Cours du vendredi 11 septembre

Bonnes pratiques en programmation

1. **Commenter le code**

   a) pour se souvenir de ce qu'il signifie quand on le relit 15 jours après
   b) pour qu'une personne qui ne connaît pas le programme et le lit comprenne facilement ce qu'il fait.

1. **Réaliser les tâches par petits morceaux** pour que le programme reste simple et facilement modifiable

1. **Tester** régulièrement le fonctionnement du programme pour ne pas avoir à relire un code très long et compliqué sans comprendre pourquoi il ne fonctionne pas comme voulu ;

1. **Lire attentivement les erreurs signalées par Python** pour trouver rapidement la source de l'erreur et la corriger ;

1. **Donner des noms explicites aux variables et aux fonctions** afin de savoir précisément ce qu'elles contiennent comme valeur ou ce qu'elles font ;

1. **Utiliser le « camel-case » pour les noms de variables et de fonction** : par exemple, `maVariable` ou `maFonctionDeTest` , afin de les lire facilement

1. **Faire des print pour voir l'évolution des variables** et être sûr qu'elles se comportent comme attendu.

Suite du cours Base de Python

6.La boucle bornée : for ...

Lire le cours sur la boucle for p6 du poly puis faire les exercices en utilisant les cellules ci dessous.

Cliquez sur la flèche à gauche pour executer une cellule et eventuellement sur le carré qui apparaîtra si vous voyez que votre boucle devient infinie.

Exercice 1 :

Compléter le script ci-dessous pour qu'il affiche la table de multiplication de 13 :

In [2]:
```
for i in range(11) :
    print("13 x",i,"=", 13*11)
```

```
13 x 0 = 143
13 x 1 = 143
13 x 2 = 143
13 x 3 = 143
13 x 4 = 143
13 x 5 = 143
13 x 6 = 143
13 x 7 = 143
13 x 8 = 143
13 x 9 = 143
13 x 10 = 143
```

Exercice 2 :

Le script ci-dessous calcule et affiche un nombre. À quelle opération ce nombre correspond-il ?

In [3]:
```
s=0
for i in range(101) :
    s = i + s
print(s)
```

```
5050
```

Il s'agit de la somme des 100 premiers entiers

Exercice 3 :

Jean a placé 5 000e sur un compte bancaire rémunéré à 2,5 % par an.

a)

Compléter le script suivant pour qu'il calcule et affiche la somme qu'il aura sur son compte après 10 ans d'augmentation.
Remarque : la `fonction round(c,2)` renvoie l'arrondi à 2 chiffres de *c*.

In [4]:
```
c = 5000
for i in range(11) :
    c = c*1.025
print(round(c,2))
```

```
6560.43
```

b)

Modifier le code pour qu'il demande le nombre n d'années d'augmentation, puis calcule et affiche la somme sur le compte de Jean après ces n années.

In [8]:
```
c=5000
a = int(input("au bout de combien d'années, souhaitez vous connaître la somme gagné
e ?"))
for i in range(a+1) :
    c = c*1.025
print(round(c,2))
```

```
au bout de combien d'années, souhaitez vous connaître la somme gagnée ?500
1179306156.12
```

7.La boucle non bornée : while ...

Lire le cours sur la boucle While p7 du poly puis faire les exercices en utilisant les cellules ci dessous.

Exercice 1 :

Un jardinier souhaite creuser un puits dans son jardin. Le prix du forage dépend de la profondeur atteinte. Le premier mètre coûte 100e, puis chaque mètre supplémentaire coûte 20e (120e pour aller à 2 m, 140e pour aller à 3 m, etc.) Pour ce forage, le budget du jardinier est de 700e.

Compléter le script suivant pour qu'il calcule et affiche la profondeur maximale atteinte.

