

Algorithmique et Python — Première STMG

20 exercices corrigés · mathlvl.fr

Énoncés et corrigés détaillés

Présentation

En gestion, Python remplace les calculs répétitifs : appliquer une remise à cent articles, suivre un capital mois après mois, compter les jours où les ventes dépassent un objectif ou dépouiller un fichier de commandes. Ces 20 exercices corrigés reprennent les bases de seconde (affectation, test, boucle, fonction), installent le compteur et l'accumulateur, puis la seule notion nouvelle de l'année, la liste. Ils finissent par le traitement de données (tableau croisé) et la simulation du hasard avec `random()`. Chaque correction suit les variables pas à pas, et toutes les sorties annoncées ont été vérifiées en exécutant les programmes.

Version en ligne, avec les corrections dépliantes :

mathlvl.fr/exercices-corriges/premiere-stmg/algorithmique-python-stmg

Exercice 1 - Suivre des affectations

On exécute les instructions suivantes :

```
prix = 120
prix = prix * 0.9
prix = prix * 1.2
```

1. Que vaut `prix` à la fin ?
2. Que représente ce calcul ?

Exercice 2 - Que renvoie cette fonction ?

On définit :

```
def ht(ttc):
    return ttc / 1.2
```

1. Quelle est l'entrée de la fonction ? sa sortie ?
2. Que renvoient `ht(150)` et `ht(60)` ? Interprète.

Exercice 3 - Un tarif selon l'âge

Un cinéma applique ce tarif :

```
def tarif(age):
    if age < 18:
        return 6
    else:
        return 9.5
```

1. Que renvoient `tarif(15)` et `tarif(18)` ?
2. Combien paie une famille de deux adultes et deux enfants de 10 et 14 ans ?

Exercice 4 - Lire une liste

Les ventes de quatre semaines sont rangées dans la liste $L = [45, 52, 38, 61]$.

1. Que valent `len(L)`, `L[1]` et `L[-1]` ?
2. Quelle instruction donne les ventes de la troisième semaine ?

Exercice 5 - Une boucle bornée

On exécute :

```
s = 0
for k in range(5):
    s = s + 2
```

1. Combien de fois la boucle tourne-t-elle ?
2. Que vaut `s` à la fin ?

Exercice 6 - Un accumulateur

On veut calculer le total et la moyenne des ventes $L = [45, 52, 38, 61]$ sans utiliser `sum` :

```
s = 0
for v in L:
    s = s + v
moy = s / len(L)
```

1. Que valent `s` et `moy` à la fin ?
2. Pourquoi initialise-t-on `s` à 0 ?

Exercice 7 - Un compteur

Les ventes de six jours sont $J = [45, 52, 38, 61, 57, 49]$. Écris un programme qui compte les jours où les ventes dépassent 50, puis donne le résultat.

Exercice 8 - Une liste par ajouts successifs

On exécute :

```
C = []
c = 2000
for n in range(4):
    C.append(round(c, 2))
    c = c * 1.02
```

1. Que contient la liste `C` à la fin ?
2. À quelle suite correspondent ses éléments ?

Exercice 9 - Une liste en compréhension

Un stock de 100 articles diminue de 15 articles par jour. Que contient la liste `[100 - 15*k for k in range(5)]` ?

Exercice 10 - Une fonction à deux entrées

Le taux de marque est la marge rapportée au prix de vente :

```
def marque(pv, pa):  
    return (pv - pa) / pv * 100
```

1. Que renvoie `marque(50, 35)` ? Interprète.
2. Pourquoi divise-t-on par `pv` et non par `pa` ?

Exercice 11 - Un seuil avec while

Un capital de 3000 € est placé à 4 % par an.

```
c = 3000  
n = 0  
while c <= 4000:  
    c = c * 1.04  
    n = n + 1
```

1. Que représente `n` à la fin ?
2. Que vaut `n` ? Vérifie par le calcul.

Exercice 12 - Filtrer une liste

Des montants de commandes sont rangés dans `M = [12, 35, 8, 50, 27]`. Que contient la liste `[x for x in M if x >= 20]` ? Quelle proportion des commandes représente-t-elle ?

