

TD2 – Boucles, conditions et proportions avec données OWID

Python pour économistes – Université de Lille, L2 Économie

Question fil rouge. Comment utiliser des boucles et des conditions pour classer des observations économiques simples ?

Livrables. Un fichier `td2_nom_prenom.py` contenant une boucle, une condition, une proportion, un graphique `co2_td2.png` et une interprétation économique courte.

Objectifs du TD

- Lire une liste de dictionnaires représentant des observations.
- Utiliser une boucle `for` pour répéter une opération.
- Écrire une condition `if/elif/else` et compter une proportion.
- Construire un graphique simple avec `matplotlib` à partir d'une boucle.
- Repérer les erreurs fréquentes de boucle, compteur et indentation.

Progression du TD

Les exercices 1 à 13 sont obligatoires. La notion statistique principale est la proportion. Les seuils utilisés sont pédagogiques : ils servent à apprendre `if`, `for` et les compteurs, sans aller trop vite vers un programme complet.

Source des données

Les valeurs de CO₂ par habitant sont des extraits pédagogiques arrondis issus d'Our World in Data.

```
1 https://ourworldindata.org/grapher/co-emissions-per-capita.csv
```

Logique 1 – Classer des observations

PSEUDO-CODE

```
1 # Objectif : classer des pays selon un indicateur
2 # Entree : liste de pays et valeurs de CO2 par habitant
3 # Pour chaque pays :
4 #     lire la valeur
5 #     comparer la valeur a un seuil
6 #     afficher une categorie
7 # Compter combien de pays sont dans chaque categorie
8 # Sortie : effectifs, proportions, interpretation
```

A. Représenter les données

Exercice 1 – Observer un dictionnaire seul

OBLIGATOIRE

Avant de manipuler plusieurs observations, lire une seule observation.

```
1 obs = {"pays": "France", "co2_pc": 4.7}
```

```
2
3 print(obs["pays"])
4 print(obs["co2_pc"])
```

Avant d'exécuter, prédire les deux affichages. Modifier ensuite la valeur de `co2_pc` à 5.1 et vérifier que seul le résultat affiché change.

Exercice 2 – Identifier une clé de dictionnaire

OBLIGATOIRE

Lire le fragment suivant avant de l'exécuter.

```
1 obs = {"pays": "France", "co2_pc": 4.7}
2
3 print(obs["pays"])
4 print(obs["population"])
```

Prédire quelle ligne fonctionne et quelle ligne produit une erreur. Exécuter ensuite le fragment, noter le type d'erreur en commentaire, puis remplacer `"population"` par une clé réellement présente dans le dictionnaire.

Exercice 3 – Créer une liste de dictionnaires

OBLIGATOIRE

Créer la liste suivante. Chaque dictionnaire représente une observation.

```
1 donnees = [
2     {"pays": "France", "co2_pc": 4.7},
3     {"pays": "Germany", "co2_pc": 7.9},
4     {"pays": "Spain", "co2_pc": 5.0},
5     {"pays": "Poland", "co2_pc": 8.4},
6     {"pays": "United States", "co2_pc": 14.9},
7     {"pays": "India", "co2_pc": 2.0},
8     {"pays": "China", "co2_pc": 8.0},
9     {"pays": "World", "co2_pc": 4.7}
10 ]
```

Afficher le nombre d'observations et la première observation.

Exercice 4 – Ajouter une observation courte

OBLIGATOIRE

Ajouter à la fin de `donnees` une observation pour `"Brazil"` avec `co2_pc = 2.3`. Relancer ensuite seulement deux contrôles :

- a. `print(len(donnees))`;
- b. `print(donnees[-1])`.

Vérifier que le reste du script pourra utiliser cette nouvelle observation sans modifier les boucles.

Exercice 5 – Parcourir les observations

OBLIGATOIRE

Avec une boucle `for`, afficher une phrase par pays :

```
1 for obs in donnees:
2     print(f"{obs['pays']} : {obs['co2_pc']} tonnes par habitant")
```

Observer que `obs` prend successivement la valeur de chaque dictionnaire de la liste. Modifier ensuite l'affichage

pour arrondir à une décimale.

B. Conditions et proportions

Exercice 6 – Classer avec une condition

OBLIGATOIRE

On utilise trois catégories pédagogiques :

- faible si `co2_pc < 5`;
- intermediaire si `5 <= co2_pc < 10`;
- eleve si `co2_pc >= 10`.

Compléter la boucle suivante pour afficher la catégorie de chaque pays.

```
1 for obs in donnees:
2     pays = obs["pays"]
3     co2 = obs["co2_pc"]
4     if co2 < 5:
5         categorie = ...
6     elif ...:
7         categorie = ...
8     else:
9         categorie = ...
10    print(pays, categorie)
```

Exercice 7 – Tester les seuils avant la boucle complète

OBLIGATOIRE

Avant d'appliquer le classement à tous les pays, tester quatre valeurs isolées : 4.9, 5.0, 9.9 et 10.0. Pour chacune, écrire en commentaire la catégorie attendue, puis vérifier avec quelques lignes utilisant `if/elif/else`.

