Objectives

After completing this lab, you will be able to:

- Create documents in MongoDB with the insert method
- Read documents by listing them, counting them and matching them to a query
- Update and delete documents in MongoDB based on specific criteria

Select the training database

```
test> use training
switched to db training
```

*Create a collection named **languages***

```
training> db.createCollection("languages")
{ ok: 1 }
```

Exercise 2 - Insert documents

insert five documents into the collection languages

```
training> db.languages.insert({"name":"java","type":"object oriented"})

... db.languages.insert({"name":"python","type":"general purpose"})

... db.languages.insert({"name":"scala","type":"functional"})

... db.languages.insert({"name":"c","type":"procedural"})

... db.languages.insert({"name":"c++","type":"object oriented"})
```

```
DeprecationWarning: Collection.insert() is deprecated. Use insertOne, insertMany, or bulkWrite.

{

  acknowledged: true,

  insertedIds: { '0': ObjectId('694d4ea32d8f721b6f9dc2a1') }

}
```

Exercise 3 - Read documents

Find the count of documents.

```
training> db.languages.countDocuments()
5
```

List the first document in the collection.

```
training> db.languages.findOne()
{
  _id: ObjectId('694d4ea32d8f721b6f9dc29d'),
  name: 'java',
  type: 'object oriented'
}
```

List all documents in the collection.

```
training> db.languages.find()
[
  {
    _id: ObjectId('694d4ea32d8f721b6f9dc29d'),
    name: 'java',
    type: 'object oriented'
  },
  {
    _id: ObjectId('694d4ea32d8f721b6f9dc29e'),
    name: 'python',
    type: 'general purpose'
  },
  {
    _id: ObjectId('694d4ea32d8f721b6f9dc29f'),
    name: 'scala',
    type: 'functional'
  },
  {
    _id: ObjectId('694d4ea32d8f721b6f9dc2a0'),
    name: 'c',
    type: 'procedural'
  },
  {
    _id: ObjectId('694d4ea32d8f721b6f9dc2a1'),
    name: 'c++',
    type: 'object oriented'
  }
]
```

List first 3 documents in the collection.

```
training> db.languages.find().limit(3)
[
  {
    _id: ObjectId('694d4ea32d8f721b6f9dc29d'),
    name: 'java',
    type: 'object oriented'
  },
  {
    _id: ObjectId('694d4ea32d8f721b6f9dc29e'),
    name: 'python',
    type: 'general purpose'
  },
  {
    _id: ObjectId('694d4ea32d8f721b6f9dc29f'),
    name: 'scala',
    type: 'functional'
  }
]
```

Query for “python” language.

```
training> db.languages.find({"name":"python"})
[
  {
    _id: ObjectId('694d4ea32d8f721b6f9dc29e'),
    name: 'python',
    type: 'general purpose'
  }
]
```

Query for “object oriented” languages.

```
training> db.languages.find({"type":"object oriented"})
[
  {
    _id: ObjectId('694d4ea32d8f721b6f9dc29d'),
    name: 'java',
    type: 'object oriented'
  },
  {
    _id: ObjectId('694d4ea32d8f721b6f9dc2a1'),
    name: 'c++',
    type: 'object oriented'
  }
]
```

lists all the documents with only name field in the output.

```
training> db.languages.find({},{"name":1})
[
  { _id: ObjectId('694d4ea32d8f721b6f9dc29d'), name: 'java' },
  { _id: ObjectId('694d4ea32d8f721b6f9dc29e'), name: 'python' },
  { _id: ObjectId('694d4ea32d8f721b6f9dc29f'), name: 'scala' },
  { _id: ObjectId('694d4ea32d8f721b6f9dc2a0'), name: 'c' },
  { _id: ObjectId('694d4ea32d8f721b6f9dc2a1'), name: 'c++' }
]
```

Exercise 4 - Update documents

Add a field to all the documents.

The 'updateMany' command is used to update documents in a mongodb collection

```
training> db.languages.updateMany({},{$set:{"description":"programming language"}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 5,
  modifiedCount: 5,
  upsertedCount: 0
}
```

```
training> db.languages.updateMany({"name":"python"},{$set:{"creator":"Guido van Rossum"}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
training> db.languages.updateMany({"type":"object oriented"},{$set:{"compiled":true}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 2,
  modifiedCount: 2,
  upsertedCount: 0
}
```

Exercise 5 - Delete documents

Delete documents based on a criteria.

Delete the **scala** language document.

```
training> db.languages.remove({"name":"scala"})
DeprecationWarning: Collection.remove() is deprecated. Use deleteOne, deleteMany,
findOneAndDelete, or bulkWrite.
{ acknowledged: true, deletedCount: 1 }
```

Delete the **object oriented** languages.

```
training> db.languages.remove({"type":"object oriented"})
{ acknowledged: true, deletedCount: 2 }
```

Delete all the documents in a collection.

```
training> db.languages.remove({})
{ acknowledged: true, deletedCount: 2 }
training>
```