

---

# **Foire Aux Question du MOOC : Apprenez à coder avec Python**

*Version - 3.0 (2020)*

**Sébastien Hoarau - Thierry Massart - Isabelle Poirier**

**févr. 29, 2020**



---

## Table des matières

---

<b>1</b>	<b>Contenu de cette Foire aux questions</b>	<b>1</b>
<b>2</b>	<b>Utilisation de PyCharm, Python Tutor et UpyLaB</b>	<b>3</b>
2.1	PyCharm . . . . .	3
2.1.1	SyntaxError dans mon code . . . . .	6
2.2	PythonTutor . . . . .	6
2.3	UpyLaB . . . . .	7
2.3.1	Problème d'accès à UpyLaB . . . . .	7
2.3.2	UpyLaB me dit que mon code est faux ! . . . . .	8
2.3.3	Pourquoi UpyLaB valide le code complet mais pas ma fonction ! . . . . .	9
<b>3</b>	<b>Foire aux questions du Module 1</b>	<b>11</b>
3.1	FAQ générale du Module 1 . . . . .	11
3.1.1	Configuration nécessaire pour installer PyCharm . . . . .	11
3.1.2	Erreur d'installlation . . . . .	12
3.2	FAQ UpyLaB du Module 1 . . . . .	13
3.2.1	UpyLaB 1.1 . . . . .	13
<b>4</b>	<b>Foire aux questions du Module 2</b>	<b>17</b>
4.1	FAQ générale du Module 2 . . . . .	17
4.1.1	Version de Python utilisée par UpyLaB . . . . .	17
4.1.2	Module turtle . . . . .	18
4.1.3	Ordre d'évaluation du modulo . . . . .	19
4.1.4	Explication du Quiz sur la mousse au chocolat . . . . .	20
4.1.5	Utiliser une fonction prédéfinie . . . . .	20
4.1.6	Nom de variables . . . . .	21
4.1.7	Intérêt des constantes . . . . .	21
4.1.8	La fonction input() . . . . .	22
4.1.9	PythonTutor . . . . .	22
4.1.10	Import de fonctions . . . . .	23
4.1.11	Aide de la fonction « help() » . . . . .	23

4.1.12	Aide sur Turtle	24
4.2	FAQ UpyLaB du Module 2	25
4.2.1	Erreurs d'arrondis	25
4.2.2	UpyLaB 2.1	26
4.2.3	UpyLaB 2.2	28
4.2.4	UpyLaB 2.3	30
4.2.5	UpyLaB 2.4	32
4.2.6	UpyLaB 2.5	35
4.2.7	UpyLaB 2.6	36
4.2.8	UpyLaB 2.7	38
<b>5</b>	<b>Foire aux questions du Module 3</b>	<b>41</b>
5.1	FAQ générale du Module 3	41
5.2	UpyLaB du Module 3	41
5.2.1	UpyLaB 3.1	41
5.2.2	UpyLaB 3.2	44
5.2.3	UpyLaB 3.3	45
5.2.4	UpyLaB 3.4	45
5.2.5	UpyLaB 3.5	46
5.2.6	UpyLaB 3.6	46
5.2.7	UpyLaB 3.7	47
5.2.8	UpyLaB 3.8	48
5.2.9	UpyLaB 3.10	48
5.2.10	UpyLaB 3.11	49
5.2.11	UpyLaB 3.12	50
5.2.12	UpyLaB 3.13	52
5.2.13	UpyLaB 3.14	58
5.2.14	UpyLaB 3.15	59
5.2.15	UpyLaB 3.17	61
5.2.16	UpyLaB 3.18	61
<b>6</b>	<b>Foire aux questions du Module 4</b>	<b>63</b>
6.1	FAQ générale du Module 4	63
6.2	UpyLaB du Module 4	63
6.2.1	UpyLaB 4.1	63
6.2.2	UpyLaB 4.3	64
6.2.3	UpyLaB 4.4	65
6.2.4	UpyLaB 4.5	67
6.2.5	UpyLaB 4.6	69
6.2.6	UpyLaB 4.7	70
6.2.7	UpyLaB 4.8	71
6.2.8	UpyLaB 4.9	72
6.2.9	UpyLaB 4.11	74
	<b>Index</b>	<b>77</b>

# CHAPITRE 1

---

## Contenu de cette Foire aux questions

---

Cette Foire Aux Questions (FAQ) regroupe, un ensemble d'échanges entre un apprenant qui a un soucis dans une partie du cours ou dans la résolution d'un exercice UpyLaB et l'équipe pédagogique.

Nous avons bien sûr anonymisé les apprenants en leur donnant un prénom d'emprunt.

Pour chaque exercice UpyLaB, cette FAQ regroupe également les erreurs fréquemment rencontrées dans les codes, leur explication et des conseils pour les éviter. Chaque partie reprend les erreurs dues aux nouveaux concepts vus dans ce module.

Pour ne pas se répéter, quand une erreur se rencontre dans plusieurs exercices, nous ne le repertorions généralement que dans le premier exercice UpyLaB où il apparaît.

Pour plus de clarté, les codes sont parfois précédés en début de chaque ligne par le numéro de la ligne (comme donné par PyCharm ou UpyLaB) et sous la ligne erronée le caractère ^ précise l'endroit où l'erreur a lieu.

Enfin, cette FAQ débute par des remarques globales sur l'utilisation des 3 outils préconisés dans ce MOOC. D'autres parties de la FAQ y seront consacrées en particulier dans la FAQ du module 1 si ces parties sont liées à des soucis d'installation.



---

### Utilisation de PyCharm, Python Tutor et UpyLaB

---

Cette section contient des remarques globales sur l'utilisation des 3 outils préconisés ou utilisés dans ce MOOC : PyCharm, Python Tutor et UpyLaB.

#### 2.1 PyCharm

Vous avez installé et créé un projet dans PyCharm, voyons comment créer et exécuter un script.

Un projet peut contenir plusieurs scripts bien sûr. Soit qui sont liés (c'est le cas dans les gros projets réels) soit qui sont indépendants mais appartiennent à une même famille... c'est le cas par exemple si je décide de faire un projet par Module de mon MOOC : je mettrai dans le projet `Module_1` tous mes exercices du Module 1 etc. On peut aussi faire un gros projet `MOOC_Python` et y mettre tous ses scripts sans distinction de module.

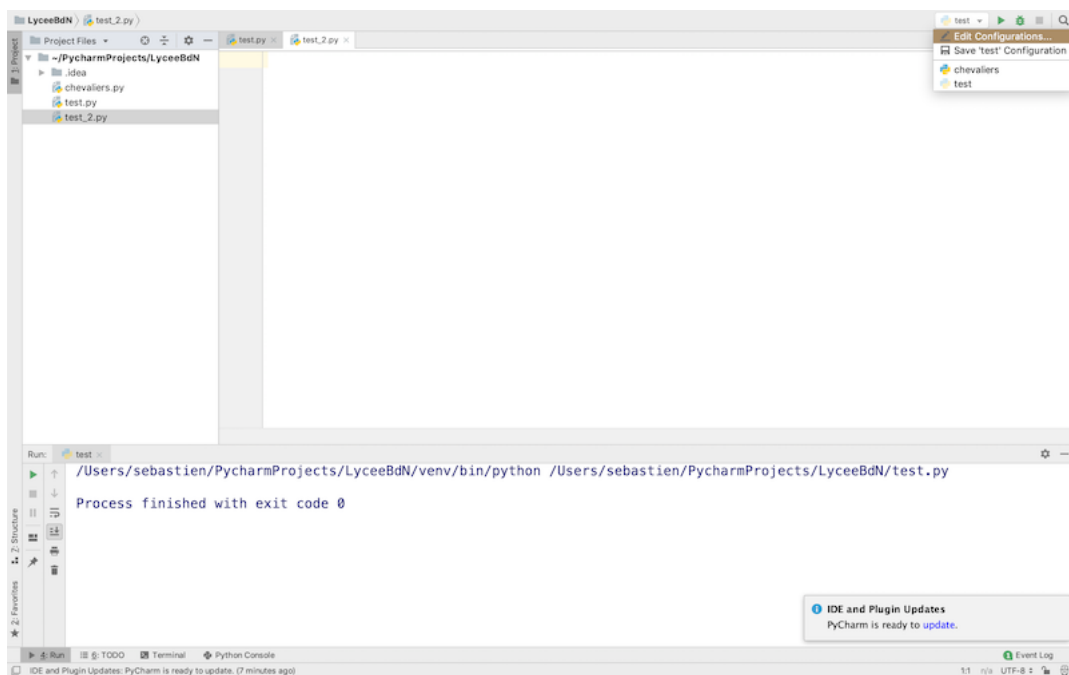
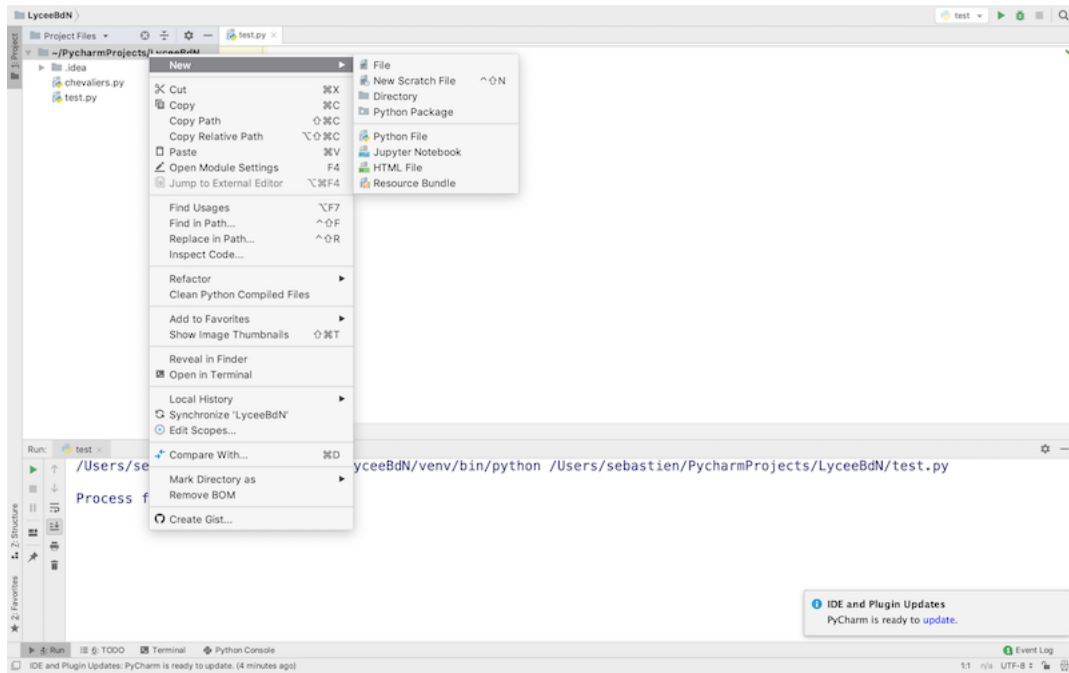
Cette organisation est bien sûr question de goût, d'habitude de travail et donc propre à chacun.

Lorsque votre projet est créé, pour y ajouter un nouveau script vous pouvez utiliser un clic droit sur le nom du projet, dans la colonne de gauche et faire `New > Python File`. Vous donnez un nom à ce script qui vient se rajouter aux autres. Dans la colonne de gauche vous avez l'ensemble des scripts. Ainsi, les deux captures suivantes montrent comment j'ai créé un nouveau script nommé `test_2.py`.

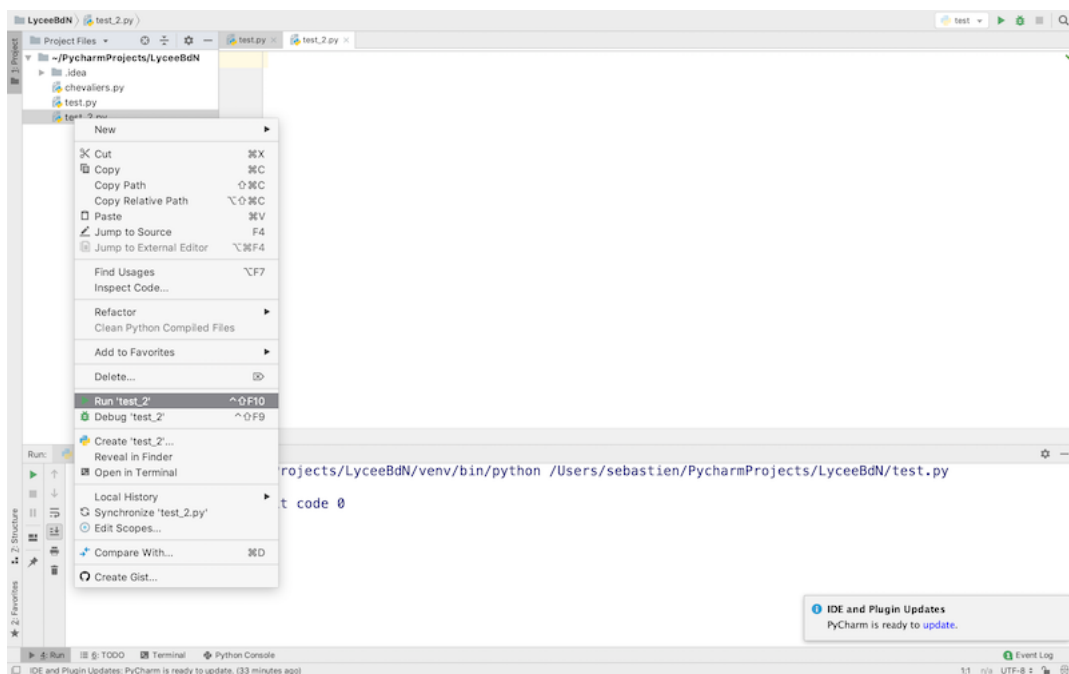
Parmi tous les scripts de votre projet, pour en exécuter un, il y a deux cas de figure :

