

DS — Python pratique (BTS SIO 1re année, Bloc B3 Cybersécurité)

La rigueur, la rédaction, le soin et le typage des fonctions sont évalués sur 2 points.

Si un de ces points n'est pas satisfait, **2 points seront retirés de la note**.

Toutes les fonctions doivent être testées. Si une fonction n'est pas testée, l'intégralité des points ne sera pas comptabilisée.

Rappels autorisés

- `int(...)`, `str(...)`, `input(...)`, `print(...)`
- Concaténation de chaînes: `"bonjour" + " monde"` → `"bonjour monde"`, `"A" + "B"` → `"AB"`, `"B" + "A"` → `"BA"`
- Pour tester qu'un caractère appartient à un ensemble (chiffres, lettres, spéciaux), parcourir la chaîne de référence avec une boucle et comparer les caractères.
- Pour vérifier qu'un caractère est dans une chaîne de caractères (mot clef **in**): `caractere in chaine` -> `"r" in "renard"` → `True`
- On peut accéder au caractère à l'indice `i` de la chaîne `chaine` avec `chaine[i]`, rappel: les indices commencent à 0. -> `"renard"[0]` → `"r"`, `"renard"[1]` → `"e"`
- Pour obtenir la taille d'une chaîne de caractères, on peut utiliser la fonction `len(chaine)`

Contexte

Les renards de la réserve doivent créer des identifiants simples pour suivre leurs terriers et messages. On vous demande d'écrire quelques fonctions utilitaires sur les chaînes de caractères.

Partie A — Fonctions à compléter (2 exercices)

A1 — Occurrences d'une lettre

Compléter `nb_occurrences(texte, lettre)` qui renvoie le nombre d'occurrences de `lettre` dans `texte`.

```
def nb_occurrences(texte: str, lettre: str) -> int:
    compteur = 0
    for c in texte:
        if ...:
            compteur = ...
    return ...
```

A2 — Contient une voyelle ?

Compléter `contient_voyelle(nom)` qui renvoie `True` si `nom` contient au moins une voyelle parmi `aeiouyAEIOUY`, sinon `False`.

```
def contient_voyelle(nom: str) -> bool:
    voyelles = "aeiouyAEIOUY"
    for c in nom:
        if ...:
            return True
    return ...
```

Partie B — Fonctions à écrire (3 exercices)

B1 — Taille sans `len`

Écrire `taille_sans_len(chaine: str) -> int` qui renvoie le nombre de caractères de `chaine` **sans utiliser** `len`. Utiliser une boucle et un compteur.

Exercice B1 (2 pt)

Écrire `nettoyer_nom(nom)` qui renvoie `nom` où les espaces sont remplacés par `-` et les autres caractères sont conservés.

B3 — Identifiant renard valide

Écrire `identifiant_renard_valide(code: str) -> bool` qui renvoie `True` si:

- longueur ≥ 6
- caractères autorisés: lettres, chiffres, `-`, `_`
- contient au moins une lettre et au moins un chiffre
- ne contient pas d'espace

Vous pouvez réutiliser `contient_deux_types(...)` et écrire une petite fonction utilitaire si besoin.