

9 janvier 2024



# Astuce python



#TIPSDATA



C'est quoi la méthode query()  
avec  pandas ?



**TOUS LES MARDIS**

Une nouvelle astuce ou rappel en data !

# Explications

C'est l'expression qui permet de filtrer les données du DataFrame grâce à une ou plusieurs conditions.

Pour indiquer si on souhaite créer un nouveau DataFrame ou remplacer l'existant.

**DataFrame.query(expr, inplace=False, \*\*kwargs)**

Ce paramètre est lié à la fonction eval() et permet de définir le moyen de parser l'expression (parser) ou de définir le moteur (engine) d'interprétation de l'expression. Il y a d'autres possibilités à consulter dans la document d'eval().

<https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.query.html>



# Les données

```
import pandas as pd
import numpy as np

# Nombre de produits à générer
num_products = 100

# Génération de données aléatoires
np.random.seed(0)
product_ids = np.arange(1, num_products + 1)
product_names = ["Product_" + str(i) for i in product_ids]
categories = np.random.choice(['Electronics', 'Clothing',
                                'Furniture', 'Toys', 'Food'], num_products)
prices = np.random.uniform(10, 500, num_products).round(2)
quantities = np.random.randint(1, 50, num_products)
sales = np.random.randint(200, 1000, num_products)

# Création du DataFrame
retail_data = pd.DataFrame({
    'ProductID': product_ids,
    'ProductName': product_names,
    'Category': categories,
    'Price': prices,
    'Quantity': quantities,
    'Sales': sales
})
```

```
print(retail_data.head())
```

	ProductID	ProductName	Category	Price	Quantity	Sales
0	1	Product_1	Food	68.90	17	204
1	2	Product_2	Electronics	155.11	9	623
2	3	Product_3	Toys	68.18	2	548
3	4	Product_4	Toys	165.81	18	482
4	5	Product_5	Toys	212.99	49	786

# Obtenir les données de la catégorie "Electronics" qui ont un prix supérieur à 100



```
ex1 = retail_data.query("Category == 'Electronics' and Price > 100")  
print(ex1)
```

	ProductID	ProductName	Category	Price	Quantity	Sales
1	2	Product_2	Electronics	155.11	9	623
9	10	Product_10	Electronics	266.39	49	573
14	15	Product_15	Electronics	337.03	31	708
17	18	Product_18	Electronics	151.81	19	526
21	22	Product_22	Electronics	416.18	3	259
23	24	Product_24	Electronics	342.13	38	866
26	27	Product_27	Electronics	481.47	3	770
31	32	Product_32	Electronics	119.31	40	750
35	36	Product_36	Electronics	352.74	18	920
43	44	Product_44	Electronics	365.37	30	668
44	45	Product_45	Electronics	255.65	47	673
46	47	Product_47	Electronics	325.56	10	600
53	54	Product_54	Electronics	312.83	18	505
62	63	Product_63	Electronics	221.40	4	607
76	77	Product_77	Electronics	311.62	12	326
80	81	Product_81	Electronics	288.86	33	312
84	85	Product_85	Electronics	232.24	29	755
85	86	Product_86	Electronics	363.81	21	263
90	91	Product_91	Electronics	186.39	20	823



# Obtenir les données avec un prix de maximum 300 et d'une quantité de minimum 10



```
max_price = 300
min_quantity = 10

result = retail_data.query("Price <= @max_price and Quantity
>= @min_quantity")

print(result.head())
```

	ProductID	ProductName	Category	Price	Quantity	Sales
0	1	Product_1	Food	68.90	17	204
3	4	Product_4	Toys	165.81	18	482
4	5	Product_5	Toys	212.99	49	786
5	6	Product_6	Clothing	41.43	36	633
7	8	Product_8	Furniture	287.63	41	939