

Solutions TP 1

Exercice 1 :

Voici quelques commandes pour commencer à utiliser l'interpréteur (dans la console). Cherchez à comprendre le résultat obtenu et les éventuels messages d'erreur.

a) Affectation d'une variable.

```
>>> x=10
>>> y=5
>>> x+y
```

b) Chaînes de caractères.

```
>>> c="bonjour"
>>> c[3]
>>> c[-1]
>>> c[2:]
>>> c[:3]
>>> c[3:5]
>>> c[-1::-1]
>>> c[:2]
```

```
>>> s='i vaut'
>>> i=1
>>> print(s,i)
```

```
>>> print ('*-'*5)
```

```
>>> print('1 < 0 est une expression ', 1 < 0)
```

c) Fonctions mathématiques : tapez les commandes suivantes et observez les résultats.

```
>>> import math
>>> help(math)
>>> import numpy as np
>>> math.pi
>>> pow(math.sin(2.5),2)+pow(math.cos(2.5),2)
>>> pow(np.sin(2.5),2)+pow(np.cos(2.5),2)
>>> math.exp(math.log(3))
>>> math.sqrt(-1)
>>> np.sqrt(-1)
```

A partir de maintenant vous allez écrire vos programmes dans des fichiers exécutables par Python. Pour cela, utilisez l'éditeur de Idle et créez un fichier `exoNumero.py` pour chaque exercice.

Exercice 2 :

Écrire un programme, qui permet à l'utilisateur de saisir successivement trois variables : une variable de type texte, une variable de type nombre entier, une variable de type nombre décimal et qui affiche leur type et leur valeur.

```
a = input('Entrez un entier')
b = input('Entrez un nombre decimal')
c = input('Entrez une chaine de caracteres')
n=int(a)
x=float(b)
print(n,' est de type ',type(n))
print(x,' est de type ',type(x))
print(c,' est de type ',type(c))
```

Exercice 3 :

Écrire un programme qui demande :

1. un angle en degrés et le convertit en radians.
2. un angle en radians et le convertit en degrés.

Tester le programme avec $\pi/4$.

```
from math import pi, degrees, radians

# Conversion degres -> radians
saisie = input("Entrez une valeur d'angle en degres (ex: pi/4) : ").strip().lower()
# strip() : supprime les espaces au debut et a la fin de la saisie
# lower() : transforme tous les caracteres en minuscules (utile si l'utilisateur ecrit '
#                                     PI/4' ou 'Pi')
# eval() : evalue l'expression saisie comme du code Python,
#          ici on autorise l'utilisation de "pi" pour ecrire des fractions comme "pi/4"
angd = eval(saisie, {"pi": pi}) # {"pi": pi} permet d'utiliser la valeur de pi dans l'
                                expression

# Conversion degres -> radians
angd = lire_angle("Entrez une valeur d'angle en degres (vous pouvez ecrire pi, pi/4, 2*pi
, etc.) : ")

angr1 = angd * pi / 180
angr2 = radians(angd)

print(angr2)
print("Un angle de", angd, "degres vaut", angr1, "radians")

# Conversion radians -> degres
angr = lire_angle("\nEntrez une valeur d'angle en radians (vous pouvez ecrire pi, pi/4, 2
*pi, etc.) : ")

angd1 = angr * 180 / pi
angd2 = degrees(angr)

print(angd2)
print("Un angle de", angr, "radians vaut", angd1, "degres")
```

Exercice 4 :

Écrire un programme qui affiche "Bonjour le monde".

```
print('Bonjour tout le monde')
```

Exercice 5 :

Écrire un programme qui permet de saisir le nom de l'utilisateur et de renvoyer "Bonjour", suivi de ce nom.

```
nom=input('Entrez votre nom : ')\nprint('Bonjour',nom)
```

Exercice 6 :

En ne gardant que 3 chiffres après la virgule, calculez le quotient de 5,357 par 2,08.

```
print (round(5.357/2.08,3))
```

Exercice 7 :

Calculez la racine carrée du nombre 200 en ne gardant que 3 décimales.

```
from math import *\nprint (round(sqrt(200),3))
```

Exercice 8 :

Utilisez le nombre donné par la fonction pi du module math pour calculer le périmètre et l'aire d'un disque de 4.56 cm de diamètre. Ne conservez que 2 décimales dans le résultats.

```
from math import *\nR=4.56\nprint (round(pi*R*R,2))\nprint (round(2*pi*R,2))
```

Exercice 9 :

Écrire un programme qui demande à l'utilisateur la saisie de a et b et affiche la somme de a et de b.

```
a=int(input('Entrez a :'))\nb=int(input('Entrez b :'))\nprint (a+b)
```

Exercice 10 :

Écrire un programme qui demande à l'utilisateur son année de naissance et qui affiche son âge. L'année courante sera mise dans une variable.

```
import datetime\ndate=datetime.datetime.now()\nannee_actuelle = date.year\nanne_naissance=int(input('annee de naissance ? '))\nage= annee_actuelle-anne_naissance\nprint('Vous avez',age, 'ans')
```

Exercice 11 :

Écrire un programme qui demande à l'utilisateur les coordonnées de deux points dans le plan et qui calcule puis affiche la distance entre ces deux points selon la formule :

