



PYTHON

DEVOIR MAISON (DM PYTHON 2)

Notes et Commentaires

Notes

Nom	Note
Alexandre	13.75
Aréna	16.00
Lilou	19.50
Lucas	19.50
Lucie	16.00
Lyvahne	5.25
Marie	20.00
Mirana	18.25
Raphaël	20.00
Simon	15.75
Yann	20.00

Alexandre

18.5/27 => 13.75/20

Warm-Up (1 / 1)

Dict Usage (2.5 / 4)

Ajoutez la clef "avg_mean":

`element["avg_mean"] = {sum(element["marks"])/len(element["marks"])}` ajoute une clef "avg_mean" comme demandé mais la valeur est la moyenne sous forme de dictionnaire et non simplement la moyenne.

Étudiants qui ont une option et qui ont au moins un 20: `if element2 is 20` n'est pas correct, il aurait fallu tester `if element2 == 20`. De plus ton code affiche plusieurs fois le message pour les étudiants qui ont plusieurs 20 dans leur note => Cf. correction

Students Generator (7 / 8)

La convention préconise de faire les `import` dans les premières lignes

Exercice	Note	Commentaire
Email Generator	2 / 2	
Option Generator	1 / 2	Tu ne respectes pas le consignes: <code>create_option</code> prend deux arguments <code>options</code> et <code>durations</code> => Cf. correction
Marks Generator	2 / 2	Tu as déjà importé <code>random</code> , tu n'as pas besoin de l'importer à nouveau
		As-tu vraiment compris ce que tu as écrit? Nous n'avons pas vu les listes de compréhension en cours, est-ce que tu saurais en refaire une sans aide ?
Student Generator	2 / 2	

eCommerce Order Report (4 / 10)

Exercice	Note	Commentaire
Nombre d'articles commandés	1 / 2	Il fallait boucler sur <code>order["products"]</code> , par exemple: <code>for p in order["products"]</code> et récupérer les quantités via l'élément parcouru. Tu aurais pu utiliser <code>range</code> mais c'était plus complexe => Cf. correction
Prix total de la commande	0 / 2	Tu aurais pu écrire le pseudo-code pour montrer que tu avais l'algo...
Frais de port	2 / 2	
Récapitulatif	1 / 4	La partie <code>Products</code> n'est pas du tout faite, pourtant elle n'utilisait pas de fonction ce n'était que de l'affichage.

Mystery Function (4 / 4)

Aréna

22/27 => 16.50 ramené à 16.00/20

Devoir rendu en retard => -0.5 sur la note sur 20

Warm-Up (1 / 1)

Dict Usage (4 / 4)

Liste des emails: L'énoncé n'était pas clair, je souhaitais simplement `print` les adresses emails.

Affichez sa moyenne: `from statistics import mean` => un peu overkill mais ça fonctionne.

Option et qui ont au moins un 20: ce que tu écris fonctionne, mais plus simplement:

`if student["option"] and 20 in student["marks"]` était plus simple et plus optimisé

Students Generator (6 / 8)

Exercice	Note	Commentaire
Email Generator	2 / 2	
Option Generator	1 / 2	<code>tmp_name = random(options)</code> : tu utilises une fonction <code>random</code> sans jamais l'importer ou la créer. Cela crée une erreur dans ton code => Cf. correction
Marks Generator	1 / 2	Même commentaire que ci-dessus avec cette fois-ci une fonction <code>randint</code> , as-tu testé ton code avant de me l'envoyer avec du retard ?
Student Generator	2 / 2	Si les fonctions ci-dessus étaient correctes ton code aurait fonctionné

eCommerce Order Report (7 / 10)

Exercice	Note	Commentaire
Nombre d'articles commandés	2 / 2	
Prix total de la commande	0 / 2	Tu as oublié de multiplier les prix par les quantités
Frais de port	2 / 2	
Récapitulatif	3 / 4	Tu as oublié le statut de <code>delivered</code>

Mystery Function (4 / 4)

Lilou

26/27 => 19.50/20

Warm-Up (1 / 1)

Dict Usage (3 / 4)

Il te manque la question `Pour chaque élève, calculez puis affichez sa moyenne`.