In [10]:
```
profondeur = 1
prix = 100
while prix <= 700 :
    prix = prix + 20
    profondeur = profondeur + 1
print("la profondeur atteinte sera de ",profondeur, " mètres")
```

```
la profondeur atteinte sera de  32  mètres
```

On peut générer des nombres aléatoires à l'aide du module `random` et de la fonction `randint` . En mettant au début du script l'instruction `import random` , on peut plus loin donner l'instruction `a = random.randint(1,6)` qui affectera à la variable a un nombre aléatoire entre 1 inclus et 6 inclus.

Compléter ce script pour qu'il compte et affiche combien de fois il faut lancer deux dés pour obtenir 12 en faisant la somme des points des dés.

In [11]:
```
import random
somme = 0
nombre = 0
while somme != 12:
    de1 = random.randint(1,6)
    de2 = random.randint(1,6)
    somme = de1 + de2
    nombre = nombre +1
print("il a fallu ", nombre, "tirages pour obtenir 12")
```

```
il a fallu  37 tirages pour obtenir 12
```

Exercice 3 :

Réaliser le programme du « plus grand–plus petit » : le script commence par tirer un nombre entier au hasard entre 1 et 1000.

Puis, il demande à l'utilisateur d'entrer un nombre. Le script affiche « trop grand » si le nombre de l'utilisateur dépasse celui à deviner, « trop petit » sinon.

Ce processus doit se répéter jusqu'à ce que l'utilisateur devine le bon nombre.

Dans ce cas, le script affiche le nombre de propositions que le joueur a faites.

Variante : on peut imposer un nombre maximum de propositions, au-delà duquel le joueur perd le jeu.

In [12]:
```
import random

nombreAdeviner = random.randint(1,100)
nombreChoisi = 0
nombreEssai=0

while (nombreChoisi != nombreAdeviner) and (nombreEssai<=5) :
    nombreChoisi = int(input("Entrez un nombre entre 1 et 100 : "))
    nombreEssai +=1
    if nombreChoisi < nombreAdeviner :
        print("trop petit !")
    elif nombreChoisi > nombreAdeviner:
        print("trop grand !")

print("Vous avez fait ",nombreEssai," essais")
print("Le nombre à deviner était : ",nombreAdeviner)
```

```
Entrez un nombre entre 1 et 100 : 5
trop petit !
Entrez un nombre entre 1 et 100 : 50
trop petit !
Entrez un nombre entre 1 et 100 : 65
trop petit !
Entrez un nombre entre 1 et 100 : 56
trop petit !
Entrez un nombre entre 1 et 100 : 23
trop petit !
Entrez un nombre entre 1 et 100 : 58
trop petit !
Vous avez fait  6  essais
Le nombre à deviner était :  93
```

8.Les fonctions : def ...

Lire le cours sur les fonctions p8 du poly puis faire les exercices en utilisant les cellules ci dessous.

Exercice 1 :

On définit la fonction exercice1 de la façon suivante : def exercice1(a) : ␣␣return a**2-a+1

In [ ]:
```
def exercice1(a) :
    return a**2-a+1
```

1. Combien cette fonction a-t-elle de paramètres ?

Vous pouvez double-cliquer sur la cellule pour l'éditer et insérer votre réponse. Pour repasser en mode texte, il faut exeecuter la cellule

1. Que sera-t-il affiché si on appelle la fonction ? Essayer d'anticiper votre réponse avant d'executer la cellule

In [ ]:
```
exercice1(2)
```

In [ ]:
```
exercice1(5)
```

In [ ]:
```
exercice1(2.1)
```

Exercice 2 :

Écrire une fonction `vitesse` qui prend deux paramètres `distance` (supposé être en km) et `temps` (en heures) et qui retourne la vitesse moyenne en km/h.

In [13]:
```
def vitesse(distance,temps):
    return distance/temps
```

In [14]:
```
vitesse(5,2)
```

Out[14]: 2.5

Exercice 3 :

On considère la fonction suivante :

In [ ]:
```
import random
def exercice3() :
    return random.randint(1,6)
```

Dans ce script, la commande `import random` met à disposition du programmeur des fonctions additionnelles de Python, parmi lesquelles `randint` . Cette fonction permet d'obtenir un nombre entier aléatoire pris entre les deux nombres donnés en paramètres de `randint` .

