Expressions numériques

2.1

Expression comportant un opérateur

2.1.1 DEFINITIONS

Une **expression** caractérise une séquence de **valeurs** associées à des **opérateurs**.

L'interpréteur Python évalue l'expression et fournit sa valeur associée. Cette action se nomme une **évaluation**.

Cas simple :

```
>>> 4 + 5
9
```

— 4 et 5 sont des valeurs.
— + est un opérateur.
— 4 + 5 est une expression.
— 9 est la valeur de l'évaluation.

Les opérations arithmétiques classiques sont symbolisées de cette façon en Python :
— + addition, − soustraction,
— ∗ multiplication, / division,
— ∗∗ puissance.

2.1.2 Priorité

L'évaluation d'une expression est **séquentielle et progressive** l'interpréteur Python effectue certaines opérations en priorité.

Voici **les priorités des opérateurs**, du plus prioritaire au moins prioritaire :

1. les parenthèses d'abord.
2. ensuite, l'opérateur de puissance **
3. puis, les opérateurs * /
4. enfin, les opérateurs + -
5. En cas d'égalité de priorité : le plus à gauche d'abord

Exemples

```
10 + (5 + 3) * 2  # parenthèses
10 + 8 * 2        # multiplication
10 + 16           # addition
26

5 + 10 / 2 * 3    # division
5 + 5.0 * 3       # multiplication
5 + 15.0          # addition
20.0
```

2.1.3 Quotient et reste de division

Rappels sur les divisions euclidiennes : il s'agit de diviser un nombre a par un nombre b, en trouvant :

— le **quotient**, c'est à dire le nombre de paquets égaux qu'on peut faire en divisant a par b.
— le **reste**, c'est à dire le nombre d'éléments restants après création des paquets.
— **Exemple** avec la division euclidienne de 10 par 5 : quotient de 2 et reste de 0.

$$\begin{array}{r|l} 10 & 5 \\ \hline -\,10 & 2 \\ \hline 0 & \end{array}$$

Opérateur Python // La syntaxe pour obtenir le **quotient** de la division entière de 10 par 5 est 10 // 5.

```
>>> 10 // 5
2
```

Opérateur Python % La syntaxe pour obtenir le **reste** de la division entière de 10 par 5 est 10 % 5.

```
>>> 10 % 5
0
```

Exemple

$$\begin{array}{r|l} 14 & 3 \\ \hline -\,12 & 4 \\ \hline 2 & \end{array}$$

```
>>> 14 % 3
4
```

```
>>> 14 % 3
2
```

Priorité Les opérateurs // et % possèdent la même priorité que la multiplication.

```
1407 % 5 // 2
2 // 2
1
```

```
1407 // 2 \% 5
703 % 5
3
```

2.2

Application à la résolution de problèmes

2.2.1 Extraire les chiffres

Chiffre u de l'unité

```
>>> 1234 % 10
4
>>> 489 % 10
9
```

Chiffre d de la dizaine

```
>>> 1234 // 10 % 10
3
>>> 489 // 10 % 10
8
```

Chiffre c de la centaine

```
>>> 1234 // 100 % 10
2
>>> 489 // 100 % 10
4
```

2.2.2 Position d'une case

Description du plateau

16 cases numérotées de 0 à 15.

```
#       C0 C1 C2 C3
# L0    00 01 02 03
# L1    04 05 06 07
# L2    08 09 10 11
# L3    12 13 14 15
```

Numéro de ligne

Il y a 4 cases par ligne et les cases commencent à 0.

Il suffit de récupérer le quotient de la division euclidienne du numéro de la case par 4.

```
>>> 4 // 4
1
>>> 7 // 4
1
```

Numéro de colonne

Il y a 4 cases par ligne et les cases commencent à 0.

Il suffit de récupérer le reste de la division euclidienne du numéro de la case par 4.

```
>>> 4 % 4  >>> 7 % 4
0          3
```

2.3

Dictionnaire et interface graphique

2.3.1 Définition d'un dictionnaire

Un dictionnaire est un **tableau associatif**. Il comporte des cases qui portent un nom et possèdent un contenu. On nomme :

— **clé (key)** le nom de la case.
— **valeur (value)** le contenu de la case.

Clé	"Alice"	"Bob"	"Charlie"
Valeur	13	8	12

2.3.2 Déclaration d'un dictionnaire

— Les éléments délimiteurs sont les accolades **{}**.
— On fournit un couple **(cle, valeur)** en utilisant la syntaxe Python suivante : **cle: valeur**.
— Les couples sont séparés par une **virgule**.

Exemple une ligne ou multiligne

```
ds = {"Alice": 13, "Bob": 8, "Charlie": 12}
```

```
ds = {
    "Alice": 13,
    "Bob": 8,
    "Charlie": 12
    }
```

2.3.3 Lecture, accéder à une valeur

On accède à une valeur en tapant le **nom du dictionnaire suivi de la clé entre crochets**.

```
>>> ds = {"Alice": 13, "Bob": 8, "Charlie": 12}
>>> ds["Bob"]
8

>>> ds["Bob l'éponge"]
KeyError: "Bob l'éponge"
```

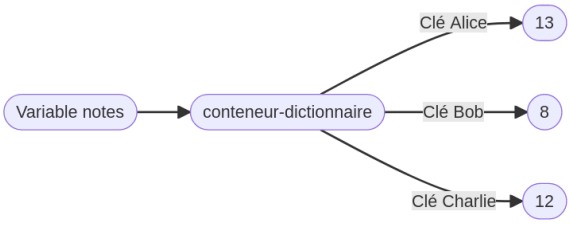

FIGURE 2.1 – Principe du dictionnaire