

# Révision MongoDB

Voici un petit exercice pour se replonger dans MongoDB, avec Docker et Python.

## Mongo dans Docker

Les conteneurs, contenant MongoDB, sont normalement déjà créés. Vous pouvez vérifier qu'ils tournent bien en tapant la commande :

```
docker ps
```

S'il y a un conteneur, vous pouvez les redémarrer à partir de **docker-compose**, dans le répertoire contenant le fichier *docker-compose.yaml* :

```
docker-compose start
```

Pour mémoire, vos travaux Docker sont sur LINUX (WSL ou installation, suivant vos préférences).

Si jamais les conteneurs n'existent plus, vous pouvez le recréer, à partir du répertoire contenant le *docker-compose.yaml* :

```
docker-compose up -d
```

Maintenant que les conteneurs sont démarrés, il faut entrer dans la base MongoDB (commande pour un conteneur qui se nomme *mongodb*):

```
docker exec -it my_mongo mongosh -u isen -p isen
```

## Travail sur la base

Nous allons refaire l'exemple *food*, qui contient une collection *fruits*.

Pour afficher les bases existantes dans MongoDB :

```
show dbs
```

Pour travailler sur une base (et la créer si elle n'existe pas) :

```
use food
```

Pour voir les collections :

```
show collections
```

Si vous avez déjà une collections *fruits*, supprimez-la (pour pouvoir travailler avec les mêmes données que cette exercice) :

```
db.fruits.drop()
```

Maintenant, nous pouvons ajouter les 3 documents suivant :

```
{name: « apple », origin : « usa », price:5}
```

```
{name : « orange », origin : « italy », quantity:30}
```

```
{name : « mango », origin : « italy », price : 3, quantity:40}
```

Pour mémoire, l'ajout se fait comme suit :

```
db.fruits.insert({name: « apple », origin : « usa », price:5})
```

## Quelques requêtes

Voici quelques requêtes à faire dans MongoDB (les réponses sont cachées juste sous la question) :

- Afficher tous les documents.
- Afficher le document ayant pour *name* : « orange ».
- Afficher tous les documents ayant pour *origin* : « italy ».
- Modifier le document orange pour passer la quantité à 34.
- Afficher les documents qui ont un prix.
- Afficher les documents dont la quantité dépasse 35.

Si toutes les requêtes fonctionnent bien, vous pouvez quitter la console Mongo. Mais ne stoppez pas les conteneurs, on va en avoir besoin avec Python.

## Avec Python

Pour utiliser une base MongoDB avec Python, vous devez avoir *pymongo* d'installé :

```
pip list
```

Si ce n'est pas le cas, utiliser le gestionnaire pour installer *pymongo* en ligne de commande.

Vous pouvez lancer VS Code dans votre répertoire de travail avec la commande suivante :

```
code .
```

Créer un nouveau script de travail, *revision.py* par exemple. Le but de l'exercice est d'être capable d'ouvrir une connexion vers le MongoDB du conteneur et d'afficher le contenu de la collection *fruits* de la base *food*.

Pour ça, voici les différentes étapes à coder :

- Charger *MongoClient* à partir de *pymongo*,
- Ouvrir une connexion au conteneur Mongo avec l'objet *MongoClient*,
- Sélectionner la base *food*,
- Lire tous les documents de la collection *fruits* et les afficher dans la console,
- Lire le document '*mango*' et l'afficher dans la console.

Pour mémoire vous pouvez utiliser *pprint()* pour afficher proprement le document dans la console.

Le code complet est sur la page suivante, c'est comme vous voulez...

```
from pymongo import MongoClient
import pprint

#attention a bien utiliser vos paramètres de connexion
client = MongoClient("mongodb://root:pass12345@127.0.0.1:27018")
db = client["food"]
fruits = db["fruits"]

documents = fruits.find({})

for document_lu in documents:
    pprint.pprint(document_lu)

mangue = fruits.find_one({"name":"mango"})

pprint.pprint(mangue)
```