Chapitre 1 : Prise en main du langage Python

I Quelques opérations simples

I.1 Les types numériques int et float

Q1. Dans un code écrit en Python, comment apporte-t-on un commentaire ?

Q2. À l'aide de Python, faire le travail suivant et commentez :

(a) Calculer la valeur de : $\mathbf{2}$, $\mathbf{0}^{13\times21}$ et de $\mathbf{2}^{13\times21}$.

(b) L'instruction `2.4**5` renvoie-t-elle le même résultat que `2.4*2.4*2.4*2.4*2.4` ?

(c) Déterminer si le nombre 271+515 est divisible par 3 ;

(d) Écrire le nombre 6 210 sous la forme $b.q + r$ où $b = 12$ et $0 \leq r < b$;

(e) Afficher le texte suivant : Bonjour, je m'appelle Jean.

Q3. Afficher la période T d'un pendule de longueur l= 50 cm. Puis, arrondir le résultat à 2 chiffres après la virgule (pour cela utiliser la fonction round(x,n) où x est le nombre à arrondir et n le nombre de chiffres retenus après le virgule)).

On rappelle que $T = 2.\pi.\left(\frac{l}{g}\right)^{0.5}$ où g = 9.81 m.s^{-2} l'accélération de la pesanteur.

Q4. Exprimer la durée τ qu'il faut à la lumière émise par le Soleil pour nous parvenir. On rappelle la distance Terre-Soleil : $D = \mathbf{150 * 10^6\,km}$. On exprimera le résultat sous la forme d'un nombre entier (valeur plancher) et on affichera la phrase suivante : « La lumière émise par le Soleil met environ ... minutes à nous parvenir »

Q5. Quelles erreurs produisent ces lignes de code. Expliquer.

```
1 13%0
2 float('bonjour')
```

I.2 L'affectation : identificateur = valeur

Q6. En vous appuyant sur les exemples suivants, et en réalisant les tests nécessaires à l'aide de la fonction type(), commentez la phrase « Le type d'une constante ou d'une variable définit les opérations qu'elle supporte ».

```
1 a = 23.6 + 87
2 b = 'bonjour ' + 'le monde'
3 print('*o' * 10 + ' Bienvenue ' + '*o' * 10)
4  c="bonjour" * 3.0
```

Q7. Déterminer les valeurs de x et y au cours des instructions suivantes et repérer les affectations simultanée, multiple et la syntaxe de la permutation (propre au langage Python).

```
# suite d' instructions 1
x = y=3
x = x+2*y
y = x-y
```

```
# suite d' instructions 2
x=3
x, y = x + 2, x*2
x *= 2
```

```
# suite d' instructions 3
x=3
y=5
x, y = y, x
```

Q8. Quelles erreurs produisent ces lignes de code. Expliquer.

```
largeur = 5. ; longueur = 10 ; hauteur = 2
aire = largeur * longeur
volume = Aire *  hauteur
```

I.3 Manipuler les fonctions

Pour toute fonction, il faudra préciser **sa signature**. La signature est l'ensemble des paramètres (avec leurs noms, positions, valeurs par défaut et annotations), ainsi que l'annotation de retour d'une fonction. C'est-à-dire toutes les informations décrites à droite du nom de fonction lors d'une définition.

Q9. Écrire une fonction **superficie** qui a comme argument le rayon r d'un disque et retourne l'aire du disque. En employant superficie, écrire une nouvelle fonction **cylindre** qui, à partir du rayon de base r et de la hauteur h retourne l'aire de la base et le volume du cylindre. Commentez *le type retourné* par chaque fonction. On prendra π **= 3,1415**.

Q10. Calculer l'aire d'un disque de rayon 2,22 cm. Calculer l'aire et le volume d'un cylindre r **= 1,5 cm** et h **= 10 cm**. Stockez dans les variables aire et volume les grandeurs calculées et retournées par superficie.

Q11. Soient les variables x et y. Écrire une fonction **norme** qui retourne le résultat de l'expression $(x^2 + y^2)^{1/k}$ où x, y et k peuvent être un nombre entier ou à virgule. Faites-en sorte que par défaut, k prenne la valeur 2.

Q12. Trouver l'erreur !

```
print ("5 * 3 =", 5*3)
#calcul du nombre de secondes dans une journée
print ("une journée a une durée égale a", 60 * 60 * 24, "secondes"
print("bonjour le monde")
```

Q13. Trouver l'erreur !

```
def carre(x) :
      '''
      calcule et affiche le carre de x
      '''
      print(x**2)
a = carre(2.6+45*3) * 5
```

I.4 Manipuler les tests et le type booléen

Q14. Taper les lignes de codes suivantes et à l'aide de tests expliquer ce que fait chaque série d'instructions.

```
# suite 1
aff1 = a =  3
aff2 = a == 3
```

```
# suite 2
a = 1234 ;  x=0
 x !=0 and a/x>1
```

```
# suite 3
'chou'  > 'fleur'
 'o' in  'hello'
```

```
# suite 4
a = 2+3.001
a == 5.001
```

```
# suite  5
e = bool(20%4)
True * 3
```

Q15. Écrire une fonction **triangle**(a,b,c) où a,b,c sont des longueurs codées sous forme d'entiers. À l'aide de ces trois longueurs, déterminer s'il est possible de construire un triangle. Déterminer ensuite si ce triangle est rectangle, isocèle, équilatéral ou quelconque. Attention : un triangle rectangle peut être isocèle.

