance

Définir et tester une fonction écrite en Python, nommée `distance` qui détermine la distance séparant deux points de coordonnées (x_1, y_1) et (x_2, y_2) d'un plan.

La documentation complète de la fonction est la suivante :

```
def distance(x1: float, y1: float, x2: float, y2: float) -> float:
    """ Retourne la distance dans le plan entre les deux points de coordonnées
    (x1, y1) et (x2, y2).

    >>> distance(0, 0, 1, 1)
    1.4142135623730951
    """
```

Exercice 8. Reprise de l'exercice 1

L'objectif de cet exercice est de retrouver une partie du comportement de la fonction `randint` du module `random` à partir de la fonction `random` de ce même module.

Définir et tester une fonction écrite en Python, nommée `tirage_entier`, telle que `tirage_entier(a)` retourne un nombre entier compris entre 1 et a (inclus), au hasard. *La fonction `tirage_entier` doit utiliser la fonction `random` du module `random`.*

Remarque. Lire attentivement la documentation de la fonction `random` afin de comprendre ce qu'elle retourne.

Rappel. La fonction intégrée `int` permet de convertir n'importe quel nombre en nombre entier.