

Les Variables : (Nombres, Booléennes, Chaines de caractères) :

1. Créer un nouveau projet sur PyCharm, et saisir le script suivant et l'enregistrer sous le nom tp1.py

```
a = 5
print ("a = ", a)

b = 5.50
print ("b = ", b)

c = 5,50,14
print ("c = ",c)

texte="Mon texte"
print (texte)

print ("type de a: ",type(a))
print ("type de b: ",type(b))
print ("type de c: ",type(c))
print ("type de texte: ",type(texte))
```

2. a. Écrire un programme, qui définit 3 variables : une variable de type **texte**, une variable de type **nombre entier**, une variable de type **nombre décimal** et qui affiche leur **type**.

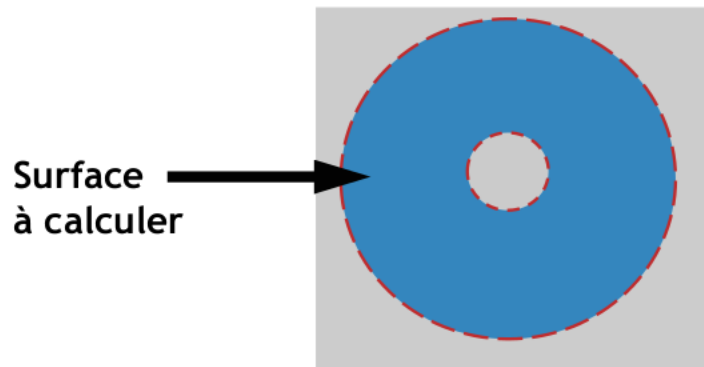
b. Affecter **dans une même ligne** les 3 variables précédemment définies.

3. Écrire un programme qui, à partir de la saisie d'un rayon et d'une hauteur, calcule le volume d'un cône droit :

$$V = \frac{1}{3} \times \pi \times r^2 \times h.$$

4. Une machine découpe dans une plaque, des disques circulaires de rayon **rExt**, percés d'un trou circulaire de rayon **rInt** avec rInt < rExt et ne débordant pas du disque.

Quelle est la surface d'un disque découpé ?



- (a) trouver les données
- (b) effectuer les calculs
- (c) afficher le résultat

-
5. Écrire un programme qui affiche le type du résultat des instructions suivantes:
- a=3
 - a= =3
-
6. Écrire un programme, qui ajoute une chaîne de caractères à un nombre entier (Exemple la chaîne "le chat" et le nombre 3 pour donner le chat 3), et puis renvoyer la taille de chaîne.
-
7. Écrire un programme qui réalise la saisie d'un nombre entier puis affiche la valeur ainsi saisie et son type. Essayer de dépasser la taille maximale des entiers. Expliquer.
-
8. Ecrire un programme qui transforme un nombre de base décimale vers la base binaire, et puis renvoyer son adresse mémoire.

Les Structures de données : les listes, les tuples et dictionnaire**I. Les listes**

9. Écrire un programme **liste_langages.py** qui initialise et affiche la liste dont les éléments sont : **Python, Java, PHP, C, C++, Français, SQL, Arabe et PLSQL**
10. Donner le nombre d'éléments de la liste.
11. Afficher la liste de manière inversée.
12. Afficher la liste de manière triée.
13. Ajouter le langage **Assembleur** dans la liste,
14. Supprimer les langages qui n'ont aucune relation avec la programmation informatique. Et puis afficher la liste après la suppression.
15. Donner le nombre des éléments de la liste après la suppression.
16. Afficher le nombre de caractères de chaque élément de la liste.
17. Vider les éléments de la liste.

II. Les tuples

18. Écrire un programme **tuple_bio.py** qui initialise et affiche le tuple dont les éléments sont : **ADN, ARN, Protéine, Virus, Bactérie**
19. Donner le nombre d'éléments du tuple.
20. Déterminer si l'élément **Corona** spécifié est présent dans le tuple.
21. Renvoyer la position de l'élément **Protéine**.

22. Afficher le nombre de caractères de chaque élément du tuple.

23. Supprimer complètement le tuple.

III. Les dictionnaires

24. Écrire un programme **dictionnaire.py** qui initialise et affiche le dictionnaire dont les éléments sont : **votre nom, votre prénom, votre âge, votre spécialité**

25. Parcourir les valeurs du dictionnaire.

26. Parcourir les clés du dictionnaire.

27. Modifier la valeur d'**âge** en réduisant 2ans.

28. Ajouter l'élément **date d'obtention du Bac** au dictionnaire.

29. Supprimer l'élément **votre spécialité**.

30. Vider le dictionnaire.

Les Structures de contrôle : les conditions et les boucles

I. Les conditions

- 31.a. Écrire un programme **conditions.py** qui demande de saisir 2 valeurs numériques et qui affiche la plus petite des 2 valeurs.
 - b. Dans le même programme précédant, afficher la plus grande des 2 valeurs déjà saisis dans la première question.
 - c. Dans le même programme précédant, afficher si les deux valeurs sont égaux.
- 32.Écrire un script **longueur_chaine.py**, qui demande de saisir 2 chaînes de caractères et qui affiche la plus grande et la plus petite des 2 chaînes (celle qui a le plus et moins de caractères).

II. Les boucles

- 33.Écrire un programme, qui affiche 9 fois "Je suis un bon programmeur en Python " à l'aide de l'instruction for.
- 34.Écrire un programme qui affiche la table de multiplications de « 2 » (de 0 jusqu'à 10) en affichant : $2 * 0 = 0$.
- 35.Écrire un programme qui affiche un joli sapin de Noël, dont la taille est donnée par l'utilisateur. Exemple pour une taille de 12 lignes :

```

      ^
     ^^^
    ^^^^^
   ^^^^^^
  ^^^^^^^
 ^^^^^^^^
^^^^^^^^^
^^^^^^^^^
^^^^^^^^^
^^^^^^^^^
^^^^^^^^^
^^^^^^^^^
^^^^^^^^^

```

Les Fonctions

36. Ecrire un script **fonctions.py**, qui demande à l'utilisateur de saisir 2 valeurs numériques. Et puis s'assurer est-ce l'utilisateur a bien saisi des valeurs numériques en utilisant les conditions.
37. Ajouter la fonction **somme** qui calcule et affiche la somme des deux valeurs saisies dans la question (1).
38. Ajouter la fonction **multiplication** qui calcule et retourne la multiplication des deux valeurs saisies dans la question (1) sans les afficher en utilisant **return**.
39. Afficher le résultat de la question (3).
40. Ajouter la fonction **division** qui calcule la division des deux valeurs saisies dans la question (1). Et puis ajouter la condition pour empêcher la division par Zéro.
41. Ajouter la fonction **table_multiplication** qui affiche la table de multiplication (du 0 jusqu'à 10) d'une variable numérique donnée en paramètre, comme mentionné ci-dessous.

```
3 * 0 = 0
3 * 1 = 3
3 * 2 = 6
3 * 3 = 9
3 * 4 = 12
3 * 5 = 15
3 * 6 = 18
3 * 7 = 21
3 * 8 = 24
3 * 9 = 27
3 * 10 = 30
```

42. Travailler la question 6 en utilisant **lambda** et **map**.