

Python: bases

1 Introduction

Commencer par vous créer un répertoire/dossier dans lequel vous mettrez vos programmes.
Pour chaque nouveau programme, il est conseillé de faire un nouveau fichier.

2 Manipulation de variables

Voici, à un moment donné, la configuration des variables, toutes de type entier :

```
A = 12
B = 19
C = 14
D = 19
```

Exercice 1 - Inversion de variables

Q.1) Quel sera la valeur de chacune de ces mêmes variables après l'exécution du programme ci-dessous (réfléchir puis programmer pour tester)

```
A = B
B = A
print(A)
print(B)
```

Q.2) Écrire le (petit) algorithme qui fera en sorte que l'ordinateur échange les contenus des variables A et B.

Q.3) Quel est l'effet de l'algorithme suivant :

```
A = A+B
B = A-B
A = A-B
```

Exercice 2 - Permutation de variables

Q.1) Écrire un (petit) programme qui fera faire à l'ordinateur une permutation circulaire des variables A, B et C. C'est-à-dire que le contenu de A doit passer dans B, celui de B dans C et celui de C dans A.

Q.2) Idem avec les variables A, B, C et D.

Q.3) Quelle serait la difficulté d'écriture d'une programme permutant un nombre arbitrairement grand de variables (e.g. 100) ?

3 Exercices en vrac

Exercice 3 - Saisies et affichage (1)

Ecrire un programme qui demande à l'utilisateur de saisir la longueur puis la largeur d'un rectangle, calcule et affiche le périmètre du rectangle.

Exercice 4 - Alternatives

Ecrire un programme python qui demande à l'utilisateur de saisir deux nombres et l'informe ensuite si leur

Python: bases

produit est négatif ou positif (le cas où le produit est nul ne sera pas pris en compte). On ne doit pas calculer le produit des deux nombres.

Exercice 5 - Alternative (2)

Ecrire un algorithme qui demande l'âge d'un enfant à l'utilisateur. Ensuite, il l'informe de sa catégorie :

- "Poussin" de 6 à 7 ans
- "Pupille" de 8 à 9 ans
- "Minime" de 10 à 11 ans
- "Cadet" après 12 ans

Exercice 6 - TVA

Q.1) Ecrire un programme qui lit un prix hors taxe saisi par l'utilisateur et affiche le prix TTC (la TVA étant à 19.6%).

Q.2) Même chose, mais on a maintenant trois types de TVA : le taux luxe à 33%, le taux normal à 19.6%, et le taux de base à 5.5%. L'utilisateur saisit son prix HT, puis vous lui demandez de saisir le taux de TVA à appliquer :

```
Indiquez le prix HT :
Indiquez le taux de TVA à appliquer (1 pour luxe, 2 pour
normal, 3 pour base)
```

Exercice 7 - Entiers

Ecrire un programme qui permet de calculer le produit des n premiers entiers positifs.

Exercice 8 - Table de multiplication

Écrire un programme qui demande un nombre de départ, et qui ensuite écrit la table de multiplication de ce nombre, présentée comme suit (cas où l'utilisateur entre le nombre 7) :

```
Table de 7 :
7 x 1=7
7 x 2 = 14
7 x 3 = 21
...
7 x 10 = 70
```

4 Autres exercices plus rigolos

Exercice 9 - Guessing game

Le jeu du *guessing game* est un jeu dans lequel le programme « pense » à un nombre entier entre 1 et 99 et c'est à l'utilisateur de le deviner avec le moins de tentative possible alors que la machine ne répond que par « plus grand », « plus petit » ou « gagné ». Implémenter le programme du Guessing game dans un fichier nommé `GuessingGame.py` avec les spécifications suivantes :

- ▶ Initialement, le programme tire un nombre au hasard entre 0 et 100.
- ▶ Une invite annonce à l'utilisateur qu'il doit donner un nombre entre 0 et 100.
- ▶ À chaque proposition de l'utilisateur, le programme répond par « plus grand », « plus petit » ou « gagné ». Si l'utilisateur n'a pas gagné, l'invite est ré-affichée.
- ▶ Tant que le joueur n'a pas gagné, il doit fournir des propositions.

- ▶ « gagné » met fin au jeu, et alors on affiche à l'utilisateur le nombre de tentatives faites.

Amélioration possible: ajouter un nombre de tentative que l'utilisateur ne doit pas dépasser (pour information, il faut au maximum 8 essais pour trouver n'importe quel nombre entier entre 0 et 99).

Aide pour tirer un nombre aléatoirement entre 1 et 99:

- ▶ ajouter l'instruction suivante en tout début de programme

```
import random
```

- ▶ Pour tirer un nombre aléatoirement entre 1 et 99 (compris) vous pouvez utiliser l'instruction ci-dessous.

```
number = random.randint(0,99)
```

Exercice 10 - Pyramide (*)

Q.1) Que fait le programme R ci-dessous ?

```
i = 0
while (i<=10) {
  j = 0
  while (j<=10) {
    if ( j%2==i%2 ) {
      cat('X') #affiche une croix (sans retour à la ligne)
    } else {
      cat('O') #affiche un rond
    }
    j = j+1
  }
  cat('\n') #retour à la ligne
  i = i+1
}
```

Q.2) Écrire une fonction qui prend un paramètre n et génère l'affichage suivant (pour $n = 5$).

Vous aurez besoin de l'expression `print('*', end='')` pour afficher une étoile sans retourner à la ligne et `print('\n')` pour forcer le retour à la ligne.

```
*
**
***
****
*****
```

Q.3) Idem avec la pyramide ci-dessous (pour $n = 5$)

```
  *
 ***
*****
*****
*****
```

Exercice 11 - Projet Euler - Problème 9

A Pythagorean triplet is a set of three natural numbers, $a < b < c$, for which, There exists exactly one Pythagorean triplet for which $a + b + c = 1000$. Find the product $a \times b \times c$.

Exercice 12 - Calcul de π

Q.1) En utilisant la formule ci-dessous, écrire un programme qui calcule une valeur approchée de π (la valeur sera approchée dans le sens où le nombre de terme calculé)

$$\frac{\pi^2}{6} = \sum_{i=1}^{\infty} \frac{1}{i^2}$$

Q.2) Modifier le programme pour afficher à chaque itération la valeur de l'accumulateur

Q.3) Modifier le programme pour qu'il affiche la différence avec la valeur « réelle » de π variable `pi` est définie par défaut dans `R`.