REPUBLIQUE DU NIGER

Ministère de l'Enseignement Supérieur, de la Recherche et de l'Innovation Technologique

École des Mines, de l'Industrie et de la Géologie (EMIG)

Introduction aux Systèmes Experts Travaux Pratiques N° 1 2025-2026 L3 Génie Logiciel

Rappel : Les bases du langage Python

Le langage Python : Il s'agit d'un langage de programmation interprété, facile à apprendre et polyvalent, largement utilisé en intelligence artificielle (IA), en science des données (Data Science), en développement web et en automatisation.

En Python, il n'est pas nécessaire de déclarer le type d'une variable. Le type est déterminé automatiquement. Par exemple :

```
age = 32 # variable de type entier (int)
taille = 1.70 # variable de type réel (float)
nom = "Amina" # variable de type chaîne de caractères (str)
cest_etudiant = True # variable de type booléen (bool)
print("Nom :", nom)
print("Âge :", age, "ans")
print("Taille :", taille, "m")
if cest_etudiant:
    print("Statut : Étudiant")
else:
    print("Statut : Non étudiant")
```

Une fonction permet de regrouper des instructions dans l'objectif dè les réutiliser. Ci-dessous la syntaxe générale :

```
def nom_fonction(parametre1, parametre2):
    instructions
    return resultat
```

Ci-dessous un exemple illustratif :

```
def somme(a, b):
    return a + b

resultat = somme(3, 5)
print(resultat)
```


Scanné avec CamScanner

Ci-dessous une fonction sans return :

```
def afficher_message():
    print("Bienvenue en Python")
```

Pour écrire une fonction, on utilise le mot clé def suivit du nom de la fonction et de deux (2) points (:).

Dans cette section, on va parler des Listes , des Tuples et des Strings. Ce sont des structures de données qui forment ce qu'on appelle des séquences. Par exemple :

#Listes

Liste1=[1,2,3] # taille 3

Liste2=["ali ", " sani " , "nana "] # taille 3

Liste3=[Liste1, Liste2] # taille 2

Listevide =[ ] # taille 0

Un tuple est une structure ordonnée et immuable (non modifiable). Ci-dessous un exemple de création des tuples :

Création d'un Tuple

Tuple =(4, 6, 7) # taille 3

Pour finir, on a les String, les chaines de caractères :

#String

prenom = "Amina " #Taille 5

Les Listes, les Tuples et les Strings sont des structures de données qui forment des séquences.

Une séquence est un ensemble d'élément ordonnée, c'est-à-dire, il y'a un premier, un deuxième, …

Dans une séquence, chaque élément a un index. Donc, les valeurs sont indexées. Grâce à ce système d'index, on peut accéder directement à certains éléments de nôtre Liste ou Tuple ou String. Par exemple, si je veux afficher le premier élément du prénom Amina (prenom = "Amina ") : print (" Le premier élément est : " prenom[0]).

Remarque 1 : Une Liste peut contenir des Listes et des Tuples. De même, un Tuple peut contenir des Tuples et des Listes. Par exemple :

Liste3=[Liste1, Liste2, Tuple] # Taille 3

Tupe2= (Liste1, Liste2, Tuple) # Taille 3

Tupe3= (Liste1, Liste2) # Taille 2

Remarque 2 : Si on veut avoir accès au dernier élément et qu'on ignore le nombre d'élément, que ça soit pour une Liste, un Tuple, un String, on utilise -1 comme index.

Scanné avec CamScanner

Remarque 3 : pour avoir accès à l'avant dernier élément, on utilise l'index -2. Ce genre de méthode est appelée indexing, qui permet d'avoir accès directement à un élément de notre séquence.

Dans la partie précédente, nous avons vu les séquences qui sont des structures de données dans lesquels, les valeurs sont rangées dans un ordre bien précis. On dit qu'elles ont un index (0, 1, 2, ...). Aujourd'hui, nous allons voir une autre structure de données qui est le dictionnaire. Contrairement aux séquences, le dictionnaire est une structure qui n'est pas ordonnée, plus précisément, les valeurs n'ont pas d'index, mais ils ont à la place ce qu'on appelle les clés. Donc, il faut voir un dictionnaire comme un ensemble d'affectation clef et valeur. Voici un exemple pour créer un dictionnaire :

traducteur={"chien": "dog"," chat ": "cat "}

print(traducteur)

Résultat : {'chien': 'dog', ' chat ': 'cat '}

Remarque : Dans un dictionnaire, chaque clef est unique.

Un autre exemple sur le dictionnaire :

inventaire={"banane": "500F"," pomme ": "700F "}

print(inventaire)

Résultat : {'banane': '500F', ' pomme ': '700F '}

Tout comme les listes, on peut créer un dictionnaire contenant d'autres dictionnaires. Par exemple :

dictionnaire={"dict_1": traducteur , " dict_2 ": inventaire }

print(dictionnaire)

Résultat :

{'dict_1': {'chien': 'dog', ' chat ': 'cat '}, ' dict_2 ': {'banane': '500F', ' pomme ': '700F '}}

En résumé, un dictionnaire stocke des données sous forme clé : valeur.

```
etudiant = {
    "nom": "Ali",
    "age": 23,
    "note": 15.5
}
```

On peut modifier et avoir accès à un dictionnaire :

```
print(etudiant["nom"])
etudiant["age"] = 24
```


Scanné avec CamScanner

Un tableau récapitulatif :

Structure	Ordonnée	Modifiable	Usage principal
Tuple	Oui	Non	Données fixes
Liste	Oui	Oui	Données dynamiques
Dictionnaire	Non (clé)	Oui	Association clé-valeur

Activité N° 1 : Réalisation d'un système expert simple pour le diagnostic du COVID-19

1. Créer un système expert de diagnostic simple basé sur des règles (Implémenter un mécanisme de raisonnement par chaînage avant)
2. Mettre en œuvre un raisonnement orienté vers les objectifs (Vérifier si une conclusion est vraie à partir de règles).
3. Associer un degré de certitude à une conclusion.

Activité N° 2 : Réalisation d'un système expert simple pour le diagnostic du cancer du sein

1. Créer un système expert de diagnostic simple basé sur des règles (Implémenter un mécanisme de raisonnement par chaînage avant)
2. Mettre en œuvre un raisonnement orienté vers les objectifs (Vérifier si une conclusion est vraie à partir de règles).
3. Associer un degré de certitude à une conclusion.

Activité N° 3 : Réalisation d'un système expert simple pour le diagnostic du paludisme

1. Créer un système expert de diagnostic simple basé sur des règles (Implémenter un mécanisme de raisonnement par chaînage avant)
2. Mettre en œuvre un raisonnement orienté vers les objectifs (Vérifier si une conclusion est vraie à partir de règles).
3. Associer un degré de certitude à une conclusion.

Scanné avec CamScanner