

# tables\_avec\_guillemets

February 26, 2021

## 1 Tables avec guillemets

Le fichier de données à gérer contient quelques désagrément avec des guillemets qui gênent la lecture lorsqu'on applique les exemples donnés par la documentation de [pandas](#).

Voici l'exemple :

```
[1]: texte = """
      Libellé","Produit","Prix","Quantite"
      "L123","meuble","1000","1"
      "L321","portable","500","2"
      "L333","lampe","100","4"
      """

      with open("exemple_fichier.txt", "w", encoding="utf8") as f:
          f.write(texte)
```

Le paramètre *encoding* de la fonction `open` est utilisé pour préciser que le texte contient des caractères non ASCII, typiquement des accents. Mais pandas gère plutôt bien ce bruit.

```
[2]: import pandas
      df = pandas.read_csv("exemple_fichier.txt", encoding="utf8")
      df
```

```
[2]:  Libellé  Produit  Prix  Quantite
      0   L123   meuble  1000         1
      1   L321 portable   500         2
      2   L333   lampe   100         4
```

On vérifie que les variables numériques sont numériques :

```
[3]: df.dtypes
```

```
[3]: Libellé      object
      Produit      object
      Prix         int64
      Quantite     int64
      dtype: object
```

On peut décider de conserver les guillemets en spécifiant le paramètre *quoting* :

```
[4]: df2 = pandas.read_csv("exemple_fichier.txt", encoding="utf8", quoting=3)
      df2
```

```
[4]:  "Libellé"  "Produit"  "Prix"  "Quantite"
      0  "L123"    "meuble"  "1000"   "1"
      1  "L321"  "portable" "500"    "2"
      2  "L333"    "lampe"   "100"    "4"
```

```
[5]: df2.dtypes
```

```
[5]: "Libellé"    object
      "Produit"  object
      "Prix"     object
      "Quantite" object
      dtype: object
```

Et si jamais les virgules sont des points-virgules, il faut préciser le paramètre *sep* :

```
[6]: texte = """
      "Libellé";"Produit";"Prix";"Quantite"
      "L123";"meuble";"1000";"1"
      "L321";"portable";"500";"2"
      "L333";"lampe";"100";"4"
      """

      with open("exemple_fichier2.txt", "w", encoding="utf8") as f:
          f.write(texte)

      df3 = pandas.read_csv("exemple_fichier2.txt", encoding="utf8", sep=";")
      df3
```

```
[6]:  Libellé  Produit  Prix  Quantite
      0  L123    meuble  1000     1
      1  L321  portable   500     2
      2  L333    lampe   100     4
```

Et si jamais le fichier initial est très grand... On veut simplement lire les premières lignes :

```
[7]: df4 = pandas.read_csv("exemple_fichier2.txt", encoding="utf8", sep=";", nrows=2)
      df4
```

```
[7]:  Libellé  Produit  Prix  Quantite
      0  L123    meuble  1000     1
      1  L321  portable   500     2
```

Ou encore une lecture par morceau de 2 lignes :

```
[8]: reader = pandas.read_csv("exemple_fichier2.txt", encoding="utf8", sep=";",
      ↪ iterator=True, chunksize=2)
      for i, extrait in enumerate(reader):
          print("extrait",i)
          print(extrait)
```

```
extrait 0
  Libellé  Produit  Prix  Quantite
0  L123    meuble  1000     1
1  L321  portable   500     2
extrait 1
```

	Libellé	Produit	Prix	Quantite
0	L333	lampe	100	4

[9] :