

TP 2 – PREMIERS PROGRAMMES EN PYTHON**Préambule :**

1. Ouvrir une session sur l'ordinateur en entrant Identifiant / Mot de passe ;
2. Lancer Spyder en cherchant Spyder dans la barre de recherche ;
3. Effacer le texte dans la fenêtre de gauche et écrire :

```
#NOM
#Prénom
#TP2

#Exercice 1
```

Objectifs de ce TP :

- Ecrire les premiers algorithmes ;
- Manipuler les entiers, les flottants et les booléens ;
- Découvrir les structures conditionnelles `if...else...` et `while ...`;

Toutes les réponses aux questions doivent être écrites dans le programme qui sera à rendre dans le dossier Restitution de devoirs de votre classe avant 18h.

Exercice 1 : le premier programme en Python

1. Recopier ces lignes de code dans votre programme.

```
print("bienvenue dans votre premier programme python")
nom=input("Comment vous appelez-vous?")
print("bonjour %s !" % nom)
age=input("quel age avez-vous?")
age=int(age)
age= age + 1
print("l'annee prochaine, vous aurez %d ans" % age)
```

2. Enregistrer votre programme dans le dossier restitution de devoir de votre classe. Le nom du fichier sera `NOM_TP2.py`.
3. Lancer le programme en appuyant sur F5. Tester le programme dans la console en bas à droite.

Les commentaires sont des phrases en français ajoutées dans le code permettant de comprendre ce que fait chaque ligne du programme. On les ajoute à la suite de chaque ligne en démarrant par le caractère #.

4. Ecrire à la suite de chaque instruction un commentaire expliquant ce que réalise chaque ligne du programme comme l'exemple ci-dessous.

```
print("bienvenue dans votre premier programme Python"># affiche la phrase...
```

Cahier des charges :

Votre programme devra vous demander votre année de naissance, calculer votre âge puis afficher la phrase « Vous avez xx ans » avec xx votre âge.

5. Compléter votre programme pour réaliser ce que demande le cahier des charges. Tester votre programme.
6. Ajouter au bout de chaque ligne un commentaire expliquant ce que fait chaque instruction.

Exercice 2 : la recette des cookies

Recette de cuisine : "Pour 12 cookies environ, battre 115 g de beurre mou avec 120 g de sucre en poudre jusqu'à ce que le mélange devienne clair. Ajouter 1 oeuf et 1 cuillère à café d'extrait de vanille. Ajouter 175g de farine, et mélanger le

tout avec 200 g de pépites de chocolat noir. Sur les plaques garnies de papier sulfurisé, déposer des petites boules de la préparation bien espacées, et faire cuire 10 minutes à 180 degrés Celsius. Laisser refroidir avant de déguster."

1. Ecrire à la suite de l'exercice 1 le commentaire `#Exercice 2`.
2. Ecrire les lignes de code qui demande à l'utilisateur le nombre de douzaine de cookies souhaitées et qui affiche en retour la quantité de chaque ingrédient : commencer par créer les variables des ingrédients puis procéder à l'affichage séquentiel des quantités.
3. Tester le programme (sélectionner les lignes puis F9).

Vous remarquez que l'écriture naïve de ce petit code est assez répétitive. Nous allons utiliser un nouveau type de variable afin d'obtenir un code plus efficace.

4. Recopier ces lignes de code à la suite de votre programme.

```
ingredient=["g de beurre", "g de farine", "g de sucre","g de pépites de chocolat"," oeuf(s)"," cuillère(s) à cafe de vanille" ]
base=[115,175,120,200,1,1]
n=int(input("combien de douzaines ?"))
print("il faut:")
for i in range(len(base)):
    phrase=str(base[i]*n)+ingredient[i]
    print(phrase)
```

5. Tester le programme (sélectionner les lignes puis F9).
6. Quel est le type des nouvelles variables `ingredient` et `base` ?
7. Combien de fois s'effectue la boucle des 3 dernières lignes ?
8. Ajouter au bout de chaque ligne des commentaires expliquant ce que fait chaque instruction.

