Launch Mongo Server

```
bhask@asus MINGW64 ~
$ mongod
{"t":{"$date":"2022-03-03T13:06:03.850+05:30"},"s":"I",  "c
":"CONTROL",  "id":23285,   "ctx":"main","msg":"Automatical
ly disabling TLS 1.0, to force-enable TLS 1.0 specify --ssl
DisabledProtocols 'none'"}
{"t":{"$date":"2022-03-03T13:06:03.850+05:30"},"s":"I",  "c
":"NETWORK",  "id":4915701, "ctx":"main","msg":"Initialized
 wire specification","attr":{"spec":{"incomingExternalClien
t":{"minWireVersion":0,"maxWireVersion":13},"incomingIntern
alClient":{"minWireVersion":0,"maxWireVersion":13},"outgoin
g":{"minWireVersion":0,"maxWireVersion":13},"isInternalClie
nt":true}}}
{"t":{"$date":"2022-03-03T13:06:04.301+05:30"},"s":"W",  "c
":"ASIO",     "id":22601,   "ctx":"main","msg":"No Transpor
tLayer configured during NetworkInterface startup"}
{"t":{"$date":"2022-03-03T13:06:04.301+05:30"},"s":"I",  "c
":"NETWORK",  "id":4648602, "ctx":"main","msg":"Implicit TC
P FastOpen in use."}
{"t":{"$date":"2022-03-03T13:06:04.303+05:30"},"s":"W",  "c
":"ASIO",     "id":22601,   "ctx":"main","msg":"No Transpor
tLayer configured during NetworkInterface startup"}
{"t":{"$date":"2022-03-03T13:06:04.303+05:30"},"s":"I",  "c
":"REPL",     "id":5123008, "ctx":"main","msg":"Successfull
y registered PrimaryOnlyService","attr":{"service":"TenantM
igrationDonorService","ns":"config.tenantMigrationDonors"}}

{"t":{"$date":"2022-03-03T13:06:04.303+05:30"},"s":"I",  "c
```

Then to use mongo

Type *mongo* in new console

```
bhask@asus MINGW64 ~
$ mongo
MongoDB shell version v5.0.6
connecting to: mongodb://127.0.0.1:27017/?compressors=disabled&gssapiSer
viceName=mongodb
Implicit session: session { "id" : UUID("c2ca5c30-42e4-4391-847e-6693f68
916b4") }
MongoDB server version: 5.0.6
================
Warning: the "mongo" shell has been superseded by "mongosh",
which delivers improved usability and compatibility.The "mongo" shell ha
s been deprecated and will be removed in
an upcoming release.
For installation instructions, see
https://docs.mongodb.com/mongodb-shell/install/
================
Welcome to the MongoDB shell.
For interactive help, type "help".
For more comprehensive documentation, see
        https://docs.mongodb.com/
Questions? Try the MongoDB Developer Community Forums
        https://community.mongodb.com
---
The server generated these startup warnings when booting:
        2022-03-03T10:59:07.891+05:30: Access control is not enabled for
 the database. Read and write access to data and configuration is unrest
ricted
```

```
> help
        db.help()                    help on db methods
        db.mycoll.help()             help on collection methods
        sh.help()                    sharding helpers
        rs.help()                    replica set helpers
        help admin                   administrative help
        help connect                 connecting to a db help
        help keys                    key shortcuts
        help misc                    misc things to know
        help mr                      mapreduce

        show dbs                     show database names
        show collections             show collections in current database
        show users                   show users in current database
        show profile                 show most recent system.profile entries with time >= 1ms
        show logs                    show the accessible logger names
        show log [name]              prints out the last segment of log in memory, 'global' is default
        use <db_name>                set current database
        db.mycoll.find()             list objects in collection mycoll
        db.mycoll.find( { a : 1 } )  list objects in mycoll where a == 1
        it                           result of the last line evaluated; use to further iterate
```

For creating new Database

Type *use <dbname>*

Then to add data type

Here products is the collection name

Insertone or insertMany keyword

```
> use ShopDb
switched to db ShopDb
> use shopDB
switched to db shopDB
> db
shopDB
> db.products.insertOne({_id:1, name:"pen", price:1.20})
{ "acknowledged" : true, "insertedId" : 1 }
> show collections
products
```

Reading Operations

```
> db.products.find()
{ "_id" : 1, "name" : "pen", "price" : 1.2 }
{ "_id" : 2, "name" : "pencil", "price" : "0.80", "stock" : 32 }
```

```
> db.products.find({name:"pen"})
{ "_id" : 1, "name" : "pen", "price" : 1.2 }
> db.products.find({price:{$gt:1}})
{ "_id" : 1, "name" : "pen", "price" : 1.2 }
```

Update and Delete Operations:

```
> db.products.updateOne({_id:2},{$set: {stock:12}})
{ "acknowledged" : true, "matchedCount" : 1, "modifiedCount" : 1 }
> db.products.find()
{ "_id" : 1, "name" : "pen", "price" : 1.2, "stock" : 32 }
{ "_id" : 2, "name" : "pencil", "price" : "0.80", "stock" : 12 }
> db.products.deleteMany({_id:1})
{ "acknowledged" : true, "deletedCount" : 1 }
> db.products.find()
{ "_id" : 2, "name" : "pencil", "price" : "0.80", "stock" : 12 }
```

Example in One GO

```
> db.items.insertOne({_id:1,name:"pencil",price:5})
{ "acknowledged" : true, "insertedId" : 1 }
> db.items.insertOne({_id:2,name:"pen", price:20})
{ "acknowledged" : true, "insertedId" : 2 }
> db.items.find()
{ "_id" : 1, "name" : "pencil", "price" : 5 }
{ "_id" : 2, "name" : "pen", "price" : 20 }
> db.items.find({price:{$gt:5}})
{ "_id" : 2, "name" : "pen", "price" : 20 }
> db.items.updateOne({_id:1},{$set:{stock:32}}
... )
{ "acknowledged" : true, "matchedCount" : 1, "modifiedCount" : 1 }
> db.items.find()
{ "_id" : 1, "name" : "pencil", "price" : 5, "stock" : 32 }
{ "_id" : 2, "name" : "pen", "price" : 20 }
> db.items.updateOne({_id:2},{$set:{stock:30}})
{ "acknowledged" : true, "matchedCount" : 1, "modifiedCount" : 1 }
> db.items.find()
{ "_id" : 1, "name" : "pencil", "price" : 5, "stock" : 32 }
{ "_id" : 2, "name" : "pen", "price" : 20, "stock" : 30 }
> db.items.deleteOne({_id:2})
{ "acknowledged" : true, "deletedCount" : 1 }
> db.items.deleteOne({_id:2})
{ "acknowledged" : true, "deletedCount" : 0 }
> db.items.find()
{ "_id" : 1, "name" : "pencil", "price" : 5, "stock" : 32 }
>
```