



L'essentiel

- Chaque objet a un type :
 - ◊ **int** : Les entiers : ils sont de taille illimitée en Python.
 - ◊ **float** : Les nombres flottants. Attention aux erreurs d'arrondis avec eux.
 - ◊ **bool** : Les booléens que nous avons déjà présentés.
 - ◊ **str** : Les textes appelés chaînes de caractères.
- On peut, lorsque cela a un sens, convertir les types d'objets en utilisant les mots-clés cités au-dessus, que l'on appelle **constructeurs**.

```
>>> bool(0)
False

>>> bool(1)
True

>>> int("123")
123

>>> float(123)
123.0

>>> int(12.3)
12
```

```
>>> int("dix")
Traceback (most recent call last):
  File "<interactive input>", line 1, in <module>
ValueError: invalid literal for int() with base 10: 'dix'

>>> "12" + "3"
'123'

>>> "12" + 3
Traceback (most recent call last):
  File "<interactive input>", line 1, in <module>
TypeError: can only concatenate str (not "int") to str

>>> int("12")+int("3")
15
```

- La fonction **input** (*phrase*) affiche la chaîne de caractères *phrase*, attend la saisie de l'utilisateur et renvoie sa réponse sous forme d'une **chaîne de caractères**.
- Attention au format de sortie du **input**, il faudra peut-être effectuer des transformations de type avant de continuer votre code selon les besoins.
- On ne place presque jamais de **input** dans une fonction : on différencie la partie fonctionnelle (algorithmique) de la partie IHM : Interface Homme Machine.

Exercices du jour

Exercice 1: Corriger le script ci-dessous pour qu'il fonctionne comme prévu :

```
naissance = input("Donner votre année de naissance: ")
age = 2024 - naissance
print("Votre âge est:", age)
```

Exercice 2: Créer un script qui :

- ▶ génère un nombre *a* au hasard entre 0 et 10;
- ▶ demande à l'utilisateur de donner un nombre entier *b* entre 0 et 10;
- ▶ affiche "Gagné" si l'utilisateur a bien deviné le nombre *a*, ou indique si son nombre est trop grand ou trop petit

Exercice 3: On considère l'équation $ax^2 + bx + c = 0$. En reprenant l'exercice 3 de la séance 3 sur les tests, écrire un script qui demande les valeurs de ces 3 nombres a , b et c et affiche le nombre de solutions et les valeurs approchées de celles-ci. Recopier et compléter le tableau ci-dessous, avant de vérifier avec votre programme :

a	b	c	solutions
1	1	1	
1	2	1	
1	3	2	
2	-2	-4	
0	2	6	
0	0	6	
0	0	0	
0	0	0	

Exercices en autonomie

Exercice 4: Une urne contient b boules blanches, r boules rouges et v boules vertes. Écrire un script qui simule le tirage d'une boule dans l'urne : en entrée, la fonction reçoit les 3 entiers b , r , v . En sortie, elle renvoie une chaîne de caractères "Blanche", "Rouge" ou "Verte".

Exercice 5: Trouverez-vous l'argument à donner à la fonction `whaou(x)` pour qu'elle renvoie "Gagné !" ?

```
def whaou(x):  
    if 0.1 + x == 0.3 :  
        return "Gagné !"  
    else :  
        return "Perdu !"
```

Aide pour les exercices

Solutions des exercices