Exercice 13 - Le meilleur mois

Les chiffres d'affaires de janvier à juin sont `V = [310, 285, 342, 298, 360, 331]`.

```
im = 0  
for i in range(len(V)):  
    if V[i] > V[im]:  
        im = i
```

1. Que contient `im` à la fin ?
2. Quel est le meilleur mois ?

Exercice 14 - Un tableau croisé

Dix salariés ont choisi une activité pour le comité d'entreprise ; chaque réponse est une liste `[genre, activité]` : F sport, H culture, F culture, F sport, H sport, H sport, F culture, H culture, F sport, H sport.

1. Écris l'instruction qui compte les femmes ayant choisi le sport, la liste s'appelant d.
2. Dresse le tableau croisé genre × activité. Le choix dépend-il du genre ?

Exercice 15 - Simuler une épreuve de Bernoulli

Un client accepte une offre avec la probabilité 0,25.

```
def accepte():
    return random() < 0.25
```

1. Pourquoi cette fonction simule-t-elle la décision d'un client ?
2. Sur 1000 clients simulés, un élève compte 257 acceptations. Commente.

Exercice 16 - Une boucle qui ne s'arrête pas

Une machine perd 10 % de sa valeur chaque année ; elle vaut 200 (milliers d'euros). Pour savoir quand elle vaudra 100 ou moins, un élève écrit :

```
u = 200
n = 0
while u > 100:
    u = u * 1.1
    n = n + 1
```

1. Pourquoi le programme ne s'arrête-t-il jamais ?
2. Corrige-le et donne le résultat.

Exercice 17 - Deux offres d'abonnement

Un abonnement A coûte 30 € le premier mois puis augmente de 2 € par mois ; un abonnement B coûte 25 € puis augmente de 5 % par mois.

```
A = 30
B = 25
m = 0
while B <= A:
    A = A + 2
    B = B * 1.05
    m = m + 1
```

1. Quelles suites modélisent les prix mensuels ?
2. Le programme affiche m = 23. Interprète.

Exercice 18 - Un programme modulaire

Une entreprise vend chaque article 120 € ; le coût de production de q articles est $2q^2 + 20q + 800$ euros.

```
def benefice(q):
    return 120*q - (2*q**2 + 20*q + 800)
```

```
best = 0
for q in range(51):
    if benefice(q) > benefice(best):
        best = q
```

1. Que contient best à la fin ?
2. Le programme trouve best = 25. Vérifie par le calcul.

Exercice 19 - Total par catégorie

Des commandes sont enregistrées sous la forme [catégorie, montant] : textile 45, chaussures 80, textile 30, accessoires 15, chaussures 65, textile 55.

1. Écris l'instruction qui calcule le total du textile, la liste s'appelant cmd.
2. Calcule le total de chaque catégorie et sa part du chiffre d'affaires.

Exercice 20 - Synthèse : estimer un risque par simulation

Trois machines tombent chacune en panne dans la journée avec la probabilité 0,1, indépendamment.

```
def journee():
    for k in range(3):
        if random() < 0.1:
            return 1
    return 0
```

1. Que renvoie journee() ?
2. Sur 10 000 journées simulées, la fréquence obtenue est 0,2683. Calcule la probabilité exacte et compare.

Corrigés détaillés

Les corrections des 20 exercices, dans l'ordre des énoncés.

Correction de l'exercice 1 - Suivre des affectations

Une affectation calcule d'abord le membre de droite avec la valeur actuelle, puis range le résultat dans la variable.

Après la deuxième ligne : $120 \times 0,9 = 108$.

