

# MongoDB Cheat Sheet (Developer Friendly)

## 1. Basics

```
show dbs          # list databases
use mydb          # switch/create database
show collections  # list collections
```

---

## 2. Insert Documents

### Single

```
db.users.insertOne({ name: "JP", age: 24 })
```

### Multiple

```
db.users.insertMany([
  { name: "A", age: 20 },
  { name: "B", age: 25 }
])
```

---

## 3. Find Documents

### Basic

```
db.users.find()
```

### With condition

```
db.users.find({ age: 20 })
```

### Projection

```
db.users.find({}, { name: 1, _id: 0 })
```

---

## 4. Comparison Operators

\$eq	equal
\$ne	not equal
\$gt	greater than
\$lt	less than
\$gte	>=
\$lte	<=
\$in	in list
\$nin	not in list

### Example

```
db.users.find({ age: { $gt: 20 } })
```

## 5. Logical Operators

\$and:	[ ... ]
\$or:	[ ... ]
\$not:	{ ... }
\$nor:	[ ... ]

### Example

```
db.users.find({ $or: [ { age: 20 }, { city: "Hyd" } ] })
```

## 6. Update Documents

### Update One

```
db.users.updateOne(  
  { name: "JP" },  
  { $set: { age: 25 } }  
)
```

### Update Many

```
db.users.updateMany(  
  { city: "Hyd" },
```

```
{ $inc: { visits: 1 } }  
)
```

---

## 7. Update Operators

```
$set      set value  
$unset    remove field  
$inc      increment  
$mul      multiply  
$rename   rename field  
$push     push into array  
$pull     remove from array  
$addToSet add only if not exists
```

---

## 8. Delete Documents

```
db.users.deleteOne({ name: "A" })  
db.users.deleteMany({ age: { $lt: 25 } })
```

---

## 9. Sorting, Limiting, Skipping

```
db.users.find().sort({ age: 1 })      # ascending  
db.users.find().sort({ age: -1 })     # descending  
  
db.users.find().limit(5)  
db.users.find().skip(10)
```

---

## 10. Indexes

### Create

```
db.users.createIndex({ name: 1 })
```

### Unique Index

```
db.users.createIndex({ email: 1 }, { unique: true })
```

## List

```
db.users.getIndexes()
```

## 11. Aggregation Framework

### Basic Pipeline

```
db.users.aggregate([
  { $match: { city: "Hyd" } },
  { $group: { _id: "$city", avgAge: { $avg: "$age" } } }
])
```

## 12. Common Aggregation Operators

\$match	filter
\$group	group by
\$project	select fields
\$sort	sort
\$limit	limit
\$skip	skip
\$lookup	join collections

### Example Join

```
db.orders.aggregate([
  {
    $lookup: {
      from: "users",
      localField: "userId",
      foreignField: "_id",
      as: "userInfo"
    }
  }
])
```

## 13. Search in Arrays

```
db.users.find({ skills: "python" })
```

```
db.users.find({ skills: { $all: ["python", "django"] } })
```

---

## 14. Regex Search

```
db.users.find({ name: { $regex: "^J", $options: "i" } })
```

---

## 15. Count Documents

```
db.users.countDocuments()
```

---

## 16. Drop Collection / DB

```
db.users.drop()  
db.dropDatabase()
```

---

## 17. MongoDB Python (PyMongo) Basics

```
from pymongo import MongoClient  
client = MongoClient()  
db = client.mydb  
  
# insert  
db.users.insert_one({"name": "JP"})  
  
# find  
for u in db.users.find({"age": {"$gt": 20}}):  
    print(u)
```