CORRIGÉ · PYTHON

P1C5 - Lisez un code Python existant, manipulez des données et gérez des erreurs simples

Code final :

```
try:
    with open("menu.csv", encoding="utf-8") as file:
        for line in file:
            parts = line.strip().split(",")
            name = parts[0]
            price = parts[1]
            print(f"{name} — {price} €")
except FileNotFoundError:
    print("Menu introuvable")
```

Sortie attendue (fichier présent) :

```
Pizza — 12 €
Salade — 8 €
Burger — 10 €
Tiramisu — 6 €
```

Sortie si le fichier est absent :

```
Menu introuvable
```

Le `try / except` entoure le `with open(...)` : si le fichier est absent, Python lève une `FileNotFoundError` et exécute directement le bloc `except`, au lieu d'afficher une erreur brute. `.strip()` retire le saut de ligne en fin de ligne. Ici, le prix reste sous forme de `str` puisqu'on ne fait que l'afficher ; s'il fallait le sommer, il faudrait ajouter `float(price)`.

> 💡 Dans le projet, vous lirez souvent des données depuis un fichier ou une source externe. Le motif try lecture / except message clair évite qu'une erreur technique interrompe brutalement l'application côté utilisateur.