

## P1C5 - Explorez un starter code, gérez fichiers et erreurs en C#

### Code final :

```
try
{
    var lines = File.ReadAllLines("menu.csv");

    foreach (var line in lines)
    {
        var parts = line.Split(',');

        if (parts.Length < 2)
        {
            Console.WriteLine($"Ligne ignorée car mal formatée : {line}");
            continue;
        }

        var name = parts[0];
        var price = parts[1];
        Console.WriteLine($"{name} - {price} €");
    }
}
catch (FileNotFoundException)
{
    Console.WriteLine("Erreur : le fichier menu.csv est introuvable.");
}
catch (Exception ex)
{
    Console.WriteLine($"Impossible de lire le fichier : {ex.Message}");
}
```

### Sortie attendue (fichier présent) :

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

### Sortie si le fichier est absent :

```
Erreur : le fichier menu.csv est introuvable.
```

`File.ReadAllLines(...)` charge toutes les lignes en une seule opération et renvoie un `string[]`. Le `try / catch (FileNotFoundException)` entoure la lecture : si le fichier est absent, C# lève l'exception et le programme passe directement dans le bloc `catch`, au lieu d'afficher une erreur brute. Avant d'accéder à `parts[0]` et `parts[1]`, on vérifie que la ligne contient bien au moins deux colonnes.



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