Après la troisième ligne : $108 \times 1,2 = 129,6$. Python affiche 129.6.

Le programme applique une remise de 10 % puis une TVA de 20 % : le prix final est 129,60 €. Les coefficients se multiplient : $0,9 \times 1,2 = 1,08$, soit +8 % au total.

Réponse. prix = 129,6 : prix TTC après une remise de 10 %.

Rappel de cours. $x = x * c$ multiplie la valeur actuelle de x par c et écrase l'ancienne valeur.

Correction de l'exercice 2 - Que renvoie cette fonction ?

L'entrée est le paramètre `ttc`, un prix toutes taxes comprises ; la sortie est la valeur renvoyée par `return`.

$ht(150) : \frac{150}{1,2} = 125$; $ht(60) : \frac{60}{1,2} = 50$.

La fonction retrouve le prix hors taxes d'un prix TTC avec une TVA de 20 % : on divise par le coefficient 1,2, on ne retire pas 20 %.

Réponse. $ht(150) = 125$ et $ht(60) = 50$: prix hors taxes.

Erreur fréquente. Retirer 20 % du prix TTC : $150 \times 0,8 = 120$, ce qui est faux.

Correction de l'exercice 3 - Un tarif selon l'âge

`tarif(15)` : $15 < 18$ est vrai, la fonction renvoie 6.

`tarif(18)` : $18 < 18$ est faux, la fonction renvoie 9,5. Un spectateur de 18 ans paie le plein tarif.

La famille paie $2 \times 9,5 + 2 \times 6 = 19 + 12 = 31$ €.

Réponse. 6 et $9,5 \cdot 31$ € pour la famille.

Correction de l'exercice 4 - Lire une liste

La liste a 4 éléments, d'indices 0 à 3 : `len(L)` vaut 4.

`L[1]` est le deuxième élément, 52 ; `L[-1]` est le dernier, 61.

La troisième semaine a l'indice 2 : `L[2]`, qui vaut 38. L'indice est toujours le rang moins 1.

Réponse. 4 ; 52 ; 61 · `L[2]`.

Erreur fréquente. Écrire `L[3]` pour la troisième semaine : c'est la quatrième.

Correction de l'exercice 5 - Une boucle bornée

`range(5)` donne les valeurs 0, 1, 2, 3, 4 : la boucle tourne 5 fois.

À chaque tour, `s` augmente de 2 : elle vaut successivement 2, 4, 6, 8, 10.

À la fin, $s = 5 \times 2 = 10$. La variable k n'est pas utilisée dans le calcul : elle sert seulement à compter les tours.

Réponse. 5 tours · $s = 10$.

Correction de l'exercice 6 - Un accumulateur

La boucle parcourt les valeurs et les ajoute une à une : s vaut 45, 97, 135, puis 196.

$\text{moy} = \frac{196}{4} = 49$: en moyenne 49 ventes par semaine.

s est un accumulateur de somme : on part de 0, qui ne change pas une somme. Pour un produit, on partirait de 1.

Réponse. $s = 196$ et $\text{moy} = 49$.

Rappel de cours. Accumulateur : $s = 0$ avant la boucle, $s = s + \dots$ dans la boucle.

Correction de l'exercice 7 - Un compteur

On initialise un compteur à 0 et on l'augmente de 1 chaque fois que la condition est vraie :

```
n = 0
for v in J:
    if v > 50:
        n = n + 1
```

Les valeurs 52, 61 et 57 dépassent 50 : n vaut 3 à la fin.

L'objectif de 50 ventes est dépassé 3 jours sur 6, soit la moitié des jours.

Réponse. 3 jours.

Correction de l'exercice 8 - Une liste par ajouts successifs

À chaque tour, on ajoute à la fin de la liste la valeur actuelle de c , puis on la multiplie par 1,02.

C vaut [2000, 2040.0, 2080.8, 2122.42].

