TP N°0-1 : Introduction à Python

UP Mathématiques*

2 novembre 2023

Exercise 1: Les variables

a) Assignez les valeurs respectives 3, 5, 7 à trois variables a, b, c. Effectuez l'opération `a-b//c`. Interprétez le résultat obtenu.

Solution. Soit les trois valeurs 3, 5 et 7 assignées, respectivement, aux variables `a`, `b` et `c`:

```
a, b, c = 3, 5, 7 # Assignation des variables
a - b//c
```

Interprétation
L'expression `a - b // c` signifie que nous voulons soustraire (signe ''-'') de la variable `a` la division entière (signe ''//'') de la variable `b` par la variable `c`. Le résultat est 3 car la division entière de 5 par 7 est égale à 0.

b) Testez les lignes d'instructions suivantes. Décrivez ce qui se passe :

```
r = 12
pi = 3.14159
s = pi * r**2
print(s)
print(type(r), type(pi), type(s))
```

Quelle est, à votre avis, l'utilité de la fonction `type()` ?

*École Supérieure PRivée d'Ingénierie et de Technologies (ESPRIT).

Solution. Soit:

```
r = 12
pi = 3.14159
s = pi * r**2
print(s)
print(type(r), type(pi), type(s))
```

Description
Ce code sert à calculer la surface $s = \pi r^2$ d'un disque de rayon r donné.

- dans les deux premières lignes, **nous assignons** les paramètres de l'équation de s (`r` et `pi`) par ses valeurs.
- à la troisième ligne, **nous attribuons** `s` par son expression.
- dans les deux dernières lignes, **nous affichons** la valeur de `s` trouvée et les types de variables utilisés dans ce code.

c) Écrivez un programme qui convertisse en degrés Celsius une température exprimée au départ en degrés Fahrenheit, ou l'inverse. La formule de conversion est :

$$T_F = T_C \times 1,8 + 32$$

Solution. Soit:

$$T_F = T_C \times 1,8 + 32$$

avec T_F estla température en degrés Fahrenheit et T_C est la température en degrés Celsius.

```
print("Conversion de degrés Celsius en degrés Fahrenheit:")
Tc1 = 25 # température en Celsius donnée
TF1 = Tc1 * 1.8 + 32
print("{} degrés Celsius vaut {} degrés Fahrenheit".format(Tc1, TF1))

print("Conversion de degrés Fahrenheit en degrés Celsius:")
TF2 = 120 # température en Fahrenheit donnée
TC2 = (TF2 - 32)/1.8
print("{} degrés Fahrenheit vaut {} degrés Celsius".format(TF2, TC2))
```

Exercise 2: Fonction `input()`

Dans tous ces exercices, utilisez la fonction `input()` pour l'entrée des données.

a) Écrivez un programme qui convertisse en mètres par seconde et en km/h une vitesse fournie par l'utilisateur en miles/heure. (Rappel : 1 mile = 1609 mètres)

Solution. Conversion de miles/heure en km/h et m/s

```
# %load sol21.py
# Conversion de miles/heure en km/h et m/s
ch = input("Veuillez entrer le nombre de miles parcourus en une heure : ")
mph = float(ch)             # conversion de la chaîne entrée en nombre réel
mps = mph * 1609 / 3600     # conversion en mètres par seconde
kmph = mph * 1.609          # conversion en km/h
# affichage :
print(mph, "miles/heure =", kmph, "km/h, ou encore", mps, "m/s")
```

b) Écrivez un programme qui calcule le périmètre et l'aire d'un triangle quelconque dont l'utilisateur fournit les 3 côtés.(Rappel : l'aire d'un triangle quelconque se calcule à l'aide de la formule :

$$S = \sqrt{d \cdot (d-a) \cdot (d-b) \cdot (d-c)}$$

dans laquelle d désigne la longueur du demi-périmètre, et a, b, c celles des trois côtés.)