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}.$$

```
from math import sqrt
a,b = input('Entrez x1,y1 separes par un espace :').split()
x1=float(a)
y1=float(b)
a,b = input('Entrez x2,y2 separes par un espace :').split()
x2=float(a)
y2=float(b)
dist=(x2-x1)*(x2-x1)+(y2-y1)*(y2-y1)
dist=sqrt(dist)
print('La distance est :',round(dist,3))
```

Exercice 12 : À vol d'oiseau

Écrire un programme qui calcule la distance entre deux points à la surface de la Terre à partir de leurs latitudes ϕ_A , ϕ_B et longitudes λ_A , λ_B . En coordonnées sphériques, la distance **angulaire**, θ_{AB} , entre ces deux points est telle que :

$$\cos \theta_{AB} = \sin \phi_A \sin \phi_B + \cos \phi_A \cos \phi_B \cos(\lambda_B - \lambda_A)$$

Paris, Londres et New-York sont respectivement situées aux coordonnées (48.86° N, 2.35° E), (51.51° N, 0.13° W) et (40.71° N, 74.01° W). Calculer la distance entre Paris et Londres puis entre Paris et New York (attention aux signes, attention aussi à la donnée manquante).

La distance à vol d'oiseau entre Paris et Londres est d'environ 344 km et celle entre Paris et New-York d'environ 5681 km. Vos résultats sont-ils cohérents ? Expliquez les éventuelles différences.

```
from math import radians,degrees,cos,sin,acos
pparis=48.86
ppa=radians(pparis)
lparis=2.35
lpa=radians(lparis)
plondres=51.51
plo=radians(plondres)
llondres=-0.13
llo=radians(llondres)
pnewyork=40.71
pne=radians(pnewyork)
lnewyork=-74.01
lne=radians(lnewyork)
rterre= 6371
dist12=rterre*acos(sin(ppa)*sin(plo)+cos(ppa)*cos(plo)*cos(llo-lpa))
print('distance Paris Londres: ',int(dist12), ' km')
dist12=rterre*acos(sin(ppa)*sin(pne)+cos(ppa)*cos(pne)*cos(lne-lpa))
print('distance Paris New-york: ',int(dist12), ' km')
print(acos(sin(ppa)*sin(plo)+cos(ppa)*cos(plo)*cos(llo-lpa)),acos(sin(ppa)*sin(pne)+cos(
    ppa)*cos(pne)*cos(lne-lpa) ))
print("les distances differentes car la Terre n'est pas parfaitement ronde mais un peu
    aplatie aux poles.")
```

Exercice 13 :

Utilisez la fonction range() pour afficher :

1. les entiers de 0 à 3;
2. les entiers de 4 à 7;
3. les entiers pairs de 2 à 8.

```
a=list(range(0,4))
b=list(range(4,8))
c=list(range(2,9,2))
print( a,"\t", b, "\t", c, "\t")
```

Exercice 14 :

Définir la liste [17, 38, 10, 25, 72], puis effectuez les actions suivantes :

1. triez et affichez la liste ;
2. ajoutez l'élément 12 à la liste et affichez la liste ;
3. renversez et affichez la liste ;
4. affichez l'indice de l'élément 17 ;
5. supprimez l'élément 38 et affichez la liste ;
6. affichez la sous-liste du 2^{ième} au 3^{ième} élément ;
7. affichez la sous-liste du début au 2^{ième} élément ;
8. affichez la sous-liste du 3^{ième} élément à la fin de la liste ;
9. affichez le dernier élément en utilisant un indicage négatif ;
10. affichez le plus grand et le plus petit élément de la liste.

```
a=[17, 38, 10, 25, 72]
a.sort()
print(a)
a.append(12)
print(a)
print(a.index(17))
a.remove(38)
print(a)
print(a[1:3])
print(a[:2])
print(a[2:])
print(a[-1])
print(max(a), min(a))
```

Exercice 15 :

Utilisez une liste en compréhension pour ajouter 3 à chaque élément d'une liste d'entiers de 0 à 5. Même question pour multiplier chaque élément par 10.

```
b=[x+3 for x in range(0,6)]
c=[10*x for x in range(0,6)]
```

Exercice 16 :

Utilisez une liste en compréhension pour ajouter 3 à chaque élément d'une liste d'entiers de 0 à 5, mais seulement si l'élément est supérieur ou égal à 2.

```
b=[x+3 for x in range(0,6) if x >= 2]
```

Exercice 17 :

Créer une liste des entiers compris entre 0 et 9. Pour accéder aux fonctions factorielle(x) et exp(x) du module math de python on peut utiliser la commande :

```
|| from math import factorial,exp
```

quelle est la différence avec les commandes :

```
|| from math import *
```

```
|| import math
```

Construire 3 nouvelles listes associées aux valeurs de $x!$, x^2 et $\exp(x)$. Afficher les résultats, on pourra utiliser la fonction `round` pour améliorer la présentation. Pour des grands nombres quelle est la fonction la plus croissante?

```
import math
a=list(range(0,10))

L1=[math.factorial(x) for x in a]
L2=[x**2 for x in a]
L3=[math.exp(x) for x in a]

print(L1, L2, L3)
```