Pour tester si un élément existe dans une liste, tu peux utiliser la méthode `in`, par exemple:

```
if 20 in [1, 10, 20]
```

Students Generator (8 / 8)

eCommerce Order Report (10 / 10)

Exercice	Note	Commentaire
Nombre d'articles commandés	2 / 2	Pourquoi ne pas avoir fait: <code>for i in order["products"]:</code> pour boucler sur les produits
Prix total de la commande	2 / 2	Même commentaire que sur l'exercice précédent
Frais de port	2 / 2	
Récapitulatif	4 / 4	

Mystery Function (4 / 4)

Lucas

26/27 => 19.50/20

Warm-Up (1 / 1)

Dict Usage (3 / 4)

Calcul / Affichage moyenne: `(note1 + note2 + note3)/3` => Ton code n'est pas générique: que se passe t'il si la liste de notes contient 1 000 000 de notes ? Il faut soit utiliser une seconde boucle for qui parcourt les notes, soit utiliser la fonction `sum` => Cf. correction `z["marks"].count(20) >= 1` : bonne idée je n'avais pas pensé à utiliser `.count()` , une des autres solutions était l'utilisation de `if 20 in z["marks"]`

Students Generator (8 / 8)

Dans tes retours de fonctions, pour simplifier la lecture tu peux directement faire:

```
def create_email_address(name):  
    return name + "@example.com"
```

Ce que tu as fait est bien entendu correct.

eCommerce Order Report (10 / 10)

Exercice	Note	Commentaire
Nombre d'articles commandés	2 / 2	
Prix total de la commande	2 / 2	
Frais de port	2 / 2	
Récapitulatif	4 / 4	

Mystery Function (4 / 4)

Lucie

21.5/27 => 16/20

Warm-Up (1 / 1)

Dict Usage (3 / 4)

Calcul / Affichage moyenne: `moy = (m["marks"][0]+m["marks"][1]+m["marks"][2])/3` =>
 Ton code n'est pas générique: que se passe t'il si la liste de notes contient 1 000 000 de notes ? Il faut soit utiliser une seconde boucle for qui parcourt les notes, soit utiliser la fonction `sum` => Cf. correction

Students Generator (8 / 8)

eCommerce Order Report (9.5 / 10)

Attention à ton indentation, tes commentaires ne sont pas au même niveau que ton code ce qui complexifie la lecture

Exercice	Note	Commentaire
Nombre d'articles commandés	2 / 2	Ta méthode d'ajouter les quantités dans une liste puis de sommer la liste fonctionne d'un point de vue métier, mais présente des problèmes d'algorithmie s'il existe une commande avec plusieurs millions de quantités => Cf. correction
Prix total de la commande	2 / 2	
Frais de port	2 / 2	
Récapitulatif	3.5 / 4	Tu modifies le dict en remplaçant <code>paid: True</code> par <code>paid: "Paid"</code> , il ne faut pas modifier les informations passées en entrée de ta fonction => Cf. correction
		Pour te simplifier la vie tu aurais pu faire plusieurs <code>print</code>

Mystery Function (0 / 4)

Simon et toi avez **exactement** les mêmes commentaires.

Lyvahne

7/27 => 5.25

Je pense qu'il serait nécessaire de faire un point pour parcourir la correction.

Warm-Up (1 / 1)

Dict Usage (0 / 4)

As-tu testé ton code avant de me l'envoyer ?

Calcul / Affichage moyenne: `for k in students[2]` => Cela sélectionne le 2ème élément de `students`, soit le deuxième dict

`print(key, ":", sum(k[key])/len(k[key]))` tu utilises une variable `key` qui n'est pas initialisée. Cela crée donc une erreur sur ton code => Cf. correction

`print students` : tu n'as pas mis les parenthèses ce qui crée une erreur sur ton code

```
for i in students:
    if duration in option:
        print(i["name", "option"])
```

Tu utilises des variables `duration` et `option` sans les avoir initialisées, cela crée donc une erreur sur ton code => Cf. correction

L'accès aux éléments d'un dictionnaire ne se fait pas via `i["name", "option"]` mais `i["name"]` et `i["option"]` comme nous ne l'avons dans la plupart des exercices.

N'hésite pas à poser des questions si certains points ne sont pas clairs.

