

## DS — Python pratique (Rattrapage 1h, BTS SIO 1re année, B3)

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`
- On peut accéder au caractère à l'indice `i` de la chaîne `chaine` avec `chaine[i]`
- Pour obtenir la taille d'une chaîne de caractères, on peut utiliser la fonction `len(chaine)`

### Partie A — Fonctions à compléter (2 exercices)

#### A1 — Compter les chiffres

Compléter `nb_chiffres(chaine)` qui renvoie le nombre de chiffres présents dans `chaine`.

```
def nb_chiffres(chaine: str) -> int:
    compteur = 0
    digits = "0123456789"
    for c in chaine:
        if ...:
            compteur = ...
    return ...
```

#### A2 — Caractères spéciaux

Compléter `contient_special(chaine)` qui renvoie `True` si `chaine` contient au moins un caractère parmi `!@#$$%^&*?`, sinon `False`.

```
def contient_special(chaine: str) -> bool:
    specials = "!@#$$%^&*?"
    for c in chaine:
        if ...:
            return True
    return ...
```

## Partie B — Fonctions à écrire (3 exercices)

### B1 — Taille du mot de passe

Écrire `taille_mot_de_passe(c: str) -> int` qui renvoie le nombre de caractères de la chaîne (sans utiliser `len(chaine)` ).

### B2 — Validation simplifiée d'un mot de passe

Écrire `est_motdepasse_valide(chaine: str) -> bool` qui renvoie `True` si:

- longueur  $\geq 6$
- contient au moins une lettre et au moins un chiffre
- ne contient pas d'espace

Vous pouvez réutiliser `nb_chiffres(...)` et écrire une petite fonction utilitaire `nb_lettres(chaine)` si besoin.

### B3 — Supprimer les espaces

Écrire `supprimer_espaces(chaine: str) -> str` qui renvoie une nouvelle chaîne où tous les espaces sont retirés.  
Construire la chaîne résultat par concaténation dans une boucle.