- 1) soit il est sélectionné comme script courant (il apparaît en haut à droite avec la petite flèche verte) auquel cas utiliser cette petite flèche ou faire `Run` dans le menu exécutera ce script ;
- 2) soit il ne l'est pas et alors le plus simple est de faire, dans la colonne de gauche, un clic droit sur le nom du script que vous voulez exécuter (disons `test_2.py` comme sur la capture ci-dessous) et choisir dans le menu contextuel `Run test_2`

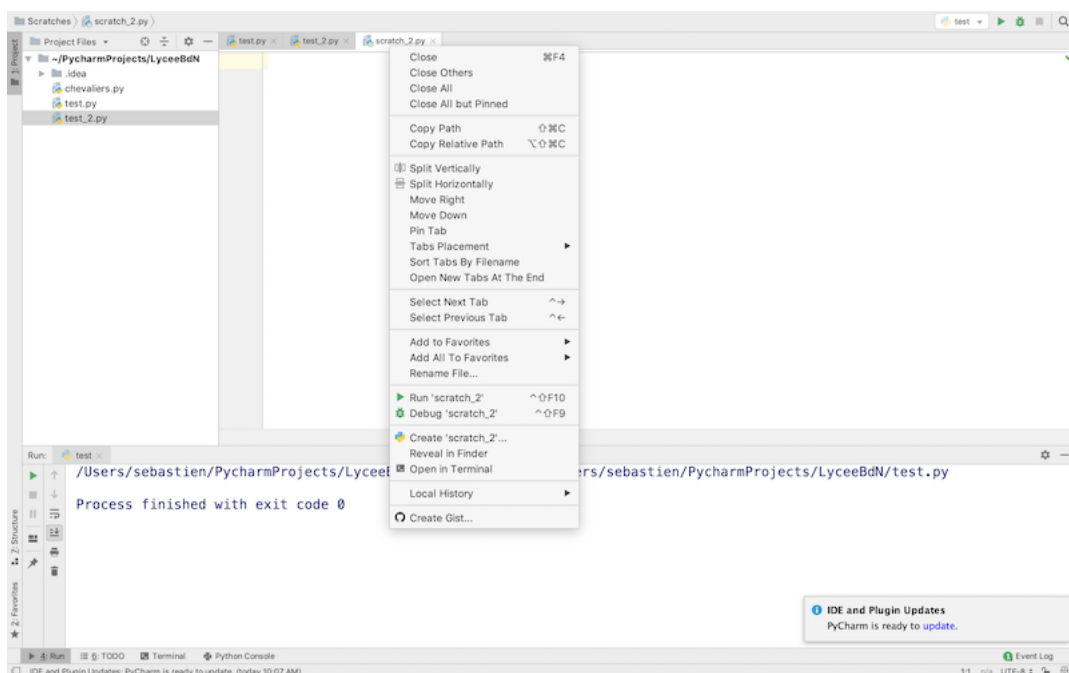
**Et les scratch file ?**







Les scratch files sont des fichiers python comme les autres mais numérotés automatiquement et rangés dans un dossier (projet) à part. Par exemple ci-dessous j'ai créé un nouveau scratch file, (ce n'est pas mon premier comme l'atteste le 2 dans le nom) et il n'apparaît pas dans mon Projet (qu'on voit à gauche). Mais si je fais apparaître le menu contextuel en cliquant-droit sur le nom `scratch_2.py`, je vois que j'ai un `Run scratch_2` et je peux donc l'exécuter.



### Conseil

De préférence, créez des fichiers python dans votre ou vos projets et de n'utilisez pas les *scratch files* de peur de ne plus savoir où se trouvent vos codes.

## 2.1.1 SyntaxError dans mon code

Si vous avez un message `syntax error` dans PyCharm qui indique par exemple la ligne suivante :

```
print(long*sin(pi/3), long*cos(pi/3))  
  
^  
  
SyntaxError: invalid syntax
```

en fait l'erreur se situe généralement à la ligne ou l'instruction précédente.

Vérifiez donc par exemple vos parenthèses, ...

Ici j'avais par exemple comme code :

```
from math import pi, cos, sin  
long = float(input())  
print(long, 0.0  
print(long*cos(pi/3), long*sin(pi/3))
```

et c'est la parenthèse manquante du premier print qui a créé l'erreur.

## 2.2 PythonTutor

Emile

Bonjour,

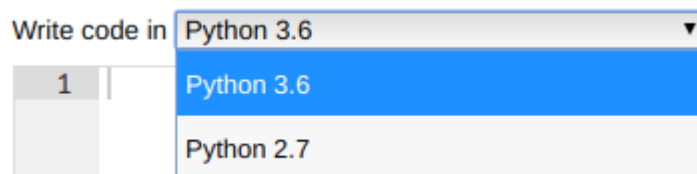
PythonTutor ne parvient pas à exécuter mon code alors que celui-ci fonctionne sur ma machine.

Merci.

Réponse de l'équipe

Bonjour,

**Vérifiez que vous utilisez la bonne version de Python (en l'occurrence la 3.6) dans PythonTutor.**



L'équipe.

## 2.3 UpyLaB

### 2.3.1 Problème d'accès à UpyLaB

Avant tout, il est toujours prudent que vous conserviez vos codes des exercices UpyLaB, pour éviter les pertes éventuelles (soyons toujours prudent).

Lors de la première session, des soucis d'accès à UpyLaB ont été observés (la fenêtre reste bloquée sur le message « Connexion en cours ... », ou « test en cours »); l'outil a été amélioré depuis.

Si ce souci apparaît, sauvez votre code et demandez de rafraîchir la page.

Si le problème persiste, réessayez un peu plus tard et en attendant de pouvoir valider vos solutions, faites vos exercices localement sur PyCharm par exemple. Les énoncés UpyLaB sont accessibles dans le support de cours fourni.

Sinon, si vous débutez avec ce MOOC et UpyLaB, le souci est peut-être lié à votre navigateur, à sa configuration ou à une extension comme ghostery qui bloque les interactions avec UpyLaB. Des soucis ont été évoqués avec les navigateurs : Safari, Firefox, Chromium et Microsoft Edge.

Pour ce genre de soucis, voici des conseils trouvés dans le forum :

**Bonjour,**

- 1) j'ai eu le souci sous safari, je suis donc passée à Google Chrome et tout fonctionne bien maintenant.
- 2) J'utilise le plugin « noscript » avec Firefox, il faut faire attention à autoriser les domaines fun-mooc.fr ET ulb.ac.be, sinon, on a également le message « Connexion en cours ... » de manière permanente.
- 3) J'utilise également Firefox (à jour) agrémenté de NoScript et ABP. J'ai rencontré le problème de connexion à UpyLaB et l'ai résolu en cliquant à gauche de la barre d'url (icône bouclier / cadenas) puis en sélectionnant « Désactiver le blocage ».
- 4) J'utilise Firefox avec un bloqueur, et il est exact qu'avec un bloqueur de pub cela ne fonctionne pas correctement.
- 5) Il faut en effet que le navigateur accepte les pop-ups
- 6) Pour les utilisateurs de Firefox paramétré pour bloquer les cookies tiers, il faut aller dans ajouter une exception pour autoriser <https://upylab.ulb.ac.be>. Préférences - Vie privée et sécurité - Gérer les permissions - mettre <https://upylab.ulb.ac.be> dans le champs Adresse du site web  
— Autoriser - Enregistrer les modifications. Recharger la page et upylab juste marche :-) Cette solution fonctionne sous Chromium,
- 7) Si vous utilisez ghostery, il faut mettre « site de confiance » pour que cela fonctionne.

## 2.3.2 UpyLaB me dit que mon code est faux !

Voici quelques astuces pour déboguer votre code.

Faites toujours attention aux espaces, tabulations, nouvelles lignes. Ces caractères ne sont pas visibles pour l'humain, mais UpyLaB les voit ! Un espace en trop ou une majuscule à la place d'une minuscule sont des erreurs fréquentes ! En effet, « Bonjour ! », « Bonjour ! », « bonjour ! » sont tous des textes différents !

Supposons qu'UpyLaB vous demande d'écrire `n` caractères '+' sur une ligne après avoir lu l'input `n` et que votre code soit

```
n = int(input())
print('+'*n)
print()    # ici j'imprime malheureusement une ligne de trop !
```

Comment faire pour voir l'erreur.

### Utilisez Python Tutor

Premier conseil : si votre code ne contient pas de module comme `turtle...`, utilisez Python Tutor comme expliqué en section 1.5 en cliquant sur le texte « Visualize your code and get live help now » et en explicitant que le code est écrit en python 3.6.

### UpyLaB me dit Le code comprend des erreurs : Ligne 15 : Erreur de syntaxe

- Mais la ligne 15 et 14 sont parfaitement syntaxiquement correctes ou pire, mon code fait moins de 15 lignes. Des lignes sont parfois ajoutées à votre code pour permettre à UpyLaB de le tester. Donc votre erreur est peut-être quelques lignes avant.
- Attention aussi à la version de Python utilisée par UpyLaB : pour la session actuelle il s'agit de Python 3.5.3 qui ne permet pas encore les f-strings introduit avec Python 3.6.

### UpyLaB me donne le message :

*Erreur serveur : Votre programme a planté ou a duré trop longtemps. Vérifiez que vos boucles terminent.*

L'exécution de votre fonction boucle probablement indéfiniment, comme l'exécution de celle-ci :

```
def ma_fonction_qui_cycle(n):
    while True:
        pass
```

### UpyLaB me dit le message : Ligne 13 « EOF » when reading a line

Si vous recevez d'UpyLaB le message « EOF when reading a line », cela signifie que UpyLaB a testé votre code et que ce dernier demande un input quand en fait UpyLaB a donné tout ce qu'il devait.

EOF signifie « end of file » : ici plus de données fournies en entrée (input)

Donc votre code fait au moins un `input()` en trop

Attention, comme des lignes de code sont ajoutées par l'outil UpyLaB pour tester votre code, l'`input()` n'est souvent pas à la ligne mentionnée mais plutôt quelques lignes avant (n'oubliez pas qu'UpyLaB n'est pas un outil de débogage ; il ne fait que valider si votre code est conforme à ce qui est requis).

### Autre astuce si mon code semble imprimer le texte demandé

Si vous voulez déboguer votre code dans PyCharm (ou UpyLaB), une astuce est d'ajouter à chaque print un caractère particulier pour voir où est la fin de la ligne

Ici :

```
n = int(input())
print('+'*n + '€') # ajoute un caractère '€' en fin de ligne
print('€') # idem
```

ou mieux, en utilisant l'argument end du print

```
n = int(input())
print('+'*n, end = "€\n") # ajoute un caractère '€' en fin de ligne
                        # suivi du passage à la ligne
print(end = "€\n")      # imprime juste le paramètre end
```

Le résultat de l'exécution avec n valant 5 sont les deux lignes

```
+++++€
€
```

ce qui met en évidence que votre code a une ligne en trop ! Après avoir corrigé, enlevez les '€' et tester votre code sur UpyLaB.

### Puis-je mettre mon code dans le forum ?

Comme précisé dans les règles du forum, aucune partie substantielle ou complète de code solution d'un exercice UpyLaB noté ne peut être diffusés. Pour pouvez décrire précisément les erreurs renvoyées par UpyLaB et même mettre des captures d'écran de ces messages.

## 2.3.3 Pourquoi UpyLaB valide le code complet mais pas ma fonction !

Dans les exercices UpyLab 4.1 à 4.4, nous vous demandons la définition de fonctions ET le code qui appelle cette fonction et imprime des résultats.

Certains d'entre vous peuvent avoir le code complet validé par UpyLaB mais pas le test UpyLaB sur la fonction elle même.

Le soucis probable est que vous n'avez pas bien appliqué le mécanisme de passage de paramètres.

Dans ce cas, nous vous invitons à bien voir ou revoir la matière sur les fonctions et en particulier la notion de paramètres formels, paramètres effectifs et la transmission de l'un à l'autre lors de l'appel.

Dans les exercices UpyLaB 4.1 à 4.4, le code de vos fonctions ne doit rien utiliser d'autre que leurs paramètres formels et les variables locales qui y sont définies !



---

### Foire aux questions du Module 1

---

#### 3.1 FAQ générale du Module 1

##### 3.1.1 Configuration nécessaire pour installer PyCharm

**Gaspard**

**Bonjour,**

Comment voir si la configuration de l'ordinateur est suffisante pour l'installation de PyCharm ?

Merci.

**Réponse de l'équipe**

Bonjour,

Cela dépend de votre système :

- Windows : Cliquez droit sur le bouton windows puis "système"
- Linux : Aller dans « moniteur système »
- Mac : Cliquez sur la pomme et dans « à propos de ce mac »

Rappel : Pour l'installation de PyCharm, il faut choisir la version gratuite (Sur le site de JetBrains mentionné ci-dessous, cliquez sur "Download Now" et puis sur "Community") qui demande les ressources suivantes : cfr la section "System Requirements" sur le site <https://www.jetbrains.com/pycharm/download/>

D'autres IDE plus légers sont disponibles si votre système ne supporte pas PyCharm : Visual Studio Code, Atom, SublimeText, etc.

L'équipe.

**Gaspard**

**Bonjour,**

Est-il nécessaire d'utiliser PyCharm pour suivre ce MOOC ?

Merci.

**Réponse de l'équipe**

Bonjour,

Bien que PyCharm ne soit pas indispensable au suivi de ce MOOC ou à l'apprentissage de Python3, c'est un outil (IDE) très complet qui va faciliter l'écriture (le développement) de code python en suivant de bonnes pratiques telle que la PEP8.