Expliquer en une phrase pourquoi le cas 5.0 doit être classé `intermediaire` et non `faible`.

Exercice 8 – Compter une proportion

OBLIGATOIRE

Compter le nombre de pays dont les émissions dépassent 5 tonnes par habitant. Calculer ensuite la proportion :

$$\frac{\text{nombre de pays au-dessus du seuil}}{\text{nombre total de pays}} \times 100.$$

Compléter d'abord ce squelette :

```
1 compteur = 0
2 for obs in donnees:
3     if obs["co2_pc"] > 5:
4         compteur = compteur + 1
5
6 proportion = ...
```

Afficher une phrase du type : 62.5 % des observations sont au-dessus du seuil.

Logique 2 – Préparer des listes pour un graphique**PSEUDO-CODE**

```

1 # Objectif : tracer une valeur par pays
2 # Entree : liste de dictionnaires
3 # 1. Créer une liste vide pour les noms de pays
4 # 2. Créer une liste vide pour les valeurs de CO2
5 # 3. Parcourir les observations une par une
6 # 4. Ajouter le pays et la valeur dans les deux listes
7 # 5. Tracer puis sauvegarder un graphique
8 # Sortie : figure png

```

Exercice 9 – Graphique des émissions par habitant**OBLIGATOIRE**

Construire deux listes avec une boucle, puis tracer un graphique en barres.

```

1 import matplotlib.pyplot as plt
2
3 noms_pays = []
4 valeurs_co2 = []
5
6 for obs in donnees:
7     noms_pays.append(obs["pays"])
8     valeurs_co2.append(obs["co2_pc"])
9
10 plt.figure()
11 plt.bar(noms_pays, valeurs_co2)
12 plt.ylabel("Tonnes de CO2 par habitant")
13 plt.title("Emissions de CO2 par habitant - extrait OWID")
14 plt.xticks(rotation=45, ha="right")
15 plt.tight_layout()
16 plt.savefig("co2_td2.png", dpi=300)

```

Exercice 10 – Créer une fonction de classement**OBLIGATOIRE**

Compléter la fonction classe_co2(valeur). Elle doit renvoyer "faible", "intermediaire" ou "eleve".

```

1 def classe_co2(valeur):
2     if valeur < 5:
3         return ...
4     elif valeur < 10:
5         return ...
6     else:
7         return ...

```

La tester d'abord avec classe_co2(4.7), classe_co2(7.9) et classe_co2(14.9), puis l'utiliser dans une boucle pour afficher le classement de tous les pays.

Exercice 11 – Débogage : boucle, compteur, indentation**OBLIGATOIRE**

Les trois fragments suivants contiennent une erreur fréquente. Pour chacun, identifier le type d'erreur probable ou le résultat logique faux, puis proposer une correction.

```

1 # Fragment A
2 for obs in donnees:
3     print(obs["pays"])

```

```
4
5 # Fragment B
6 compteur = compteur + 1
7
8 # Fragment C
9 compteur = 0
10 for obs in donnees:
11     if obs["co2_pc"] > 5:
12         compteur = compteur + 1
```

Ne pas intégrer ces fragments fautifs au script final.

C. Interpréter

Exercice 12 – Question économique

OBLIGATOIRE

Ajouter quatre commentaires à la fin du script :

- quel pays de l'extrait a la valeur la plus élevée ?
- la France est-elle au-dessus ou au-dessous de la valeur mondiale ?
- quelle proportion de l'extrait dépasse 5 tonnes par habitant ?
- pourquoi ne faut-il pas confondre CO₂ par habitant et CO₂ total ?

Exercice 13 – Contrôle final

OBLIGATOIRE

Le script doit contenir au minimum : une liste de dictionnaires, une boucle for, une condition if/elif/else, un compteur, une proportion, une fonction, un graphique matplotlib et une interprétation.

D. Entraînement facultatif

Pour aller plus loin 1 – Dix petits entraînements

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

- Changer le seuil de 5 à 7 tonnes et recalculer la proportion.
- Créer une liste contenant seulement les pays classés élevé.
- Calculer la moyenne de co2_pc avec une boucle, sans sum().
- Ajouter un pays et vérifier que le code fonctionne encore.
- Afficher les pays au-dessous de la valeur mondiale.
- Compter séparément les pays faible, intermédiaire et élevé.
- Écrire une fonction au-dessus_seuil(valeur, seuil).
- Trier manuellement deux pays : lequel a la valeur la plus élevée ?
- Écrire une phrase sur la limite d'un classement par seuil.
- Expliquer pourquoi une proportion dépend de l'échantillon choisi.

À retenir

- Une boucle for permet d'appliquer la même logique à chaque observation.
- Une condition transforme une comparaison booléenne en décision.
- Un compteur doit être initialisé avant la boucle, puis modifié au bon niveau d'indentation.
- Un graphique peut se préparer en construisant explicitement les listes à afficher.
- Une proportion dépend toujours de l'ensemble d'observations choisi.