

Type ou class des objets

En Python comme dans tous les langages informatiques, on distingue les types d'objets et en fonction de ces types d'objets, on propose des fonctions ou des méthodes qui leur sont propres.

```
>>> type(2)
<class 'int'>
>>> type(5.3)
<class 'float'>
>>> type("azer")
<class 'str'>
```

- int : pour `integer` désigne les entiers naturels,
- float : pour `flottants` désigne les nombres décimaux,
- str : pour `string` désigne les chaînes de caractères

Le codage des flottants posant problème en informatique, on fera particulièrement attention aux passages dans ce monde étrange :

```
>>> 10/2
5.0
>>> type(10/2)
<class 'float'>
>>> 5+10/2
10.0
```

Chaque type d'objets possède des opérations qui lui sont propres comme par exemple l'addition :

```
>>> 3+8
11
>>> 3+8.0
11.0
>>> "azer"+"ty" # on parle de concaténation de chaînes de caractères
'azerty'
```

Le mélange de différents types d'objets peut poser problème :

```
>>> 5+10/2
10.0
>>> 5+"aze"
Traceback (most recent call last):
  File "<pyshell>", line 1, in <module>
TypeError: unsupported operand type(s) for +: 'int' and 'str'
```

On peut modifier le type d'un objet à l'aide de commandes simples :

```
>>> int(7.9) # donne la partie entière et convertit en int
7
>>> float(6)
6.0
>>> int(7.0)
7
>>> str(2)
'2'
>>> int("23")
23
```

Quelques opérations à connaître :

Avec les int

```
>>> 3*5 # la multiplication
15
>>> 5**3 # les puissance
125
>>> 5/3 # la division décimale
1.6666666666666667
>>> 8/4 # attention, la division décimale fait passer dans les floats
2.0
>>> 8//3 # le quotient de la division euclidienne
2
>>> 8%2 # le reste de la division euclidienne
0
```

Avec les chaînes de caractères :

```
>>> "aze"+"rty"
'azerty'
>>> "aze"*5
'azeazeazeazeaze'
```