D'autres IDE plus légers sont disponibles si votre système ne supporte pas PyCharm : Visual Studio Code, Atom, SublimeText, etc.

L'équipe.

### 3.1.2 Erreur d'installation

**Georges**

**Bonjour,**

J'ai un problème de DLL avec Python3 dans PyCharm. Que puis-je faire ?

Merci.

**Réponse de l'équipe**

Bonjour,

Ce problème est fort probablement lié à une mise à jour manquante de votre système. Pour corriger cela, téléchargez la mise à jour depuis de lien : <https://support.microsoft.com/en-us/help/2999226/update-for-universal-c-runtime-in-windows> en sélectionnant votre système (Windows Vista, 7, 8 ou 8.1)



et la version de votre système (32 bits = x86 ou 64 bits = x64). Pensez à redémarrer votre système également.

L'équipe.

## 3.2 FAQ UpyLaB du Module 1

### 3.2.1 UpyLaB 1.1

LUC

**Bonjour,**

Ayant été confronté au souci, j'indique ici un détail sur l'exécution du premier exercice UpyLaB. Si cela peut servir à d'autres...

Après donc plusieurs essais infructueux me renvoyant toujours la même réponse d'erreur (je ne me souviens plus du libellé exact, mais quelque chose comme « erreur de syntaxe ligne 1 »), j'ai fini par entrer l'instruction

```
print("Bonjour UpyLaB !")
```

dans la fenêtre UpyLab en ligne.

Eh oui, ayant mal compris les modalités de l'exercice, j'entrais en fait ladite instruction dans le logiciel PyCharm installé sur mon ordinateur et copiais ensuite la réponse obtenue dans la fenêtre UpyLaB : forcément, ça bloquait !

**Réponse de l'équipe**

Bonjour LUC,

Hé oui il s'agit effectivement de résoudre l'exercice demandé dans PyCharm puis, une fois que vous êtes satisfait, de copier-coller l'ensemble de votre **code Python** dans la fenêtre UpyLaB. Ce n'est effectivement pas le résultat obtenu par l'exécution PyCharm qu'il faut mettre dans cette fenêtre.

### Erreurs de syntaxe fréquentes

#### 1.1a

```
1  print("Bonjour UpyLaB !")  
    ^
```

**Explication :** Le code (ici l'instruction `print`) ne commence pas en début de ligne.

**Conseil :** Nous verrons au module 3 que certaines instructions Python ne débutent pas en début de ligne (car elles sont imbriquées dans d'autres structures de programme). Jusque là, les instructions Python débutent toujours au début de la ligne.

### 1.1b

```
1 print(Bonjour UpyLaB !)
```

```
1 print("Bonjour UpyLaB !)
```

```
1 print("Bonjour UpyLaB !')
```

**Explication et conseil :** ne pas oublier les doubles apostrophes en début et fin du texte Bonjour UpyLaB !

### 1.1c

```
1 print "Bonjour UpyLaB !"
```

**Explication et conseil :** ne pas oublier les parenthèses après `print` et en fin. **Note :** si votre ce code fonctionne sans erreur dans votre environnement PyCharm, cela signifie que vous utilisez un mauvais interpréteur (Python 2.7 par exemple). Dans ce cas, vous devez modifier l'interpréteur que vous utilisez (interpréteur Python 3.8 par exemple) car dans ce cours nous apprenons Python 3 et pas Python 2 !

## Non validation par UpyLaB

Voici quelques exemples de réponses non validées par UpyLaB pour l'exercice UpyLaB 1.1.

### 1.1d

```
1
```

**Explication et conseil :**

Validez un code vide risque de ne pas bien fonctionner;-)

### 1.1e

```
1 print("Bonjour UpyLab !")
```

**Explication et conseil :**

```
print("bonjour UpyLaB !")
      ^
print("Bonjour UpyLaB! ")
              ^
print("Bonjour UpyLaB ! ")
              ^
print("Bonjour upylab !")
      ^   ^   ^
print("Bonjour UpyLab !")
              ^
print("Bonjour UpylaB !")
              ^
```

autant de textes qui ne donnent pas la bonne réponse car différent de ce qui est demandé.

### 1.1f

```
1 print("Bonjour UpyLaB ! ")
      ^
```

#### Explication et conseil :

Ici nous avons une **erreur difficile à détecter** : notez que pour cet exemple erroné, UpyLaB affiche la différence

L'appel à votre programme a renvoyé:

Bonjour UpyLaB !

Le résultat attendu était :

Bonjour UpyLaB !

Nous sommes d'accord, cela semble la même chose. Et bien non : si vous surlignez avec votre souris le premier texte, vous avez le texte affiché `Bonjour UpyLaB !` avec une espace en fin tandis que le second texte n'en a pas ; les textes sont donc différents !

## AUTRES CONSEILS

L'exercice UpyLaB 1.1 peut être solutionné en copiant dans l'énoncé le texte `print("Bonjour UpyLaB !")` et en le collant dans la fenêtre UpyLaB avant de le valider.

L'interpréteur Python et UpyLaB exigent que le code soit correct et que tous les détails soient identiques à ce qui est demandé.



## Foire aux questions du Module 2

### 4.1 FAQ générale du Module 2

Cette section comprend les questions posées et réponses au sujet de la matière abordée dans le module 2 du cours.

#### 4.1.1 Version de Python utilisée par UpyLaB

MORGANE

**Bonsoir,**

Quelle version de Python UpyLaB utilise-t-il pour valider les codes ?

#### **Réponse de l'équipe**

Bonjour Morgane,

L'interpréteur Python utilisé actuellement dans UpyLaB tourne avec Python 3.5.3.

Ce n'est pas un souci pour tout ce qui est vu ici.

Malheureusement, pour ceux qui utilisent des concepts introduits dans Python 3.6 ou version ultérieures, cela peut poser problème.

L'utilisation de f-strings, introduits dans la version 3.6, pose problème à UpyLaB qui voit le code comme une erreur de syntaxe.

De même le comportement des dictionnaires diffère légèrement entre les versions Python 3.5 et Python 3.8 au niveau de l'ordre des éléments insérés ; cela peut impacter certains exercices qui fonctionneraient dans PyCharm (par exemple avec Python 3.8) mais qui ne seraient pas validés avec UpyLaB.

Nous ne pouvons mettre la dernière version durant cette session, sans risquer une interruption de service majeure. Désolés pour ceci.

### Réponse de Morgane

Merci pour votre réponse

## 4.1.2 Module turtle

### RODRIGUE

Bonsoir,

J'essaye d'exécuter un code qui utilise le module turtle et j'ai l'erreur suivante :

```
/Users/rodrigue/venv/bin/python
/Users/rodrigue/Library/Preferences/PyCharmCE2018.3/scratches/scratch_64.py
Traceback (most recent call last): File
"/Users/rodrigue/Library/Preferences/PyCharmCE2018.3/scratches/scratch_64.py",
line 3, in <module> turtle.up() AttributeError: module
'turtle' has no attribute 'up'

Process finished with exit code 1
```

### Réponse de l'équipe

Bonjour Rodrigue,

Le message d'erreur produit par l'interpréteur Python indique, en bas de message, que le module turtle n'a pas d'attributs `up`. Je vois que le fichier contenant ton script s'appelle `scratch_64.py`, ce qui ne devrait pas poser de souci, mais es-tu sûr que tu n'as pas un script que tu as appelé `turtle.py` ?

Si c'est le cas, c'est ce script qui est importé lors de l'instruction `import turtle` en lieu et place du module `turtle` contenant toutes les méthodes désirées.

Renomme ce fichier différemment d'un module que tu utilises (donc non plus `math.py`, `random.py`, ...) et ton code devrait fonctionner.

### Réponse de Rodrigue

Effectivement, j'avais nommé un fichier `turtle.py`, ce qui n'était donc pas une bonne idée.

Merci !

## 4.1.3 Ordre d'évaluation du modulo

DEDA

Bonjour à toutes et tous,

Je comprends mal la correction de la dernière question de la section 2.2. Celle avec l'ordre d'exécution des opérateurs par rapport à modulo.

Je lis que dans `2 % 2 * 2`, le résultat est 0

`2 % 2` est évalué avant `* 2` (`*` et `%` associatifs à gauche et ont la même priorité).

Vu que `2 % 2` est évalué avant la multiplication je serais tenté de dire que modulo a une priorité supérieure à la multiplication. Par ailleurs, je pense mal comprendre la notion d'associativité à gauche. Dois-je comprendre que cela veut dire que `a * b * c` est équivalent à `(a * b) * c`?

### réponse de Steve

Bonjour Deda,

l'ordre de priorité des opérateurs arithmétiques est :

1° exponentiations

2° divisions, multiplications et modulo

3° additions et soustractions

dans un même niveau de priorité, l'évaluation va de gauche à droite sauf pour l'exponentiation qui est une exception où `5 ** 2 ** 3` est égal à `5 ** (2 ** 3)`.

Les parenthèses servent à modifier cet ordre si nécessaire.

[https://www.tutorialspoint.com/python/operators\\_precedence\\_example.htm](https://www.tutorialspoint.com/python/operators_precedence_example.htm)

#### 4.1.4 Explication du Quiz sur la mousse au chocolat

ANTOINE

Bonjour,

Je ne comprends pas les calculs présents dans le quizz sur la mousse au chocolat.

Merci d'avance pour votre aide.

##### réponse de l'équipe

Bonjour Antoine, Quand on a 3 oeufs pour 4 personnes et qu'on veut savoir combien on en a besoin pour 7 personnes, il suffit d'utiliser la règle de trois. Pour cela, on calcule le nombre d'oeufs pour 1 personne d'abord en divisant 3 oeufs par 4 (car 3 oeufs étaient nécessaires pour 4 personnes, il faut donc  $\frac{3}{4}$  d'oeuf pour 1 personne) et ensuite on multiplie cela par 7 (pour 7 personnes). Nous obtenons un chiffre avec une décimale donc il suffit d'arrondir avec la fonction round que nous avons vue. Cela revient à multiplier les 3 oeufs par  $\frac{7}{4}$  et nous arrondissons : `round(3 * 7 / 4)`.

J'espère que cette explication est assez claire.

L'équipe.

#### 4.1.5 Utiliser une fonction prédéfinie

JOSEPH

Bonjour,

Comment utiliser la fonction "round" prédéfinie dans Python ?

Merci.

##### Réponse de l'équipe

Bonjour,

Pour savoir comment utiliser une fonction prédéfinie de Python, il suffit de taper "help(nom de la fonction)" dans la console (comme mentionné dans la vidéo de la section 2.5). Pour plus d'information sur le fonctionnement d'une fonction et comment créer les vôtres, rendez-vous au module 4 ou patientez si ce dernier n'est pas encore disponible.

L'équipe.



### 4.1.6 Nom de variables

SIMON

Bonjour,

Après avoir défini la variable `min = 4`, je ne peux plus utiliser la fonction `min(5, 10)`, pourquoi ?  
Merci.

#### Réponse de l'équipe

Bonjour,

C'est normal, vous avez redéfini `min` (qui était une fonction prédéfinie Python) en entier valant 4. Vous ne pouvez donc plus utiliser la fonction.

Pour cela, il est déconseillé d'utiliser des noms de variables ayant déjà une signification dans Python comme `min`, `max`, `print`.

Voici une liste des mots et noms de fonctions réservés de Python : [https://fr.wikibooks.org/wiki/Programmation\\_Python/Tableau\\_des\\_mots\\_r%C3%A9serv%C3%A9s](https://fr.wikibooks.org/wiki/Programmation_Python/Tableau_des_mots_r%C3%A9serv%C3%A9s)

L'équipe.

### 4.1.7 Intérêt des constantes

Léon

Bonjour,

Je ne comprends pas l'intérêt des « constantes ».

Merci.

#### Réponse de l'équipe

Bonjour,

Une constante est une variable donc la valeur ne change pas durant l'exécution du code. L'intérêt d'une constante peut vous paraître assez limité pour l'instant mais elle se révèle très pratique pour définir une valeur fixée telle que la valeur de la TVA, la taille maximale d'une fenêtre, le nombre pi, etc. Ainsi, on peut très facilement réutiliser ces variables. Si on utilise la variable `TVA` à plusieurs endroits de notre programme et qu'un jour on souhaite augmenter la TVA de 21 à 22%, il suffira de modifier une seule variable plutôt que de changer la valeur 21 qu'on utilise partout et de relancer notre code. C'est également plus clair que

d’avoir des “nombres magiques” un peu partout dans le code : 21 pour la TVA, 1080 pour la définition de l’écran, ...

Ceci est expliqué dans la section 4.4 que vous pourrez lire plus tard

L’équipe.

### 4.1.8 La fonction input()

Sarah

Bonjour,

La fonction input() (du module 2.5) peut-elle prendre d’autres types que le float ?

Merci.

#### Réponse de l’équipe

Bonjour,

Oui, vous pouvez entrer du texte (“string” ou “chaîne de caractères”) également. Dans tous les cas, que vous entriez des nombres (entiers ou à virgule) ou du texte, la fonction input() convertira votre input vers un string, il sera donc nécessaire de transformer votre input en entier ou flottant si vous voulez effectuer des actions dessus. N’hésitez pas à tester par vous même ! Exemple : que se passe-t-il si vous faites int(input(“Nombre : “)) et que vous donnez un string à l’input ? Est-ce logique ?

L’équipe.

