

Chapitre n° 2: Les Bases de la syntaxe

(Partie 03)

L1 Mathématiques appliquées

Introduction

- Les nombres jouent un rôle fondamental dans les mathématiques et dans la programmation. La maîtrise des outils numériques permet d'effectuer des calculs, de vérifier des résultats théoriques et d'explorer des concepts mathématiques de manière plus dynamique.
- Le langage **Python** permet d'effectuer facilement toutes sortes d'opérations arithmétiques et respecte les règles de priorité des calculs. Il propose également une bibliothèque mathématique puissante, **math**, qui offre des fonctions avancées telles que les racines carrées, les puissances et les logarithmes.

Ainsi, Python devient un véritable outil de calcul scientifique au service des mathématiques.

Les opérations arithmétiques العمليات الحسابية الأساسية

- Python permet d'effectuer les mêmes opérations qu'une calculatrice :

Opération	Symbole en Python	Exemple	Résultat
Addition (الجمع)	+	10 + 5	15
Soustraction (الطرح)	-	10 - 3	7
Multiplication (الضرب)	*	4 * 2	8
Division (القسمة)	/	9 / 2	4.5
Division entière (القسمة الصحيحة)	//	9 // 2	4
Reste de division (الباقي)	%	9 % 2	1
Puissance (القوة)	**	3 ** 2	9

Les opérations arithmétiques العمليات الحسابية الأساسية

Remarque :

- La **division /** donne toujours un **nombre à virgule** (عدد عشري float).
- La **division entière //** enlève la **partie décimale** (الجزء العشري).

Exemple en Python:

```
a = 10
b = 3
print ("a + b =", a + b)
print ("a - b =", a - b)
print ("a * b =", a * b)
print ("a / b =", a / b)
print ("a // b =", a // b)
print ("a % b =", a % b)
print ("a ** b =", a ** b)
```

Résultat

```
a + b = 13
a - b = 7
a * b = 30
a / b = 3.3333333333333335
a // b = 3
a % b = 1
a ** b = 1000
```

Priorité des opérateurs ترتيب العمليات

Comme en **mathématiques**, **Python** respecte un **ordre précis d'exécution** :

Priorité	Opérations	Symboles
1	Les parenthèses	()
2	La puissance	**
3	Multiplication, Division, Modulo, Division entière	*, /, %, //
4	L'addition et la soustraction	+, -

Exemple en Python:

```
x = 10 + 3 * 2
```

```
print(x)
```

16 (multiplication avant
addition)

```
x = (10 + 3) * 2
```

```
print(x)
```

26 (parenthèses d'abord)

```
x = 15 // 4 % 2
```

```
print(x)
```

1

La bibliothèque math مكتبة الرياضيات

- Pour des **calculs plus complexes**, Python dispose d'une **bibliothèque** (appelée **math**, qui regroupe de nombreuses **fonctions mathématiques**): **racines ,sinus logarithmes** etc.
- Pour l'utiliser, il faut l'**importer**: **import** math

Principales fonctions de math أهم دوال مكتبة الرياضيات

<code>math.sqrt(x)</code>	Square root	<code>math.sqrt(16)</code>	4.0
<code>math.pow(x,y)</code>	Power	<code>math.pow(2, 3)</code>	8.0
<code>math.exp(x)</code>	Exponential	<code>math.exp(1)</code>	2.71
<code>math.log(x)</code>	logarithm naturel (ln)	<code>math.log(10)</code>	2.30
<code>math.log10(x)</code>	logarithm base 10	<code>math.log10(100)</code>	2.0
<code>math.ceil(x)</code>	Ceiling	<code>math.ceil(4.2)</code>	5
<code>math.floor(x)</code>	Floor	<code>math.floor(4.9)</code>	4
<code>math.fabs(x)</code>	floating absolute	<code>math.fabs(-7)</code>	7.0
<code>math.factorial(n)</code>	$n! = n \times (n-1) \times \dots \times 1$	<code>math.factorial(6)</code>	720
<code>math.comb(n, k)</code>	$\binom{n}{k} = \frac{n!}{k! \times (n-k)!}$	<code>math.comb(5, 2)</code>	10

Principales fonctions de math أهم دوال مكتبة الرياضيات

Exemples en Python:

```
import math
```

```
x = 25
```

```
y = 3
```

```
print ("Racine carrée de x =", math.sqrt(x)) # الجذر التربيعي x
```

```
print ("x puissance y =", math.pow(x, y)) # x مرفوع للقوة y
```

```
print ("Logarithme de x =", math.log(x)) # اللوغاريتم الطبيعي x
```

Résultat obtenu:

Racine carrée de x = 5.0

x puissance y = 15625.0

Logarithme de x = 3.2188758248682006

Logique et Comparaison

Ces opérateurs sont fondamentaux pour créer des conditions. Ils retournent un résultat booléen (True ou False).

A. Les Opérateurs de Comparaison

Test	Symbole	Exemple (a=10,b=3)	Résultat
Égal à	==	10 == 3	False
Différent de	!=	10 != 3	True
Strictement Inférieur à	<	10 < 3	False
inférieur ou égal à	<=	10<=3	False
Strictement Supérieur à	>	10 > 3	True
Supérieur ou égal à	>=	10 > =3	True

B. Les Opérateurs Logiques

Indispensables pour combiner plusieurs conditions mathématiques (ex: $0 < x < 10$).

Opérateur	Nom	Description	Exemple
and	ET	Vrai si toutes les conditions sont vraies	$(x > 0) \text{ and } (x < 10)$
or	OU	Vrai si au moins une condition est vraie	$(x < 0) \text{ or } (x > 0)$
not	NON	Inverse le résultat (Vrai devient Faux)	$\text{not } (x == 10)$

Structures de Contrôle - Les Conditions

Le bloc **if / elif / else** permet d'exécuter du code selon des tests logiques.

Exemple : Étude du signe d'un nombre

```
x = 15
if x > 0:
    print("Le nombre est positif")
elif x < 0:
    print("Le nombre est négatif")
else:
    print("Le nombre est nul")
```

Résultat obtenu:

Le nombre est positif

Structures de Contrôle - Les Boucles

Les boucles permettent de répéter des instructions mathématiques.

1. La boucle for (Itération définie)

Utilisée pour répéter un calcul un nombre précis de fois, souvent avec range().

Exemple :

```
# Somme des entiers de 1 à 3
somme = 0
for i in range(1, 4):
    somme = somme + i**2
print("La somme est :", somme)
```

Résultat obtenu:

La somme est : 14

Structures de Contrôle - Les Boucles

2. La boucle while (Itération indéfinie)

Répète l'action **tant qu'une condition** est remplie.

Exemple : Trouver n tel que $2^n > 1000$

```
n = 0
```

```
while 2**n <= 1000:
```

```
    n = n + 1
```

```
print("Le plus petit n est :", n)
```

Résultat obtenu:

Le plus petit n est : 10

Conclusion (Résumé du cours)

- ✓ Les opérations arithmétiques permettent d'effectuer tous les calculs de base en **Python**.
- ✓ La priorité des opérateurs garantit le respect des règles mathématiques dans les calculs.
- ✓ Les opérateurs logiques et de comparaison permettent de formuler des conditions mathématiques.
- ✓ Les structures de contrôle (if, boucles for et while) facilitent l'automatisation des calculs répétitifs.
- ✓ Avec la bibliothèque math, Python peut effectuer des calculs plus complexes utilisés en sciences mathématiques.