Solution. Périmètre et Aire d'un triangle quelconque:

```
# %load sol22.py
# Périmètre et Aire d'un triangle quelconque

from math import sqrt

a = float(input("Veuillez entrer le côté a : "))
b = float(input("Veuillez entrer le côté b : "))
c = float(input("Veuillez entrer le côté c : "))
d = (a + b + c)/2.0                 # demi-périmètre
s = sqrt(d*(d-a)*(d-b)*(d-c))    # aire (suivant formule)

print("Longueur des côtés =", a, b, c)
print("Périmètre =", d*2, "Aire =", s)
```

Exercise 3: Corriger l'erreur dans le code

a) Le code suivant renvoie une erreur. Trouver et corriger l'erreur:

```
prenom = input('Entrez votre prénom : ')
age = input('Entrez votre age : ')
annee_naissance = 2018 - age
print("Bonjour Mr/Mme", prenom)
print("vous êtes né en", annee_naissance)
```

Indication.

- utiliser la fonction `type()`
- utliser la fonction `int()`

Solution. La solution est comme suit:

```
In [71]: prenom = input('Entrez votre prénom : ')
    ...: age = input('Entrez votre age : ')
    ...: type(age) # ici str
    ...: age = int(age)
    ...: type(age) #ici int
    ...: annee_naissance = 2018 - age
    ...: print("Bonjour Mr/Mme", prenom)
    ...: print("vous êtes né en", annee_naissance)

Entrez votre prénom : Foulen

Entrez votre age : 25
Bonjour Mr/Mme Foulen
vous êtes né en 1993
```

b) Afficher le messge suivant: Bonjour Mr/Mme `prenom`, votre age est `age` et vous êtes né en `annee_naissance`.

Indication. Remplacer les points par ce qui convient dans le code:

```
print(" Bonjour Mr/Mme {}, votre age est {}....".format(...))
```

Solution. La solution s'écrit:

```
print(" Bonjour Mr/Mme {}, votre age est {} et vous êtes né en {}".format(prenom, age, annee_naissar
```

Exercise 4: Polynômes creux

L'objectif de cet exercice est la manipulation des polynômes creux à une seule variable.

Un polynôme creux est un polynôme dont certains coefficients sont nuls.

Un polynôme est construit à partir de monômes.

Un monôme est une expression de la forme ax^n où a ($a \neq 0$) est le coefficient du monôme et n ($n \geq 0$) son degré.

Un monôme est représenté par un dictionnaire à un élément dont la clé est le degré n et la valeur est le coefficient a.

Exemple :

Le monôme $8x^2$ est représenté par le dictionnaire `{2:8}`.

Un polynôme creux est alors défini comme une association de monômes de degrés différents.

Exemple :

Le polynôme $-x^4 + 8x^2 - 5x$ est représenté par le dictionnaire `{2:8, 1:-5, 4:-1}`.

Le dictionnaire `{0:1, 5:1, 8:1}` représente le polynôme $x^8 + x^5 + 1$.

a) Compléter le script de la fonction `ajout_monome(P,monome)`, sachant que `P` est un dictionnaire représentant un polynôme. On rappelle que cette méthode ajoute un monôme saisi au clavier (en faisant les contrôles nécessaires) si le paramètre `monome` est nul ou ajoute le monôme nommé `monome` sinon.

```
def ajout_monome(P,monome={}):
    """
    Cette méthode ajoute un monôme saisi au clavier à P
    si le paramètre monome est nul ou ajoute le monôme nommé monome sinon
    """
    if len(monome)==0:
        # Partie à compléter
    else:
        degre=list(monome.keys())[0]
        coeff=list(monome.values())[0]
        assert degre>=0
        assert type(degre)==int
        assert type(coeff)==int or type(coeff)==float
        assert len(monome)==1
        P.update(monome)
```