### 4.1.9 PythonTutor

Emile

Bonjour,

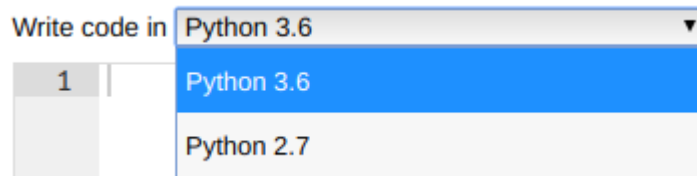
PythonTutor ne parvient pas à exécuter mon code alors que celui-ci fonctionne sur ma machine.

Merci.

#### Réponse de l’équipe

Bonjour,

Vérifiez que vous utilisez la bonne version de Python (en l'occurrence la 3.6) dans PythonTutor.



L'équipe.

#### 4.1.10 Import de fonctions

Sévrine

Bonjour,

Comment faire pour n'importer que la fonction cosinus du module math par exemple ?

Merci.

Réponse de l'équipe

Bonjour,

Il suffit d'écrire : `from math import cos` (Comme indiqué dans la première vidéo de la section 2.6.)

L'équipe.

#### 4.1.11 Aide de la fonction « help() »

Elinne

Bonjour,

Pourquoi j'obtiens une erreur lorsque je tape "help(math)" ?

Merci.

Réponse de l'équipe

Bonjour,

Car “math” est un module, une librairie de Python. Il faut d’abord l’importer avec l’instruction “import math” puis taper “help(math)”. (Comme indiqué dans la première vidéo de la section 2.6.)

L’équipe.

#### 4.1.12 Aide sur Turtle

Anthony

Bonjour,

J’ai un problème avec turtle : la tortue ne va pas dans la bonne direction. Comment faire ?

Merci.

##### Réponse de l’équipe

Bonjour,

Faites attention à la direction de la tortue en regardant vers où la flèche pointe. Au début, la tortue va regarder vers la droite (0°). Si vous la faites tourner de 90° vers la droite, elle regardera vers le bas, si vous la faites tourner de 90° vers le haut, elle regardera vers le haut. Mais attention, si elle regarde déjà vers le haut, tourner la tortue à gauche de 90° va la faire regarder vers la gauche. Lorsqu’on lui dit de tourner à gauche/droite, c’est par rapport à sa direction actuelle.

L’équipe.

Alexandre

Bonjour,

La fenêtre avec la tortue s’est directement fermée. Comment faire pour la garder ?

Merci.

##### Réponse de l’équipe

Bonjour,

N’oubliez pas l’instruction finale “turtle.done()” dans le cas d’un script afin de garder la fenêtre ouverte. (Comme indiqué dans la deuxième vidéo du module 2.6.)

L’équipe.

**Roger**

**Bonjour,**

Dans turtle, la forme n'a pas été coloriée.

Merci.

**Réponse de l'équipe**

Bonjour,

N'oubliez pas de commencer par `turtle.begin_fill()` et de finir par `turtle.end_fill()`

L'équipe.

## 4.2 FAQ UpyLaB du Module 2

Cette section comprend les questions posées et réponses au sujet des exercices UpyLaB proposés dans le module 2 du cours.

### 4.2.1 Erreurs d'arrondis

**ALINE**

**Bonjour,**

Plusieurs exercices UpyLaB du module 2 (et certains de modules ultérieurs) demandent de calculer et parfois d'afficher des valeurs fractionnaires (float). Les calculs ne sont pas totalement précis et donc UpyLaB renvoie des résultats avec des erreurs d'arrondi. Par exemple avec `a = 100`, `a * cos(pi / 3)` donne `50.0000000000000014` et `-a * cos(5 * pi / 3)` donne `-49.99999999999998` à la place respectivement de `50.0` et `-50.0`. Est-ce un souci ?

J'ai utilisé `round(res, 3)` avec `res` valant `-77.94228634059945` (ce qui donne `77.942`, et UpyLaB ne valide pas mon exercice.

**Réponse de l'équipe**

Bonjour ALINE,

L'exerciceur UpyLaB tient compte dans les calculs des erreurs de troncature et d'arrondi, généralement

avec une tolérance de l'ordre de  $1.0e-5$ ; donc 50.0 ou 50.0000000000000014 ou encore -49.99999999999998 seront jugés égaux par l'outil.

Par contre entre -77.942 et -77.94228634059945 la différence est trop importante et due à des calculs différents. L'utilisation de `round( , 3)` dans votre code est donc à proscrire ici.

Bonne continuation.

## 4.2.2 UpyLaB 2.1

JÉRÔME

Bonjour,

Lorsque je fais le premier exercice d'assignation, le code fonctionne dans Python Tutor sans signalement d'erreurs mais ne fonctionne pas pour la chaîne de caractères dans upylab. Je l'ai testé dans spyder sous anaconda. Merci de votre aide.

Réponse de l'équipe

Bonjour JÉRÔME,

Les détails ont de l'importance. Ici le nom des variables, le texte précis, ...

Bon travail.

Réponse de MORGANE

Oui même le choix majuscule/minuscule... tout doit être exactement comme demandé.

Réponse de JÉRÔME

Merci à tous. J'avais effectivement mis `bonjour` à la place de `Bonjour`.

WILLY

Bonjour,

J'ai du mal à utiliser `input`. En effet quand j'écris `input` sans paramètre rien ne s'affiche le programme attend. Tandis que si je mets `input("produit = "+str(x*x))` ça répond à la question mais il s'arrête encore et rien ne se passe. Merci

### Réponse de l'équipe

La fonction `input` est utilisée pour lire des valeurs saisies au clavier par l'utilisateur. Il est donc normal que ton programme se fige : il attend tout simplement que tu saisisse quelque chose. Si tu veux afficher quelque chose, c'est la fonction `print` qu'il faut utiliser.

### Réponse de WILLY

Mais dans cet exercice, on nous demande de ne pas utiliser la fonction `print`.

### Réponse de l'équipe

Effectivement. Dans cette activité, il s'agit simplement d'affecter des valeurs à des variables. Il ne doit donc y avoir ni appel à `print`, ni appel à `input` dans le code que l'on soumet à UpyLaB.

Cependant, si tu veux exécuter ton code dans PyCharm et visualiser le résultat, il te faudra là ajouter des appels à `print`.

Bon courage !

### Réponse de WILLY

Merci pour la réponse. Cela fonctionne.

## Non validation par UpyLaB

### 2.1a

```
div_entiere = 36 / 5
```

#### Explication et conseil :

En Python 3, `36 / 5` donne la valeur `7.2` et la division entière utilise l'opérateur `//` (`36 // 5` donne `7`). Notez que si `36 / 5` donne la valeur `7` dans votre environnement PyCharm ou l'environnement de développement (IDE) que vous utilisez (IDLE, Thonny, Spyder, Eclipse, Pydev, ...) cela signifie que vous utilisez un mauvais interpréteur (Python 2.7 par exemple). Dans ce cas, vous devez modifier l'interpréteur que vous utilisez (interpréteur Python 3.8 par exemple) car dans ce cours nous apprenons Python 3 et pas Python 2!

### 2.1b

```
pi = 3,14159
    ^
```

### Explication et conseil :

En Python, la partie fractionnaires des nombres `float` (fractionnaires) est précédée par un point `.` et pas une virgule `,`. Nous verrons que la virgule est utilisée pour définir des tuples. Ainsi, on peut voir qu'après avoir fait `pi = 3,14159`, l'instruction `print(pi)` imprime le tuple (ici le couple) d'entiers `(3, 14159)`

### 2.1c

```
print('x = 36')
...
print('mon_texte = "Bonjour"')
```

Ne passe pas le test UpyLaB !

### Explication et conseil :

L'exercice demande de faire des assignations à des variables et non des print.

## 4.2.3 UpyLaB 2.2

### RÉJANE

Bonjour,

J'ai toujours une erreur UpyLaB sur les 4 lignes du genre :

```
L'appel à votre programme sur l'input ``1.0↔2.0↔`` a renvoyé:
taper la valeur de a : taper la valeur de b : la moyenne est :
```

```
Le résultat attendu était : 1.5
```

### Réponse de l'équipe

Bonjour Réjane,

Lisez bien les consignes de l'énoncé : il ne faut pas mettre de texte supplémentaire dans les `input` et dans les `print` :

```
a = int(input())
print(a)
```

Et non :

```
a = int(input('Valeur de a :'))
print('a = ', a)
```



### Réponse de RÉJANE

Merci beaucoup.

LUC

Bonsoir,

J'ai bien deux variables `a` et `b` lues par `input` et qui sont des `float`. Puis je calcule la moyenne par `a + b / 2` mais UpyLaB refuse de valider.

### Réponse de l'équipe

Bonsoir Luc,

Attention à la priorité des opérateurs.

### Réponse de LUC

Effectivement la division est plus prioritaire que l'addition : il me fallait des parenthèses. Merci.

## Erreurs de syntaxe

### 2.2a

```
a = float(input())
b = float(input())
(a + b) / 2
print(res)
    ^
```

**Explication et conseil :** La variable `res` n'a pas été définie et Python ne sait donc pas quoi imprimer à la ligne 4. À la ligne 3, Python fait le calcul demandé, mais le résultat n'est assigné à aucune variable et est donc inaccessible dans la suite du programme.

### 2.2b

```
float(input(a))
    ^
```

**Explication et conseil :** Ceci n'est pas une erreur de syntaxe à proprement parler, à condition que la variable `a` ait été définie en amont, mais ce n'est pas comme cela que s'utilise la fonction `input`. Cette fonction retourne

un résultat qui doit être affecté, à l'aide de l'opérateur d'affectation =, à une variable si on veut conserver cette valeur.

### 2.2c

```
a = float((input()))
```

**Explication et conseil :** Cette instruction contient 3 parenthèses ouvrantes et 2 parenthèses fermantes. De ce fait, elle est syntaxiquement incorrecte et il faut supprimer la seconde parenthèse ouvrante.

**De façon générale, il vous faudra toujours vérifier, quand vous utilisez des fonctions, que chaque parenthèse ouvrante est bien contrebalancée par une parenthèse fermante correspondante.**

### Non validation par UpyLaB

Voici quelques exemples de réponses non validées par UpyLaB pour l'exercice UpyLaB 2.2.

### 2.2d

```
a = float(input())
b = float(input())
res = (a + b) / 2
```

**Explication et conseil :** Ce code n'affiche rien !

### 2.2e

```
a = 2.0
b = 3.0
print((a + b) / 2)
```

**Explication et conseil :** Il est attendu de calculer la moyenne arithmétique de deux nombres saisis par l'utilisateur. Ces valeurs doivent être lues, grâce à des appels à la fonction `input`. Les valeurs 2.0 et 3.0 ne sont que des exemples.

### 2.2f

```
print((a + b) // 2)
```

**Explication et conseil :** Quel opérateur Python permet de calculer une division décimale ?

## 4.2.4 UpyLaB 2.3

BRUNO

Bonjour,

Je me mélange un peu les pinceaux entre *console* et *scratch*. Quand on fait `print('Bonjour')` dans la console ou dans un fichier scratch on obtient la même chose. Par contre, on ne peut pas faire d'opération (par exemple `2 + 2`) dans le fichier scratch.

### Réponse de l'équipe

Bonjour Bruno,

`2+2` est une expression arithmétique, qu'un interprète Python comprend... quand on la tape dans une console, l'interprète pour nous montrer qu'il sait faire ce calcul, nous affiche en retour la valeur de cette expression.

Quand on écrit un script, on va écrire une succession d'instructions. Si on tape juste une expression mais sans préciser ce qu'on veut en faire, Python n'en fait rien.

Ainsi,

avec un script qui contient :

```
2+2
```

Je peux l'exécuter, il n'y aura pas d'erreur mais il ne se passera rien. Je n'ai pas dit à l'interprète quoi faire avec ce `2+2`... Si je fais :

```
print(2+2)
```

Cette fois j'ai une instruction (l'appel à la fonction `print`) qui utilise cette expression `2+2`... et Python va m'afficher 4. J'espère que cela clarifie les choses pour vous, la différence entre **expression** et **instruction**.

### Réponse de BRUNO

Merci beaucoup. Je comprends mieux oui.

## Non validation par UpyLaB

Voici quelques exemples de codes parmi les réponses non validées par UpyLaB pour l'exercice UpyLaB 2.3.

### 2.3a

```
a = float(input(3.5))
```

**Explication et conseil :** Mauvaise utilisation de l'`input`. Ce code affiche 3.5 avant de demander une valeur en entrée. Donc soit le code soumis à UpyLaB demande d'entrer une valeur que l'on veut affecter à la variable `a` ce qui correspond à :

```
a = float(input())
```

soit le code veut simplement assigner à a une valeur fractionnaire valant 3.5 mais alors il s'agit d'une simple assignation :

```
a = 3.5
```

### 2.3b

```
print((a * b) / c)
```

**Explication et conseil :** Ceci ne correspond pas au calcul demandé dans l'énoncé UpyLaB 2.3 !

### 2.3c

```
print((b * c) // a)
```

**Explication et conseil :** Quel opérateur Python permet de calculer une division décimale ?

## 4.2.5 UpyLaB 2.4

LIONEL

Bonjour,

Sur l'exercice 2.4 mon code génère l'erreur suivante :

```
ImportError: No module named '_tkinter', please install the python3-tk
```

Comment la résoudre ?

### Réponse de l'équipe

Bonjour Lionel,

Je dirais simplement que UpyLaB ne demande pas et **n'autorise pas** turtle ni tk.

### Réponse de LIONEL

Effectivement j'ai bien relu l'énoncé et terminé l'exercice. Merci.

ALINE

**Bonjour,**

Est-ce qu'il y a une différence de précision de calcul sous unix et sous windows? Je ne comprends pas comment vous arrivez avec une précision différente entre `a*cos(60)` qui vaut 45.000000000000001 et `-a*cos(60)` qui vaut -44.999999999999998. Le résultat aurait été plus facile si vous aviez demandé un arrondi à n décimales.

