

## Cours n° 3: Variables et types de données المتغيرات وأنواع البيانات

1 ère année Sciences économiques, de gestion  
et commerciales

# Plan du cours

- 1 Introduction
- 2 تتعريف المتغيرات وقواعد التسمية  
Définir et nommer les variables
- 3 أنواع البيانات  
Les types de données de base
- 4 كتابة برامج بسيطة  
Exercices pratiques
- 6 Conclusion

# Introduction

En Python, tout programme commence par la **manipulation de données**.  
التعامل مع البيانات.

Pour stocker et utiliser ces données, on utilise des **variables**.  
المتغيرات

# تعريف المتغيرات ؟ Qu'est-ce qu'une variable ?

- Une **variable** est un **nom que l'on donne à une valeur** pour pouvoir la réutiliser.  
اسم نمنه لقيمة معينة
- C'est comme une **boîte** dans laquelle on range une information.  
يشبه صندوقاً نضع بداخله معلومة

**Exemple simple :**

nom = "Yasmine"

age = 19

Ici :

On a une boîte appelée nom qui contient "Yasmine"

Et une autre appelée age qui contient 19

On pourra ensuite les afficher, les modifier ou les utiliser dans des calculs.

# تعريف المتغيرات ?

Dans la vie d'un gestionnaire, on manipule constamment des **valeurs** : **قيم مختلفة**

- ☞ **le prix d'un produit,** سعر المنتج
- ☞ **la quantité vendue,** الكمية المباعة
- ☞ **le revenu total,** إجمالي الإيرادات
- ☞ **ou encore le nom d'un client.** أو اسم الزبون

**Les variables** (المتغيرات) en Python permettent de représenter ces informations (المعلومات) sous forme **numérique** (عددية) ou **textuelle** (نصية) et de les traiter automatiquement (معالجتها تلقائياً).

# قواعد التسمية

- ☞ Doit commencer par une **lettre** ou un **underscore** (\_) بشرطه سفلية (\_) أو بحرف
- ☞ Pas d'espace ni de caractères spéciaux الرموز الخاصة الفراغات
- ☞ Sensible à la casse (Age ≠ age)

## Exemples valides et invalides :

nom\_client = "Ahmed" # ✓

2nom = "Ali" # X # Commence par un chiffre يبدأ برقم

nom client = "Sara" # X # Espace interdit يمنع الفراغ

# Règles de dénomination قواعد التسمية

## Remarques:

☞ Une variable peut être réutilisée : إعادة استخدام المتغير

`x = 5`

`x = x + 3`

`print(x)`

**Résultat:** 8

☞ Affectation multiple: إسناد عدة قيم في نفس السطر (إسناد متعدد)

`a, b, c = 1, 2, 3`

`print(a, b, c)`

**Résultat:** 1 2 3

## أنواع البيانات: Il existe 4 types principaux :

Type	Signification	Exemple	Description
str	chaîne de caractères سلسلة المحارف	"Bonjour", 'Ali'	Texte
int	Entier عدد صحيح	5, -10, 100	Nombre sans virgule
float	Décimal عشري	3.14, -0.5	Nombre à virgule
bool	Booléen منطقي	True, False	Valeurs logiques

### Exemples en Python:

```
x = 10      # int
y = 3.14    # float
nom = "Sara"  # str
est_etudiant = True # bool
```

# أنواع البيانات Les types de données de base

✓ **Vérifier le type :** `type()` : باستعمال الدالة : التحقق من نوع البيانات

**Exemple:** `print(type(x))`

**Résultat :** `<class 'int'>`

➤ **Conversions entre types (casting) :** التحويل بين الأنواع

# De str à int (int) (لكي نستعمله في الحساب، يجب تحويله إلى نوع)

`age = "20"`

`age = int(age)`

`print(age+5)`

**Résultat :** 25

إذا أردنا إضافة فاصلة عشرية

# De int à float إلى العدد، نحوله إلى نوع

`x = 5`

`x = float(x)`

`print(x)`

**Résultat :** 5.0

عند طباعة عدد داخل جملة  
نصية، يجب أولاً تحويله إلى  
نص.

# De float à str

`prix = 12.5`

`prix = str(prix)`

`print("le prix est de "+prix)`

**Résultat :** le prix est de 12.5

# أنواع البيانات Les types de données de base

## □ Les chaînes de caractères (str) سلاسل المحارف

- Le type str (abréviation de *string*) représente un **texte** écrit **النصوص** **entre guillemets** (" " ou ' ').
- On peut les **concaténer**, connaître leur **longueur** et **modifier** **leur forme** (ex. en majuscules).  
الدمج معرفة الطول.  
تغيير شكلها

### Exemples en Python:

```
prenom = "ali"
```

```
nom = "ben"
```

```
print(prenom + " " + nom) # Concaténation # دمج النصوص
```

```
print(len(prenom)) # Longueur de la chaîne # طول السلسلة
```

```
print(prenom.upper()) # Conversion en majuscules # تحويل إلى حرف كبيرة
```

## أنواع البيانات / Les types de données de base

## □ Les nombres

- **Le type int** (abréviation de *integer*) sert à représenter des  **nombres entiers**, positifs ou négatifs, sans virgule. Exemples:  
5, 0, -10, 2025      الأعداد الصحيحة      بدون فاصلة عشرية

## Exemples en Python:

année = 2025

température = -3

- **Le type float** (abréviation de *floating point number*) représente les nombres avec une partie décimale (virgule). Exemples : 3.14, -0.5, 2.0 الأعداد العشرية

## Exemples en Python:

prix = 12.5

poids = 65.8

# Les types de données de base أنواع البيانات

## □ Les booléens (bool) القيم المنطقية

Le type bool (abréviation de *boolean*) sert à représenter une valeur **vraie ou fausse**; True (vrai) ou False (faux).

### Exemples en Python:

```
est_ouvert = True
```

```
print("Le magasin est ouvert ?", est_ouvert)
```

### Résultat :

Le magasin est ouvert ? True

# كتابة برامج بسيطة — Exercices pratiques

## Exercices proposés :

### Ex n°1: Affichage d'informations personnelles

nom = "Omar"

عرض المعلومات الشخصية

age = 21

ville = "Jijel"

```
print("Nom :", nom)
```

```
print("Âge :", age)
```

```
print("Ville :", ville)
```

### Résultat:

Nom : Omar

Âge : 21

Ville : Jijel

# كتابة برامج بسيطة — Exercices pratiques

## Ex n°2: Afficher un texte avec une variable

nom = "Fatima"

عرض نص يحتوي على متغيرات

age = 19

```
print("Je m'appelle", nom, "et j'ai", age, "ans.")
```

### Résultat:

Je m'appelle Fatima et j'ai 19 ans.

## Exercices pratiques — كتابة برامج بسيطة

### Ex n°3: Interaction avec l'utilisateur التفاعل مع المستخدم

```
nom = input("Entrez votre nom : ")
```

```
age = input("Entrez votre âge : ")
```

```
print("Bienvenue", nom, "!")
```

```
print("Vous avez", age, "ans.")
```

#### Résultat:

Entrez votre nom : Yasmine

Entrez votre âge : 19

Bienvenue Yasmine !

Vous avez 19 ans.

## Conclusion (Résumé du cours)

- ✓ Une variable = espace mémoire qui stocke une valeur
- ✓ Types principaux : int, float, str, bool
- ✓ Règles de nommage simples mais importantes
- ✓ Les variables sont la base de tout programme Python