PREMIÈRE / EXERCICES CORRIGÉS

Algorithmique et programmation en Python

Astrolab Cours · astrolabcours.fr

Ex. 1 Suivre un programme à la main

●○○ Application

Que contiennent `a` et `b` à la fin de ce programme ?

```
a = 3
b = 5
a = a + b
b = a - b
a = a - b
```

Indice

Fais un tableau avec une ligne par instruction.

Ligne par ligne

Après les deux premières lignes, a = 3 et b = 5. Puis a = 8. Puis b = 8 − 5 = 3. Puis a = 8 − 3 = 5.

← *une ligne à la fois*

Conclusion

a = 5 et b = 3 : le programme a échangé les contenus des deux variables.

Ex. 2 Une fonction avec une condition

●○○ Méthode

Écrire une fonction `mention(note)` qui renvoie « TB » si la note est au moins 16, « B » si elle est au moins 14, « AB » si elle est au moins 12, et « aucune » sinon.

Indice

Utilise `if`, `elif` et `else`, en commençant par la plus haute mention.

L'ordre des tests

On teste d'abord 16, sinon une note de 17 serait classée « AB » dès le premier test de 12.

© Astrolab Cours · astrolabcours.fr/maths/premiere/algorithmique/ · Tous droits réservés : usage personnel uniquement, republication et usage commercial interdits

Le programme

```
def mention(note):
    if note >= 16:
        return "TB"
    elif note >= 14:
        return "B"
    elif note >= 12:
        return "AB"
    else:
        return "aucune"
```

Ex. 3 Terme d'une suite ●●○ Méthode

Soit $u_0 = 5$ et $u_{n+1} = 0{,}5\,u_n + 3$. Écrire une fonction `terme(n)` qui renvoie u_n, puis calculer `terme(3)` à la main.

Indice

Une boucle `for` qui fait `n` tours.

Le programme

```
def terme(n):
    u = 5
    for k in range(n):
        u = 0.5 * u + 3
    return u
```

À la main

$u_1 = 5{,}5$, $u_2 = 5{,}75$, $u_3 = 5{,}875$.

← trois tours de boucle

Conclusion

terme(3) renvoie 5.875.

Ex. 4 Algorithme de seuil ●●○ Méthode

Un capital de 2 000 € est placé à 3 % par an. Écrire un programme qui affiche le nombre d'années nécessaires pour que le capital dépasse 3 000 €. Quel résultat affiche-t-il ?

Indice

Augmenter de 3 %, c'est multiplier par 1,03.

© Astrolab Cours · astrolabcours.fr/maths/premiere/algorithmique/ · Tous droits réservés : usage personnel uniquement, republication et usage commercial interdits

Le programme

```
c = 2000
n = 0
while c <= 3000:
    c = 1.03 * c
    n = n + 1
print(n)
```

Le résultat

$2\,000 \times 1{,}03^{13} \approx 2\,937$ et $2\,000 \times 1{,}03^{14} \approx 3\,025$.

Conclusion

Le programme affiche 14.

Ex. 5 Le piège du range

●●○ Piège

Un élève veut calculer $S = 1 + 2 + \cdots + 10$ et écrit :

```
S = 0
for k in range(10):
    S = S + k
print(S)
```

Qu'affiche son programme ? Le corriger.

Indice

Quelles valeurs prend `k` ?

L'erreur

k prend les valeurs de 0 à 9. Le programme calcule $0 + 1 + \cdots + 9 = 45$, il oublie 10.

La correction

Il faut `range(1, 11)`, qui va de 1 à 10 :

```
S = 0
for k in range(1, 11):
    S = S + k
print(S)
```

← on s'arrête avant 11

© Astrolab Cours · astrolabcours.fr/maths/premiere/algorithmique/ · Tous droits réservés : usage personnel uniquement, republication et usage commercial interdits

Conclusion

Le programme corrigé affiche 55.

Ex. 6 Liste et moyenne ●●● Type bac

Écrire une fonction `moyenne(L)` qui renvoie la moyenne des nombres d'une liste `L`, puis une fonction `au_dessus(L)` qui renvoie la liste des valeurs strictement supérieures à la moyenne. Tester avec `L = [12, 8, 15, 9, 16]`.

Indice

Parcours la liste avec `for x in L:`.

La moyenne

```
def moyenne(L):
    s = 0
    for x in L:
        s = s + x
    return s / len(L)
```

Les valeurs au-dessus

```
def au_dessus(L):
    m = moyenne(L)
    R = []
    for x in L:
        if x > m:
            R.append(x)
    return R
```

← on réutilise la première fonction

Le test

La moyenne vaut $\frac{60}{5} = 12$, donc au_dessus(L) renvoie [15, 16].

© Astrolab Cours · astrolabcours.fr/maths/premiere/algorithmique/ · Tous droits réservés : usage personnel uniquement, republication et usage commercial interdits