

OBJECTIFS : Les objectifs de ce premier TP sont :

- de manipuler les opérations // et %
- de se familiariser avec les types de variables : str, int ou float
- d'écrire quelques programmes utilisant les fonctions *input()* et *print()*

DOCUMENT A RENDRE : Ce travail est évalué. Vous en rédigerez un compte-rendu numérique en utilisant un logiciel de traitement de texte (*Word ou Libre Office*). Le fichier constitué sera appelé *tp1.doc* ou *tp1.odt* et devra être transféré en fin d'activité **par l'intermédiaire** du site *nsibranly.fr* : se loguer et transférer en utilisant le code **tp1** . Ce compte-rendu contiendra :

- les réponses aux différentes questions posées,
- les captures d'écran **des morceaux de codes écrits et celles des résultats des exécutions** données dans le shell. Pour faire ces captures, utiliser *l'Outil Capture d'écran* de Windows.

Pour commencer, ouvrir dans Pyzo un fichier nommé « *premiers_programmes.py* » .

1. Manipuler les opérations // et %

- Réaliser la division euclidienne de 11 par 2 à la main. Quel quotient et quel reste obtient-on ? Dans la console exécuter l'opération `11 // 2` , puis `11 % 2` . Retrouve-t-on les résultats calculés à la main ?
- Une entreprise a produit 14037 masques en une journée. Ces masques sont vendus dans des sachets contenant 15 masques. Combien de sachets ont été remplis et combien de masques reste-t-il ? (ne pas faire les calculs à la main mais en utilisant les opérateurs // et % de python).

2. Se familiariser avec les types de variables

Exécuter les instructions suivantes dans le *shell* ou dans l'éditeur et écrire le résultat de l'exécution :

- `print(2020)`
- `print("2020")`
- `pi = 3.14159`
`type(pi)`
- `pi = "nombre pi"`
`type(pi)`

- `pi = 3.14159`
`texte = "le nombre pi est "`
`type(pi)`
`type(str(pi))`
`print(texte+str(pi))`
- `saisie = input("Entre le nombre pi : ")`
`pi = float(saisie)`
`print("Le nombre pi est :", pi)`
- `a = 5`
`b = "10"`
`print(a+b)`
`print(str(a)+b)`
`print(a+int(b))`
`print(a+float(b))`
- `a = "bon"`
`b = "jour"`
`print(a+b)`

Questions :

- 1- Que permet de faire la fonction native de python : `str()`
- 2- Que permet de faire la fonction native de python : `int()`
- 3- Que permet de faire la fonction native de python : `float()`
- 4- Peut-on ajouter 2 chaînes de caractères ?

3. Ecrire des programmes utilisant les fonctions `input()` et `print()`

Ecrire le code python des programmes présentés ci-dessous.

Ajouter à chaque fois, **un commentaire** (précédé d'un #) précisant ce que ce bout de code réalise.

- Ecrire un programme qui saisit l'année de naissance après affichage du texte
`"Donne moi ton année de naissance : "`
et qui affiche ensuite le message :
`"Tu as eu ou tu auras encore ans en 2022"`

- Ecrire un programme qui saisit une valeur de somme d'argent en € pour ensuite afficher la même somme convertie en \$US (Voir le cours actuel du \$US sur internet).
Pour l'écriture du message de sortie, utiliser une structure du type :

```
somme_en_euro = .....  
somme_en_dollars =  
print(somme_en_euro , " € font " , somme_en_dollars , " $US")
```
- Ecrire un programme qui saisit le prix HT d'un produit pour ensuite en afficher le prix TTC.
Pour l'écriture du message de sortie, utiliser une structure du type :

```
prix_ttc = .....  
message = "Le prix TTC est de :" + str(prix_ttc) + " €"  
print(message)
```
- Ecrire un programme qui saisit le diamètre puis la hauteur d'un cylindre en cm, pour ensuite afficher le volume de ce cylindre en cm^3 .
- Ecrire un programme qui saisit l'âge de l'utilisateur en années entières (16 par exemple) pour ensuite afficher ce même âge en secondes.
- Ecrire un programme qui saisit l'âge de l'utilisateur en secondes, pour ensuite afficher ce même âge en années entières et jours additionnels (par exemple : 16 ans et 207 jours).