Chapitre n°0c

Programmation fonctionnelle

Exercices d'application directe du cours

Exercice n°0c.1 ★

Interaction entre variables globales et variables locales

Pour chacun des blocs d'instructions suivants, prévoir l'affichage obtenu puis le vérifier en utilisant un terminal Python.

Q1

Code Python

```
>>> x = 42

>>> def foo(x):
...     x = 0
...     return x

>>> foo(1), x
```

Q2

Code Python

```
>>> x = 42

>>> def foo(y):
...     y = 0
...     return y

>>> foo(1), y
```

Q3

Code Python

```
>>> x = 42

>>> def foo(y):
...     return x

>>> foo(1), x
```

Q4

Code Python

```
>>> x = 42

>>> def foo(y):
...     x = 0
...     return y

>>> foo(1), x
```

Q5

Code Python

```
>>> x = 42

>>> def foo(y):
...     x = y
...     return y

>>> foo(1), x
```

Q6

Code Python

```
>>> x = 42

>>> def foo(y):
...     y = x
...     return y

>>> foo(1), x
```

Exercice n°0c.2

★★★

Effets de bord

Pour chacun des blocs d'instructions suivants, prévoir l'affichage obtenu puis le vérifier en utilisant un terminal Python. Pour tout l'exercice, on supposera les variables `L` et `M` (re)déclarées de la manière suivante :

Code Python

```
>>> L = [12, 67, 29]
>>> M = ["a", "b", "c"]
```

avant chaque question.

On considère également que les fonctions `foo` et `bar` sont déclarées de la manière suivante :

Code Python

```
# Déclaration de la fonction
<foo>
def foo(L):

    L[0] = 42

    return L[::-1]
# Déclaration de la fonction
<bar>
def bar(M):

    L = M

    return foo(L)
```

Q1

Code Python

```
>>> A = foo(L)
>>> print(L)
```

Q2

Code Python

```
>>> L = foo(L)
>>> print(L)
```

Q3

Code Python

```
>>> A = bar(M)
>>> print(A)
```

Q4

Code Python

```
>>> A = bar(M)
>>> print(L)
```

Q5

Code Python

```
>>> A = bar(M)
>>> print(M)
```

Exercice n°0c.3 ★

Partie de tarot

Lors d'une partie de tarot à cinq, les scores de chacun des cinq joueurs (nommés Alice, Bob, Charles, Delphine et Étienne) sont stockés sous la forme d'une liste `L` construite de la manière suivante :

- le premier élément de la liste `L` est le tuple $(A_1, B_1, C_1, D_1, E_1)$ où X_1 désigne le score du joueur X à l'issue du premier tour de jeu ;
- le deuxième élément de la liste `L` est le tuple $(A_2, B_2, C_2, D_2, E_2)$ où X_2 désigne le score du joueur X à l'issue du deuxième tour de jeu ;
- etc.

On supposera la liste `L` déjà déclarée à l'issue de la partie.

Q1 Proposer une suite d'instructions permettant de déterminer le nombre n de tours qu'a comporté la partie de tarot.

Q2 Écrire une fonction `scores` prenant comme argument une liste `L` représentant le résultat de la partie de tarot (analogue à la liste `L` décrite précédemment), et retournant un tuple constitué de cinq listes `A`, `B`, `C`, `D` et `E`. Chacune de ces cinq listes est de la forme $(X_1, ..., X_n)$.

Au tarot, la somme des scores des différents joueurs est toujours nulle (c'est d'ailleurs un élément permettant de vérifier qu'aucune erreur de décompte de points n'a été commise au cours de la partie).

Q3 Écrire une fonction `verification` prenant comme arguments cinq listes `A`, `B`, `C`, `D` et `E` (représentant respectivement les listes $(X_1, ..., X_n)$ des scores d'Alice, de Bob, de Charles, de Delphine et d'Étienne au cours de la partie), et retournant le booléen `VRAI` si aucune erreur n'a été effectuée lors du décompte de points, et le booléen `FAUX` sinon.

Exercice n°0c.4 ★★

Moyennes diverses et variées

La moyenne de Hölder d'un ensemble $x_1, ..., x_n$ de réels strictement positifs est définie par :

$$H_p(x_1, ..., x_n) = \left(\frac{1}{n} \times \sum_{i=1}^{n} (x_i) \right)^p$$

avec $p \in \mathbb{R}^*$.

Q1 Quelle moyenne retrouve-t-on dans le cas particulier $p = 1$?

Q2 Écrire une fonction `holder` prenant comme arguments une liste `X` (représentant la suite $(x_i)_i$ de réels strictement positifs) et un flottant `p` (représentant le réel p non-nul), et retournant la moyenne de Hölder du n-uplet $(x_1, ..., x_n)$.

Q3 Écrire une fonction `arithmetique` prenant comme arguments une liste `X` (représentant la suite $(x_i)_i$ de réels strictement positifs), et retournant la moyenne arithmétique du n-uplet $(x_1, ..., x_n)$. On utilisera la fonction `holder`.

Q4 Comment pourrait-on modifier la définition de la fonction `holder` de manière à ce que :

- le résultat de l'appel `holder(X, p)` ne soit pas modifié ;
- le résultat de l'appel `holder(X)` soit égal au résultat de l'appel `arithmetique(X)` ?

Exercice n°0c.5 ★★

Échantillonnage d'un intervalle

On souhaite écrire une fonction `frange` analogue à la fonction `range`, mais fonctionnant également avec des bornes réelles.

Q1 Écrire une fonction `frange` prenant comme arguments trois flottants `debut`, `fin` et `pas`, et retournant la suite des valeurs comprises entre `debut` et `fin` par pas de `pas`. Afin de mimer le comportement de la fonction `range`, la valeur `fin` sera exclue de la liste de sortie et la valeur par défaut de `pas` sera prise égale à 1.

Q2 Écrire une fonction `frangeplus` reproduisant les caractéristiques suivantes de la fonction `range` :

- lorsqu'on fournit trois arguments à la fonction `frangeplus`, celle-ci se comporte comme la fonction `frange` de la question précédente (les arguments sont – dans l'ordre – `debut`, `fin` et `pas`) ;
- lorsqu'on ne fournit que deux arguments à la fonction `frangeplus`, ceux-ci désignent dans l'ordre les arguments `debut` et `fin`, le pas étant pris par défaut égal à 1 ;
- lorsqu'on ne fournit qu'un seul argument à la fonction `frangeplus`, celui-ci désigne la variable `fin`, la variable `debut` étant pris égale à 0 et la variable `pas` étant prise égale à 1.