



Chapitre 1

Introduction à Python

Pour les étudiants en génie civil

Dr. Hachimi Dahhaoui
Université de Tlemcen

mardi 7 octobre 2025

Objectifs & Plan

Objectifs du chapitre

- Comprendre l'interpréteur Python
- Connaître les types de données de base
- Appliquer Python à des calculs de génie civil
- Réaliser des exercices interactifs

Plan du cours

1. Installation et interpréteur
2. Types de données (nombres, textes, listes)
3. Exercices et devinettes
4. Application en génie civil

Pourquoi Python ?

- Syntaxe simple et lisible
- Large écosystème de bibliothèques
- Portabilité (Windows, Linux, macOS)
- Parfait pour prototyper rapidement
- Gratuit et open source

Python en Génie Civil

- Calculs de structures (résistance des matériaux, RDM)
- Analyse des sols et géotechnique
- Optimisation des matériaux (béton, acier)
- Simulation de phénomènes physiques
- Automatisation de rapports et calculs

Installation & Outils

- Télécharger Python depuis python.org
- Utiliser un IDE (VS Code, PyCharm, Jupyter)
- Vérifier l'installation : `python --version`
- Installer les bibliothèques nécessaires (`pip install`)

Interpréteur vs Script

- Mode script : exécuter un fichier .py complet
- Mode interactif (REPL) : taper des commandes une par une
- Écrire du code dans un éditeur puis l'exécuter
- Utiliser `print()` pour afficher les résultats

```
$ python3 script.py  
<sortie du script>
```

```
$ python3  
>>> print('Bonjour')  
Bonjour
```

Démonstration interactive

```
>>> 2 + 3
5
>>> 'Python'.upper()
'PYTHON'
```

REPL signifie Read-Eval-Print-Loop :

- Read : lire la commande
- Eval : l'exécuter
- Print : afficher le résultat
- Loop : recommencer

Arguments en ligne de commande

`sys.argv` contient les arguments passés au script

`sys.argv[0]` : nom du fichier

`sys.argv[1]`, `sys.argv[2]` : arguments supplémentaires

```
import sys
print(sys.argv)
# Exécution : python script.py fichier.txt 2025
```


Environnement & Variables

- Python est dynamiquement typé : pas besoin de déclarer les types
- `sys.path` contient les chemins de recherche des modules
- On peut inspecter les variables avec `type()`
- Utiliser `PYTHONPATH` pour ajouter des répertoires

```
a = 5  
b = 3.2  
c = a * b  
print(c)
```

Encodage & Commentaires

- Python 3 utilise l'encodage UTF-8 par défaut
- Pas besoin de déclarer `# -*- coding: utf-8 -*-`
- Pour un encodage différent, ajoutez une déclaration au début
- Les commentaires commencent par `#` et ne sont pas exécutés

```
# -*- coding: iso-8859-15 -*-  
print('Exemple avec accents : façade')
```

Types de données

Nombres

Entiers (int), réels (float), complexes (complex)

Textes

Chaînes de caractères (str)

Listes

Séquences modifiables d'éléments (list)

Nombres & Opérateurs

- Addition : $a + b$
- Soustraction : $a - b$
- Multiplication : $a * b$
- Division : a / b (résultat float)
- Division entière : $a // b$ (entier)
- Modulo : $a \% b$ (reste)
- Puissance : $a ** b$

```
print(8 / 5)      # 1.6
print(8 // 5)     # 1
print(8 % 5)      # 3
print(2 ** 4)     # 16
```

Exemple : Masse volumique

Calcul de densité d'un matériau :

densité = masse / volume

Dans cet exemple, masse = 1500 g et volume = 1.2 dm³.

La densité est donc 1250 g/dm³

```
masse = 1500 # grammes  
volume = 1.2 # dm3  
densite = masse / volume  
print('Densité =', densite, 'g/dm3')
```

Exercice : Volume d'un cylindre

Consigne :

- Demander un rayon r et une hauteur h
- Calculer le volume : $V = \pi \times r^2 \times h$
- Afficher le résultat avec 2 décimales

```
import math
r = float(input('rayon (m) : '))
h = float(input('hauteur (m) : '))
V = math.pi * r**2 * h
print('Volume =', round(V, 2), 'm3')
```

Chaînes de caractères

- Définition : str = séquence de caractères
- Utiliser des guillemets simples ou doubles
- Chaînes multi-lignes avec triple quotes
- Les chaînes sont immuables

```
texte = 'Bonjour'  
multiligne = """Ligne1  
Ligne2"""  
print(texte, multiligne)
```

Chaînes : indexation & tranches

- Accéder à un caractère : `mot[i]` ($i \geq 0$)
- Dernier caractère : `mot[-1]`
- Sous-chaîne : `mot[a:b]` (b exclu)
- Omettre a ou b pour début/fin

```
mot = 'Python'  
print(mot[0])    # P  
print(mot[-1])   # n  
print(mot[1:4])  # yth
```