**Réponse de l'équipe**

Bonjour ALINE,

Que vous écriviez votre code dans Upylab sous Windows ou sous Linux ne change pas le résultat, car le code est exécuté sur un serveur différent de votre ordinateur. L'exerciseur UpyLaB tient compte dans les calculs des erreurs de troncature et d'arrondi (donc 45.0 ou 45.000000000000001 seront jugés égaux par l'outil).

Par contre entre -77.942 et -77.94228634059945 la différence est trop importante et due à des calculs différents. Ici probablement l'utilisation de `round( , 3)` dans votre code.

Bonne continuation.

**KIM**

**Bonsoir,**

Je suis complètement bloqué à l'exercice UpyLaB 2.5. Je serai capable de créer un petit programme sans problème, mais je suis nul en géométrie. Je ne vois pas quelles formules permettent de donner les coordonnées des points par rapport à la longueur du rayon.

**Réponse de XAVIER**

Bonjour Kim,

Il faut effectivement se replonger dans le cercle trigonométrique... Vous pouvez aller sur Wikipédia (ou autre) et chercher « cercle trigonométrique ». On vous donne les valeurs remarquables en fonction de  $\pi$  (avec l'équivalence en degré). Par exemple,  $60^\circ$  correspond à  $\pi/3$  (puisque  $180^\circ$  c'est  $\pi$  !). Pour trouver les coordonnées d'un point : pour x, c'est le rayon \* cos de l'angle et pour y, le rayon \* sin de l'angle...

Bon courage !

**Réponse de KIM,**

J'ai résolu l'exercice ! Merci pour votre aide !!

### ARTHUR

Bonsoir,

Je n'arrive pas à afficher les deux coordonnées x, y sur une même ligne et séparées par une espace comme dans le résultat. Les miennes s'affichent les unes en dessous des autres. Y a-t-il une fonction particulière pour cela ?

#### Réponse de l'équipe

Bonjour Arthur,

Pour cela, écris les deux arguments dans le même appel à la fonction `print` :

```
print("hello", "world")
```

#### Réponse d'AMINE,

Il est mentionné dans la doc de la fonction `print` accessible depuis un interprète Python par `help(print)` qu'un séparateur qui est l'espace par défaut est intercalé entre les valeurs. On peut changer ce séparateur. Par exemple pour une virgule :

```
print(a, b, sep=',')
```

### MORGANE

Bonjour,

Je reviens vers vous car ma formule n'arrondit pas de la même façon alors comment faire.

L'appel à votre programme sur l'input ``90↔`` a renvoyé :

```
(90.0, 0.0) (45.000000000000001, 77.94228634059948) (-44.99999999999998, 77.94228634059948)
(-90.0, 0.0) (-45.000000000000001, -77.94228634059948) (-44.99999999999998, -77.94228634059948)
```

Le résultat attendu était :

```
90.0 0.0 45.000000000000001 77.94228634059948 -44.99999999999998 77.94228634059948
-90.0 0.0 -45.000000000000001 -77.94228634059948 45.000000000000001 -77.94228634059948
```

**Réponse de l'équipe,**

Le problème ne provient pas des arrondis, puisque UpyLaB est assez tolérant à leur égard. Mais ton programme affiche des parenthèses qui ne sont pas attendues. Essaie sans elles. Regarde la différence entre `print((1, 2))` et `print(1, 2)`.

**Astuce de l'équipe**

Pensez à donner à vos fonctions sinus et cosinus des angles en radian. Utilisez donc `math.cos(math.pi/3)` plutôt que `math.cos(60)`.

Par ailleurs,  $180^\circ = 1 \cdot \pi$  radian, donc  $240^\circ = 4/3 \cdot \pi$  radian. Donc  $300^\circ = ?$

Attention, les valeurs finales demandées doivent toutes être des float. Donc si vous devez afficher la valeur 0, il faut écrire 0.0 !

## 4.2.6 UpyLaB 2.5

LÉA

**Bonjour,**

On demande d'écrire un programme, qui lit deux valeurs entières x et y strictement positives. J'arrive à faire tout le programme sauf avoir des nombres strictements positifs. Quelqu'un aurait-il une idée ?

**Réponse de FABRICE**

Bonjour Léa,

À ce stade du cours, il ne nous est pas demandé de vérifier la validité des entrées. Nous devons simplement récupérer deux entiers grâce aux fonctions `input` et `int` en sachant qu'UpyLaB, au moment de tester notre programme, ne fournira que des entiers positifs.

**Réponse de Léa**

Merci Fabrice, c'est compris.

KIM

Bonjour,

Je suis coincé :

L'appel à votre programme sur l'input ``2↔7↔1.5↔25.0↔`` a renvoyé :

```
-5.0 3.5 26.5 3.0 1.0 0.25 1.6875 0.7071067811865476
```

Le résultat attendu était :

```
-5 3.5 26.5 3.0 1.0 0.25 1.6875 0.7071067811865476
```

Je ne comprends pas d'où peut venir l'erreur. Pourquoi dans le résultat attendu n'y a t'il aucun chiffre après la virgule contrairement aux autres ?

### Réponse de l'équipe

Bonjour,

Le type des résultats est important aussi. -5 et -5.0 ne sont pas de même type. Éventuellement revois la matière (int, float).

## Non validation par UpyLaB

Voici quelques exemples de codes parmi les réponses non validées par UpyLaB pour l'exercice UpyLaB 2.5.

### 2.5a Pour l'avant-dernière expression à imprimer

```
print ( (x+y) *z/4*x)
```

**Explication et conseil :** Qu'affiche l'instruction `print(8 / 4 * 2)` d'après vous ? Testez-le si vous pensez que le résultat est ``1 et revoyez la notion d'associativité des opérateurs.

### 2.5b Pour la première expression à imprimer

```
x = float(input())
y = float(input())
print(x - y)
```

**Explication et conseil :** Lisez toujours bien les consignes. En particulier, quel type doivent avoir les variables `x` et `y` ? De façon générale, quel type doit avoir chacun des résultats (int, float, ...)?

## 4.2.7 UpyLaB 2.6

AMINE



**Bonjour,**

Bonsoir, je n'arrive pas à comprendre la différence ; L'appel à votre programme a renvoyé :

```
Hello World
Aujourd'hui
C'est "Dommage!"
Hum \o/
```

Le résultat attendu était :

```
Hello World
Aujourd'hui
C'est "Dommage!"
Hum \o/
```

Il ne me semble pas y avoir de problème d'espace. ... pourtant mon programme n'est pas validé.

**Réponse de l'équipe**

Bonjour Amine,

Tu n'as pas laissé d'espace à la fin ?

**Réponse d'Amine**

Merci, c'est un classique mais je ne le voyais pas : à la ligne 3 entre les deux " " finaux. Cette fois cela fonctionne.

**XAVIER**

**Bonjour,**

Je n'arrive pas du tout à écrire la 3e ligne :

```
C'est "Dommage!"
```

Cela doit être avec les " " mais si je remplace par des ' ' ça bloque aussi puisqu'il apparaît dans "c'est ". J'ai bien vérifié les espaces, je ne vois vraiment pas ce qui ne va pas ! Comment faire ?

**Réponse de l'équipe**

Bonjour Xavier,

La difficulté vient bien du fait qu'il faut afficher à la fois une apostrophe et des guillemets. Comment peut-on

faire pour afficher un caractère spécial sans que Python l'interprète comme un élément de la syntaxe ?  
Bon débogage.

### Non validation par UpyLaB

Voici quelques exemples de morceaux de codes parmi les réponses non validées par UpyLaB pour l'exercice UpyLaB 2.6.

#### 2.6a

```
print('Hello World ')\nprint(" Aujourd'hui")\nprint(" C'est \"Dommage !\" ") \nprint(" Hum \\o/ ")
```

**Explication et conseil :** Une espace en trop ou en moins peut facilement apparaître dans de tels textes. Il faut être méticuleux, ou si possible faire des copier / coller à partir de l'énoncé pour éviter les erreurs de recopiage. Par ailleurs, les éléments nécessaires pour résoudre ce problème (ici afficher du texte avec des caractères particuliers) peuvent souvent être trouvés dans les explications du cours. Ici, en section 2.7.1, des explications sur les caractères d'échappement aident à la réalisation de l'exercice.

### 4.2.8 UpyLaB 2.7

#### JULIE

Bonjour,

J'obtiens des erreurs dues aux arrondis sur le dernier chiffre. J'ai essayé la fonction `round` sans succès.

#### Réponse de l'équipe

Bonjour Julie,

Dans cet exercice, il ne faut pas utiliser la fonction `round`. D'autre part n'oublie pas de bien utiliser la fonction `float()` avec le `input()`.

#### Réponse de JULIE

Merci beaucoup.

### Non validation par UpyLaB

Voici un exemple de morceau de code parmi les réponses non validées par UpyLaB pour l'exercice UpyLaB 2.7.

**2.7a**

```
...  
print(4/3*3.14*rayon**3)
```

**Explication et conseil :** Même si ce n'est pas explicitement indiqué dans les consignes, utilisez toujours la valeur de `pi` la plus précise. Importez cette valeur du module `math`.



## Foire aux questions du Module 3

### 5.1 FAQ générale du Module 3

Cette section comprend les questions posées et réponses au sujet de la matière abordée dans le module 3 du cours.

### 5.2 UpyLaB du Module 3

#### 5.2.1 UpyLaB 3.1

**ARTHUR**

**Bonjour,**

Dans l'exercice 3.1 je suis bloqué : mon code soumis donne cette erreur :

L'appel à votre programme sur l'input  $3 \leftrightarrow 5 \leftrightarrow 4$  a renvoyé :

Le résultat attendu était : Aucun print

**Réponse de l'équipe**

Le problème est que `print ( " " )` ne va rien afficher de visible, mais tout de même ajouter une ligne vide. Ce qui n'est pas ce qui est attendu ici.

Reprends la consigne : il n'y a rien à imprimer dans certains cas, pourquoi vouloir utiliser print alors ?

### Réponse d'Arthur

Donc je laisse vide ma section `else...` mais cette fois j'ai une erreur de syntaxe. Si j'écris `continue` j'ai aussi des erreurs.

### Réponse de l'équipe

As-tu vraiment besoin d'une clause `else` ici ?

### Réponse d'Arthur

Effectivement, j'ai retiré le `else` et tout va bien. Merci.

## OCEANE

Bonjour,

Pour l'exercice 3.1, UpyLaB valide certains tests mais pas d'autres ; par exemple :

L'appel à votre programme sur l'input `3↔3↔3` a renvoyé :

`3 3 3`

Le résultat attendu était :

`3`

### Réponse de l'équipe

Bonjour Océane,

Dès que ton code affiche une valeur, il doit s'arrêter. Je te conseille de regarder comment fonctionne la partie `elif` d'une instruction `if` ; cela pourra sûrement t'aider.

### Réponse d'Océane

ha oui, j'ai compris mon erreur. Merci !

## ALI

**Bonjour,**

Pour l'exercice 3.1, UpyLaB valide certains tests mais pas d'autres ; par exemple :

L'appel à votre programme sur l'input  $3 \leftrightarrow 4 \leftrightarrow 3$  a renvoyé :

Le résultat attendu était :

3

**Réponse de l'équipe**

Bonjour Ali,

Clairement ton code ne teste pas tous les cas. Pourquoi n'utilises-tu pas Python Tutor pour voir ce qui se passe avec les données 3, 4 et 3 ?

**Réponse Ali**

ok je vais faire cela, merci !

**LOUISE**

**Bonjour,**

Lors de la correction de l'exercice UpyLaB 3.1, tous les résultats sont bons sauf celui ci-dessous. Je fais comment ? Mon résultat est celui attendu, mais ce n'est pas validé.

L'appel à votre programme sur l'input ``20`` a renvoyé :

Aucun print

Le résultat attendu était :

Aucun print

**Réponse de l'équipe**

Bonjour Louise,

Quand UpyLaB écrit qu'il ne faut aucun print, cela signifie que ton code ne doit pas faire de print, alors qu'ici ton code imprime le texte "Aucun print".

## 5.2.2 UpyLaB 3.2

LOUIS

**Bonjour,**

Lors de la correction de l'exercice UpyLaB 3.2, tous les résultats sont bons sauf celui ci-dessous. Je fais comment ? Mon résultat est celui attendu, mais ce n'est pas validé.

```
L'appel à votre programme sur l'input ``20`` a renvoyé:
```

```
Aucun résultat
```

```
Le résultat attendu était :
```

```
Aucun résultat
```

**Réponse de l'équipe**

Bonjour Louis,

Quand UpyLaB écrit qu'il ne faut aucun résultat, cela signifie pas de print, alors que ton code imprime le texte "Aucun résultat".

SEVERINE

**Bonjour,**

Lors de la correction de l'exercice UpyLaB 3.2, tous les résultats sont bons sauf celui ci-dessous.

```
L'appel à votre programme sur l'input "0" n'a rien renvoyé.
```

```
Le résultat attendu était :
```

```
Il va faire froid
```

**Réponse de l'équipe**

Bonjour Séverine,

Pour trouver le code qui solutionne un problème, tous les cas doivent être analysés. La plupart des problèmes ont leurs cas limites. L'exemple ici illustre parfaitement cela. Que faut-il faire quand la température est de zéro (0) degré ?

Bon débogage



### 5.2.3 UpyLaB 3.3

AGATHE

Bonjour,

Dans Pycharm et UpyLaB, les lignes suivantes de mon code :

```
print(a.b)
print(a^2+a . b)
```

provoquent une erreur :

```
int object has no attribute b
```

Je ne comprends pas.

#### Réponse de l'équipe

Attention Agathe, la multiplication et la puissance ne se notent pas comme cela en Python. Peut-être devrais-tu revoir le cours sur les opérateurs arithmétiques.

