

TP Python : Fonctions manipulant des nombres

- Auteur : Gwénaél LAURENT
- Date : 17/09/2024



Licence Creative Commons

Exo_1 : Surface d'un disque

Définissez une fonction qui calcule la surface d'un disque :

- Nom de la fonction : `calculer_aire_disque`
- Un paramètre : `r`, le rayon du disque, de type float
- Une valeur de retour : la surface du disque

Rappel : la surface d'un disque de rayon R est de πR^2

Note : la valeur approchée de π fait partie du module de math. Pour l'utiliser, il faut d'abord importer la variable `pi`. Exemple d'utilisation :

```
>>> from math import pi
>>> print(pi)
3.141592653589793
```

Dans le programme principal, affichez le résultat de l'**appel de la fonction** quand on lui passe en argument le rayon sous forme de :

- un nombre littéral
- une variable contenant un float
- une variable contenant un int

N'oubliez pas les *docstring* du script et de la fonction !

Exo_2 : Note sur 20

Vous avez sûrement déjà reçu un note de DS qui était noté sur un nombre de point différent de 20 😊
Comment faire pour connaître la note correspondante sur 20 ?

Définissez une fonction qui retourne la note sur 20 :

- Nom de la fonction : `note_sur_20`
- 2 paramètres : `note` et `total`, type float
- Une valeur de retour : la note /20

Testez votre fonction.

Exemple de résultat à obtenir :

```
Saisissez votre note : 35
Le devoir était noté sur combien de point ? 40
35.0/40.0 équivaut à 17.50/20
```

Exo_3 : Prix du trajet en voiture

Vous allez réaliser un programme qui calcule le prix de l'essence pour un trajet en voiture.

Définissez une fonction qui calcule le prix du trajet :

- Nom de la fonction : `prix_du_trajet`
- 4 paramètres :
 - `conso_v` : la consommation de la voiture en litres/100km
 - `prix_l` : le prix d'un litre d'essence
 - `kms` : le nombre de kilomètres parcourus
 - `nb_pers` : le nombre de personne dans la voiture
- Une valeur de retour : le prix du trajet pour une personne

Exemple de résultat à obtenir :

```
Consommation de la voiture (litres/100km) : 6
Prix d'un litre d'essence (€): 1.546
Nombre de kilomètres parcourus : 200
Nombre de personne dans la voiture : 3
Le trajet revient à 6.18€ par personne.
```

Exo_4 : Volume d'une boîte

Définissez une fonction `vol_boite(larg, long, haut)` qui renvoie le volume d'une boîte parallélépipédique dont on fournit les trois dimensions largeur, longueur, hauteur en arguments (en cm).

Etape 1 : Le calcul

Créez votre script avec la définition de la fonction et l'appel de cette fonction.

Exemple de résultat à obtenir :

```
Largeur (cm): 5.2
Longueur (cm): 7.7
Hauteur (cm): 3.3
Le volume de la boîte est de 132.13cm3
```

Etape 2 : Paramètres avec valeurs par défaut

Modifiez la fonction de manière à ce qu'elle puisse être appelée avec trois, deux, un seul, ou même aucun argument.

Utilisez pour ceux ci des valeurs par défaut égales à 10

Testez votre fonction avec tous les nombres d'arguments possibles.

Exo_5 : Convertisseur Euros -> Dollars

Réalisez un programme qui affiche la conversion de devises Euros -> Dollars. Pour être pratique, votre programme affichera la conversions pour plusieurs prix.

Exemple de résultat à obtenir entre 10€ et 30€ :

```
10€ -> 11.10$  
15€ -> 16.65$  
20€ -> 22.20$  
25€ -> 27.75$  
30€ -> 27.75$
```

Créez une procédure `afficher_conversions()` qui affiche les conversions de devises

- Paramètres :
 - `taux` : taux de change exprimé en décimal (ex : 1.11 signifie que 1€ = 1.11\$)
 - `debut` : première valeur en € à convertir
 - `fin` : valeur max en € à convertir
 - `pas` : écart entre chaque valeurs en € (par défaut = 10)
- Pas de valeur de retour

Exo_6 : Teste de primalité

Un test de primalité est un algorithme permettant de savoir si un nombre entier est premier.

Un **nombre premier** est un entier naturel qui admet exactement deux diviseurs distincts entiers et positifs. Ces deux diviseurs sont 1 et le nombre considéré, puisque tout nombre a pour diviseurs 1 et lui-même (comme le montre l'égalité $n = 1 \times n$), les nombres premiers étant ceux qui ne possèdent pas d'autre diviseur. Par exemple, le nombre entier 7 est premier car 1 et 7 sont les seuls diviseurs entiers et positifs de 7.

Les nombres 0 et 1 ne sont donc ni premiers ni composés : 1 n'est pas premier car il n'a qu'un seul diviseur entier positif et 0 non plus car il est divisible par tous les entiers positifs.

Les 25 nombres premiers inférieurs à 100 sont :

2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, 59, 61, 67, 71, 73, 79, 83, 89, et 97.

Le test de primalité d'un nombre N se fera en suivant la **méthode naïve** suivante :

- On commence par supposer qu'il est premier
- On teste s'il est divisible par l'un des entiers compris entre 2 et $N-1$.
- Dès qu'on trouve un diviseur, alors N n'est pas premier. S'il n'y a pas de diviseur, N est premier.

Ecrivez un programme qui affiche si un nombre saisi est un nombre premier, ou pas. Le test de primalité sera réalisé dans une fonction `est_premier()`.

Exemples de résultat :

```
Entrez le nombre à tester : 1979
1979 est un nombre premier
```

```
Entrez le nombre à tester : 2048
2048 n'est pas un nombre premier
```