

## I. Happy apiculteur – D’heures en heures, l’apiculteur se meurt –

(Mo)

Dans ce TP nous allons partir d’un problème donné au baccalauréat ES 2015 (Pondichéry) pour découvrir les **boucles for**.

Énoncé :

Un apiculteur souhaite étendre son activité de production de miel à une nouvelle région. En juillet 2014, il achète 300 colonies d’abeilles qu’il installe dans cette région.

Après renseignements pris auprès des services spécialisés, il s’attend à perdre 8 % des colonies durant l’hiver. Pour maintenir son activité et la développer, il a prévu d’installer 50 nouvelles colonies chaque printemps.



On va étudier l’évolution du nombre de colonies à l’aide d’une fonction python `colonies(nb_annees)`, qui devra renvoyer une estimation du nombre de colonies d’abeilles durant l’année en considérée.

1. Commencez par écrire une fonction python `colonies()` (sans variable) renvoyant le nombre de colonies en 2014 après l’avoir stockée dans une variable que vous nommerez `nb_colonies`.
2. Testez votre code puis appelez-moi pour vérifier.
3. Complétez l’affectation suivante représentant l’évolution du nombre de colonies d’une année à l’autre :

```
nb_colonies ← nb_colonies ... ..
```

 Pseudo-Code

4. Modifiez la fonction `colonies()` en y insérant une adaptation python du pseudo-code précédent.
5. Combien y-a-t’il de colonies en 2015?

-----  
-----

## II. Avec une boucle for

(Mo)

Pour continuer de suivre l'évolution du nombre de colonies, il serait nécessaire d'effectuer le même calcul un certain nombre de fois.

Heureusement, il est possible d'exécuter un code en boucle avec le mot clef **for**.

1. Testez les codes python suivants et pour chacun d'eux indiquer le résultat obtenu ainsi qu'une explication du pourquoi.

a. 

```
1 | for k in range(10):
2 |     print(k)
```

-----  
 -----

b. 

```
1 | for k in range(5,7):
2 |     print(k)
```

-----  
 -----

c. 

```
1 | for k in ["maman", "papa", "tonton", "tata"]:
2 |     print("bonjour", k, "!")
```

-----  
 -----

2. En vous inspirant de vos réponses précédentes, reprenez la fonction `colonies(nb_annees)` afin qu'elle renvoie le nombre de colonies pour les **nb\_annees** premières années.

Le résultat attendu doit ressembler à :

```
>>> colonies(5)
2014 : 300
2015 : 326.0
2016 : 349.92
2017 : 371.9264
2018 : 392.17228800000004
```

3. Combien y aura-t-il de colonies en 2050? .....

4. Que remarque-t-on?

-----  
 -----

5. Appelez-moi pour vous féliciter.