

## 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)`

### Partie A — Code à compléter (8 pts)

#### Exercice A1 (1 pt)

Compléter `nb_points(texte)` qui renvoie le nombre de `.` dans `texte`.

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

#### Exercice A2 (1 pt)

Compléter `nb_arobase(texte)` qui renvoie le nombre de `@` dans `texte`.

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

### Exercice A3 (1.5 pt)

Compléter `contient_minuscule(texte)` qui renvoie `True` si `texte` contient au moins une minuscule (a-z).

```
def contient_minuscule(texte: str) -> bool:
    alphabet_min = "abcdefghijklmnopqrstuvwxyz"
    for c in texte:
        if ...:
            return True
    return ...
```

### Exercice A4 (1.5 pt)

Compléter `contient_separateur(texte)` qui renvoie `True` si `texte` contient `-` ou `_` ou `.`.

```
def contient_separateur(texte: str) -> bool:
    seps = "-_."
    for c in texte:
        if ...:
            return True
    return ...
```

### Exercice A5 (1.5 pt)

Compléter `identifiant_valide(identifiant)` qui renvoie `True` si:

- `@` n'apparaît pas
  - au moins une lettre ou un chiffre
  - caractères autorisés: lettres, chiffres, `-`, `_`, `.`
- Utiliser les fonctions précédentes.

```

def identifiant_valide(identifiant: str) -> bool:
    alphabet = "abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ"
    digits = "0123456789"
    if nb_arobase(identifiant) > ...:
        return ...
    a_lettre = False
    a_chiffre = False
    for c in identifiant:
        if c in alphabet:
            a_lettre = True
        elif c in digits:
            a_chiffre = True
        elif c in "-_." :
            ...
        else:
            return ...
    if a_lettre or a_chiffre:
        return ...
    return ...

```

### Exercice A6 (1.5 pt)

Compléter `score_identifiant(identifiant)` qui calcule un score:

- +2 si longueur  $\geq 7$
- +2 si `contient_separateur` est vrai
- +2 si `contient_minuscule` est vrai
- +2 si `nb_arobase(identifiant) == 0`
- +2 si `nb_points(identifiant)  $\geq 1$`

```
def score_identifiant(identifiant: str) -> int:
    score = 0
    if len(identifiant) >= ...:
        score = score + ...
    if contient_separateur(...):
        score = score + ...
    if contient_minuscule(...):
        score = score + ...
    if nb_arobase(...) == ...:
        score = score + ...
    if nb_points(...) >= ...:
        score = score + ...
    return ...
```

## Partie B — Fonctions à écrire (10 pts)

### Exercice B1 (2 pt)

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

### Exercice B2 (2 pt)

Écrire `sans_espace(email)` qui renvoie `True` si `email` ne contient aucun espace, sinon `False`.

Parcourir la chaîne et retourner `False` dès qu'un espace ( `' '` ) est rencontré; si aucun espace n'est trouvé, retourner `True`.

### Exercice B3 (2 pt)

Écrire `est_email_simple(email)` qui renvoie `True` si:

- contient au moins un `@`
- contient au moins un `.`
- ne contient pas d'espace

### Exercice B4 (2 pt)

Écrire `generer_token(nom, jour)` qui renvoie une chaîne `TK-` + deux premières lettres de `nom` en majuscules + `-` + `jour` répété 2 fois (ex: `jour=12` → `1212`).

### Exercice B5 (2 pt)

Écrire `force_estimation(identifiant)` qui renvoie `"faible"`, `"moyen"` ou `"fort"` selon `score_identifiant(identifiant)` (`<6`, `6-8`, `≥9`).