
IngeSUP - TD 07 - Les fonctions 1

Entrée []:



“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 [10]:



```
# Complétez le code ci-dessous

from math import pi

def aireDisque(r):
    aire = r**2*pi
    return aire

# 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))
```

```
c'est parti
50.26548245743669
De quel rayon vous voulez l'aire ?56
c'est parti
9852.03456165759
```

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 [2]:



```
# 1. Complétez le code ci-dessous

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

# 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 = float(input("rayon ?"))
hauteur = float(input("base ?"))
print(volumeCylindre(rayon, hauteur))
```

```
56.548667764616276
rayon ?3
base ?6
169.64600329384882
```

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 [3]:



```
def affiche_bonjour():
    nom=input("quel est votre nom ?")
    prenom=input("quel est votre prénom ?")
    print("bonjour", prenom, nom)

affiche_bonjour()
affiche_bonjour()
```

```
quel est votre nom ?ùnfvj bq
quel est votre prénom ?mkjfvbaer
bonjour mkjfvbaer ùnfvj bq
quel est votre nom ?ajnerùgj nae
quel est votre prénom ?amejrbeajùr
bonjour amejrbeajùr ajnerùgj nae
```

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 [4]:



```
def nombreVoyelles(texte):
    compteur=0
    for i in texte:
        if i in "aeiouy":
            compteur+=1
    return compteur

print(nombreVoyelles("bonjour, comment ça va ?"))
print(nombreVoyelles("super merci"))
```

7
4

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 [1]:



```
def puissance(x,n):
    resultat=x**n
    return resultat

print(puissance(3,4))
```

81

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 [2]:



```
def puissanceBis(x,n):
    resultat=1
    for i in range(n):
        resultat *= x
    return resultat

print(puissanceBis(3,4))
```

81

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 [8]:



```
def trinome(a,b,c,x):
    return a*x**2+b*x+c

print(trinome(2,-1,0,6))
```

66

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 [1]:



```
def racine(a,b,c):
    delta=b**2-4*a*c
    if delta>0:
        print("deux racines")
    elif delta==0:
        print("une racine")
    else:
        print("pas de racine")

racine(2,-1,0)
```

deux racines

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 [12]:



```
def racine(a,b,c):
    delta=b**2-4*a*c
    if delta>0:
        return ((-b-(delta)**(1/2))/(2*a),(-b+(delta)**(1/2))/(2*a))
    elif delta==0:
        return -b/(2*a)
    else:
        return None

print(racine(2,-1,0))
```

(0.0, 0.5)

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 [14]:



```
def ET(a,b):
    if a==True and b==True:
        return True
    else:
        return False

#si je me force a ne pas utiliser la fonction and

def ET(a,b):
    if a==True:
        if b==True:
            return True
        else:
            return False
    if a ==False:
        return False

print(ET(1==2,3==1+2))

#on va essayer d'utiliser la fonction ET qu'on vient de faire
def OU(a,b):
    if ET(a==False,b==False):
        return False
    else:
        return True

def NON(a):
    if a==True:
        return False
    else:
        return True
```

False

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 [15]:



```
def palindrome(mot):
    L=[]
    for i in mot:
        L.append(i)
    L2=L.copy()
    L2.reverse()
    if L==L2:
        return True;
    else:
        return False

print(palindrome("radar"))

#variantes sur les indices :
def palindrome(mot):
    for i in range(len(mot)//2) :
        if mot[i] != mot[-i-1] :
            return False
    return True

print(palindrome('bb'))

#variante encore plus rapide :
def palindrome(chaine):
    chaine2 = chaine[::-1]
    if chaine2 == chaine:
        return True
    else:
        return False
print(palindrome("KAYAK"))
```

True
True
True

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 [28]:



```
def strip_voyelle(chaine):
    for i in "aeiouy":
        chaine=chaine.replace(i, "")
    return chaine

print(strip_voyelle("salut"))

#variante:
def strip_voyelle(chaine):
    resultat=""
    for i in chaine:
        if i not in "aeiouy":
            resultat+=i
    return resultat

print(strip_voyelle("salut"))
```

slt
slt

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 [48]:



```
def conversion_euros_vers_dollars(montant):
    return 1.02*montant

print(conversion_euros_vers_dollars(1500))
```

1530.0

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 [50]:



```
def conversion_euros(montant, devise):  
    if devise=="dollar":  
        return montant*1.02  
    elif devise=="livre":  
        return montant*0.87  
    #on peut mettre plein de devises  
  
print(conversion_euros(150,"livre"))
```

130.5

Corrigé du TD 07

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