Exercice 3 : le jour de la semaine

Ecrire un programme qui calcule le numéro du jour de l'année à partir du numéro de la semaine et du numéro du jour dans la semaine. Le programme affichera par exemple :

```
Numero de semaine? 2
Numero jour de la semaine entre 1 et 7? 3
-> Jour numero 10
```

1. Ecrire à la suite de l'exercice 1 le commentaire `#Exercice 3`.
2. Ecrire votre code.
3. Tester le code (sélectionner les lignes puis F9).

Exercice 4 : instruction conditionnelle

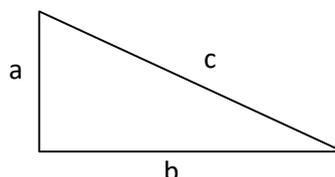
1. Ecrire à la suite de l'exercice 1 le commentaire `#Exercice 4`.
2. Recopier le code suivant en respectant les indentations :

```
z=int(input("z?"))
if (z>0):
    print("entier strictement positif")
```

3. Tester le programme avec des valeurs d'entiers >0 ou >0 :
4. Ecrire au bout de chaque ligne un commentaire expliquant ce que fait chaque instruction.
5. Compléter le programme pour qu'il affiche « entier strictement négatif » si c'est le cas.
6. Compléter le programme pour qu'il affiche « entier nul » si c'est le cas.

Exercice 5 : triplet Pythagorien.

On considère un triangle rectangle de côtés a , b et c :



1. Ecrire à la suite du code le commentaire `#Exercice 5`.
2. Ecrire, sous la forme d'une équation $f(a,b,c)=0$, la condition pour laquelle le triangle est rectangle.

3. Ecrire un programme qui demande à l'utilisateur les longueurs a,b et c puis affiche f(a,b,c).
4. Tester votre programme (sélectionner les lignes +F9).
5. Compléter votre programme pour afficher « Triplet Pythagoricien ! » si c'est a,b,c vérifient $f(a,b,c)=0$ ou « Triplet non Pythagoricien ! » dans le cas contraire.
6. Tester votre programme avec (a,b,c) = (3,4,5).
7. Tester votre programme avec (a,b,c) = (1,4,6).

Exercice 6 : Divisible ou pas ?

1. Ecrire à la suite du code le commentaire #Exercice 6.
2. Ecrire un code permettant demandant 2 entiers à l'utilisateur et qui renvoie True si le premier est divisible par le second et False dans le cas contraire.
3. Tester votre programme (sélectionner les lignes puis F9).

Exercice 7 : résolution du trinôme du second degré

Cahier des charges :

Vous devez écrire un programme qui demande à l'utilisateur les coefficients a, b et c d'un trinôme du second degré et qui renvoie les solutions de l'équation suivantes :

$$a.x^2 + b.x + c = 0$$

1. Ecrire à la suite du code le commentaire #Exercice 7.
2. Vous aurez besoin de la fonction $\sqrt{\quad}$. Taper `import numpy as np` dans la console en bas à droite.
3. Taper `np.sqrt(9)` dans la console et observer le résultat.
4. Ecrire votre programme.
5. Tester votre programme avec (a,b,c) = (1,2,1), il doit renvoyer la racine double -1.
6. Tester votre programme avec (a,b,c) = (2,-1,-6), il doit renvoyer les racines réelles -1.5 et 2.
7. Tester votre programme avec (a,b,c) = (1,1,1), il doit afficher « pas de solution réelles ».

Exercice 8 : année bissextile

Le pseudo-code suivant décrit comment déterminer si une année est bissextile ou non :

```
#Programme qui permet de vérifier les années bissextiles
PROGRAMME bissextile
#Déclaration des variables
VAR
année:entier,
encore:entier
#Début de programme#
DEBUT
encore=1
TantQue encore=1 Faire :
    Afficher "Saisir l'année"
    annee=valeur saisie
    Si reste(annee/4)==0 et reste(annee/100)!=0 Faire :
        Afficher "Année bissextile"
    Sinon si reste(annee/400)==0 Faire :
        Afficher "Année bissextile"
    Sinon Faire :
        Afficher "Année non bissextile"
    FinSi
    Afficher "Recommencer ? 1=oui 0=non"
    encore=valeur saisie
FinTantQue
FIN
```

1. Ecrire à la suite du code le commentaire #Exercice 8.
2. Traduire ce pseudo-code en python en sachant que la structure TantQue s'écrit :


```
while condition :
    instruction1
    instruction2
    ...
```
3. Tester votre programme avec 2022 et 2024.