

## Devoir bilan

### Bases de la programmation Python

Durée : 1 heure

Les calculatrices sont autorisées.

*Consignes.*

- Attendre la fin de la distribution avant de consulter le sujet.
- Écrire ses nom et prénom sur chaque feuille.
- Répondre aux questions directement sur l'énoncé.

**Exercice 1.** Types et valeurs de base (6 points)

Le langage Python offre quatre types de valeur de base rappelés dans le tableau ci-dessous.

Type	Permet de représenter en Python ...	Exemples de valeur
int	... les nombres entiers positifs et négatifs.	-1, 3, 321
float	... les nombres décimaux.	1.5, -0.1, 2.0
str	... du texte (chaînes de caractères).	'Bonjour !', "vert"
bool	... les booléens (Vrai et Faux).	True, False

Pour chaque expressions ci-dessous, préciser sa valeur (le résultat de l'évaluation par Python) et son type.

N°	Expression Python	Valeur	Type
1	3 + 5		
2	2 * 2.5		
3	20 / 4		
4	2 / 4		
5	2 ** 4		
6	'Bon' + 'jour'		
7	'hello' + 'hello' + 'hello'		
8	3 * 'hello'		
9	3 * 5		
10	3 * '5'		
11	1 == 2		
12	3 <= 5		

**Exercice 2.** Affectations de variable (4 points)

On considère le programme ci-dessous.

```

1  duree = 3
2  distance = 75
3  vitesse_moy = distance / duree
4  message = "Vitesse moyenne en km/h :"
5  print(message)
6  print(vitesse_moy)

```

Compléter le tableau ci-dessous listant les informations des variables du programme précédent après son exécution.

Nom de variable	Valeur en fin d'exécution du programme	Type

**Exercice 3.** Boucles **for** (5 points)

1. On considère le programme Python ci-dessous.

```
1 for _ in range(4):  
2     print("Bonjour,")  
3 print("comment ça va ?")
```

Compléter ci-dessous l’affichage obtenu dans le terminal après exécution du programme précédent. Attention, toutes les lignes ci-dessous ne sont pas forcément utiles.

```
Bonjour  
.....  
.....  
.....  
.....  
.....
```

2. Voici l’affichage obtenu dans le terminal après l’exécution d’un programme.

```
1  
et puis  
2  
et puis  
3  
et puis  
4  
et puis  
terminé
```

Compléter le programme ci-dessous afin que son exécution produise l’affichage précédent. *Indication.* Attention à être précis sur les indentations !

```
1 for nbre in range(.....):  
2     print(.....)  
3 .....  
4 .....
```

*Suite page suivante*

**Exercice 4.** Boucles `while` et branchement conditionnel (5 points)

1. On rappelle que trois cas différents peuvent se présenter lorsqu'un programme rencontre une boucle `while` :

- *Cas 1.* La boucle `while` se termine.
- *Cas 2.* La boucle `while` ne se termine pas.
- *Cas 3.* La boucle `while` n'est pas exécutée.

Pour chaque programme ci-dessous indiquer dans quel cas on se trouve et donner l'affichage éventuel obtenu. Si aucun affichage n'est obtenu, marquer la case *Affichage* d'une croix.

<p>(a) <div style="border: 1px solid black; padding: 5px; display: inline-block; width: 150px;"> <pre>n = 5 while n &lt; 10:     n = n + 1 print(n)</pre> </div></p>	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr><td style="text-align: center;">Cas</td></tr> <tr><td style="height: 15px;"> </td></tr> <tr><td style="text-align: center;">Affichage</td></tr> <tr><td style="height: 15px;"> </td></tr> </table>	Cas		Affichage	
Cas					
Affichage					
<p>(b) <div style="border: 1px solid black; padding: 5px; display: inline-block; width: 150px;"> <pre>n = 10 while n &lt; 10:     n = n + 1 print(n)</pre> </div></p>	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr><td style="text-align: center;">Cas</td></tr> <tr><td style="height: 15px;"> </td></tr> <tr><td style="text-align: center;">Affichage</td></tr> <tr><td style="height: 15px;"> </td></tr> </table>	Cas		Affichage	
Cas					
Affichage					
<p>(c) <div style="border: 1px solid black; padding: 5px; display: inline-block; width: 150px;"> <pre>n = 5 while n &gt;= 0:     n = n - 1 print(n)</pre> </div></p>	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr><td style="text-align: center;">Cas</td></tr> <tr><td style="height: 15px;"> </td></tr> <tr><td style="text-align: center;">Affichage</td></tr> <tr><td style="height: 15px;"> </td></tr> </table>	Cas		Affichage	
Cas					
Affichage					
<p>(d) <div style="border: 1px solid black; padding: 5px; display: inline-block; width: 150px;"> <pre>n = 5 while n &gt;= 0:     n = n + 1 print(n)</pre> </div></p>	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr><td style="text-align: center;">Cas</td></tr> <tr><td style="height: 15px;"> </td></tr> <tr><td style="text-align: center;">Affichage</td></tr> <tr><td style="height: 15px;"> </td></tr> </table>	Cas		Affichage	
Cas					
Affichage					

2. On considère le programme suivant qui simule un jeu de dé.

```

1 from random import randint
2
3 face = randint(1, 6)
4 print("Face obtenue :", face)
5 if face < 6:
6     print("Perdu !")
7     if face == 5:
8         print("Pas loin !")
9 else:
10    print("Gagné !")

```

*Remarque.* L'appel `randint(1, 6)` renvoie un nombre entier aléatoire compris entre 1 et 6, les nombres 1 et 6 étant inclus.

Pour chaque cas ci-dessous, indiquer l'affichage obtenu. Attention, dans les cadres de réponse, toutes les lignes disponibles ne sont pas forcément nécessaires.

(a) *Cas 1.* La variable `face` prend la valeur 4 en ligne 3.

Face obtenue : ...

.....

.....

.....

(b) *Cas 2.* la variable `face` prend la valeur 5 en ligne 3.

Face obtenue : ...

.....

.....

.....