

Correction « FOR »

Python - salah.cc

M. SALAH

1. Écrire une fonction colonies() (sans variable en argument) renvoyant le nombre de colonies en 2014

L'énoncé nous dit que l'apiculteur possède **300 colonies** en 2014.
On veut simplement créer une fonction qui **retourne** ce nombre.

Code :

```
def colonies():  
    nb_colonies = 300 # nombre de colonies en 2014  
    return nb_colonies
```

- **nb_colonies** stocke la valeur 300.
- La fonction renvoie cette valeur.

2. Tester le code puis appeler l'enseignant

exécuter la fonction et vérifier qu'elle renvoie bien 300.

Code :

```
print(colonies()) # ==> Doit afficher 300
```

3. Compléter l'affectation (pseudo-code) représentant l'évolution du nombre de colonies d'une année à l'autre

Rappel des informations :

1. On **perd 8%** des colonies durant l'hiver.

○ => Cela signifie qu'il nous reste **92%** des colonies d'origine
($100\% - 8\% = 92\%$).

○ Traduit en code : `nb_colonies = nb_colonies * 0.92`

2. On **ajoute 50** nouvelles colonies au printemps.

○ Traduit en code : `nb_colonies = nb_colonies + 50`

Pour passer d'une année à la suivante est donc :

```
nb_colonies ← nb_colonies * 0.92 + 50
```

4. Modifier la fonction `colonies()` pour prendre en compte plusieurs années

```
def colonies(nb_annees):
```

```
    nb_colonies = 300 (=> nombre de colonies en 2014 - initialisation)
```

```
    for i in range(nb_annees):
```

```
        nb_colonies = nb_colonies * 0.92 + 50
```

```
    return nb_colonies
```

5. Combien y a-t-il de colonies en 2015 ?

L'année **2015** correspond à **1 an** après 2014.

On calcule donc l'évolution **sur 1 an** :

1. Départ : 300 colonies (année 2014)

2. Perte de 8% pendant l'hiver :

$$300 \times 0.92 = 276$$

3. Ajout de 50 colonies au printemps :

$$276 + 50 = 326$$

$$276+50=326$$

Résultat : en 2015, il y a 326 colonies.

Pour le vérifier avec du code Python :

```
print(colonies(1)) => doit afficher 326
```

Part 2 - « FOR »

Python - salah.cc

M. SALAH

1. Exemples de boucles :

`range(10) → 0 à 9`

`range(5,7) → 5, 6`

`for k in [« maman", "papa", "tonton", "tata"]:` → on parcourt la liste donnée.

2. Fonction `colonies(nb_annees)` :

```
def colonies(nb_annees):  
    nb_colonies = 300  
    for i in range(nb_annees):  
        annee = 2014 + i  
        print(annee, ":", nb_colonies)  
        nb_colonies = nb_colonies * 0.92 + 50
```

3.

Nombre de colonies en 2050 :

exécuter colonies (37) pour voir la dernière ligne affichée (équivalente à l'année 2050).

4.

Le nombre de colonies tend vers 625 au fil des ans.