

Exercice 1

On utilise le système d'instructions suivant :

```
# Python
1. for i in .. .. :
2.     print(i)
```

1. L'affichage produit est alors:

```
1. T
2. G
3. V
```

Quel était l'itérable placé dans la boucle for à la ligne 1 du script? "TGV" ou bien ["T","G","V"]

2. L'affichage produit est maintenant:

```
1. "ab"
2. "abc"
3. "abcd"
```

Quel était l'itérable placé dans la boucle for à la ligne 1 du script? ["ab","abc","abcd"]

3. Le programme est donné:

```
# Python
1. for k in [2012 , 2019 , 2024]:
2.     if k%2 == 0:
3.         print(k)
```

Quel sera l'affichage?

```
1. 2012
2. 2024
```

Exercice 2

1. Transformer le script utilisant une boucle **for** en un nouveau script utilisant une boucle **while**

programme avec boucle for	programme avec boucle while
<pre># Python 1. for i in range(4) : 2. a = i * 2 3. print(a)</pre>	<pre># Python 1. i = 0 2. while i < 4: 3. a = i * 2 4. print(a) 5. i = i + 1</pre>

2. Montrer, par des arguments logiques, que pour le script de droite (boucle while), le programme va bien terminer (et ne pas boucler à l'infini):

Le variant de boucle, c'est i , qui démarre avec la valeur $i = 0$. La condition d'exécution, c'est $i < 4$, donc la boucle terminera lorsque i atteindra 4. Or, à chaque itération, i augmente de une unité. Donc, au bout de 4 itérations, la boucle termine.

Exercice 3

Le programme suivant va construire une chaîne de caractères à partir des rangs ascii des caractères, placés dans une liste L. A la fin, on affiche la suite de caractères. Compléter le script

```
L = [110,39,111,117,118]
chaîne = "" # à compléter
for rang in L:
    car = chr(rang) # à compléter
    chaîne = chaîne + car # à compléter
print(chaîne) # à compléter
```

Exercice 4

1. Écrire une fonction $f_1(x)$ qui calcule la fonction mathématique $f_1(x) = x^2 + 5$

```
def f(x):
    return x*2 + 5
```

2. Ecrire l'instruction qui calcule $f_1(x)$ pour $x = 6$

```
f(6)
```

3. Que vaut $f_1(10)$?

```
25
```

4. Écrire une fonction `fahrenheit_vers_celsius` qui convertit une température de Fahrenheit en Celsius

```
1. def fahrenheit_vers_celsius(F):
2.     return F * 5/9 - 160/9
3.
4.
5.
```

Aide: utiliser une des 2 formules suivantes: $F = C \times 9/5 + 32$ et $C = F * 5/9 - 160/9$

5. Ecrire l'appel de la fonction `fahrenheit_vers_celsius` avec la température 71°Fahrenheit.

```
6. fahrenheit_vers_celsius(71)
```

Exercice 5

Un site impose à ses clients des critères sur leur mot de passe. Pour cela, il utilise la fonction `valider` suivante (en python):

```
# Python
1. def valider(mot_de_passe):
2.     test = False
3.     for c in mot_de_passe:
4.         if c in '!#0,;:&$%$*':
5.             test = True
6.     return test
```

1. Compléter le tableau de suivi avec l'appel à la fonction `valider('best@')`

iteration n°	c	c in '!#@,;:&\$%\$*'	test à la fin de l'itération
1	"b"	False	False
2	"e"	False	False
	"s"	False	False
	"t"	False	False
5	"@"	True	True

2. Que retourne la fonction avec les mots de passe suivants: (donner le résultat sans justifier)

`valider('paptap23')` -> **False**

`valider('2!@59fgds')` -> **True**

Exercice 6

On donne ci-dessous la table des caractères ASCII et leurs points de code.

- Combien de caractères contient la table ascii? **128**
- Sachant que le caractère A est représenté en ASCII par la séquence binaire 100 0001 :
 - Donner le point de code, exprimé en binaire pour le caractère entouré dans le tableau:
..
 - Donner le point de code en numération hexadécimale: ..
 - puis en numération décimale: ..
- Il existe un encodage de type ascii étendu, sur 8 bits, permettant d'afficher aussi les caractères latins accentués. Pourquoi ce type d'encodage n'est-il plus utilisé, et pourquoi lui préfère-t-on l'UNICODE?
- L'utf-8, permet-il l'encodage de tous les caractères, y compris ceux en Grec?
- Présenter en quelques lignes ce qu'est l'UNICODE et l'utf-8.