

Programmer en Python
Licence 2, Mathématiques
V. Monbet
TD 1. Variables et affectations
Corrigé

Exercice 1

Examiner la série de commandes ci-dessous et prédire le résultat de chacune des instructions.

```
>>> x=1
>>>print(x)
>>> x=1
>>> y=2
>>> x=x+y
>>> y=x**y
>>> print(y)
>>> print(x,y)
```

Exercice 2

1. Décrivez le plus clairement et le plus complètement possible ce qui se passe à chacune des trois lignes de l'exemple ci-dessous :

```
>>> largeur = 20
>>> hauteur = 5 * 9.3
>>> largeur * hauteur
```

Corrigé : on crée deux variables `largeur` et `hauteur` auxquelles on affecte respectivement l'entier 20 et le réel 5×9.3 . Puis on calcule le produit `largeur` par `hauteur`. Le résultat s'affiche à l'écran.

2. Assignez les valeurs respectives 3, 5, 7 à trois variables a, b, c.

Corrigé :

```
>>> a,b,c = 3,5,7
```

Quel résultat devrait s'afficher quand on exécute l'instruction `a - b/c`? **Corrigé :**
3

3. Assignez les valeurs respectives 3.0, 5.0, 7.0 à trois variables a1, b1, c1. Quel résultat devrait s'afficher quand on exécute l'instruction `a - b/c`?

```
>>> a1,b1,c1 = 3.0,5.0,7.0
>>> a1 - b1/c1
2.2857
```

4. Pourquoi n'obtient-on pas le même résultat ? **Corrigé** : dans le premier cas on travaille dans l'ensemble des entiers. Dans le second cas, on travaille dans l'ensemble des réels.

Exercice 3

Remarque. La commande ci-dessous permet d'échanger deux noms de variables.

```
>>> x,y=y,x
```

1. Écrire une liste d'instructions qui permette d'affecter la valeur $\sin(\pi/4)$ à x , $\cos(\pi/4)$ à y et de calculer $z = x^2 + y^2$ puis d'afficher la valeur contenue dans la variable z .

Corrigé

```
x = math.sin(math.pi/4)
y = math.cos(math.pi/4)
x**2+y**2
```

2. Pourquoi serait-il plus "logique" d'affecter le \cos à x et le \sin à y ? **Corrigé** : cohérence des noms de variables avec les conventions mathématiques.

Exercice 4

```
x, y = 2.3, 15
z = 4*x + y/5
t = [1,1.5,2,2.5,3]
s = [0.1,0.2,0.3,0.4,0.5]
d = (t-s)**2
```

1. Dans les commandes ci-dessus quelles sont les variables créées ? **Corrigé** : x , y , z , t , s , d
2. Quel est le type affecté automatiquement à chacune des variables ? **Corrigé** : $x = \text{float}$, $y = \text{integer}$, $z = \text{float}$, $t = \text{list}$, $s = \text{list}$, $d = \text{error}$...

On ne peut pas faire d'opérations mathématiques sur des listes. il faut créer des vecteurs (`numpy.array`)

```
import numpy as np
t = np.array( [1,1.5,2,2.5,3])
s = np.array([0.1,0.2,0.3,0.4,0.5])
d = (t-s)**2
```

ou faire une boucle sur les listes (cette solution est davantage en adéquation avec le potentiel de python qui est un langage propocé à l'utilisation des listes)

```
t = [1,1.5,2,2.5,3]
s = [0.1,0.2,0.3,0.4,0.5]
d = list()
```

```
for i in range(len(t)):
    d.append((t[i]-s[i])**2)
print(d)
```

autre solution plus élégante (et plus rapide?)

```
d = [(x-y)**2 for x,y in zip(t,s)]
```

3. Quelles valeurs prennent les variables `z` et `d`?

Exercice 5

Pour une auto école, on va développer un programme qui permet de compter le nombre de mois écoulés entre l'inscription à la formation et la réussite au code. Pour chaque élève on veut stocker son prénom, son nom, son âge, le mois de l'inscription (codé 1 à 12) et le mois de réussite du code (codé 1 à 12).

