

🎵 **Mise en scène :** Tu travailles sur une application de streaming musical. Pour gérer les données des artistes et des chansons, tu vas utiliser Python. Chaque exercice te fait manipuler un type de variable différent.

Exercice 1 — Reconnais les types

Voici des données extraites d'une application de streaming. Pour chaque valeur, indique le type Python (`int`, `float`, `str`, `bool`, `list`, `dict`).

Valeur	Type Python
"Taylor Swift"	
1989	
4.8	
True	
["Anti-Hero", "Shake It Off", "Blank Space"]	
{"artiste": "Bad Bunny", "streams": 8.2e9}	

Exercice 2 — Opérations surprenantes

Un développeur teste son application. Prédis ce qu'affiche Python pour chaque instruction, **sans utiliser de calculatrice**. Attention aux types !

Partie A — Concaténations

💡 **Rappel :** la fonction `str()` convertit un entier (ou un flottant) en chaîne de caractères. **Exemple :** `str(42)` donne `"42"` (type `str` et non plus `int`).

Instruction	Résultat affiché
<code>>>> "Dua" + " " + "Lipa"</code>	
<code>>>> "Top " + str(50)</code>	
<code>>>> "Ha" * 3</code>	
<code>>>> "2" + "0" + "2" + "4"</code>	

Partie B — Calculs musicaux

Lis le programme ci-dessous ligne par ligne et complète le tableau.

```
duree_titre = 3.45          # durée en minutes
nb_titres = 12
artiste = "The Weeknd"
duree_album = duree_titre * nb_titres
description = artiste + " - " + str(nb_titres) + " titres"
succes = duree_album > 40
```

Variable	Valeur	Type
duree_album		
description		
succes		

Exercice 3 — Erreurs en studio

Un stagiaire a écrit le code ci-dessous, mais il y a **3 erreurs de type**. Retrouve-les, explique chaque erreur et propose une correction.

```
titre = "Flowers"
annee = 2023
note = 9.5

# Ligne 5 : afficher le titre et sa note
print(titre + note)          # erreur 1

# Ligne 8 : calculer la note sur 20
note_sur_20 = note / "2"     # erreur 2

# Ligne 11 : afficher le titre et l'année
titre_complet = titre + annee # erreur 3
```

Erreur 1 — Ligne : ____

Erreur 2 — Ligne : ____

Erreur 3 — Ligne : ____

★ Exercice 4 BONUS — Crée ta propre playlist

En Python, complète le code suivant pour représenter **une vraie playlist de ton choix**. Chaque ligne comporte un commentaire qui indique le type attendu.

```
nom_playlist = _____ # str : le nom que tu donnes à ta
playlist
nb_titres = _____ # int : le nombre de titres
duree_moy = _____ # float : durée moyenne d'un titre (en
min)
publique = _____ # bool : True si la playlist est
publique
titres = _____ # list : 3 titres de la playlist

# Calcule la durée totale et stocke-la dans une variable
duree_totale = _____
```

★ Question bonus — les f-strings

💡 Un **f-string** permet d'insérer directement une variable dans une chaîne, sans `str()`. Exemple :

```
artiste = "Beyoncé"
nb_albums = 7
print(f"{artiste} a sorti {nb_albums} albums.")
# Affiche : Beyoncé a sorti 7 albums.
```

En utilisant un f-string, écris une instruction `print()` qui affiche : **"Ma playlist « [nom] » dure [duree_totale] minutes."** en utilisant tes variables.