Students Generator (2 / 8)

1ère ligne: `from random import choice, randint`

6ème ligne: `from random import randint`

Tu importes deux fois la méthode `randint` du package `random`

Exercice	Note	Commentaire
Email Generator	2 / 2	Tu ne respectes pas le consigne, la fonction ne prend qu'un argument qui est <code>name</code>
Option Generator	0 / 2	<code>for value in option.values(x):</code> , tu n'as pas besoin de boucle <code>for</code> => Cf. correction
Marks Generator	0 / 2	Tu initialises une variable <code>create_marks</code> ligne 78 et tu crées une fonction nommée <code>create_marks</code> ligne 79, cela crée une interférence entre la fonction et la méthode qui crée le bug lors à la ligne 91
		Il aurait fallu initialiser une liste de notes dans la fonction <code>create_marks</code> et retourner cette liste de notes
Student Generator	0 / 2	Cf. correction

eCommerce Order Report (0 / 10)

Tu aurais pu faire le pseudo-code pour aller chercher quelques points

Mystery Function (4 / 4)

Marie

27/27 => 20/20

Warm-Up (1 / 1)

Dict Usage (4 / 4)

Calcul / Affichage moyenne: `moyenne = somme / 3` => Une façon de rendre ton code plus générique serait `moyenne = somme / len(element["marks"])` de façon à ce que tu n'es pas à reprendre ton code si jamais le nombre de notes change

Students Generator (8 / 8)

Tu importes le package `time` sans l'utiliser => Stockage mémoire inutile

eCommerce Order Report (10 / 10)

Exercice	Note	Commentaire
Nombre d'articles commandés	2 / 2	
Prix total de la commande	2 / 2	Tu utilises <code>,</code> et <code>+</code> pour afficher les string. Même si les deux opérateurs font la même chose, essaye de ne t'en tenir qu'à un seul.
Frais de port	2 / 2	Même commentaire que sur l'exercice précédent
Récapitulatif	4 / 4	Même commentaire que sur l'exercice précédent

Mystery Function (4 / 4)

Mirana

24.5/27 => 18.25/20

Warm-Up (1 / 1)

Dict Usage (4 / 4)

Students Generator (7 / 8)

La convention préconise de faire les `import` dans les premières lignes

Exercice	Note	Commentaire
Email	2 / 2	

Exercice	Note	Commentaire
Generator		
Option Generator	1 / 2	On ne sait pas à quoi correspondent tes arguments <code>lst</code> et <code>lst2</code> , par exemple les nommer <code>option_names</code> et <code>option_durations</code> a plus de sens
		Retour attendu: <code>{"name": "python", "duration": 9}</code> . Retour de ta fonction: <code>{"python": 9}</code>
Marks Generator	2 / 2	Attention à la façon dont tu nommes tes variables: tu nommes ta liste <code>list</code> , mais <code>list</code> est un mot clef Python, comme <code>dict</code> , <code>str</code> , <code>int</code> ,
Student Generator	2 / 2	

eCommerce Order Report (8.5 / 10)

Exercice	Note	Commentaire
Nombre d'articles commandés	2 / 2	Attention à la façon dont tu nommes tes variables: tu nommes ta dict <code>dict</code> , mais <code>dict</code> est un mot clef Python, comme <code>list</code> , <code>str</code> , <code>int</code> ,
		La description de ton argument <code>dict</code> n'est pas correcte, par exemple <code>Order from which to extract products quantity</code> précise pourquoi il est nécessaire de passer l'argument
		Ta méthode d'ajouter les quantités dans une liste puis de sommer la liste fonctionne d'un point de vue métier, mais présente des problèmes d'algorithmie s'il existe une commande avec plusieurs millions de quantités => Cf. correction
Prix total de la commande	2 / 2	Mêmes commentaires que sur l'exercice précédent concernant le nom de tes variables, la doctstring et la méthode de calcul
Frais de	1 / 2	Tu ne respectes pas la consigne: l'argument passé est un <code>int</code>

Exercice	Note	Commentaire
port		et non un <code>dict</code>
		Mêmes commentaires que sur l'exercice précédent concernant le nom de tes variables et la doctstring
Récapitulatif	3.5 / 4	Tu modifies le dict en remplaçant <code>paid: True</code> par <code>paid: "Paid"</code> , il ne faut pas modifier les informations passées en entrée de ta fonction => Cf. correction
		Pour te simplifier la vie tu aurais pu faire plusieurs <code>print</code>
		<code>"None"</code> à la fin de l'affichage => <code>print(order_summary(order))</code> affiche le retour de la fonction, si tu fais <code>order_summary(order)</code> ce qui est la même chose mais sans le <code>print</code> tu n'auras plus le <code>None</code> à la fin

