

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 ?

`11 // 2 = 5` et `11 % 2 = 1`

- 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).

`14037 // 15 = 935` et `14037 % 15 = 12`

Donc 935 sachets et il reste 12 masques

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)` ----- 2020
- `print("2020")` ----- 2020
- `pi = 3.14159`
`type(pi)` ----- 2020
- `pi = "nombre pi"`
`type(pi)` ----- <class 'float'>
- `pi = 3.14159`
`texte = "le nombre pi est "`
`type(pi)` ----- <class 'float'>
`type(str(pi))` ----- <class 'str'>
`print(texte+str(pi))` ----- le nombre pi est 3.14159
- `saisie = input("Entre le nombre pi : ")`
`pi = float(saisie)`
`print("Le nombre pi est :", pi)` ----- le nombre pi est 3.14 (si 3.14 a été saisi)
- `a = 5`
`b = "10"`
`print(a+b)` ----- entraine une erreur
`print(str(a)+b)` ----- "510"
`print(a+int(b))` ----- 15
`print(a+float(b))` ----- 15.0

- a = "bon"
b = "jour"
print(a+b) ----- "bonjour"

Questions :

- 1- Que permet de faire la fonction native de python : `str()`
Renvoie le nombre mis en argument sous forme de chaîne de caractère
- 2- Que permet de faire la fonction native de python : `int()`
Renvoie la chaîne de caractère mise en argument sous forme d'un entier
- 3- Que permet de faire la fonction native de python : `float()`
Renvoie la chaîne de caractère mise en argument sous forme d'un nombre réel
- 4- Peut-on ajouter 2 chaînes de caractères ? *oui, les chaînes de caractères sont fusionnées.*

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 une valeur de somme d'argent en € pour ensuite afficher la même somme convertie en \$US (Voir le cours actuel du \$US sur internet).

```
1 somme_en_euro = float(input("Somme en € : "))
2
3 somme_en_dollars = somme_en_euro / 0.84
4
5 print(somme_en_euro , "€ font " , somme_en_dollars , "$US")
6
```

En exécutant, on a :

```
>>> (executing file "<tmp 1>")
Somme en € : 10
10.0 € font 11.904761904761905 $US
```

- Ecrire un programme qui saisit le prix HT d'un produit pour ensuite en afficher le prix TTC.

```
1 prix_ht = float(input("Prix HT : "))
2
3 prix_ttc = prix_ht * 1.2
4
5 message = "Le prix ttc est de "+str(prix_ttc)+" €"
6 print(message)
```

En exécutant, on a :

```
>>> (executing file "prix_TTC.py")
Prix HT : 10
Le prix ttc est de 12.0 €
```

- 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 .

```
1 # saisi des valeurs
2 d = float(input("Diamètre en cm : "))
3 h = float(input("Hauteur en cm : "))
4 # calcul du volume
5 r = d / 2      # r est le rayon
6 volume = (3.14 * r**2) * h
7 # affichage du résultat
8 message = "Le volume est de "+str(volume)+" cm3"
9 print(message)
```

En exécutant, on a :

```
>>> (executing file "calcul_volume_cylindre.py")
Diamètre en cm : 2
Hauteur en cm : 4
Le volume est de 12.56 cm3
```

- Ecrire un programme qui saisit l'âge de l'utilisateur en années entières (16 par exemple) pour ensuite afficher ce même âge en secondes.

```
1 # saisi des valeurs
2 age = float(input("Entre ton âge : "))
3
4 # calcul en secondes
5 age = age * 365.25 * 24 * 60 * 60
6 # affichage du résultat
7 message = "Depuis ta naissance, il s'est déjà écoulé " + \
8           str(age) + " secondes"
9 print(message)
```

En exécutant, on a :

```
>>> (executing file "age_en_secondes.py")
Entre ton âge : 16.5
Depuis ta naissance, il s'est déjà écoulé 520700400.0 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).

```
1 age = float(input("Entre ton âge en secondes : "))
2
3 age = age / (60*60*24) # calcul en jours
4 annee = age // 365.25
5 reste = age % 365.25
6 # affichage
7 message = "Tu as "+str(annee)+" ans et "+str(reste)+" jours"
8 print(message)
```

En exécutant, on a :

```
>>> (executing file "age_en_annees.py")
Entre ton âge en secondes : 1000000000
Tu as 31.0 ans et 251.32407407407482 jours
```