

1- Python – Opérateurs et Chaînes de caractères



Quentin

Pour plus d'informations ...
www.icuresearch.eu

Les Opérateurs Arithmétiques :

+

Addition

Types :
Entier, Réel,
Chaîne de
caractère

Exemple :
1 + 1 = 2
"a" + "b" = "ab"

-

Soustraction

Types :
Entier, Réel

Exemple :
2 - 1 = 1

*

Multiplication

Types :
Entier, Réel,
Chaîne de
caractère

Exemple :
2 * 5 = 10
2 * "a" = "aa"

**

Puissance

Types :
Entier, Réel

Exemple :
2 ** 5 = 32

/

Division

Types :
Entier, Réel

Exemple :
2 / 5 = 0.4

//

**Division
entière**

Types :
Entier, Réel

Exemple :
9 // 5 = 1

%

Modulo

Types :
Entier, Réel

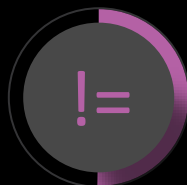
Exemple :
9 % 5 = 4

Les Opérateurs Logiques :



Égalité

Différence



**Infériorité /
Supériorité**

Supérieur, Inférieur,
Supérieur ou égal et
Inférieur ou égal

"ET" Logique

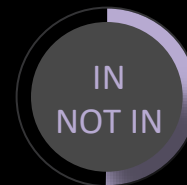


"OU" Logique

Symbole :
Pipe

**Présence ou Absence
dans une séquence**

Vérifier si une valeur
se trouve dans une
séquence.



Les chaînes de caractères :

Quelques fonctions :

Exemple de chaîne de caractères : str = "bonjour;au revoir"

Nombre de caractères
dans une chaîne

len(str) → 17

Récupérer une sous-
chaîne

str[2] → "n"
str[0:4] → "bonj"

Majuscule
Minuscule

Maj = str[0].upper() + str[1:]
→ "Bonjour;au revoir"
Maj[0].lower() + str[1:]
→ "bonjour;au revoir"

Séparer une chaîne dans
une liste selon un
séparateur

str.split(';') → ['bonjour', 'au
revoir']

F-string : Insertion
d'expression dans des
chaînes de caractère via
une syntaxe minimale

Name = "Patrick"
Age = 25
print(f"Votre nom est {Name}
et vous avez {Age} ans.")
→ "Votre nom est Patrick et
vous avez 25 ans."

Position de la première
occurrence d'une
chaîne

str.find("on") → 1
str.find("z") → -1

Remplacer une sous-
chaîne

str.replace("o", "j")
→ "bjnjur;au revjir"

Enlever les espaces à
droite et à gauche de la
chaîne

Espace = " bonjour "
Espace.strip()
→ "bonjour"

Compter le nombre
d'occurrence d'une
chaîne

str.count("o") → 3

R-string : Permet de ne
pas traiter '\ ' comme un
caractère
d'échappement quand
on veut une chaîne
contenant '\ ' comme un
caractère littéral

print(r'bonjour \n au
revoir')
→ 'bonjour \n au revoir'
print('bonjour \n au
revoir')
→ "bonjour
au revoir"

Module « re »

Permet d'appliquer des expressions régulières sur une chaîne ([Documentation](#))

Rechercher un motif dans un
texte → `re.search()`

```
Animaux = "Lion Tigre"
if re.search("Tigre", Animaux):
    print("Ok") → Ok
```

Récupérer une liste de toutes les
correspondances → `re.findall()`

```
str = "a=1.12 et b=2.72"
re.findall("[0-9]+\.[0-9]+", str)
→ ["1.12", "2.72"]
```

Remplacer une sous-chaîne par
une autre définie → `re.sub()`

```
str = "a=1.12 et b=2.72"
re.sub("[0-9]+\.[0-9]+", "2", str)
→ "a=2 et b=2"
```