Q16. Écrire une fonction **appreciation** qui convertit une note scolaire N entrée par l'utilisateur sous forme de points (par exemple 27 sur 85 , N vaut alors 32%) passée en argument, en une note standardisée suivant le code ci-après :

La note	$N \geq 80$	$60 \leq N < 80$	$50 \leq N < 60$	$40 \leq N < 50$	$N < 40$
Appreciation	*A*	*B*	*C*	*D*	*E*

Appreciation(27, 85) renvoie E

Taper les lignes de code suivantes :

```
A = 51
If a> 0 :
        print('positif')
elif a<0 :
        print('négatif')
else :
        print('nul')
```

II Les boucles

Un type très utilisé que nous verrons ultérieurement en détail est le type liste :

Taper les lignes de code suivantes :

```
L= ['a', 1, 5 , 'bc'] # création d'une liste de 4 termes
 print(len(L))
 for elt in L :
        print(L)
L2 =[] # création d'une liste vide
for k in range(len(L)-1,-1,-1) : # parcourt dans le sens décroissant des indices
       print (k, L[k])
       L2.append(L[k]) # rajout dans L2 de l'élément L[k]
 print(L2)
```

Q17. Écrivez un programme qui parcourt un par un tous les éléments d'une liste de mots (par exemple : ['chat', 'chien', 'pomme', 'poire', 'bonjour', 'bonsoir']) pour générer deux nouvelles listes. L'une contiendra les mots comportant un maximum de 5 caractères, l'autre les mots comportant au moins 6 caractères.

Q18. À l'aide de la fonction range :

(a) énumérer les entiers de 2 à 60 par pas de 3 ;

(b) énumérer les entiers de 0 à 5 (exclu) par pas de 1 ;

(c) énumérer les entiers de 10 à 0 (inclus) par pas de -1 ;

(d) énumérer les entiers de 5 à 25 par pas de 5 (les deux bornes incluses) ;

Q19. Construire les listes suivantes par compréhension de liste :
Exemple : taper les lignes

```
L3 = [x**3 for x in range(7) if x%3==0]
print(L3)
L4 = [(a, b) for a in range(3) for b in range(4)]
print(L4)
```

(a) Liste des carrés des entiers pairs compris entre 0 et 9.

(b) Liste des multiples de 12 compris entre 0 et 100.

(c) Liste des diviseurs d'un entier naturel *N (prendre N=20 pour tester)* .

(d) Donner la liste des couples **(***a, b***)** tel que $a + b = 10$. a et b sont chacun compris entre 0 et9.

(e) Résoudre l'équation $x, y, z \in \mathbb{N}, 0 < x, y, z \leq 100, x^2 + y^2 = z^2$

Q20. Pour cette question, on considère que la liste étudiée est une liste de nombres (entiers et/ou à virgule).

(a) Écrire une fonction **somme**(liste) qui calcule et retourne la somme de tous les termes contenus dans la liste de nombre passée en paramètre.

(b) Écrire une fonction **moyenne**(liste) qui calcule et retourne la moyenne des nombres contenus dans la liste.

Q21. Soit une liste L, définie par la commande suivante : `>>> L = [10.4, 20.21, 30.0, 6, 2, 4, 201.3, 0.98, 0]`
Testez les différentes opérations suivantes. Regardez à chaque fois le contenu de L.

Solution:

```
>>> L = [10.4, 20.21, 30.0, 6, 2, 4, 201.3, 0.98, 0]
>>> len(L)
9
>>> L.append(30.0) ; print(L) # on ajoute 30.0 à la fin de L
[10.4, 20.21, 30.0, 6, 2, 4, 201.3, 0.98, 0, 30.0]
>>> L[0] # accès à l'élément placé en position 0 dans L
10.4
>>> L[ :5] # vue des 5 derniers éléments dans L
[10.4, 20.21, 30.0, 6, 2]
>>> L[ 5 :] # vue des 5 premiers éléments dans L
[4, 201.3, 0.98, 0, 30.0]
>>> L[ 2 : 7 ] # vue des éléments d'indice 2 (inclus) à 6 (7 exclu)
[30.0, 6, 2, 4, 201.3]
>>> L[: : 2] # vue de tous les éléments de L par pas de 2
[10.4, 30.0, 2, 201.3, 0]
>>> L[3] = 1 # modification de l'élément positionné en troisième place
>>> print(L) # L a été modifiée
[10.4, 20.21, 30.0, 1, 2, 4, 201.3, 0.98, 0, 30.0]
```

Q22. Trouvez l'erreur !

```
1 >>> jour = ['lundi', 'mardi', 'mercredi', 1800, 20.357, 'jeudi', 'vendredi']
2 >>>print(jour[7])
```