

Examen Python (2h) - L2 Économie

Matériel autorisé : vos notes de cours/TP (pas d'Internet, pas d'IA).

Rendez votre code en `.py`.

Conseil gestion du temps : A (20 min) - B (35 min) - C (35 min) - D (30 min).

Consignes

- Donnez des **réponses courtes et précises**. Pour le code, commentez brièvement (#).
- Quand un **résultat numérique** est demandé, précisez l'**unité** (si pertinent) et l'arrondi.
- Créez un fichier `Exam_NOM_prenom.py`.

Partie A - Fondamentaux Python (6 points)

A1. (1 pt) Que renvoient les expressions suivantes ? Expliquez en 1 courte phrase.

```
9/4, 9//4, 9%4, "Py"*4, "statistiques"[2:7], bool("")
```

A2. (1 pt) Dans un terminal, que font les commandes : `ls -l`, `cd dossier/` ? (1 phrase chacune).

A3. (1 pt) Un cas où un **script** est préférable à l'**interpréteur**, et un cas inverse (1 phrase par cas).

A4. (1 pt) Soit `s="transition"`. Donnez : `s[1:5]`, `s[-4:]`, `s[::-2]` et expliquez brièvement la logique.

A5. (2 pt) Affichez "Le produit coûte 8,50 euros" à partir de `prix=8.5` en utilisant une **f-string** (2 décimales).

Partie B - Listes, dictionnaires, petites fonctions (7 points)

B1. (2 pts) **Moyenne simple**. Écrivez une fonction `moyenne(L)` qui :

- renvoie la moyenne des valeurs numériques de `L`,
- renvoie `None` si la liste est vide.

Testez-la avec : `[10, 12, 15, 9]`.

B2. (2 pts) **Nettoyage de texte**. À partir de :

```
phrase = "Les données économiques sont utiles, mais parfois complexes !"
```

Écrivez du code qui :

- met la phrase en minuscules,
- remplace `,. ; : ! ?` par des espaces,
- découpe en mots,
- affiche le **nombre total de mots**.

B3. (1.5 pt) **Tri simple**. À partir de :

```
produits = [
    {"nom": "riz", "prix": 2.4},
    {"nom": "pates", "prix": 1.9},
    {"nom": "quinoa", "prix": 4.2}
]
```

Donnez une seule ligne Python pour trier la liste **du moins cher au plus cher**.

B4. (1.5 pt) Filtrer des codes postaux. À partir de :

```
codes = ["59000", "75001", "59A00", " 34000 ", "4400"]
```

Produisez une nouvelle liste contenant uniquement les codes composés de **exactement 5 chiffres**.

Partie C - CSV, pandas, stats & graphiques (4 points)

Téléchargez à nouveau la base de données du TP4 `emissions_aviation.csv`. Importez le fichier CSV.

C1. (3 pt) Donnez le **minimum**, le **maximum**, et l'**étendue** (max-min)

C2. (0,5 pt) Standardisez la variable :

$$z_i = \frac{x_i - \text{moyenne}}{\text{écart-type}}$$

Affichez les **3 pays avec les plus fortes valeurs standardisées**.

C3. (0,5 pt) Créez une nouvelle colonne indiquant si l'émission d'un pays est **supérieure à la médiane** (1) ou non (0). Tracez un **diagramme en barres** du nombre de pays dans chaque catégorie (0 et 1) avec titres et axes explicites.

Partie D - Réflexion avec Python (3 pts)

D1. (1 pt) Utilisez une boucle `for` pour afficher les multiples de 3 strictement inférieurs à 30 et inférieurs à 90.

D2. (2 pts) Calculez la somme des entiers **impairs** entre 1 et 200 (inclus) à l'aide d'une boucle `for` et d'une condition `if`.