Indication. L'appel à la fonction `ajout_monome` peut être utilisé comme suivant :

```
In[1]: P = {0:1, 5:1, 8:1} # x^8 + x^5 + 1
In[2]: ajout_monome(P)
donner le degré du monôme : 2
donner le coefficient du monôme : 1

Out[2]: {0: 1, 5: 1, 8: 1, 2: 1.0}
```

ou bien

```
In[3]: ajout_monome(P, {2:8})
Out[3]: {0: 1, 5: 1, 8: 1, 2: 8}
```

```
def ajout_monome(P,monome={}):
    """
    Cette fonction ajoute un monôme saisi au clavier à P
    si le paramètre monome est nul ou ajoute le monôme nommé monome sinon
    """
    if len(monome)==0:
        # Réponse à Q1.
        while True:
            try:
                degree = int(input("donner le degré du monôme : "))
                coeff = float(input("donner le coefficient du monôme : "))
                if degree >=0:
                    break
            except:
                print("erreur de saisie") # ou pass ou continue
```

```
            P.update({degree:coeff})
        else: # Si monome est non vide
            degre=list(monome.keys())[0] # extraction du degré
            coeff=list(monome.values())[0] # extraction du coefficient
            assert degre>=0
            assert type(degre)==int
            assert type(coeff)==int or type(coeff)==float
            assert len(monome)==1
            P.update(monome)
        return P
```

Solution.

b) Écrire le code de la fonction `degree(P)` qui prend en paramètre le polynôme `P` représenté sous forme d'un dictionnaire et renvoie son degré.

```
def degree(P):
  # à compléter
```

```
def degree(P):
    return max(P.keys())
```

Solution.

```
In[3]: degree(P)
Out[3]: 8
```

c) Écrire le code de la fonction `call(P,x0)` qui retourne la valeur du polynôme P passé en paramètre sous forme d'un dictionnaire pour un réel `x0` donné.

```
def call(P, x0):
  # à compléter
```

```
def call(P, x0):
    val = 0
    for degree in P.keys():
        val += P[degree]*x0**degree
    return val
```

Solution.

```
In[4]: P
Out[4]: {0: 1, 5: 1, 8: 1, 2: 8}
In[5]: call(P, 2)
Out[4]: 321
```

d) Écrire le code de la fonction `add(P,P1)` qui retourne le polynôme somme des deux polynômes `P` et `P1`.

```
def add(P, P1):
    # à compléter
```

Indication. aucun monôme nul ne doit apparaître dans le polynôme résultat.

```
def add(P, P1):
    p = {}
    for degree in P.keys() :
        if not (degree in P1.keys()):
            ajout_monome(p,{degree:P[degree]})
        elif P[degree] != - P1[degree] :
            p.update({degree :P[degree] + P1[degree]})
    for degree in P1.keys():
        if not (degree in P.keys()):
            ajout_monome(p,{degree:P1[degree]})
    return p
```

Solution.

e) En déduire une fonction `diff(P,P1)` qui retourne le polynôme différence des deux polynômes P et P1. **Remarque :** aucun monôme nul ne doit apparaître dans le polynôme résultat.

f) Écrire le code de la fonction `mul(P,P1)` qui retourne le polynôme produit de deux polynômes. **Remarque :** aucun monôme nul ne doit apparaître dans le polynôme résultat.

g) Écrire le code de la fonction `derive(P)` qui retourne le polynôme dérivé de P.

h) Écrire le code de la fonction `affiche(P)` qui retourne la chaîne de caractères représentant l'expression du polynôme ordonné par ordre décroissant.

Pour le polynôme représenté par `{4:4, 0:4, 12:6, 9:1, 7:-1}`, la chaîne retournée est : `"6 * X^12 + X^9 - X^7+ 4 * X^4 + 4"`

i) Écrire le code de la fonction `primitive(P)`, qui retourne le polynôme représentant la primitive. On suppose que la constante d'intégration est nulle.

j) On définit, l'intégrale d'un polynôme creux P en x entre les bornes a et b, par :

$$S = \int_a^b P dx$$

Écrire le script de la fonction, nommée `integrale`, permettant de retourner la valeur de S à partir d'un polynôme P, de type `PolynomeCreux`, et des bornes d'integration a et b réels.

k) Écrire une fonction `courbe()` affichant la dans un même repère un polynôme P, sa primitive et sa dérivée.