UE121 - Introduction à la programmation

HAUTE ÉCOLE DE NAMUR-LIÈGE-LUXEMBOURG

Technologie de l'informatique - bloc 1

Sécurité des systèmes - bloc 1

Exercices 1 : bases

Objectifs

- apprendre à manipuler les notions élémentaires : variables, affectations, entrées/sorties
- apprendre à utiliser les expressions arithmétiques et booléennes…
- veiller au Clean Code,
- veiller à la portabilité du programme

A. Introduction

Les séries d'exercices visent à mettre en pratique les notions vues lors des ateliers et des séances de mise en commun. Il est important de faire un maximum d'exercices pour vous familiariser avec l'IDLE et avec le langage Python.

Au cas où vous ne termineriez pas les exercices durant la séance, nous vous conseillons de les achever chez vous au plus tôt et surtout avant l'atelier suivant.

Cette première série concerne la notion de variables et le choix judicieux de leur nom. Il est aussi important de maîtriser la notion de type malgré le fait qu'il soit dynamiquement et implicitement attribué à la variable en fonction de la valeur affectée. Vous utiliserez également les méthodes `print` et `input` afin d'afficher ou de récupérer diverses informations.

Note.

Dans les exemples qui suivent, les passages en texte normal sont à sortir tels quels.

Les parties en *italique souligné* correspondent aux entrées de l'utilisateur.

Les portions en **gras** varient en fonction des entrées.

Sur votre partition de travail (`U:`), créez un répertoire appelé Python. Ensuite, pour chaque exercice ci-après, ajoutez-y un fichier intitulé `Ex1-XX.py` avec `XX` à remplacer par le numéro de l'exercice en question.

B. Exercices à réaliser dans l'IDLE

Exercice 1

Écrivez le script qui convertit un montant en Euros vers les Dollars. Le montant en Euros est demandé à l'utilisateur et récupéré dans une variable dont le nom est choisi judicieusement. Le taux de change est le suivant 1 EUR = 1.09 USD. Le taux de change est également mémorisé dans une « variable », mais sa valeur ne change pas au cours du script. Choisissez à nouveau un nom adéquat.

On affiche ensuite à l'écran " ... Euros vaut ... Dollars " avec les valeurs correspondantes.

À l'écran doit apparaitre par exemple :

```
Montant en Euros : 45
45 Euros vaut 49.05 Dollars
```

Exercice 2

Reprenez le script précédent mais cette fois le taux de change est demandé à l'utilisateur et est récupéré dans une variable. Faites toujours attention au choix du nom de la variable et à la façon dont il est écrit.

Exercice 3

Écrivez le script qui demande et récupère une température en kelvin, et qui l'imprime en Celsius et en Fahrenheit.
kelvin = degre_celsius + 273.15
degre_celsius = (degre_fahrenheit – 32) * 5 / 9

À l'écran doit apparaitre par exemple :

```
kelvin : 300
300 K = 80.3333333 °F = 26.85 °C
```

Exercice 4

Écrivez le script qui demande et récupère deux nombres et en affiche la moyenne.

Dans une première version, utilisez une variable pour chacun des deux nombres obtenus.

Essayez maintenant en n'utilisant qu'une seule variable.

À votre avis, y a-t-il une façon de faire préférable parmi ces deux versions ? Pourquoi ?

Exercice 5

Écrivez un script qui demande et récupère un rayon (en mètres), calcule et affiche la circonférence et l'aire d'un disque.

Référez-vous à la documentation de Python et utilisez la constante prédéfinie correspondant à PI.

Circonférence : 2 * π * rayon
Aire : π * rayon^2

À l'écran doit apparaitre par exemple :

```
Rayon : 5
Circonférence : 31.42 mètres
Aire : 78.53982 mètres carrés
```

Exercice 6

Écrivez un script qui demande et récupère deux valeurs dans deux variables, et, qui inverse la valeur des deux variables. Il faut donc faire en sorte que la valeur de la première variable devienne celle de la deuxième variable et que celle de deuxième variable devienne celle de première variable.

Affichez les valeurs des deux variables avant et après l'inversion. Attention, on a bien dit d'inverser les valeurs des variables, pas juste leur affichage. À l'écran doit apparaitre par exemple :

```
Valeur 1 : 5
Valeur 2 : 9
Valeur 1 : 5 - Valeur 2 : 9
Valeur 1 : 9 - Valeur 2 : 5
```

Exercice 7

Écrivez un script qui demande et récupère un nombre entier. Écrivez l'expression la plus lisible permettant de savoir s'il est pair ou impair. Vous devez avoir comme réponse du Shell soit `True` si celui-ci est pair, soit `False` sinon.

Exercice 8

Écrivez un script qui demande et récupère un nombre positif. À l'aide des opérateurs modulo et division entière, afficher séparément les parties entière (int) et décimale (float) d'un nombre positif entré au clavier. À l'écran doit apparaitre par exemple :

```
Nombre : 8.25

8.25 = 8.0 + 0.25
```

Testez votre programme pour plusieurs nombres et vérifiez les résultats obtenus.

Exercice 9

Écrivez un script qui demande et récupère un nombre de secondes et le transforme en heures, minutes et secondes. À l'écran doit apparaitre par exemple :

```
Secondes : 12863
12863 secondes = 3 heures 34 minutes 23 secondes
```

Exercice 10

Même problème que l'exercice précédent mais en transformant en semaines, jours, heures, minutes et secondes. À l'écran doit apparaitre par exemple :

```
Secondes : 86521647
12863 secondes = 143 semaines 0 jours 9 heures 47 minutes 27 secondes
```

Exercice 11

Écrivez un script qui demande et récupère un nombre entier (entre 0 et 15) et affiche la représentation binaire d'un nombre en commençant par le bit de poids fort (MSB).

Exercice 12

Même problème que l'exercice précédent mais en commençant par le bit de poids faible (LSB).

Quelle est la différence principale de votre algorithme avec cette approche ?

Exercice 13

Dans le cadre d'un jeu de stratégie, on désire savoir s'il est possible de construire des nouveaux bâtiments. Connaissant le nombre de bâtiments à construire `nb_batiments`, les ressources nécessaires à la construction d'un bâtiment `ressources_necessaires` et les ressources disponibles `ressources_disponibles`, écrivez les instructions (1 instruction par variable) permettant de déclarer et d'initialiser les variables suivantes :

- `nb_batiments_valide`, qui indique si le nombre de bâtiments est compris entre 1 et 15 ;
- `ressources_insuffisantes`, qui indique si les ressources disponibles ne sont pas suffisantes à la construction des bâtiments ;
- `peut_construire`, qui indique si la construction est réalisable, c'est-à-dire que le nombres de bâtiments à construire est valide et que les ressources sont suffisantes.

Écrivez le script qui permet de demander et de récupérer les valeurs des trois variables utilisées ci-dessus, et ensuite de vérifier si les bâtiments peuvent être construits en réutilisant les différentes instructions que vous avez écrites ci-avant.