Ce sont les quatre premiers termes de la suite géométrique $c_n = 2000 \times 1,02^n$: un capital de 2000 € placé à 2 % par an. Par exemple $2000 \times 1,02^3 \approx 2122,42$.

Réponse. [2000, 2040.0, 2080.8, 2122.42] : capital à 2 % par an.

Correction de l'exercice 9 - Une liste en compréhension

La compréhension calcule $100 - 15k$ pour $k = 0, 1, 2, 3, 4$.

On obtient [100, 85, 70, 55, 40] : le stock au début de chacun des cinq premiers jours.

Ce sont les termes de la suite arithmétique de premier terme 100 et de raison -15 : la compréhension traduit directement la formule $u_k = 100 - 15k$.

Réponse. [100, 85, 70, 55, 40].

Correction de l'exercice 10 - Une fonction à deux entrées

$\text{marque}(50, 35) : \frac{50 - 35}{50} \times 100 = \frac{15}{50} \times 100 = 30$.

Un article acheté 35 € et vendu 50 € a un taux de marque de 30 % : la marge représente

30 % du prix de vente.

Le taux de marque rapporte la marge au prix de vente ; rapportée au prix d'achat, elle donnerait le taux de marge, $\frac{15}{35} \times 100 \approx 42,9\%$. L'ordre des paramètres compte : pv d'abord, pa ensuite.

Réponse. 30 : taux de marque de 30 %.

Correction de l'exercice 11 - Un seuil avec while

La boucle multiplie le capital par 1,04 tant qu'il ne dépasse pas 4000 €, en comptant les années : à la fin, n est le nombre d'années nécessaires pour dépasser 4000 €.

Le programme s'arrête avec n = 8.

Vérification : $3000 \times 1,04^7 \approx 3947,80$ est encore inférieur à 4000, et $3000 \times 1,04^8 \approx 4105,71$ le dépasse.

Réponse. n = 8 : le capital dépasse 4000 € au bout de 8 ans.

Correction de l'exercice 12 - Filtrer une liste

La compréhension garde les éléments qui vérifient la condition $x \geq 20$: 35, 50 et 27.

La liste vaut [35, 50, 27] : ce sont les commandes d'au moins 20 €.

Elles représentent $\frac{3}{5} = 0,6$, soit 60 % des commandes. En Python :

`len([x for x in M if x >= 20]) / len(M)`.

Réponse. [35, 50, 27] · 60 % des commandes.

Correction de l'exercice 13 - Le meilleur mois

im garde l'indice du plus grand élément rencontré : il vaut 0, puis 2 (car 342 > 310), puis 4 (car 360 > 342).

À la fin, im = 4 et V[4] = 360.

L'indice 4 correspond au cinquième mois : le meilleur mois est mai, avec 360.

Réponse. im = 4 : mai (360).

Correction de l'exercice 14 - Un tableau croisé

On compte les réponses qui vérifient les deux critères :

`len([r for r in d if r[0] == "F" and r[1] == "sport"])`

qui vaut 3.

En répétant pour chaque couple : femmes, 3 sport et 2 culture ; hommes, 3 sport et 2 culture.

Dans les deux groupes, la fréquence du sport vaut $\frac{3}{5} = 0,6$, égale à la fréquence globale $\frac{6}{10}$: dans cet échantillon, le choix ne dépend pas du genre.

Réponse. 3 · F : 3 et 2 ; H : 3 et 2 ; pas de lien avec le genre.

Correction de l'exercice 15 - Simuler une épreuve de Bernoulli

`random()` renvoie un nombre au hasard dans $[0; 1[$. La condition `random() < 0.25` est vraie quand ce nombre tombe dans $[0; 0,25[$, ce qui arrive avec la probabilité 0,25 : la fonction renvoie True (acceptation) avec la probabilité voulue.

La fréquence observée est $\frac{257}{1000} = 0,257$, proche de 0,25.