SAUL

Bonjour,

UpyLaB ne valide pas le cas où  $c$  a une valeur négative.

Que puis-je faire ?

#### Réponse de l'équipe

Bonjour Saul,

Tu dois bien relire l'énoncé et analyser les différents cas possibles. Que doit faire ton code si  $c$  a une valeur négative ?

Après avoir bien compris les différents cas possibles, tester ton code sur une valeur correspondant à chacun des cas, en utilisant Python Tutor, pourra t'aider

Bon débogage

### 5.2.4 UpyLaB 3.4

NASIA

**Bonjour à l'équipe,**

Pour certaines valeurs, UpyLaB ne valide pas mon code. Pourtant tout semble identique.

L'appel à votre programme sur l'input 37 a renvoyé :

False

Le résultat attendu était :

False

**Réponse de l'équipe**

Bonjour Nasia,

L'exercice demande d'afficher le résultat d'un test. Mais toi tu affiches le texte « False » probablement suivi d'un ou plusieurs caractères espaces.

Enlève les espaces ou mieux fait le print du résultat du test (dont les valeurs possibles sont `True` ou `False`).

### 5.2.5 UpyLaB 3.5

VINCENT

**Bonjour,**

Ma vérification fonctionne lorsque les données a et b ne sont pas divisibles entièrement. Mais la réponse est `True` même si les chiffres sont divisibles entièrement. La réponse `False` n'apparaît jamais. Il semble que les 2 dernières lignes du programme ne sont jamais atteintes. Quelqu'un peut-il me dire pourquoi svp ?

**Réponse de Marc**

Bonjour Vincent,

Si tu utilises un `or` c'est normal, car alors dès qu'un des deux nombres divise l'autre le résultat sera `True`. Il faut bien faire attention à la règle : `not (A or B)` est équivalent à `not A and not B`.

### 5.2.6 UpyLaB 3.6

SOPHIE

**Bonjour,**

J'ai un problème sous UpyLaB lorsque je teste ma réponse, il me met ce message d'erreur alors que tout est ok sous Python Tutor :

```
L'appel à votre programme sur l'input ``1.0↔2.0↔`` a été interrompu par une  
↳exception:  
Ligne 2: invalid literal for int() with base 10: '1.0
```

Pourriez-vous m'aider. Bonne journée !

### Réponse de l'équipe

Bonjour Sophie,

Dans cet exercice, les données ne sont pas forcément des entiers. Si tu essaies de transformer en `int` une chaîne comme `'1.0'` alors il se produit une exception. Il faut donc utiliser la fonction `float()` ici.

## 5.2.7 UpyLaB 3.7

**SALIM**

**Bonjour,**

Dans cet exercice, on parle de mise, de couleur, de numéro. Je m'embrouille avec toutes ces valeurs et je ne comprends plus le sujet. Déjà que je n'y connais rien en jeu de hasard, alors en plus avec un programme dans un jeu de hasard c'est incompréhensible. Pouvez-vous m'aidez ? Merci.

### Réponse de l'équipe

Bonjour Salim,

Ton programme commence par récupérer deux valeurs entières. L'une correspond au pari du joueur : un nombre entre 0 et 12 pour dire que le joueur a parié que ce nombre sortirait au tirage ou alors 13 pour dire que c'est un nombre impair qui sortirait, ou bien 14 pour un nombre pair, ou bien 15 pour dire que c'est un nombre rouge (voir le dessin du tapis dans l'énoncé pour voir quels sont les nombres rouges) ou encore 16 pour un nombre noir. L'autre valeur correspond au tirage : un nombre entre 0 et 12.

Ensuite il te faut faire des tests pour voir combien le joueur va remporter. Les trois possibilités étant : 120 euros si le joueur a parié sur le numéro sorti au tirage, 20 euros s'il a trouvé la couleur ou la parité et 0 dans tous les autres cas.

### 5.2.8 UpyLaB 3.8

LISE

**Bonjour,**

Je n'arrive pas à réaliser l'exercice 3.8 sur les figures géométriques avec les lettres. En effet, dans les `x = float(input())` il est possible de mettre des chiffres mais je ne vois pas comment le programme peut lire des lettres majuscules. J'aurais besoin d'un peu d'aide et d'explications. Merci d'avance.

**Réponse de l'équipe**

Bonjour Lise,

La fonction `input` va en fait lire une chaîne de caractères depuis l'entrée. Si cette chaîne est un nombre, on a donc besoin de la convertir en nombre en utilisant les fonctions `int` ou `float` selon le cas.

Si on veut lire des caractères, il n'y a rien à convertir et on utilise la fonction `input` seule. Essaie donc avec `x = input()`.

Bon courage !

### 5.2.9 UpyLaB 3.10

KARIN

**Bonjour,**

Comment puis-je déterminer le nombre de valeurs totales pour ensuite calculer la moyenne ?

**Réponse de l'équipe**

On ne connaît pas ce nombre à l'avance, mais on peut compter les valeurs au fur et à mesure qu'on les lit. Utiliser des variables supplémentaires peut être utile du coup.

Bon codage !

JÉRÔME

**Bonjour,**

En ce qui me concerne j'ai un peu `_filouté_` pour enlever de la somme et du nombre d'entreprises la valeur

sentinelle. Après la boucle, j'ai écrit une instruction qui dit que si l'input est égal à -1 alors il faut augmenter le nombre d'employés de 1 et réduire le nombre d'entreprises de 1.

Mais il y a certainement une solution beaucoup plus propre à cela pour faire que la valeur sentinelle ne soit pas prise en compte !

### Réponse de l'équipe,

Bonjour Jérôme,

Effectivement le canevas suivant pourrait t'aider :

```
x = int(input())
while _une_condition_sur_x:
    #
    # faire un traitement...
    #
    x = int(input())
res = ...
```

## 5.2.10 UpyLaB 3.11

**SALIM**

**Bonjour,**

une personne pourrait me donner un coup de pousse sur l'exercice 3.11 ? Je n'arrive pas à aller à la ligne pour afficher les XXXX, merci d'avance.

### Réponse de l'équipe

Bonjour Salim, Si tu utilises plusieurs appels à la fonction print, les affichages correspondants se feront chacun sur leur propre ligne. Tu peux aussi ajouter `\n` dans la chaîne de caractères à afficher à l'endroit où tu veux insérer le saut de ligne. Bon courage !

**SOPHIE**

**Bonjour ;**

je ne vois pas la différence entre mon résultat et le résultat d'UpyLab. Pourtant UpyLab considère mon résultat comme faux. Merci pour votre aide.

### Réponse de l'équipe

Bonjour Sophie,

Vérifie que tu n'affiches pas des espaces superflues en fin de ligne, ou une éventuelle dernière ligne vide.  
Bon courage !

## 5.2.11 UpyLaB 3.12

CHLOÉ

Bonjour,

J'ai la même figure, mais décalée d'un cran sur la droite, comme si j'avais une espace en trop à chaque ligne.

### Réponse de Karin

La fonction `print()` intercale une espace par défaut entre chaque valeur affichée et un retour à la ligne à la fin. Les premières fois, je te conseille de remplacer les espaces que tu veux insérer devant tes 'X' par un autre caractère pour mieux visualiser les éventuelles espaces superflues.

### Réponse d'Albert

Merci à vous. Je ne trouvais pas mon problème. Je me disais que pour afficher n espaces avant mon 'X' je pouvais utiliser

```
print(' ' * n, 'X')
```

En me disant que si n vaut 0 j'afficherai zéro espace. Mais cela m'affichait quand même une espace, je viens de comprendre qu'en fait le `print` met une espace avant ma deuxième valeur. Et j'ai réussi en faisant :

```
print(' ' * n + 'X')
```

Une autre façon est d'explicitier que dans le `print`, le paramètre `sep` qui donne la chaîne de caractères à mettre entre chaque valeur affichée, est la chaîne vide :

```
print(' ' * n, 'X', sep='')
```

JÉRÔME

**Bonjour,**

UpyLab me dit que les réponses sont fausses alors que celles qu'il me propose sont toutes identiques aux miennes ! Mon programme fonctionne comme il faut dans mon interpréteur Python.

### Réponse de l'équipe

Bonjour Jérôme,

Vérifie bien que ton programme n'affiche pas une ligne supplémentaire à la fin. Pour cela, une petite astuce. Ajoute une dernière instruction `print("A")`. Si ce "A" s'accroche à la dernière ligne de "X", pas de problème. Il faut aussi vérifier le nombre d'espaces et vérifier qu'il n'y a pas une espace supplémentaire à chaque ligne. Cette espace superflue pourrait être due à la fonction `print`. En effet, si on passe plusieurs arguments à cette fonction, ceux-ci seront par défaut séparés par une espace. On peut modifier ce comportement en précisant une valeur pour le paramètre `sep` de la fonction :

```
print("Bon", "jour", sep = "")
```

ou éviter de passer deux arguments à la fonction en concaténant les chaînes de caractères :

```
print("Bon" + "jour")
```

Bon courage !

**ALINE**

**Bonjour,**

Est-ce qu'il est possible d'être un peu aidée sur la conception de l'exercice 3.13 ?.

J'ai compris qu'il fallait utiliser la concaténation entre les deux signes mais je n'ai pas réussi à voir comment mettre ça forme.

Merci à vous.

### Réponse de l'équipe

Bonjour Aline,

L'idée consiste effectivement à concaténer une chaîne formée d'espaces et une chaîne formée de 'X'. Il suffit maintenant de voir combien d'espaces et combien de 'X' par ligne. N'oublie pas que l'on peut utiliser l'opérateur `*` pour afficher plusieurs fois la même chaîne. " " \* 3 va afficher 3 espaces consécutives par exemple.

Bon courage !

## 5.2.12 UpyLaB 3.13

MORGANE

**Bonsoir,**

personnellement, je ne comprends pas l'exemple du nombre négatif. Comment savoir combien de data il doit y avoir ? Dans l'exemple avec -1, il y a 5 termes de somme. Je ne comprends vraiment pas comment sont générés les data pour ce cas. Quelqu'un a une piste pour m'aiguiller ?

**Réponse de l'équipe**

Bonjour Morgane,

C'est justement le point de cet exercice, nous ne connaissons pas à l'avance le nombre de données à lire. Ce que tu as pu faire dans l'exercice 3.10 devrait t'aider pour ce cas-là.

Bon courage !

**Réponse de Morgane**

Merci pour votre aide, cela m'a bien aidé. Maintenant je bute sur le problème de l'erreur à cause du F qui n'est pas un nombre entier !! Je cherche et je vais trouver !! Merci

**Réponses de l'équipe**

Bonjour Morgane,

L'idée pourrait être de ne chercher à convertir l'entrée en `int` qu'à partir du moment où on est sûr qu'il ne s'agit pas de la lettre 'F'.

Bon courage !

**Réponse de Morgane**

J'ai bien fini par trouver. Merci pour votre aide.

AMINE

**Bonjour,**

Sur le programme, j'ai également une erreur avec la chaîne de caractères F, je n'arrive pas à voir comment



je peux convertir la chaîne de caractères. Voici ci-dessous le message d'erreur d'UpyLaB.

```
L'appel à votre programme sur l'input ``-1↔8↔F↔`` a été interrompu par une  
↳exception:  
Ligne 13: ValueError: invalid literal for int() with base 10: 'F'
```

Merci pour votre aide.

### Réponse de l'équipe

Bonjour Amine,

En fait, il suffit de ne pas le convertir en `int`

Lorsqu'on appelle la fonction `input`, celle-ci retourne ce qui est lu sous la forme d'une chaîne de caractères quel que soit le type d'entrée, numérique ou non. Ici, puisque l'entrée peut ne pas être un nombre, il faut utiliser `input` seule, puis tester si la valeur lue est différente de 'F' avant d'effectuer la conversion en entier avec `int`.

J'espère que c'est assez clair. Bon courage !

**XAVIER**

Bonjour a tous,

Je rencontre un problème avec cet exercice dans le cas où la valeur est négative. Ce qui me pose un gros problème est la lettre F, je n'arrive pas à faire de test avec ; j'ai essayé avec comme conseillé (ex : `data = input()`), ensuite tester si `data == 'F'` mais à chaque fois j'ai un problème de type de data. Quelqu'un pourrait-il me mettre sur la voie ?

Merci d'avance !!

### Réponses de l'équipe

Bonjour Xavier,

Vérifie ton code au niveau de la conversion des valeurs que tu reçois. N'oublie pas le fait que `input()` renvoie une valeur de type `str`, et qu'une conversion de ce type vers un entier dans les fonctions `int()` ou `float()` ne peut aboutir que si la valeur entre paramètre correspond à celle du type vers laquelle on effectue la conversion. Par exemple : `data = int("F")`, produira une erreur.

### Réponse de Xavier

Merci à vous !

Grâce à vos commentaires j'ai compris ; en fait ce qu'il me manquait c'était la bonne syntaxe pour trans-

former mon `data = input()` en `int` pour pouvoir additionner ensuite. `data = int(data)` était la solution pour moi !

Merci !

**KIM**

**Bonjour,**

Je rencontre des difficultés, notamment la partie avec la fin de commande "F". J'ai l'erreur :

```
TypeError: can only concatenate str (not "int") to str
```

Lorsque je fais `data = input()` dans la boucle, il prend bien les valeurs numériques et la lettre F mais je n'arrive pas à additionner les nombres rentrés pour obtenir leur somme sans obtenir le message. Et inversement si je fais un `int(input)`.

Merci d'avance pour vos lumières, en espérant avoir été clair sur mon problème sans divulguer mon code.

**Réponse de l'équipe**

Bonjour Kim,

L'opérateur `+` va opérer différemment selon le type des opérandes. Si tu « ajoutes » deux chaînes de caractères, il va les concaténer.