Mystery Function (4 / 4)

Raphaël

27/27 => 20/20

Warm-Up (1 / 1)

Dict Usage (4 / 4)

`def has_twenty(marks):` Bonne idée la création de fonction que tu utilises pour les deux exercices. C'est exactement à ça que servent les fonctions donc parfait. En revanche tu t'es un peu complexifié la tâche puisque tu pouvais plus simplement faire

`if 20 in student["marks"]` . Ta solution reste toutefois correcte.

`if has_twenty(student["marks"]) == True:` , tu peux directement écrire

`if has_twenty(student["marks"]):` mais ta solution reste correcte

Students Generator (8 / 8)

La convention préconise de faire les `import` dans les premières lignes

eCommerce Order Report (10 / 10)

Exercice	Note	Commentaire
Nombre d'articles commandés	2 / 2	<code>order ([dico]):</code> le type est <code>dict</code> , et non <code>dico</code>
Prix total de la commande	2 / 2	
Frais de port	2 / 2	
Récapitulatif	4 / 4	Variable <code>product_kind</code> initialisée ligne 133 mais jamais utilisée

Mystery Function (4 / 4)

Simon

21/27 => 15.75/20

Warm-Up (1 / 1)

Dict Usage (3 / 4)

Il est préférable d'utiliser `if x is not None` plutôt que `if x != None`

Tu as une erreur sur la dernière ligne: `NameError: name 'a' is not defined`, ta dernière ligne devrait être: `print(x["name"], "is taking the hard way")`

Students Generator (7.5 / 8)

Tu n'as pas écrit les docstring qui détaillent le traitement de tes fonctions: -0.5 points

eCommerce Order Report (9.5 / 10)

Tu n'as pas écrit les docstring qui détaillent le traitement de tes fonctions: -0.5 points

Exercice	Note	Commentaire
Nombre d'articles commandés	2 / 2	
Prix total de la commande	2 / 2	
Frais de port	2 / 2	
Récapitulatif	4 / 4	Tu modifies le dict en remplaçant <code>paid: True</code> par <code>paid: "Paid"</code> , il ne faut pas modifier les informations passées en entrée de ta fonction => Cf. correction
		Pour te simplifier la vie tu aurais pu faire plusieurs <code>print</code>
		Tu utilises plusieurs fois tes fonctions <code>get_articles_price</code> et <code>get_shipping_costs</code> alors que les valeurs d'entrée n'ont pas été modifiées. Dans ce cas il est recommandé de stocker le résultat des fonctions dans des variables puis d'utiliser ces variables. Cela évite d'avoir à faire le traitement plusieurs fois

Mystery Function (0 / 4)

Lucie et toi avez **exactement** les mêmes commentaires.

Yann

27/27 => 20/20

Warm-Up (1 / 1)

Dict Usage (4 / 4)

`if not values["option"]:` n'est pas nécessaire

Students Generator (8 / 8)

Exercice	Note	Commentaire
Email Generator	2 / 2	
Option Generator	2 / 2	
Marks Generator	2 / 2	Différence <code>choice(range(0,21)) / randint()</code> ? Aucune idée, <code>randint</code> est optimisé pour la génération de nombre "aléatoire" alors que <code>choice(range(0,21))</code> doit créer une liste temporaire (<code>range(0,21)</code>) puis choisir dans cette liste via <code>choice</code>
Student Generator	2 / 2	

eCommerce Order Report (10 / 10)

Exercice	Note	Commentaire
Nombre d'articles commandés	2 / 2	Ta méthode d'ajouter les quantités dans une liste puis de sommer la liste fonctionne d'un point de vue métier, mais présente des problèmes d'algorithmie s'il existe une commande avec plusieurs millions de quantités => Cf. correction
Prix total de la commande	2 / 2	Même commentaire qu'exercice ci-dessus
Frais de port	2 / 2	

Exercice	Note	Commentaire
Récapitulatif	4 / 4	

Mystery Function (4 / 4)