

19 septembre 2023

Astuce python

#TIPSDATA



La fonction reduce()



TOUS LES MARDIS

Une nouvelle astuce ou rappel en data!

Explications

La librairie de la fonction
reduce

Une fonction d'agrégation à
écrire

`functools.reduce(function, iterable[, initializer])`

La séquence (liste,
dictionnaire, etc.)

Possibilité de donner une
valeur d'initialisation

Exemple 1

Concaténer tous les caractères d'une liste

```
from functools import reduce

carac = ["d", "a", "t", "a"]

def concat_all(x, y):
    return x + y

text = reduce(concat_all, carac)

print(text)
```

✓ 0.0s

data

Exemple 2

Doubler tous les caractères d'une chaîne de caractères

On ne sait jamais, ça peut être utile...

```
from functools import reduce

text = "Data Engineering"

def double_carac(x, y):
    return x + y*2

new_text = reduce(double_carac, text, "")

print(new_text)
```

✓ 0.0s

DDaattaa EEnnnggiinneeeerriinnngg

Exemple 3

Additionner tous les nombres au carré d'une liste

```
from functools import reduce

list = [1,2,3,4]

def sum_all(x, y):
    return x + y**2

final = reduce(sum_all, list, 0)

print(str(final))
```

✓ 0.0s

Exemple 4

Agréger le CA des ventes au produit

```
from functools import reduce

# Liste des transactions
transactions = [
    {'product_id': 1, 'amount': 100.5},
    {'product_id': 2, 'amount': 50.0},
    {'product_id': 1, 'amount': 200.5},
    {'product_id': 3, 'amount': 30.0},
    {'product_id': 2, 'amount': 60.0},
    {'product_id': 3, 'amount': 30.0},
]

def aggregate_sales(total_sales, transaction):
    product_id = transaction['product_id']
    amount = transaction['amount']

    # Si le produit existe déjà dans le total, on ajoute le montant de
    # la transaction à son total.
    # Sinon, on crée une nouvelle entrée pour ce produit avec le
    # montant de la transaction.
    if product_id in total_sales:
        total_sales[product_id] += amount
    else:
        total_sales[product_id] = amount

    return total_sales

# Utilisation de reduce pour obtenir le total des ventes par produit.
# Il est nécessaire de lui fournir une valeur initiale qui est {}
total_sales_per_product = reduce(aggregate_sales, transactions, {})

print(total_sales_per_product)
# Résultat : {1: 301.0, 2: 110.0, 3: 60.0}
```