Si tu « ajoutes » deux nombres, il va retourner leur somme.

Le problème, c'est que tu ne peux pas mélanger. Ajouter une chaîne de caractères et un nombre n'a pas de sens. La difficulté ici, c'est qu'on manipule deux types d'entrées, des nombres et un caractère. C'est pourquoi l'indice propose d'utiliser la fonction `int()` dans le cas où l'entrée est différente de 'F'.

Bon courage !

**Réponse de Xavier**

Merci pour votre réponse. Cela m'a permis de débloquent cet exercice. Grace à vos conseils et en utilisant Python Tutor, j'ai remarqué que je n'arrive pas à passer les `data` en `int`. Tout simplement parce que j'écrivais dans mon code `int(data)` au lieu de `data = int(data)`.

Erreur de débutant.

**ARTHUR**

**Bonjour,**

Cet exercice me donne aussi du fil à retordre. Avec l'instruction `while data == 'F'` : je n'arrive pas

à rentrer dans la boucle et je ne vois pas trop comment faire autrement . Merci d'avance pour votre aide :o)

### Réponse de l'équipe

Bonjour Arthur,

Avec la condition `data == 'F'`, le programme n'entrera dans la boucle que si `data` contient la valeur 'F'. Ne serait-ce pas plutôt le contraire que tu voudrais faire, étant donné que 'F' est la dernière valeur à lire ?

### Réponse d'Arthur

Merci je vais essayer ça parce que pour le moment je n'ai réussi qu'en utilisant `while True : et break`

**KARIN**

Bonjour,

Je bloque sur la condition `data == 'F'` pour terminer le calcul. Mon programme fait bien la somme de tous les nombres que je saisis mais pour terminer la boucle, si j'écris simplement `F`, il me retourne un message d'erreur. Si j'écris `'F'` le programme fonctionne mais ce n'est pas ce qui demandé. Le problème doit se situer au niveau du type de la variable `data`, mais je ne comprends pas (j'ai essayé d'ajouter `data = str(input())` par exemple) . Je comprends d'autant moins que dans Python Tutor mon programme se termine sans problème en écrivant `F` directement. Si vous avez des pistes.

Merci

### Réponse de l'équipe

Bonjour Karin,

vosre programme doit lire une donnée `n` et tester si elle vaut 'F' ou sinon traduire en `int` et faire des calculs et recommencer

### Réponse de Karin

oui, c'est bien ce que j'essaye de faire. Cependant le programme doit utiliser la même variable `n` aussi bien pour les nombre à saisir que pour le 'F'. J'utilise `n = input()`, ça fonctionne bien pour tous les nombres mais lorsque je saisis `F` (sans guillemets) le programme plante, il ne semble pas reconnaître le type string de la variable.

### Réponse de l'équipe

Bonjour Karin,

`n = input()` va fonctionner aussi bien pour des entrées numériques que des lettres. Dans tous les cas, cette fonction retourne un objet chaîne de caractères. Le souci doit plutôt provenir au moment où tu souhaites convertir cette valeur en `int`, c'est-à-dire au moment où tu utilises la fonction `int()`. Cela ne sera pas possible si `n` est égal à la lettre 'F' et c'est pour cela qu'il faut d'abord tester ceci avant de convertir la chaîne lue en nombre et l'ajouter à la somme.

Bon courage !

### JÉRÔME

Bonjour,

Dans l'exercice, j'aimerais tester si la valeur entrée correspond à la lettre F. J'essaie, en vain : `if x = "F" if x = F` Et ça bloque. Quelqu'un peut-il m'indiquer la syntaxe exacte ? Merci.

### Réponse de l'équipe

Bonjour Jérôme,

Le test est `if x == 'F' :`

### ALBERT

Bonjour,

Sur l'exercice, je bloque avec l'histoire du F. Voici mon erreur : Je ne saisis pas les types de données. Comment faire comprendre à la machine que soudainement je n'ai plus de nombre mais une lettre ? Doit-on jouer avec `int()` et `str()` en `input` ? Comment dire tant que je n'ai pas une lettre en `input`, j'additionne ? merci

### Réponse de l'équipe

Bonjour, Tu as parfaitement raison Albert. La plupart du temps, nous castons au moment de lire la donnée  
`data = int(data)` est possible.

### Réponse d'Albert

rhaaa ! je pensais que la nuit m'aiderait à trouver la solution. Et bien non ! je bloque toujours. J'y suis

presque, mais je m'embrouille, avec le `while`, le `if`, le `not` et ce fichu `F` ! Un indice supplémentaire ?

### Réponse de l'équipe

Bonjour Alert,

Ne cherche pas à convertir la valeur lue en `int` avant d'avoir vérifié que ce n'est pas la lettre '`F`'. En gros, lis l'entrée avec `input()`, teste si c'est différent de '`F`' et alors seulement convertis en `int`.

Bon courage !

### Réponse d'Albert

**Je crois que mon souci vient du fait que je n'arrivais pas à tester** juste ce `F`. Subitement ça a fonctionné.

Des fois, la chance. Merci !

ALINE

Bonsoir,

Je suis revenue sur l'exercice et je suppose que les données d'entrée ne sont pas une liste. Dommage ce serait plus facile. Je bute sur la déclaration du '`F`' en donnée d'entrée. J'ai mis une entrée `data = input()` sans déclarer le type de données. Mais dans la suite je dois déclarer la condition complémentaire du type de `data`. Or Python ne comprend pas `type(data) = int()` ou `data = int()` ou `input = int()`. J'ai à peu près tout essayé mais il y a un truc qui m'échappe ou que j'ai raté dans la déclaration du type de donnée. Comment puis-je ajouter une condition d'être ou ne pas être `int` ou `str`. Je suis bloquée.

Merci

### Réponse de l'équipe

Il ne s'agit pas de déclarer quoi que ce soit. Il s'agit de simplement récupérer une donnée. si cette donnée est un '`F`' c'est qu'on a terminé sinon c'est que c'est quelque chose qui peut se transformer en entier et s'ajouter avec ce qu'on a déjà;-)

### Réponse d'Aline

Merci, j'ai compris.

### 5.2.13 UpyLaB 3.14

AMINE

Bonjour,

J'ai ça comme message d'erreur :

```
L'appel à votre programme sur l'input ``50↔24↔11↔17↔20↔18↔`` avec les_
↪arguments ``100`` a renvoyé:
```

```
Trop grand
Trop grand
Trop petit
Trop petit
Trop grand
Perdu !
Le secret était 18.
```

Le résultat attendu était :

```
Trop grand
Trop grand
Trop petit
Trop petit
Trop grand
Gagné en 6 essais !
```

Comment se fait-il que le programme fasse la vérification sur 100 et en même temps que le résultat attendu soit 18 ? Merci d'avance.

#### Réponse de l'équipe

Bonjour Amine,

L'argument avec lequel UpyLaB teste notre code n'est pas le nombre secret à deviner. C'est la valeur qu'il passe à la fonction `random.seed()` et qui fait que les fonctions du module `random` vont générer la même suite de nombres « aléatoires » à chaque fois.

SABINE

Bonjour,

J'ai un peu de mal avec cet exercice. Il semblerait que l'initialisation de la valeur aléatoire pose problème. Le résultat obtenu n'est jamais celui attendu.

Comment fait-on pour intégrer argument dans notre code ?

### Réponse de l'équipe

Bonjour Sabine,

Nous n'avons pas à intégrer `argument` dans notre code, c'est UpyLaB qui utilise cette variable pour initialiser différents tests qu'il fait subir à notre programme.

Si tu veux tester ton programme dans PyCharm dans les mêmes conditions, ajoute l'instruction `random.seed(argument)` en remplaçant `argument` par les valeurs utilisées par UpyLaB soit 5 puis 32 puis 27...

Mais n'oublie pas de retirer cette instruction avant de soumettre ton code à UpyLaB.

XAVIER

Bonsoir,

Pour réaliser l'exercice j'ai été « obligé » d'utiliser un `BREAK` dans un de mes `if`; est-ce une mauvaise pratique ?

### Réponse de l'équipe

Bonsoir Xavier,

Dans l'absolu utiliser un `break` n'est pas forcément mauvais. Toutefois, dans le cadre d'une initiation, il est préférable de s'en passer. La boucle `for` n'est peut-être pas la plus adaptée ici. Pourquoi ne pas essayer le canevas suivant :

```
on_continue = True
while on_continue:
    if ...
        on_continue = False
    ...
```

## 5.2.14 UpyLaB 3.15

MARINE

Bonjour,

J'ai un problème avec l'exercice de l'écureuil. Quand ce-dernier atteint la cible, le programme affiche la dernière valeur alors que la règle veut que le programme imprime « Cible atteinte » à la place.

**Réponse de Lise**

Bonjour MARINE,

Tu pourrais peut-être envisager de tester la valeur courante **avant** de procéder à son affichage.

**Réponse de Marine**

Ca marche ! Merci beaucoup.

**TOM**

**Bonjour,**

J'ai quelques problèmes concernant cet exercice. Mon programme fonctionne pour saut = 9 ; position\_cible = 7 et pour saut = 8 ; position\_cible = 7.

Mais pour (10;20), (50;66), (14,4), (20;6) et (24,6) mon programme ne marche pas. On est pas censé arrêter le programme dans tous les cas si on dépasse au maximum deux fois le barreau 0 ?

**Réponse de l'équipe**

L'idée est que l'écureuil peut faire autant de tours qu'il veut, mais s'arrêtera quand il revient en 0 **ou** atteint la cible.

**Réponse de Tom**

Ah d'accord, je n'avais pas compris cela. C'est validé maintenant.

**ARTHUR**

**Bonsoir,**

J'ai des impressions au dessus de 100 alors que j'utilise modulo.

**Réponse de Luc**

Bonsoir Arthur. Le modulo étant prioritaire sur l'addition, il faut penser à utiliser des parenthèses :

```
(valeur + increment) % 100
```



**Réponse de Arthur**

Merci. Ca fonctionne.

### 5.2.15 UpyLaB 3.17

LUCIE

Bonjour,

Dans cet exercice, je ne comprends pas le calcul à faire effectuer par python. Je comprends qu'il faut récupérer avec une boucle les valeurs impaires des factorielles et leur calcul. Mais je ne comprends pas cette notion de constante epsilon. Merci.

**Réponse de l'équipe**

On pourrait continuer indéfiniment le calcul des différents termes de la série. Ici, on nous demande de les arrêter, et de donner une approximation de la valeur de  $\sin(x)$  lorsque le terme à ajouter a une valeur suffisamment petite (il doit être inférieur en valeur absolue à cette constante donnée dans l'énoncé).

**Réponse de Lucie**

D'accord merci pour cette précision.

### 5.2.16 UpyLaB 3.18

MARC

Bonjour,

Je suis totalement coincé sur cet exercice : je ne sais pas du tout comment l'aborder.

**Réponse de l'équipe**

Bonjour Marc

Une idée, possible, est de décomposer la construction d'une ligne, numéro  $i$  : un certain nombre d'espaces puis des chiffres en miroir autour d'un chiffre.



---

### Foire aux questions du Module 4

---

#### 6.1 FAQ générale du Module 4

Cette section comprend les questions posées et réponses au sujet de la matière abordée dans le module 4 du cours.

#### 6.2 UpyLaB du Module 4

##### 6.2.1 UpyLaB 4.1

**XAVIER**

**Bonjour**

J'ai bien compris qu'il fallait renvoyer le booléen `True` ou `False` mais j'ai dû zapper un truc, car depuis le début des cours, lorsqu'on me demande d'imprimer `True` ou `False`, j'envoie la chaîne de caractères et ça marche... !? Alors quelle est la commande pour retourner le booléen ? Merci.

**Réponse de l'équipe**

Bonjour Xavier,

`True` et `'True'` sont deux objets différents : le premier est un booléen et le second une chaîne de caractères. De même que `1` et `'1'` sont deux objets différents (le premier est un entier). Et si dans chacun de ces cas le *print* imprimera la même chose à l'écran, pour une fonction qui fait un *return* les choses sont bien différentes.

Pour une fonction booléenne, il faudra faire `return True` ou `return False`.

## 6.2.2 UpyLaB 4.3

VALÉRIE

Bonjour,

Je ne comprends rien à cet exercice.

### Réponse de l'équipe

Bonjour Valérie,

Dans un premier temps il s'agit d'écrire une fonction `premier` qui retourne `True` si le nombre qu'on lui passe est premier. Pour tester si un nombre est premier, il faut vérifier qu'aucun des nombres compris entre 2 et lui-même n'est un diviseur de ce nombre. En fait, on peut se contenter de le vérifier pour les nombres inférieurs ou égaux à la racine carrée du nombre.

On peut donc envisager d'utiliser une boucle et regarder si le reste de la division euclidienne est égal à 0.

Dans un second temps, il faut écrire un programme qui lit un entier `x` puis affiche tous les nombres premiers strictement inférieurs à `x`.

### Réponse de VALÉRIE

Merci, je vois mieux le but à atteindre.

MAËL

Bonsoir,

Je ne comprends pas pourquoi UpyLaB me répond cela. Pourriez-vous m'éclairer ?

```
L'entête de la fonction premier est correctement défini fail L'appel à votre_
↪programme
sur l'input ``39`` a renvoyé:

False False 2 3 False 5 False 7 False False False 11 False 13 False False_
↪False 17 False 19 False False False 23 False False False False False 29_
↪False 31 False False False False False 37 False

Le résultat attendu était :

2 3 5 7 11 13 17 19 23 29 31 37
```

### Réponse de l'équipe

Bonjour Maël,

Manifestement, ton programme affiche `False` pour les nombres qui ne sont pas premiers. Or il faudrait ne rien afficher pour ceux-ci.

