

Python - Bases

Fonctions de base

print()

Affiche du texte ou des variables à l'écran.

Exemples:

```
>>> print("Bonjour")
Bonjour (type => str)

>>> print(42)
42 (type => int)

>>> print("La somme de 4 + 3 est ", 7, ".")
La somme de 4 + 3 est 7
```

input()

Demande une entrée à l'utilisateur. Retourne une chaîne de caractères (str) .

```
nom = input("Quel est ton nom ? ")
# L'utilisateur entre une valeur qui est stockée dans 'nom'
```

La fonction `input` est une fonction qui renvoie quelque chose. Il faut donc bien penser à stocker dans une variable ce qu'elle renvoie.

type()

Renvoie le type de la variable passée en argument.

Exemple:

```
print(type(42))          # Affiche: <class 'int'>
print(type("Bonjour"))  # Affiche:
```

Les types de base et leurs opérateurs

int (entier)

- Opérateurs mathématiques:
 - Addition : `+`
 - Soustraction : `-`
 - Multiplication : `*`
 - Division (résultat à virgule) : `/`
 - Division entière : `//`
 - Modulo : `%`
 - Puissance : `**`

Exemples:

```
a = 10
b = 3

print(a + b) -> 13 | print(a * b) # 30
print(a - b) -> 7 | print(a / b) # 3.3333...
print(a // b) -> 3 | print(a % b) # 1
print(a ** b) ->
```

str (chaîne de caractères)

- Opérateurs sur les chaînes de caractères:
 - Concaténation : `+`
 - Répétitions : `*` (avec un entier)

Exemples:

```
prenom = "Alice"
nom = "Dupont"

print(prenom + " " + nom) # Affiche: Alice Dupont
print(prenom * 3)         # Affiche: AliceAliceAlice
```

bool

- Valeurs possibles : `True` , `False`
- Opérateurs logiques :
 - ET logique : `and`
 - OU logique : `or`
 - NON logique : `not`

Exemples:

```
x = True
y = False

print(x and y) # False
print(x or y)  # True
print(not x)   # False
```

- Comparaisons:
 - Égalité : `==`
 - Différent de : `!=`
 - Plus grand que : `>`
 - Plus petit que : `<`
 - Plus grand ou égal : `>=`
 - Plus petit ou égal : `<=`

Exemple:

```
a = 10
b = 20

print(a == b) # False
print(a < b)  # True
```