

TD1 – Premiers scripts et statistiques descriptives simples

Python pour économistes – Université de Lille, L2 Économie

Question fil rouge. Comment écrire un premier script Python pour décrire quelques indicateurs économiques et sociaux issus d'Our World in Data ?

Livrables. Un fichier `td1_nom_prenom.py` qui s'exécute sans erreur, affiche des statistiques simples, crée `esperance_vie_td1.png` et contient trois phrases d'interprétation.

Objectifs du TD

- Créer, sauvegarder et exécuter un fichier Python `.py`.
- Utiliser `print()`, des variables, des types simples et des listes.
- Produire un premier graphique très guidé avec `matplotlib`.
- Lire un premier message d'erreur et formuler une correction.
- Produire une statistique descriptive simple, un graphique et une courte interprétation.

Progression du TD

Les exercices 1 à 12 sont obligatoires. Le but principal est Python : variables, types, listes, calculs, affichage lisible, commentaires, premier graphique et premier débogage guidé. La statistique sert seulement de support.

Source des données

Les valeurs utilisées dans ce TD sont des extraits pédagogiques arrondis de graphiques Our World in Data. Pour l'instant, elles sont écrites directement dans le script afin de ne pas dépendre de `pandas`.

```
1 https://ourworldindata.org/grapher/life-expectancy.csv
2 https://ourworldindata.org/grapher/population.csv
```

Logique 1 – Décrire une petite liste d'indicateurs

PSEUDO-CODE

Compléter la logique avant d'écrire le code.

```
1 # Objectif : decire une liste de valeurs OWID
2 # Entree : une liste de pays et une liste de valeurs
3 # 1. Afficher les donnees de depart
4 # 2. Calculer le nombre d'observations
5 # 3. Calculer la moyenne
6 # 4. Reperer la valeur minimale et maximale
7 # 5. Afficher une phrase avec une unite
8 # Sortie : statistiques + interpretation courte
```

A. Dossier, script, exécution

Exercice 1 – Préparer un dossier de travail

OBLIGATOIRE

Créer un dossier TD1 contenant un script `td1_nom_prenom.py`. Le nom doit se terminer exactement par `.py` : c'est l'extension qui indique que le fichier contient du code Python.

```
1 mkdir TD1
2 cd TD1
3 pwd
4 python3 td1_nom_prenom.py
```

Procéder dans cet ordre :

- a. créer le fichier `td1_nom_prenom.py` ;
- b. écrire `print("test TD1")` ;
- c. sauvegarder le fichier ;
- d. exécuter le fichier avec la commande ci-dessus ;
- e. modifier le message, sauvegarder, puis relancer.

Dans le script, ajouter un commentaire qui explique la différence entre le dossier courant et le fichier exécuté. Si le terminal répond que le fichier est introuvable, vérifier d'abord le dossier courant avec `pwd` et le nom exact du fichier.

Exercice 2 – Premières variables économiques

OBLIGATOIRE

Dans le script, créer les variables suivantes puis afficher leur valeur et leur type.

```
1 pays = "France"
2 annee = 2021
3 esperance_vie = 82.5
4 population_millions = 67.7
```

Afficher ensuite une phrase lisible avec une f-string :

```
1 print(type(pays))
2 print(type(annee))
3 print(type(esperance_vie))
4 print(f"En {annee}, {pays} compte environ {population_millions} millions d'habitants.")
```

Exercice 3 – Prédire puis modifier une affectation

OBLIGATOIRE

Lire le fragment suivant sans l'exécuter.

```
1 population_millions = 67.7
2 population_millions = 68.0
3 print(population_millions)
```

Écrire en commentaire la valeur qui sera affichée, puis exécuter pour vérifier. Modifier ensuite seulement la deuxième ligne pour afficher 68.2. Cet exercice sert à comprendre qu'une nouvelle affectation remplace l'ancienne valeur.

Exercice 4 – Premier débogage guidé

OBLIGATOIRE

Copier chaque ligne fautive une par une dans le script, exécuter, lire le message, puis corriger. Ne pas laisser de ligne fautive dans le rendu final.

```
1 print("Bonjour")
2 print(pay)
3 print("Population : " + population_millions)
```

Pour chaque erreur, noter en commentaire :

- le type d'erreur affiché par Python ;
- la ligne indiquée ;
- l'hypothèse de correction.

Indice : les trois problèmes portent sur une parenthèse, un nom de variable et un mélange entre texte et nombre.

B. Listes et statistiques simples

Exercice 5 – Créer un extrait OWID simplifié

OBLIGATOIRE

Créer deux listes alignées :

```
1 pays = ["France", "Germany", "Spain", "Italy", "Poland", "Sweden"]
2 esperance_vie = [82.5, 80.6, 83.0, 82.9, 76.5, 83.1]
```

Les valeurs sont arrondies et exprimées en années. Afficher la première valeur, la dernière valeur et le nombre de pays. Avant d'exécuter, prédire ce que renvoient `pays[0]`, `esperance_vie[-1]` et `esperance_vie[10]`. Tester ensuite les deux premières expressions. Pour la troisième, expliquer en commentaire pourquoi elle produirait une `IndexError`.