L'idée est de parcourir les entiers inférieurs au nombre `n`, et tester pour chacun d'eux s'il est premier en utilisant la fonction `premier` qu'on demande d'écrire. Si le résultat est `True`, alors on imprime le nombre. Mais on ne fait rien si le résultat est `False`.

### Réponse de MAËL

Merci, cela fonctionne maintenant.

## 6.2.3 UpyLaB 4.4

OLIVIER

Bonjour,

Je rencontre deux soucis : un `None` qui se glisse à la fin de ma suite de Fibonacci et UpyLaB qui me dit que ma fonction ne retourne pas le bon résultat :

L'appel à votre programme sur l'input ``8`` a retourné:

```
0
1
1
2
3
5
8
13
21
```

None

fibonacci(12) n'a pas retourné 144

### Réponse de l'équipe

Bonjour Olivier,

Vérifiez bien que votre fonction `fibonacci` effectue un `return` du résultat et non un simple `print`

### Réponse d'OLIVIER

Le problème était bien celui là. Merci.

## CHLOÉ

### Bonjour

J'ai un problème avec le UpyLaB 4.3. Je teste un code pour lequel tous les tests sont ok sauf le premier, pour lequel le commentaire est le suivant :

L'appel à votre programme sur l'input ``17`` a renvoyé:  
0 1 1 2 3 5 8 13 21 34 55 89 144 233 377 610 987 1597

Le résultat attendu était :

0 1 1 2 3 5 8 13 21 34 55 89 144 233 377 610 987

Sauf que, quelques lignes plus loin (sur le même code, donc) le même input est apparemment testé avec succès :

fibonacci(17) a retourné 1597

**Réponse d'Arthur,**

Bonjour Chloé,

Il faut se rappeler que `for i in range (n)` commence à 0 et termine à n-1. Ça peut peut-être aider...

**Réponse de Chloé,**

C'est bien ce que j'ai dans mon code, Arthur. Mon problème, comme je l'expliquais initialement, est que UpyLaB semble attendre deux valeurs différentes pour le même input.

**Réponse de l'équipe**

Bonjour Chloé,

Les tests ne testent pas tous la même chose. Le premier teste notre code en entier, donc avec l'affichage de tous les nombres de la suite *fibonacci*(*i*) pour *i* allant de 0 à *x* **exclu**.

Les autres tests ne vérifient que la fonction. Si ces tests sont validés, c'est que ta fonction semble faire ce qui est attendu d'elle.

Le souci est donc dans le nombre d'affichages que tu fais dans le programme complet. Et la remarque d'Arthur prend tout son sens du coup.

**Réponse de Chloé**

Ah, je comprends mieux. Cela fonctionne maintenant. Merci.

## 6.2.4 UpyLaB 4.5

### CHLOÉ

**Bonjour**

Je ne comprends pas les résultats que me donne UpyLaB :

```
Ok L'entête de la fonction alea_dice est correctement défini
```

```
Ok Bon résultat pour alea_dice(54384): False
```

```
Fail Résultat attendu pour alea_dice(98088): True
```

```
Ok Bon résultat pour alea_dice(77718): True
```

Pourquoi est-ce que pour 98088 mon résultat n'est pas correct ? Il l'est pour tous les autres.

### Réponse de l'équipe

Bonjour Chloé

Testez votre fonction `alea_dice(98088)` dans PyCharm ou pythontutor, vous devriez obtenir `True`. Si ce n'est pas le cas, essayez votre code dans PyCharm ou pythontutor en remplaçant les 4 instructions :

```
random.seed(s)
d1 = random.randint(1,6)
d2 = random.randint(1,6)
d3 = random.randint(1,6)
```

par :

```
d1, d2, d3 = 1, 2, 4
```

Vous devriez obtenir `True`, de même avec `2, 1, 4` ou `4, 1, 2` etc.

### Réponse de CHLOÉ

Ah d'accord, j'ai compris mon erreur : ma fonction ne retournait `True` que pour les valeurs 4, 2, 1 dans cet ordre.

AMINE

Bonjour,

Voici le message d'erreur que j'obtiens :

```
Résultats ok L'entête de la fonction alea_dice est correctement défini
fail Le code comprend des erreurs:
Ligne 25: name 'random' is not defined
```

Si je mets `from random import *` j'ai un autre message d'erreur. Quelqu'un a une explication ? Merci d'avance.

### Réponse de SABINE

Bonjour Amine,

As-tu essayé avec `import random` ?



### Réponse de AMINE

Merci Sabine. Cela fonctionne.

### Réponse de l'équipe

Pour compléter, ton `from random import *` fonctionne à condition de ne plus préfixer les fonctions avec le nom du module c'est-à-dire utiliser `randint()` et `seed()` à la place de `random.randint()` et `random.seed()`.

## 6.2.5 UpyLaB 4.6

SABINE

### Bonsoir,

Dans cet exercice, mon résultat est une liste. J'ai trouvé qu'on pouvait transformer une liste en string (et inversement) grâce à la fonction *join*. Mais je n'y arrive pas pour autant.

### Réponse de l'équipe

Bonjour Sabine,

Dans cet exercice il s'agit bien de retourner les cinq valeurs trouvées, mais pas dans une liste, ni dans une chaîne de caractères mais grâce à l'instruction :

```
return x20, x10, x5, x2, x1
```

### Réponse de Sabine

Merci. J'ai pu renvoyer les 5 valeurs. Mais Upylab me dit maintenant :

```
Votre programme a été interrompu par une exception: Ligne 1: EOF when reading_
↪ a line.
```

J'ai bien lu le post sur ce sujet. Mais je ne comprends toujours pas pourquoi. Mes valeurs input sont le montant à payer (prix) et celui payé.

### Réponse de l'équipe

Votre fonction ne doit rien demander comme input. Reprenons l'énoncé pas à pas :

Écrivez une fonction `rendre_monnaie` qui prend en paramètres un entier prix et cinq valeurs entières `x20`, `x10`, `x5`, `x2`, `x1` : Combien de paramètres au total reçoit la fonction *rendre\_monnaie* ? Comment doit donc être faite sa définition ?

La fonction doit renvoyer cinq valeurs représentant la somme qu'il faut rendre au client, décomposée en billets et pièces (dans le même ordre que précédemment) : `res20`, `res10`, `res5`, `res2`, `res1`.

S'il manque de l'argent, la fonction renverra cinq valeurs *None* : attention au cas particulier du client qui ne donne pas suffisamment d'argent.

### Réponse de Sabine

Merci à vous. Cette fois cela fonctionne.

PAUL

Bonjour,

J'ai des erreurs qui semblent aléatoires.

### Réponse de l'équipe

Bonjour Paul,

Dans cet exercice où on fait un traitement similaire pour les cinq variables `res20`, `res10`, `res5`, `res2`, `res1`, faire un copié-collé est peut-être intéressant. Toutefois, il faut bien vérifier que l'on effectue ensuite les changements qui doivent être faits. Cela est souvent source d'erreurs difficiles à détecter car il n'y a pas d'erreur de syntaxe et le bug peut ne pas intervenir sur tous les jeux de données.

### Réponse de PAUL

Merci ! Effectivement un `//10` apparaissait deux fois : une fois au bon endroit et une seconde fois à la place d'un `//2`.

## 6.2.6 UpyLaB 4.7

AMINE

Bonjour,

L'énoncé de l'exercice 4.8 est assez simple à comprendre, aucune difficulté apparente. Néanmoins, lorsque je souhaite tester sur UpyLab, j'obtiens 3 bonnes réponses et ensuite un message d'erreur du type :

Votre programme a été interrompu par une exception : Ligne 25 : sum() missing 1 required positional argument : "b"

Y'a t-il une subtilité dans l'énoncé, ou un « piège » à éviter au vu de la simplicité du code ?

### Réponse de l'équipe

Non il n'y a pas de piège, simplement une nouveauté : les arguments optionnels. À revoir dans le chapitre consacré.

## 6.2.7 UpyLaB 4.8

MATHIEU

Bonjour,

Salut à tous, avez-vous une idée sur ces messages ? J'ai testé mon code sur pythontutor mais sur UpyLAB ça donne :

```
Résultats ok L'entête de la fonction rac_eq_2nd_deg est correctement défini
fail Votre programme a été interrompu par une exception:
Ligne 25: TypeError: 'float' object is not iterable
fail Résultat attendu pour rac_eq_2nd_deg(1.0,1.0,-2.0) : (-2.0, 1.0)
ok Bon résultat pour rac_eq_2nd_deg(3.0,5.0,3.0) : ()
ok Bon résultat pour rac_eq_2nd_deg(-4.5,3.5,-3.5) : ()
ok Bon résultat pour rac_eq_2nd_deg(2.0,1.0,1.0) : ()
```

### Réponse de l'équipe

Bonjour Mathieu

Je pense que cela vient de ton utilisation de la fonction tuple. Celle-ci s'attend en effet à avoir comme argument un objet itérable, comme une liste, pour le transformer en tuple. Or, tu lui passes des objets de type float, d'où le message d'erreur.

Pour créer un tuple formé d'une ou plusieurs valeurs données, tu n'as pas besoin de cette fonction. La syntaxe (a, b) crée un tuple de deux éléments, et (a,) d'un seul élément. On peut même se passer des parenthèses, c'est en fait la virgule qui indique que l'on crée un objet de type tuple.

Tu auras juste besoin de la fonction tuple pour retourner un tuple vide, avec tuple() sans argument.

### Réponse de Mathieu

Merci, bien. Ca fonctionne maintenant.

## 6.2.8 UpyLaB 4.9

KIM

### Bonjour

Mes résultats sont validés pour les tests du programme « L'appel à votre programme a renvoyé... » mais ils sont erronés pour les tests de la fonction « Résultat attendu pour catalan(x)... »

### Réponse de l'équipe

Bonjour Kim,

Cela signifie que votre fonction en soi est mal définie. A priori elle reçoit ce dont elle a besoin pour fonctionner.

Vérifiez bien que c'est ce vous faites et que vous n'utilisez pas par exemple des variables globales comme dans l'exemple ci-dessous :

```
def ma_fonction(x):  
    var = g(n)  
    return var  
  
n = int(input())  
print(ma_fonction(n))
```

Dans cet extrait de code, dans la définition de `ma_fonction` il y a une faute de frappe : le `g(n)` aurait dû être un `g(x)`. Mais comme la variable `n` existe, définie au niveau du code principal, il n'y a pas d'erreur pour l'interprète Python. Pourtant il y a fort à parier que ce code ne fait pas ce qu'il devrait.

### Réponse de Kim

Effectivement, j'utilisais la variable du code appelant dans ma fonction. Maintenant, c'est corrigé et l'exercice est réussi.

FLORENCE

**Bonjour,**

Mon code ne contient aucune variable de type float et pourtant j'obtiens les résultats suivants :

L'appel à la fonction a retourné:

16796.0

Le résultat attendu était:

16796

D'où cela peut-il venir ?

**Réponse de l'équipe**

Bonjour Florence,

La division (/) retourne toujours un float. Essayez la division entière : //

**Réponse de FLORENCE**

Tous mes résultats sont faux maintenant.

**Réponse de l'équipe**

Attention lorsque vous utilisez la division entière. En effet si par exemple pour faire :

`2/4 * 2/4 * 2 * 2`

Vous faites :

`2//4 * 2//4 * 2 * 2`

Vous obtiendrez 0 à la place de 1. Il faut donc plutôt faire toutes les multiplications, puis les divisions :

`2 * 2 * 2 * 2 // 4 // 4`

**Réponse de FLORENCE**

Merci j'ai revu mon code en suivant ce conseil et cela fonctionne maintenant.

## 6.2.9 UpyLaB 4.11

CLAIRE

Bonjour,

Je me perds dans l’affichage et la mise à jour de la variable score. J’ai une variable globale que ma fonction `bat` modifie mais rien n’est validé avec UpyLaB.

Réponse de l’équipe

Bonjour Claire

Il vaut mieux éviter les variables globales. Essaie de faire une fonction qui prend le coup de chacun des joueurs et retourne la valeur de la manche (1, 0, -1). Il faudra alors faire l’affichage en fonction de ce résultat et mettre à jour la variable du score.

Réponse de Claire

Je me rapproche de la solution mais il me reste des croix rouges.

Réponse de l’équipe

Attention aux affichages, `PIERRE` n’est pas la même chose que `Pierre`. Ne pas confondre le nom des variables avec les chaînes de caractères à imprimer. Ainsi,

```
PIERRE = 0
FEUILLE = 1
CISEAUX = 2
```

permet de définir des **constantes** et par convention, on met leur nom en majuscules. Mais l’affichage demande d’écrire la chaîne de caractères `'Pierre'` ce qui n’a rien à voir.

Réponse de Claire

D’accord, je comprends mieux mon erreur.

OLIVIER

Bonjour,

Je suis un peu perdu : nous avons un seul `input` pour le joueur et cinq appels à `random.randint` pour le choix de l'utilisateur ? Les résultats laissent penser que le choix du joueur (`input`) est différent à chaque manche. En fait, je ne comprends pas à quoi sert `s = int(input())` au début de la fonction ? Merci par avance pour vos clarifications.

### Réponse de Virginie

Bonjour Olivier

Dans cet exercice, il faut 5 `inputs` pour ce que joue le joueur, et 5 appels à la fonction `random.randint` pour le choix de ce que l'ordinateur fait.

Alors il faut faire simultanément un `input` (pour le joueur) et un choix `random` (pour l'ordinateur).

Le `s = int(input())` au début de la fonction sert à fixer la graine du module `random`. Cette graine forcera les différentes exécutions du programme à fournir la même suite de valeurs aléatoires. C'est important pour qu'UpyLaB puisse vérifier que votre programme fonctionne bien.





### F

f-strings, [18](#)

### T

turtle, [18](#)

### V

version de Python, [17](#)