Parmi les nombres suivants, lesquels ne peuvent pas être retournés par la fonction exercice3 ?

```
2     3,1     7     1,412     6     1     0

seul 2 6 et 1 sont valides
```

Exercice 4 :

Écrire une fonction `solde` qui prend en paramètres un prix et un taux de pourcentage, et qui retourne le prix soldé.

In [ ]:
```
def solde(prix,rabais):
    return prix-prix*rabais/100
```

Exercice 5 :

La population d'un village était de 3000 habitants en 2017. En moyenne, la ville perd 2 % de ses habitants chaque année.

Écrire une fonction `population` qui prend en paramètre une durée en années, et qui calcule la popuplation du village après cette duréee.

Vérifiez en exécutant votre fonction que l'instruction `population(2)` renvoie la valeur 2881,2.

In [15]:
```
#fonction qui prend en paramètre une durée exprimée en année
#calcule la population apres cette durée

def population(duree) :
    pop = 3000
    for i in range(1,duree+1):
        pop = pop*0.98
    return pop

population(2)
```

Out[15]:
```
2881.2
```

Écrire un programme qui détermine au bout de quelle durée la population sera inférieure ou égale à 1500 habitants, et affiche cette durée.

In [16]:
```
popu=3000
duree = 0
while popu > 1500:
    popu=popu*0.98
    duree +=1
print(duree)
```

```
35
```

Exercice 6 :

Une patinoire propose deux formules de tarification : Formule A : chaque entrée coûte 5,25e. Formule B : on paye un abonnement à l'année de 12e, et chaque entrée coûte alors 3,50e.

Écrire deux fonctions `tarifA` et `tarifB` pour calculer le prix à payer en fonction du nombre d'entrées.

In [17]:
```
def tarifA(nbEntree):
    return nbEntree*5.25


def tarifB(nbEntree):
    return 12+nbEntree*3.5
```

Utiliser ces fonctions pour déterminer au bout de combien d'entrées la formule B est la plus avantageuse.

In [19]:
```
i=0
while tarifA(i)<tarifB(i):
    i+=1
print("Le tarif B est plus avantageux au bout de ", i, " entrées.")
```

```
Le tarif B est plus avantageux au bout de  7  entrées.
```

9.Les chaînes de charactères

Cette partie est facultative. A faire si il vous reste du temps dans la séance.

Exercice 1 :

Faire sur la feuille

Exercice 2 :

La chaîne de caractères abc est définie par la commande `abc='trois lettres'` . Complétez les affichages obtenus en réponse aux instructions ci-dessous :

In [ ]:
```
print(abc)
```

In [ ]:
```
print('abc')
```

In [ ]:
```
print('abc')
```

In [ ]:
```
1+2+3
```

In [ ]:
```
abc+'d'
```

Exercice 3 :

On a commencé un programme qui demande une phrase à l'utilisateur, puis qui compte combien il y a d'espaces dans cette phrase. Complétez ce programme et testez-le :

In [20]:
```
phrase = input("Tapez votre phrase  : ")
n=0
for c in phrase :
    if c == " "  :
        n= n+1
print("Il y a", n, "espaces dans votre phrase")
```

```
Tapez votre phrase  : a b c
Il y a 2 espaces dans votre phrase
```

Exercice 4 :

Inspirez-vous du programme de l'exercice précédent pour créer un script qui compte la fréquence de 'e' (majuscules ou minuscules) dans une phrase.

Attention, il faut compter les 'é' 'è' 'ê' et 'ë' comme des 'e' !

Approfondissement : Calculez la fréquence de chacune des voyelles, voire la fréquence de chacune des lettres présente dans le texte.

In [22]:
```
phrase = input("Tapez votre phrase  : ")
n=0
for c in phrase :
    if c in ["e","E","é","è","ê","ë"]  :
        n= n+1
print("Il y a", n, "voyelle e dans votre phrase")
```

```
Tapez votre phrase  : e E
Il y a 2 voyelle e dans votre phrase
```

Exercice 5 :

Les fonctions `upper()` , `lower()` et `capitalize()` s'appliquent à une chaîne de caractères.