Exercice 6 – Modifier une liste sans la désaligner

OBLIGATOIRE

Ajouter "Japan" à la liste `pays` et 84.5 à la liste `esperance_vie`. Afficher ensuite :

- le nombre de pays ;
- le nombre de valeurs d'espérance de vie ;
- la phrase Les deux listes ont la même longueur : True si les deux longueurs sont identiques.

Supprimer temporairement la valeur 84.5, relancer, puis expliquer en commentaire pourquoi deux listes alignées doivent toujours avoir la même longueur.

Exercice 7 – Moyenne, minimum, maximum

OBLIGATOIRE

À partir de la liste `esperance_vie`, calculer :

- le nombre d'observations avec `len(...)` ;
- la moyenne avec `sum(...) / len(...)` ;
- le minimum avec `min(...)` ;
- le maximum avec `max(...)`.

Afficher chaque résultat avec une unité claire : années.

Logique 2 – Passer d'une liste à un graphique

PSEUDO-CODE

```
1 # Objectif : représenter des valeurs par pays
2 # Entree : liste de pays, liste de valeurs
3 # 1. Créer une figure
4 # 2. Tracer une barre par pays
5 # 3. Ajouter un titre et une unité sur l'axe vertical
6 # 4. Sauvegarder la figure dans un fichier png
7 # Sortie : graphique lisible
```

Exercice 8 – Premier graphique avec matplotlib**OBLIGATOIRE**

On trace un graphique volontairement simple. Copier le code, l'exécuter, puis modifier le titre.

```
1 import matplotlib.pyplot as plt
2
3 plt.figure()
4 plt.bar(pays, esperance_vie)
5 plt.ylabel("Années")
6 plt.title("Espérance de vie - extrait OWID")
7 plt.xticks(rotation=45, ha="right")
8 plt.tight_layout()
9 plt.savefig("esperance_vie_td1.png", dpi=300)
```

Si Python affiche `ModuleNotFoundError: No module named 'matplotlib'`, le script est lisible mais la bibliothèque manque dans l'environnement utilisé.

Exercice 9 – Modifier le graphique sans changer les données**OBLIGATOIRE**

Reprendre le graphique précédent et modifier seulement :

- le titre ;
- le nom de l'axe vertical ;
- le nom du fichier sauvegardé, par exemple `esperance_vie_td1_version2.png`.

Avant de relancer, sauvegarder le script. Vérifier ensuite que deux fichiers `.png` distincts existent dans le dossier du TD.

Exercice 10 – Pourcentage d'évolution**OBLIGATOIRE**

Pour la France, on donne un extrait simplifié de l'espérance de vie :

```
1 france_2000 = 79.1
2 france_2021 = 82.5
```

Calculer la variation en années, puis le taux d'évolution en pourcentage :

$$\frac{\text{valeur finale} - \text{valeur initiale}}{\text{valeur initiale}} \times 100.$$

Afficher le résultat avec deux décimales.

C. Interpréter sans surcharger

Exercice 11 – Phrase d'interprétation

OBLIGATOIRE

Ajouter trois commentaires à la fin du script :

- a. une phrase qui donne la moyenne de l'extrait ;
- b. une phrase qui compare la France à cette moyenne ;
- c. une limite : pourquoi six pays ne suffisent pas à décrire le monde ?

Exercice 12 – Contrôle final du script

OBLIGATOIRE

Le script rendu doit contenir au minimum : une variable str, une variable int, une variable float, une liste, une moyenne, un graphique matplotlib, un pourcentage d'évolution, une f-string et trois phrases d'interprétation.

D. Entraînement facultatif

Pour aller plus loin 1 – Dix petits entraînements OWID

Choisir au moins deux questions si le parcours obligatoire est terminé.

- a. Ajouter "Japan" et une valeur d'espérance de vie dans les deux listes.
- b. Créer une liste de populations en millions pour les mêmes pays.
- c. Calculer la population totale de l'extrait.
- d. Calculer la part de la France dans cette population totale.
- e. Afficher pays[0] et esperance_vie[0] dans une même phrase.
- f. Tester `esperance_vie[0] > 80` et interpréter le booléen obtenu.
- g. Remplacer une valeur par une valeur manquante fictive 0, puis expliquer pourquoi cela fausse la moyenne.
- h. Créer une variable `source = "Our World in Data"` et l'afficher dans les résultats.
- i. Arrondir la moyenne avec `round(moyenne, 1)`.
- j. Écrire une phrase expliquant pourquoi les unités sont importantes.

À retenir

- Un script Python est un fichier sauvegardé, exécuté depuis un dossier courant.
- Une variable associe un nom à une valeur ; son type détermine les opérations possibles.
- Une liste ordonne plusieurs valeurs et s'utilise avec des indices.
- matplotlib permet de transformer une liste de valeurs en figure sauvegardée.
- Un message d'erreur se lit en partant du type d'erreur, puis de la ligne indiquée.