
IngeSUP - TD 07 - Les fonctions 1

“Sur la terre tout à une fonction, chaque maladie une herbe pour la guérir, chaque personne une mission.”

Sagesse indienne

Exercice 07.1

Ecrivez une fonction qui renvoie l'aire d'un disque en fonction de la valeur du rayon.

Indication: Pour importer la constante `pi` en Python on fera appel à la bibliothèque `math` en tapant l'instruction suivante, répliquée dans la cellule ci-dessous:

```
from math import pi
```

Entrée []:



```
# Complétez le code ci-dessous

from math import pi

def aireDisque(r):
    aire = ...
    return ...

# Noubliez pas de tester la fonction !
print(aireDisque(4))

# Ou essayez de tester la fonction via input
val = int(input("De quel rayon vous voulez l'aire ?"))
print(aireDisque(val))
```

Exercice 07.2

A l'aide de la fonction `aireDisque()` codée précédemment:

1. Ecrivez une fonction qui renvoie le volume d'un cylindre en fonction du rayon et de la hauteur.

Indication: Le volume V d'un cylindre est égal au produit de π (Pi) par le rayon au carré et par la hauteur :

$$V = \pi \times r^2 \times h$$

2. Testez votre fonction avec un cylindre dont le disque est de rayon 3 et de hauteur 2.

Entrée []:



```
# 1. Complétez le code ci-dessous

def volumeCylindre(r,h):
    base= ...      # Appel à la fonction aireDisque sur r et relation avec h
    return base

# 2. Tester la fonction avec un rayon de 3 et une hauteur de 2
print(volumeCylindre(3,2))

# 3. Tester la fonction en demandant à l'utilisateur via input
rayon = ...
hauteur = ...
print(volumeCylindre(rayon, hauteur))
```

Exercice 07.3

Codez une fonction appelée `affiche_bonjour()` qui demande à l'utilisateur son prénom et son nom de famille et qui affiche ensuite « Bonjour » suivi du prénom de l'utilisateur et de son nom de famille.

Entrée []:



```
# VOTRE CODE ICI
```

Exercice 07.4

Ecrivez une fonction `nombreVoyelles()` qui compte le nombre de voyelles (minuscules, sans accents) dans une chaîne de caractères passée en paramètres.

Entrée []:



```
# VOTRE CODE ICI
```

Exercice 07.5

On se propose d'écrire une fonction qui calcule la valeur d'un réel x élevé à la puissance n .

1. Ecrivez une fonction `puissance(x,n)` qui prend en paramètre un réel x et une puissance n et qui renvoie x^n .

Entrée []:



```
# VOTRE CODE ICI
```

2. Modifiez la fonction précédente pour qu'elle détermine la valeur d'un réel x élevé à la puissance n à l'aide d'une boucle **for**.

Entrée []:



```
# VOTRE CODE ICI
```

Exercice 07.6

Le but de cet exercice est de coder en Python certaines des opérations qui concernent les polynômes du second degré.

1. Écrivez une fonction `trinome()` qui dépend de quatre paramètres a , b , c et x et qui renvoie la valeur du trinôme $ax^2 + bx + c$.

Par exemple `trinome(2,-1,0,6)` renvoie 66.

Entrée []:



```
# VOTRE CODE ICI
```

2. Écrivez une fonction `racine()` qui, en fonction de trois paramètres a , b et c , affiche le nombre de racines dont dispose le polynôme $ax^2 + bx + c$.

Indication: La fonction ne renvoie aucune valeur. L'affichage peut se faire par le recours à des **print**, Par exemple : "Ce polynôme contient deux racines."

Indication 2: N'oubliez pas le calcul du discriminant...

Entrée []: ▶

VOTRE CODE ICI

3. Cette fois on veut connaître la valeur des racines en question. Pour ça on veut que le programme renvoie trois types de valeurs différentes.

* Si le polynôme n'a pas de racines on renvoie **None** (c'est un mot-clé de Python qui exprime le vide).

* Si le polynôme a une seule racine on la renvoie.

* Si le polynôme a deux racines on les renvoie toutes les deux séparées par une virgule (tuple).

(On ne s'intéresse qu'aux racines réelles).

Modifiez la fonction précédente pour faire ce calcul.

Entrée []: ▶

VOTRE CODE ICI

Exercice 07.7

Voici la table de vérité des opérateurs Booléen ET, OU et NON :

A	B	Résultat ET
F	F	F
F	V	F
V	F	F
V	V	V

A	B	Résultat OU
F	F	F
F	V	V
V	F	V
V	V	V

A	Résultat NON
F	V
V	F

Écrivez des fonctions `ET()` , `OU()` , `NON()` , qui, à l'aide de structures conditionnelles, prennent deux booléens `a` et `b` en paramètres (`NON` ne prend qu'un seul booléen en paramètre) et renvoient `True` si selon la table de vérité leur combinaison est vraie et `False` si selon la table de vérité leur combinaison est fausse.

Entrée []:



```
# VOTRE CODE ICI
```

Exercice 07.8

On définit un palindrome comme une chaîne de caractères qui se lit de la même manière de gauche à droite et de droite à gauche.

Écrivez une fonction `palindrome()` qui retourne `True` si le mot qu'on lui donne en paramètre est un palindrome et `False` si ça n'est pas le cas.

Par exemple, votre fonction retournera `True` pour « RADAR » ou « KAYAK » mais `False` pour « exercice ».

Entrée []:



```
# VOTRE CODE ICI
```

Exercice 07.9

Écrire une fonction `strip_voyelles()` qui prend en paramètres une chaîne de caractères quelconque et renvoie la même chaîne sans ses voyelles.

Entrée []:



```
# VOTRE CODE ICI
```

Exercice 07.10

1. Écrivez une fonction `conversion_euros_vers_dollars()` qui dépend d'un paramètre et qui, pour une somme d'argent `montant`, exprimée en euros, renvoie sa valeur en dollars (on prendra 1 euro = 1.15 dollar).

Entrée []:



```
# VOTRE CODE ICI
```

2. Écrivez une fonction `conversion_euros()` qui dépend d'un paramètre `montant` et d'une monnaie `devise` et qui convertit la somme `montant` donnée en euros, dans la devise souhaitée.

Exemples de devises : 1 euro = 1.15 dollars ; 1 euro = 0.81 livres ; 1 euro = 130 yens.

Par exemple, `conversion_euros(100,"livre")` renvoie 81.

Entrée []:



```
# VOTRE CODE ICI
```

Corrigé du TD 07

Vous pouvez retrouver le corrigé de ce TD [ici \(Corrig%C3%A9s/Corrig%C3%A9_TD%2007.ipynb\)](#).