

The background is a dark charcoal grey. It is decorated with a variety of colorful, hand-drawn style lines and shapes. These include straight lines, curves, loops, and geometric forms like rectangles and circles in shades of blue, green, yellow, orange, red, and purple. Some shapes resemble stylized icons, such as a document with lines, a speech bubble, and a person silhouette. The overall effect is a vibrant, abstract pattern.

Yassmina Radi & Abdessamad Zgor

UTILISANT MONGODB AVEC PYTHON



OBJECTIVES

01 Connection entre MongoDB et Python

02 Opérations CRUD avec Pymongo

03 PdMongo -- une Shell Mongo pour les Data Analysts

CONNECTION A MONGODB AVEC PYTHON

MongoDB est une base de données NoSQL populaire, utilisée pour stocker des données sous forme de documents flexibles au format JSON. PyMongo est une bibliothèque Python qui permet d'interagir avec MongoDB. Voici une présentation des étapes pour se connecter à MongoDB et effectuer des opérations CRUD (Create, Read, Update, Delete) avec PyMongo.

PyMongo

PyMongo est une distribution Python comprenant des outils pour travailler avec MongoDB, et c'est la méthode recommandée pour interagir avec MongoDB depuis Python. Cette documentation vise à expliquer tout ce que vous devez savoir pour utiliser PyMongo.

```
import pymongo
# Établir une connexion à MongoDB
# Remplacez l'URL par votre propre URL MongoDB
client = pymongo.MongoClient("mongodb://localhost:27017/")
# Sélectionner une base de données
mydb = client["nom_de_votre_base"]
# Sélectionner une collection dans la base de données
mycollection = mydb["nom_de_votre_collection"]
return go(f, seed, [])
}
```

OPERATIONS CRUD AVEC PYMONGO

I. CREATE (Créer)

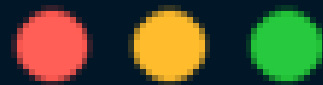
Insérer des données dans la collection :



```
# Insérer un document  
data = {"nom": "Exemple", "âge": 30, "ville": "Testville"}  
insert_result = mycollection.insert_one(data)
```

II. Read (Lire)

Récupérer des données depuis la collection :

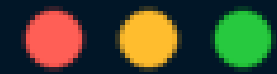


```
# Trouver un document
result = mycollection.find_one({"nom": "Exemple"})
print(result)

# Récupérer plusieurs documents
results = mycollection.find({"ville": "Testville"})
for result in results:
    print(result)
```

III. Update (Mettre à jour)

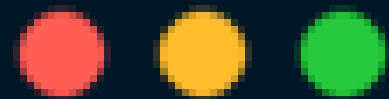
Mettre à jour un document existant :



```
# Mettre à jour un document  
mycollection.update_one({"nom": "Exemple"}, {"$set":{"âge": 35}})
```

IV. Delete (Supprimer)

Supprimer un document existant:



```
# Supprimer un document  
mycollection.delete_one({"nom": "Exemple"})
```