

03- Types des expressions (Programmation)



I – Types d'expressions

Pour récupérer le type d'une expression, on peut utiliser la fonction native `type()`.

Dans un premier temps, nous allons voir quatre types :

- Les **integers** : **type int** en Python. Ce sont les entiers.
- Les **nombres à virgules flottantes** : **type float** en Python. Ce sont des nombres

contenant une virgule. Attention, 5.0 est bien un flottant même si mathématiquement il faut juste 5.

- Les **chaînes de caractères** ou **strings** : **type str** en Python.
- Les **booléens** : **type bool**. Ce sont des expressions ne pouvant valoir que **True** ou **False**.

II – Strings : afficher des caractères

2.1 Création d'un string

Un string commence avec des guillemets doubles " et finit par des guillemets doubles " :
`"Hello World !"`

Mais on peut également utiliser des simples ' :
`'Hello World !'`

Attention : même type en ouverture et en fermeture.

2.2 Caractère d'échappement

L'antislash \ est le caractère d'échappement de Python dans les chaînes de caractères :

- `\t` veut dire tabulation.
- `\n` veut dire passage à la ligne.
- `\"` veut dire d'afficher " .
- `\\` veut dire d'afficher \ .

2.3 Fonction native print

La fonction native `print()` permet notamment (mais pas que) d'afficher un résultat dans la console à destination d'un humain.

```
>>> print("Nom\nPrenom\nAge")
Nom
Prenom
Age
```

2.4 Concaténation de strings

```
>>> "Cou" + "cou"
'Coucou'
```

```
>>> "Cou" * 2
'CouCou'
```

2.5 Compter les caractères

On utilise la fonction native `len()`, comme length en anglais, longueur.

```
>>> len('Coucou')
6
```

2.6 Tester la présence

On utilise le mot-clé `in` de cette façon :

```
>>> 'soja' in "Sucre ; huile ; eau"
False
```

```
>>> 'huile' in "Sucre ; huile ; eau"
True
```

III – Expressions booléennes

3.1 Booléens

Les booléens sont des données ne peuvent prendre que deux valeurs : **True** ou **False** en Python.

3.2 Expressions booléennes

Expression dont l'évaluation renvoie... un booléen.

```
'Est-ce que 5 est supérieur à 12 ?'
>>> 5 > 12
False
```

```
'Est-ce que 12 est supérieur ou égal à 5 ?'
>>> 12 >= 5
True
```

```
'Est-ce que 5 est égal à 6 ?'
>>> 5 == 6
False
```

```
'Est-ce que 5 est différent de 6 ?'
>>> 5 != 6
True
```

```
'Est-ce que Crocodile est plus long que Paquebot ?'
>>> len("Crocodile") > len("Paquebot")
True
```