1. Complétez les exemples suivants pour voir ce qu'elles font.

In [4]:
```
p="Il FAit bEAu auJOUrd'hui !"
```

In [ ]:
```
p.upper()
p.lower()
p.capitalize()
```

1. Complétez ce script pour faire la fiche d'identité d'une personne, en mettant son nom en majuscules et son prénom avec une lettre capitale :

In [24]:
```
nom = input("Quel est votre nom ? ")
prenom = input("Quel est votre prénom ? ")
age = input("Quel est votre âge ? ")
print("NOM :", nom.upper() )
print("Prénom : ", prenom.capitalize() )
print("Âge : ", age )
```

```
Quel est votre nom ? sophie
Quel est votre prénom ? l^$l$1
Quel est votre âge ? 20
NOM : SOPHIE
Prénom :  L^$l$1
Âge :  20
```

Exercices de révsion

Exercice 1 :

Écrire un programme en Python qui récupère au clavier le rayon d'un cercle et qui affiche à l'écran son périmètre

In [ ]:
```
import math

rayon = float(input("Entrez le rayon du cercle : "))
print("le périmètre du cercle est : ", 2*math.pi*rayon)
print("l'aire du cercle est de : ", math.pi*rayon**2)
```

Exercice 2 :

compléter le programme ci dessous qui compte le nombre d'entiers entre 10 et 100 (l'utilisateur saisi en console) qui ont le **même** chiffre des unités ou des dizaines. Attention : le programme n'est pas correctement indenté. A vous de corriger l'indentation pour que cela fonctionne.

In [27]:
```
myst  = float(input("Entrez le chiffre mystère à compter : "))
nbEntier = 0
for entier in range(10, 100) :
    unite = entier % 10
    dizaine = entier // 10
    if unite == myst or dizaine == myst :
        nbEntier = nbEntier + 1
print(nbEntier)
```

```
Entrez le chiffre mystère à compter : 2
18
```

Exercice 3 :

Écrire un programme en Python qui récupère l'âge d'une Personne, si cette personne est étudiante ou non ainsi que si cette personne est chômeur ou non puis qui affiche le bon prix de la place de cinéma.

In [28]:
```
age = int(input("Quel est ton âge ?")) # par exemple 18
estEtudiant = input("Es tu étudiant ?") # par exemple True
estChomeur = input("Es tu chômeur ?")

if (age<16 or estEtudiant=="oui" or estChomeur=="oui"):
    prix = 4.5
else : prix = 6.5
print (prix)
```

```
Quel est ton âge ?25
Es tu étudiant ?oui
Es tu chômeur ?non
4.5
```

Exercice 4 :

Écrivez un programme qui affiche les 20 premiers termes de la table de multiplication par 53. Lerésultat commencera comme ceci : 1 x 53 = 53 2 x 53 = 106 3 x 53 = 159

Aide : Il existe une fonction de formatage qui permet d'aligner joliment les nombres en colonne.

Par exemple, `print('{:4d}'.format(a))` Cela écrira l'entier a sur 4 espaces et le calera sur la droite.

In [29]:
```
i=1
while i<=20 :
    print('{:2d}'.format(i), "x 53 = ",'{:4d}'.format(53*i))
    i += 1
```

```
 1 x 53 =   53
 2 x 53 =  106
 3 x 53 =  159
 4 x 53 =  212
 5 x 53 =  265
 6 x 53 =  318
 7 x 53 =  371
 8 x 53 =  424
 9 x 53 =  477
10 x 53 =  530
11 x 53 =  583
12 x 53 =  636
13 x 53 =  689
14 x 53 =  742
15 x 53 =  795
16 x 53 =  848
17 x 53 =  901
18 x 53 =  954
19 x 53 = 1007
20 x 53 = 1060
```