L3 TP AIS — PYTHON

Séance 1 · Les bases de Python

Fil rouge : construire un premier outil d'audit de serveur · Durée indicative : 2 h

Pourquoi Python en administration ?

L'administrateur système et réseau manipule des informations, vérifie des configurations et automatise des tâches répétitives. Python sera notre boîte à outils pour transformer des données en décisions.

Entrées	Traitement	Décision	Sortie
nom du serveur, RAM, service	calculer / convertir	conforme ou à vérifier	message ou rapport

Objectifs de la séance

- exécuter un script Python et afficher une information ;
- utiliser variables, types et conversions ;
- lire une donnée saisie au clavier ;
- écrire une condition avec if, elif, else ;
- combiner des tests et produire un mini-audit.

Règle du jeu — À chaque séance, nous enrichirons un outil réaliste : inventaire, analyse, puis exploitation de données système.

1. Premier programme : afficher

Un script est lu de haut en bas. La fonction print() affiche une valeur à l'écran.

```
print("Bonjour")
print("Bienvenue dans votre premier programme Python")
```

Commentaires

```
# Cette ligne explique le programme
print("Audit démarré")  # information affichée
```

À faire maintenant

1. Créer le fichier seance1.py.
2. Afficher votre prénom, puis « Audit AIS en cours ».
3. Modifier le message pour afficher le nom fictif d'un serveur.

À retenir — Les guillemets indiquent du texte. Python respecte la casse : print et Print ne désignent pas la même chose.

2. Variables et types

Une variable associe un nom à une valeur. Choisissez des noms explicites, en minuscules, avec des underscores.

```
nom_serveur = "web-01"
ram_go = 8
charge_cpu = 0.72
service_actif = True

print(nom_serveur)
print(type(ram_go))
```

Type	Exemple	Usage AIS
str	"web-01"	nom, IP, état
int	8	RAM, nombre de connexions
float	0.72	charge, pourcentage
bool	True	service actif ?

Opérations utiles

```
ram_go = 8
ram_go = ram_go + 4
print(ram_go)  # 12
```

Attention — = affecte une valeur ; == compare deux valeurs.

3. Entrées utilisateur et conversions

input() retourne toujours du texte (str). Il faut convertir si l'on veut calculer ou comparer un nombre.

```
nom = input("Nom du serveur : ")
ram_go = int(input("RAM disponible (Go) : "))
charge = float(input("Charge CPU (0 à 1) : "))
```

Fonction	Rôle	Exemple
str()	convertir en texte	str(42)
int()	convertir en entier	int("8")
float()	convertir en décimal	float("0.72")

Mini-test

Pourquoi ce programme pose-t-il problème ? Corrigez-le.

```
ram = input("RAM : ")
print(ram + 2)
```

Bon réflexe — Avant de convertir une saisie, précisez à l'utilisateur le format attendu.

4. Décider avec des conditions

Une condition exécute un bloc seulement si son test est vrai. L'indentation (4 espaces ou 1 tab) fait partie du programme.

```
ram_go = 4

if ram_go < 8:
    print("Mémoire insuffisante")
elif ram_go == 8:
    print("Configuration minimale")
else:
    print("Mémoire correcte")
```

Comparer des valeurs

Opérateur	Signification	Exemple
==	égal à	etat == "actif"
!=	différent de	port != 22
< >	inférieur / supérieur	charge > 0.85
<= >=	inférieur/supérieur ou égal	ram >= 8

Erreur classique — Un bloc mal indenté déclenche une erreur ou modifie le comportement. Indentez toujours de la même manière.

5. Combiner des conditions

Utilisez and quand tous les tests doivent être vrais ; or quand un seul suffit ; not pour inverser un booléen.

```
service_actif = True
charge_cpu = 0.91

if service_actif and charge_cpu > 0.85:
    print("Alerte : service actif mais charge élevée")
```

Expression	Résultat attendu
True and False	False
True or False	True
not True	False

Exercice rapide

Écrivez une condition qui affiche « Accès refusé » si le rôle est différent de "admin" ou si le compte est désactivé.

Lire une condition — On peut prononcer le code : « si le service est actif ET que la charge est supérieure à 0,85 ».

6. TP guidé · Audit d'un serveur

Contexte : vous recevez quelques indicateurs sur un serveur. Écrivez un script audit_serveur.py qui aide à repérer une situation à vérifier.

Étape 1 — Recueillir les données

4. Demander le nom du serveur.
5. Demander la RAM disponible (en Go) et la charge CPU (entre 0 et 1).
6. Demander si le service principal est actif : répondez oui ou non.

Étape 2 — Afficher un résumé

Affichez, par exemple : « Serveur web-01 · RAM : 8 Go · CPU : 72 % ».

Étape 3 — Première décision

Si la RAM est strictement inférieure à 8 Go, afficher un avertissement. Sinon, afficher que la mémoire est suffisante.

> **Indice —** Pour afficher un pourcentage à partir de 0.72, calculez charge_cpu * 100.

TP guidé · suite

Étape 4 — Ajouter les contrôles

7. Si le service n'est pas actif, afficher « CRITIQUE : service arrêté ».
8. Sinon, si la charge CPU est supérieure à 0.85, afficher « À SURVEILLER : charge élevée ».
9. Sinon, afficher « OK : indicateurs normaux ».

Étape 5 — Un verdict lisible

Votre script doit toujours afficher un seul verdict global. Ajoutez un message qui contient le nom du serveur.

Je vérifie mon travail

Jeu de test	Verdict attendu
web-01, RAM 16, CPU 0.35, oui	OK
db-01, RAM 16, CPU 0.92, oui	À SURVEILLER
api-01, RAM 4, CPU 0.40, oui	Avertissement RAM (au minimum)
web-02, RAM 16, CPU 0.20, non	CRITIQUE

Consigne — Testez plusieurs cas. Un programme « juste » avec un seul jeu de données peut être faux dans les autres cas.

7. Défi · score de sécurité

Ajoutez un score initial de 100. Retirez des points selon les indicateurs, puis affichez un niveau.

Situation	Points retirés
service inactif	50
RAM < 8 Go	20
charge CPU > 0.85	15

Niveaux proposés : score ≥ 80 : « bon » ; de 50 à 79 : « à surveiller » ; < 50 : « critique ».

Bonus

- Refuser une réponse autre que oui/non.
- Afficher un résumé final sur plusieurs lignes.
- Ajouter un test sur l'espace disque libre (en %).

But du défi — Chercher une solution claire et testable ; le code le plus court n'est pas toujours le plus compréhensible.

8. Travail personnel

Exercice 1 — Compte utilisateur

Demander un identifiant et un nombre de tentatives. Afficher une alerte à partir de 3 tentatives.

Exercice 2 — Port de service

Demander un numéro de port. Indiquer s'il est dans l'intervalle 1–65535. Bonus : signaler 22, 80 et 443.

Exercice 3 — Température

Demander une température CPU. Afficher « normale » (< 70), « élevée » (70–84), ou « critique » (≥ 85).

Débogage

Ce programme doit signaler une charge élevée. Il contient au moins trois erreurs. Identifiez-les puis corrigez-le.

```
charge = input("Charge CPU : ")
if charge > 0.85
print("Charge élevée")
else:
    print("Charge normale")
```

Pour la séance 2 — Nous répéterons des actions avec les boucles et nous manipulerons des collections de serveurs. Gardez votre fichier audit_serveur.py.