Premiers élèves

- Jean Dupont, 20 ans, inscription 01/2017, réussite 02/2017
- Jeanne Flavier, 18 ans, inscription 02/2017, réussite 05/2017
- Floria Carpon, 19 ans, inscription 04/2017, réussite 06/2017
- Eric Vert, 27 ans, inscription 07/2017, réussite 07/2017

1. Choisir les noms de variables nécessaires pour stocker les informations.
2. Choisir un(e) élève et écrire les instructions qui permettent d'affecter à aux variables les bonnes informations.

Corrigé

```
nom = 'Dupont'
prenom = 'Jean'
age = 20
inscription = 01 # on stocke uniquement le mois
reussite = 02 # idem
```

3. Pour l'élève choisi(e) comment peut-on calculer le nombre de mois pendant lesquels il a préparé son code.
Proposer une liste d'instructions permettant de réaliser ce calcul.

Corrigé

```
duree = reussite-inscription
```

4. Critiquer le programme suivant (en particulier la façon dont sont définis les noms de variables)

```
NOM = "Dupont"
Prénom = "Jean"
âge = 20
mois_inscription = 1
mois_reussite = 2
```

```
TempsTotal = mois_reussite-mois_inscription
print(TempsTotal)
```

5. Créer les listes `noms`, `prenoms` et les vecteurs `ages`, `inscriptions`, `reussites` contenant respectivement les noms, prénoms, âges, mois d'inscription et de réussite.

Pour les vecteurs, vous pourrez procéder de la façon suivante

```
import numpy as np # importation de la librairie numpy
age = np.array([20,18,29,27]) # création du vecteur et remplissage
np.mean(age) # calcul de l'age moyen
```

6. Calculer l'âge moyen des élèves ainsi que le temps moyen nécessaire à l'obtention du code.

Corrigé

```
noms = ['Dupont', 'Flavier', 'Carpon', 'Vert']
prenoms = ['Jean', 'Jeanne', 'Floria', 'Eric']
age = np.array([20,18,29,27]) # création du vecteur et remplissage
inscriptions = np.array([1,2,4,7])
reussites = np.array([2,5,6,7])
durees = reussites-inscriptions
durees_moy = np.mean(durees)
```

```
elev = 2
print prenoms[elev], " ", noms[elev], " obtient son code en ",
      durees[elev], " mois"
```

Exercice 6

On souhaite rédiger un programme élémentaire qui calcule le prix d'une commande de bières pour un festival. Les trois valeurs suivantes sont représentées par les variables :

`nbr` = entier désignant le nombre de fûts commandés
`prix` = prix unitaire d'un fût
`reduc` = coefficient (entre 0 et 1) représentant la réduction dont bénéficie le client

Le programme affiche le montant de la facture `m = nbr * prix * reduc`.

1. Affecter les variables comme indiqué ci-dessus dans le cas d'une commande de 27 fûts dont le prix unitaire est de 22,95 euros pour un client bénéficiant de 5 % de réduction.
2. Quelle instruction permet d'afficher le résultat dans l'interpréteur ?

Corrigé

```
nbr = 27
prix = 22.95
reduc = 0.05
print "Le montant à réglé est",nbr*prix*(1-reduc)
```

Exercice 7

Je suis ligoté sur les rails en gare d'Arras. Écrire un programme qui affiche un tableau me permettant de connaître l'heure à laquelle je serai déchiqueté par le train parti de la gare du Nord à 9h (il y a 170 km entre la gare du Nord et Arras).

Le tableau prédira les différentes heures possibles pour toutes les vitesses de 100 km/h à 300 km/h, par pas de 50 km/h, les résultats étant arrondis à la minute inférieure.

Vous pourrez utiliser les fonctions `linspace` or `np.arange`. Par exemple les 4 commandes ci-dessous donnent le même résultat

```
>>> x = np.array([0.1, 0.3, 0.5, 0.7, 0.9, 1.1])
y1 = np.linspace(0.1, 1.1, 6)
y2 = np.arange(0.1, 1.1, 0.2)
y3 = np.arange(0.1, 1.11, 0.2)
```

Corrigé

```
vitesse = np.linspace(100.0, 300.0, 6)
heure = 9+170/vitesse
```