L'écart de 0,007 est dû à la fluctuation d'échantillonnage : sur un grand nombre de clients simulés, la fréquence se rapproche de la probabilité.

Réponse. Condition vraie avec la probabilité $0,25 \cdot 0,257$, proche de 0,25.

Correction de l'exercice 16 - Une boucle qui ne s'arrête pas

L'élève multiplie par 1,1 : la valeur augmente (220, 242...) et reste toujours supérieure à 100. La condition du while n'est jamais fausse : c'est une boucle infinie.

Perdre 10 %, c'est multiplier par 0,9. On remplace la quatrième ligne par $u = u * 0.9$.

Le programme s'arrête avec $n = 7$: $200 \times 0,9^6 \approx 106,29$ et $200 \times 0,9^7 \approx 95,66$. La machine vaut 100 milliers d'euros ou moins au bout de 7 ans.

Réponse. La valeur augmente au lieu de baisser · avec $u = u * 0.9$, $n = 7$.

Rappel de cours. Dans une boucle while, le corps doit faire évoluer la variable testée vers la sortie.

Correction de l'exercice 17 - Deux offres d'abonnement

A est une suite arithmétique : $a_m = 30 + 2m$. B est une suite géométrique : $b_m = 25 \times 1,05^m$.

La boucle avance d'un mois tant que B reste moins cher ou au même prix que A. Elle s'arrête au premier mois où B devient plus cher.

Au bout de 23 mois, A coûte $30 + 2 \times 23 = 76$ € et B environ $25 \times 1,05^{23} \approx 76,79$ € : B est devenu plus cher. Pour un engagement de moins de 23 mois, B est plus avantageux chaque mois ; ensuite, sa hausse en pourcentage le rend plus cher.

Réponse. A arithmétique, B géométrique · B devient plus cher que A à partir du 23^e mois.

Correction de l'exercice 18 - Un programme modulaire

La boucle teste les quantités entières de 0 à 50 et garde dans best celle qui donne le plus grand bénéfice : c'est un balayage.

Le bénéfice s'écrit $B(q) = -2q^2 + 100q - 800 = -2(q - 10)(q - 40)$.

Ses racines sont 10 et 40 ; le sommet est au milieu, en $q = 25$, et $B(25) = -2 \times 15 \times (-15) = 450$ €. Le programme et le calcul concordent : 25 articles, 450 € de bénéfice.

Réponse. best = 25 : bénéfice maximal de 450 €.

Correction de l'exercice 19 - Total par catégorie

On cumule les montants des commandes de la catégorie :

```
sum(c[1] for c in cmd if c[0] == "textile")
```

qui vaut $45 + 30 + 55 = 130$.

De même : chaussures $80 + 65 = 145$, accessoires 15. Le total général vaut $130 + 145 + 15 = 290$ €.

Parts : textile $\frac{130}{290} \approx 44,8$ %, chaussures $\frac{145}{290} = 50$ %, accessoires $\frac{15}{290} \approx 5,2$ %. Les chaussures font la moitié du chiffre d'affaires avec seulement deux commandes sur six.

Réponse. Textile 130 € (44,8 %), chaussures 145 € (50 %), accessoires 15 € (5,2 %).

Correction de l'exercice 20 - Synthèse : estimer un risque par simulation

La fonction simule les trois machines une à une : dès qu'une panne survient, elle renvoie 1 ; si aucune ne tombe en panne, elle renvoie 0. Elle simule l'événement « au moins une panne dans la journée ».

Par l'événement contraire, « aucune panne » : $P = 1 - 0,9^3 = 1 - 0,729 = 0,271$.

La fréquence simulée, 0,2683, est très proche de 0,271 : l'écart de 0,0027 s'explique par la fluctuation d'échantillonnage. Il y a au moins une panne un peu plus d'une journée sur quatre.

Réponse. 1 s'il y a au moins une panne · probabilité exacte 0,271, proche de 0,2683.