

TP4-Solution

Exercice 1

Copier, exécuter et analyser le code suivant :

```
# Liste d'articles scientifiques
articles = [
    {"title": "ChatGPT for Natural Language Tasks", "author": "OpenAI",
"year": 2023},
    {"title": "Tesla Autopilot Safety Analysis", "author": "Tesla Inc.",
"year": 2022},
    {"title": "Smart Grid Management using AI", "author": "A. Smith",
"year": 2023},
    {"title": "Siri and Alexa Performance Study", "author": "Apple &
Amazon", "year": 2021},
    {"title": "Google Translate Improvements", "author": "Google",
"year": 2020}
]
# Affichage des articles
for art in articles:
    print("Title:", art["title"])
    print("Author:", art["author"])
    print("Year:", art["year"])
    print("-"*30)
```

- 1) Ajouter une nouvelle information *"keyword"* (Exemple *"AI"*) pour chaque article.
- 2) Pourquoi utilise-t-on des dictionnaires pour représenter les articles ?
- 3) Ajouter un nouvel article à la liste.
- 4) Afficher uniquement les articles publiés après 2021.

Exercice 1 (2)

1)

```
for art in articles:  
    art["keyword"] = "AI"
```

2)

Parce qu'ils permettent de structurer les données avec des clés (title, author, year) → plus lisible et organisé.

3)

```
articles.append({  
    "title": "Deep Learning Advances",  
    "author": "Y. LeCun",  
    "year": 2024  
})
```

Exercise 1 (3)

4)

```
for art in articles:  
    if art["year"] > 2021:  
        print(art["title"])
```

Exercice 2

Recopier et exécuter le code suivant (*articles* est défini dans l'exercice précédent).

```
keyword = "AI"
results = [art for art in articles if keyword in art["title"]]
print("Articles trouvés avec", keyword, ":")

for res in results:
    print(res["title"])
```

- 1) Que signifie: *keyword in art["title"]*
- 2) Expliquer la ligne suivante: *results = [art for art in articles if keyword in art["title"]]*
- 3) Modifier le code pour rechercher plusieurs mots-clés.
- 4) Que se passe-t-il si aucun article ne correspond ?
- 5) Afficher aussi l'auteur et l'année.
- 6) Rendez la recherche insensible à la casse (par exemple : « AI », « ai » et « Ai » doivent être considérés comme identiques).
- 7) Pourquoi est-il important de filtrer les articles avant de les lire entièrement ?

Exercice 2

1)

Vérifie si le mot-clé existe dans le titre.

2)

Crée une liste contenant les articles dont le titre contient le mot-clé

3)

```
keywords = ["AI", "ChatGPT"]
results = [art for art in articles if any(k in art["title"]
for k in keywords)]
```

4)

La liste results sera vide [].
print(results)

Exercice 2 (2)

5)

```
for r in results:  
    print(r["title"], "-", r["author"], "-", r["year"])
```

6)

```
results = [art for art in articles if keyword.lower() in  
art["title"].lower()]
```

7)

Pour gagner du temps et se concentrer uniquement sur les articles pertinents.

Exercice 3

Recopier et exécuter le code suivant.

```
article_text = """
ChatGPT is a conversational AI that can answer questions, write texts,
and summarize articles.
It is widely used in education and research.
"""

summary = " ChatGPT is an AI that answers questions, writes and
summarizes texts."
print("Summary:", summary)
```

- 1) Quelle est la différence entre *article_text* et *summary* dans ce code ?
- 2) Comment pourrait-on générer un résumé automatique en Python ?
- 3) Pourquoi est-il important de résumer les articles avant de les utiliser ?
- 4) Que se passerait-il si le texte original est très long ?

Exercice 3

- 1)
article_text est le contenu original ; summary est une version courte.
- 2)
En utilisant des bibliothèques Python comme sumy ou gensim.
- 3)
Comprendre rapidement le contenu sans lire tout le texte.
- 4)
Utiliser un résumé automatique ; un résumé manuel peut être trop long.

Exercice 4

Etant donné le code Python suivant : (Remarque : *articles* est défini dans l'Exercice 1).

```
keywords = ["AI", "Smart Grid", "Translate"]
for kw in keywords:
    count = sum(1 for art in articles if kw in art["title"])
    print(f"Number of articles containing '{kw}':", count)
```

- 1) Que fait l'instruction `sum(1 for art in articles if kw in art["title"])` ?
- 2) Pourquoi le comptage des mots-clés est-il utile dans une revue de littérature ?
- 3) Comment peut-on afficher la liste des articles pour chaque mot-clé ?
- 4) Comment peut-on trier les mots-clés par nombre d'articles en ordre décroissant ?
- 5) Que se passe-t-il si un article contient plusieurs mots-clés à la fois ?

Exercice 4

1)

Compter combien d'articles contiennent le mot-clé

Explication simple

- `for art in articles` → on parcourt tous les articles
- `if kw in art["title"]` → on vérifie si le mot-clé est dans le titre
- `1` → on ajoute 1 chaque fois que c'est vrai
- `sum(...)` → additionne tous les 1

2)

Permet d'identifier les thèmes les plus fréquents et importants.

Exercice 4 (2)

3)

Solution 1

```
for kw in keywords:
    print("Mot-clé :", kw)
    for art in articles:
        if kw in art["title"]:
            print(" -", art["title"])
```

Solution 2

```
for kw in keywords:
    arts = [art["title"] for art in articles if kw in art["title"]]
    print(kw, ":", arts)
```

Exercice 4 (3)

4) Plusieurs Solution

```
counts = {}
# Compter les occurrences
for kw in keywords:
    count = 0
    for art in articles:
        if kw in art["title"]:
            count += 1
    counts[kw] = count

# Trier avec sorted
sorted_keywords = sorted(counts, key=counts.get, reverse=True)

# Affichage
for kw in sorted_keywords:
    print(kw, ":", counts[kw])
```

Exercice 4 (4)

Explication

```
counts = {}
```

On crée un dictionnaire vide pour stocker mot-clé, nombre d'articles

```
sorted_keywords = sorted(counts, key=counts.get,  
reverse=True)
```

On trie les mots-clés :

- ✓ `counts.get` : on regarde les nombres
- ✓ `reverse=True` : du plus grand au plus petit

Exercice 4 (5)

- 5)**
L'article sera compté plusieurs fois, une fois pour chaque mot-clé correspondant.