TP Découverte du langage Python 1

I Lancer l'éditeur Python

Ouvrir l'éditeur de code Python qui s'appelle Thonny.

II Dans la console (Shell)

La console se reconnaît facilement. C'est elle qui contient le chevron >>> qui est l'invite de Python (prompt en anglais) et qui signifie que Python attend une commande.

Exercice 1

Lancer les instructions suivantes dans la console en appuyant sur Enter à chaque fin de ligne et regarder puis écrire le résultat

```
>>> print ( "Hello world !")
>>> x=3
>>> x
>>> 4+5
>>> 5/2
>>> 5//2
>>> print("la valeur de x est : ", x)
```

1a. print affiche :

- ☐ seulement ce qui est entre les guillemets
- ☐ ce qui est entre les guillemets et le nom des variables
- ☒ ce qui est entre les guillements ou/et le contenu des variables

1b. 5//2 correspond

- ☐ à une division décimale
- ☒ au quotient de la division entière
- ☐ au reste de la division entière

III Les différents type de variables

Voici les différents types de variables que vous devez connaître (lire, ne pas tester) :

Type	Notation Python	Exemples
Nombres **entiers relatifs**	int( )	> int(−5.5) −5 > type(2) <class 'int'>
Nombres **flottants** (décimaux = nombres à virgules)	float( )	> type(2.0) <class 'float'>

Les **chaînes de caractères (string)**	str( )	> type(" a ") <class 'str'>
Les **booléens** (True ou False) (True ou False ou expression qui est soit vraie soit fausse)	bool( )	> type(False) <class 'bool'> > 10 < 2 False > type(2<3) <class 'bool'>
Les **listes**	list( )	> type([1,2]) <class 'list'>

Exercice 2

En utilisant le tableau précédent, donner le type de a :

a	a = 2	a = 2.0	a = 2+3	a = 2+3.0	a="bonjour"	a = False	a = 2 < 3	a = "2<3"	a = [2,3]	a="2.1 "
type(a)	int	float	int	float	str	bool	bool	str	list	str

IV Tester et comparer des variables

Une variable booléenne est le résultat (True ou False) d'une phrase ou d'un test logique. Exemple : le test logique (ou la comparaison) a<b peut être True ou False, tout comme le test a==b . Python est capable d'effectuer toute une série de comparaisons entre le contenu de deux variables, telles que :

= = égal à
! = différent de
> supérieur à
> = supérieur ou égal à
< inférieur à
< = inférieur ou égal à

```
# A essayer dans la console PYTHON
>>> 2<3
True
>>> 3 = = 2
False
>>> 3 = 2 # Attention : le symbole égal = est une affectation,
          # cette écriture renvoie une erreur. Essayez 3 == 2
SyntaxError: cannot assign to literal
```

Exercice 3

Prévoir puis testez les résultats suivants : Résultats :

 # Dans la console PYTHON tapez l'expression puis entrée:

a = 2
b = 3
c = 5

a == b ..... : False
a+b == c ..... : True
a < b ..... : True
a <= c ..... : True
a==b **and** a==2 ..... : False
a==b **or** a==2 ..... : True
type(a == c) ..... : Bool

V Variables et affectations simultanées

Exercice 4

a) Taper le programme suivant dans la **zone d'édition** de Thonny.

```
a = 2
b = -5
a,b = a+b, a-b
print("Maintenant a= ", a ," et b = ", b)
```

Enregistrer et nommer le fichier **affectations.py** dans un dossier PYTHON lui même dans un dossier NSI que vous aurez créés dans votre H travail. Faites RUN avec la flèche verte ou en pressant F5.

b) On considère un autre programme ci-dessous écrit en Python. Exécutez-le.

```
a=2
b=-5
a=a+b
b=a-b
print("Maintenant a= ", a ," et b = ", b)
```

En comparant les résultats des 2 programmes, expliquez ce que permet de faire la ligne :
a,b = a+b, a-b du premier programme :

Cela permet de faire une double affectation sns que les valeurs de a ou b changent

On considère l'algorithme suivant écrit en pseudo code :

L1 U ← 500
L2 N ← 0
L3 U ← 0.7×U +300
L4 N ← N +1
L5 Afficher U , N

1. Compléter le tableau suivant afin de déterminer les valeurs affichées en sortie.

Ligne	Valeur de U	Valeur de N
L1	500	----------
L2		0
L3	650	
L4		1
L5	650	1

2. Écrire en Python ce programme en utilisant le moins de lignes possible

```
U = 500
N = 0
U = 0.7*U + 300
N = N + 1
print(U,N)
```

et mieux :

```
U, N = 500, 0
U, N = 0.7*U + 300, N +1
print(U,N)
```