

# MongoDb Cheatsheet

---

A comprehensive list of the most commonly used MongoDB commands for beginners (compatible with MongoDB 4.2, 5.x, 6.x, and 7.x). Assume we are working in a collection named comments in a database of your choice.

---

## 1. Database Commands

---

### View all databases

```
show dbs
```

### Create a new or switch databases

```
use dbName
```

### View current database

```
db
```

### Delete current database

```
db.dropDatabase()
```

---

## 2. Collection Commands

---

### Show collections

```
show collections
```

### Create a collection named 'comments'

```
db.createCollection('comments')
```

### Drop a collection named 'comments'

```
db.comments.drop()
```

---

## 3. Document (Row) Commands

---

### Show all documents in a collection

```
db.comments.find()
```

### Show all documents (pretty format)

```
db.comments.find().pretty()
```

### Find the first document matching a filter

```
db.comments.findOne({name: 'Harry'})
```

## Insert one document

```
db.comments.insertOne({  
  name: 'Harry',  
  lang: 'JavaScript',  
  member_since: 5  
})
```

## Insert many documents

```
db.comments.insertMany([  
  {name: 'Harry', lang: 'JavaScript', member_since: 5},  
  {name: 'Rohan', lang: 'Python', member_since: 3},  
  {name: 'Lovish', lang: 'Java', member_since: 4}  
])
```

## Search by field

```
db.comments.find({lang: 'Python'})
```

## Limit the number of results

```
db.comments.find().limit(2)
```

## Skip results (useful for pagination)

```
db.comments.find().skip(2).limit(2)
```

## Count the number of matching documents

```
db.comments.countDocuments({lang: 'Python'})
```

---

## 4. Update Commands

---

### Update one document

```
db.comments.updateOne(  
  {name: 'Shubham'},  
  {$set: {name: 'Harry', lang: 'JavaScript', member_since: 51}},  
  {upsert: true}  
)
```

### Update many documents

```
db.comments.updateMany(  
  {lang: 'JavaScript'},  
  {$set: {verified: true}}  
)
```

### Increment a field

```
db.comments.updateOne(  
  {name: 'Rohan'},  
  {$inc: {member_since: 2}}  
)
```

### Rename a field

```
db.comments.updateOne(  
  {name: 'Rohan'},  
  {$rename: {member_since: 'member'}}  
)
```

---

## 5. Delete Commands

---

### Delete one document

```
db.comments.deleteOne({name: 'Harry'})
```

### Delete many documents

```
db.comments.deleteMany({lang: 'Java'})
```

---

## 6. Query Operators

---

### Comparison Operators

```
db.comments.find({member_since: {$lt: 90}}) // less than
db.comments.find({member_since: {$lte: 90}}) // less than or equal
db.comments.find({member_since: {$gt: 90}}) // greater than
db.comments.find({member_since: {$gte: 90}}) // greater than or equal
db.comments.find({member_since: {$ne: 5}}) // not equal
```

### Logical Operators

```
db.comments.find({$and: [{lang: 'Python'}, {member_since: {$gt: 2}]}})
db.comments.find({$or: [{lang: 'Python'}, {lang: 'Java'}]})
db.comments.find({lang: {$in: ['Python', 'Java']}})
db.comments.find({lang: {$nin: ['C++', 'Rust']}})
```

### Sorting

```
db.comments.find().sort({member_since: 1}) // ascending
db.comments.find().sort({member_since: -1}) // descending
```

## Projection (select specific fields)

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

---

## 7. Indexes (Basics)

---

### Create an index

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

### View indexes

```
db.comments.getIndexes()
```

### Drop an index

```
db.comments.dropIndex({name: 1})
```

---

## 8. Aggregation (Basics)

---

### Count by field

```
db.comments.aggregate([
  {$group: {_id: "$lang", total: {$sum: 1}}}
])
```

## Average by field

```
db.comments.aggregate([
  {$group: {_id: "$lang", avgExperience: {$avg: "$member_since"}}}
])
```