Immutabilité & méthodes

- Les chaînes ne peuvent pas être modifiées
- Pour remplacer un caractère : créer une nouvelle chaîne
- Méthodes utiles : `s.upper()`, `s.lower()`, `s.replace()`

```
mot = 'Python'  
# Erreur : mot[0] = 'J'  
mot = 'J' + mot[1:]  
print(mot)  
print('bonjour'.upper())
```

Exercice : Message personnalisé

Consigne :

- Demander le prénom de l'utilisateur
- Afficher : Bonjour, [prénom], bienvenue dans Python !

```
nom = input('Prénom : ')
print('Bonjour,', nom, 'bienvenue dans Python
!')
```

Listes : introduction

- Une liste est une séquence ordonnée d'éléments
- Définie avec des crochets : []
- Peut contenir des types variés (int, str, float, etc.)
- Indice commence à 0

```
fruits = ['pomme', 'banane', 'orange']  
print(fruits[0])  
print(len(fruits))
```

Manipulation de listes

- Ajouter un élément : `list.append(x)`
- Insérer : `list.insert(i, x)`
- Supprimer : `list.remove(x)` ou `list.pop(i)`
- Concaténer : `list1 + list2`

```
l = [1,2,3]
l.append(4)
print(l)
l.remove(2)
print(l)
```

Tranches de listes

- Obtenir une sous-liste : `l[a:b]`
- Omettre a ou b pour début/fin
- Copie de liste : `l[:]` • Pas (step) : `l[a:b:step]`

```
squares = [1,4,9,16,25]  
print(squares[1:4])  
print(squares[-2:])
```

Exercice : résistances

Créez une liste de résistances (MPa) : [23.5, 25.2, 24.9, 26.1]

- Remplacez la troisième valeur par 25.0
- Ajoutez 27.3 à la fin
- Calculez la moyenne des résistances

```
res = [23.5, 25.2, 24.9, 26.1]
res[2] = 25.0
res.append(27.3)
mean = sum(res) / len(res)
print('Moyenne =', round(mean,2), 'MPa')
```

Application : moyenne des résistances

Cette fonction calcule la moyenne des valeurs d'une liste de résistances :

`mean = sum(liste) / len(liste)`

Vous pouvez réutiliser cette fonction pour d'autres grandeurs (densité, contraintes...).

```
def moyenne(liste):  
    return sum(liste) / len(liste)  
  
res = [23.5, 25.2, 25.0, 26.1, 27.3]  
print('Moyenne =', round(moyenne(res), 2),  
      'MPa')
```

Devinette 1

Vous empilez 10 briques de 0,20 m de hauteur chacune.

Quelle est la hauteur totale ?

Réfléchissez, puis vérifiez avec Python !

```
n = 10
h = 0.20
hauteur_totale = n * h
print('Hauteur totale =', hauteur_totale, 'm')
```


Devinette 2

Une force de 3000 N est appliquée sur une section de 0,005 m².
Quelle est la contrainte ?

```
force = 3000  
section = 0.005  
stress = force / section  
print('Contrainte =', stress, 'Pa')
```

Devinette 3

**Vous devez couler 500 kg de ciment. Les sacs pèsent 50 kg chacun.
Combien de sacs faut-il ?**

```
poids_total = 500
poids_sac = 50
nb_sacs = poids_total // poids_sac
print('Nombre de sacs =', nb_sacs)
```

Quiz rapide

1. Python est-il compilé ou interprété ?
2. Quelle commande permet de quitter l'interpréteur ?
3. Que fait l'opérateur // ?
4. Quelle différence entre une chaîne et une liste ?
5. Peut-on modifier une chaîne ?

Synthèse & prochaines étapes

- Vous savez lancer Python et utiliser le REPL
- Vous maîtrisez les bases : nombres, chaînes, listes
- Vous pouvez écrire des scripts simples pour le génie civil
- Prochaine étape : structures de contrôle (if, for, while)

Applications Génie Civil

- Automatisation des calculs (sections, volumes, densités)
- Traitement des résultats d'essais (moyennes, écarts)
- Simulation et modélisation de structures
- Génération de rapports automatisés

Conclusion & appel à l'action

Python est un outil puissant pour l'ingénieur.

Continuez à expérimenter, tester et créer vos propres scripts.

Posez des questions et partagez votre code avec vos camarades.

Rendez-vous au prochain chapitre !

Merci !
Des questions ?

Contact : hachimi.dahhaoui@univ-tlemcen.dz

Ressources supplémentaires

- [Documentation officielle Python : docs.python.org](https://docs.python.org)
- [Tutoriels Real Python \(interpréteur, encodage\)](#)
- [Cours DataCamp sur Python](#)
- [Guides sur l'encodage et Unicode](#)

Fin du Chapitre 1

“Le meilleur moyen de prévoir l’avenir est de le créer.”
— Peter Drucker

À très bientôt pour la suite !