Mongo Cheatsheat

To Run Mongo Server: **In Terminal:** `mongod`

To use the mongo shell in terminal

In Terminal - new tab must run with `mongod`

- After starting `mongo` in another tab open a new tab and type
- `mongo`
- prompt should change from bash `$` to `>`

Show DBs

- `show dbs`

Show which db is in use

- `db`

Switch to a database (will also create the database, if it does not exist)

- `use databseName`

Create a collection

- `db.createCollection('collectionName')`

Show collections in database

- `show collections`

Show data or show documents

- `db.collectionName.find()`

- Make it pretty `db.collectionName.find().pretty()

Find something specific

- From the docs `db.collectionName.find ( <field> : <value> )`
  - How you would actually type it out:
  - `db.collectionName.find( { school: "General Assembly"})`
- More from the docs
  : https://docs.mongodb.com/manual/reference/method/db.collection.find/

Find one

- `db.collectionName.findOne()`
- find one and show only the name and school fields
  - `db.collectionName.findOne({}, { name: 1 , school :1 })`
- find one and exclude the id field
  - `db.collectionName.findOne( {} , { _id : 0 })`

Remove set of documents

- db.collectionName.remove ({state : 'NY'})

Drop Collection - Warning! Cannot be undone!

- db.collectionName.drop()

Drop database - Warning! Cannot be undone!

- use `dbToBeDropped`
- `db.dropDatabase()`

Exit mongo shell

^ C or `exit` or `quit()`
Shut down `mongod` (Mongo Server)
^ C
Gracefully exit out of terminal (still need to close window after executing, but will delete expired sessions) `exit`
Problem exiting `mongod`?

In terminal

1. ps -A | grep mongod
2. find the line that just mentions mongod, but not grep
3. take note of the number on the left
4. type kill 1774 or whatever that number is. Try mongod again.
5. If that doesn't work, go to /data/db and rm mongod.lock. Try mongod again.

Basic Crud operations:

https://docs.mongodb.com/manual/crud/

A Handful of Collection Methods

See Options Table below, to understand the parameters in each method

From the Mongo Documentation

Mongo Command	what it does
`db.collection.count(query, options)`	returns the count of documents that would match a find query
`db.collection.deleteOne(filter, options)`	removes a single document from a collection that matches the fiter
`db.collection.deleteMany(filter, options)`	removes all documents that match the filter
`db.collection.distinct(field, query, options)`	finds distinct values for a specific field across a collection and returns the results in an array
`db.collection.drop()`	drops the collection
`db.collection.find(query, projection)`	finds matching documents based on query
`db.collection.findAndModify({options})`	object that has the following(and more)options `query: sort: remove: update: new: fields: upsert:` see below for options table.

Mongo Command	what it does
	Finds one and modifies based on the parameters (see below)
`db.collection.findOne({query , projection})`	Finds one document that satisfies the query
`db.collection.findOneAndDelete(filter, options)`	Finds one document that satisfies the query and has options for projection and sort (and more) see below
`db.collection.findOneAndReplace(filter, replacement, options)`	finds one and modifies based on the parameters (see below)
`db.collection.findOneAndReplace(query, projection)`	finds one and modifies based on the parameters (see below)
`db.collection.findOneAndUpdate(filter, update, options)`	finds one and modifies based on the parameters (see below)
`db.collection.insert(<document or array of documents>, options)`	inserts one or an array of documents
`db.collection.insertOne(<document>, options)`	inserts one document
`db.collection.insertMany([<doucment1>, <document2>, ... ] , options)`	inserts many documents in an array
`db.collection.remove(query, {justOne ,options})`	removes a document that matches the query. `justOne` option is by default true
`db.collection.renameCollection(target, dropTarget)`	`target` is new collection name (string), `dropTarget` boolean -default is false
`db.collection.update(query, update, options)`	must use update operators to set values or else document(s) will be destroyed (see below)
`db.collection.updateOne(filter, update, options)`	must use update operators to set values or else document will be destroyed (see below)
`db.collection.updateMany(filter, update, options)`	must use update operators to set values or else document(s) will be destroyed (see below)

Common Options

What these parameters mean in the methods listed above

Option	type	example	what it does
fields	{<field1> : <value> ,<field2> : <value> ... }	{school : 'General Assembly ' , program: 'WDIR'}	the subset of fields to return
filter	(<field> : <value>)	{ school: “General Assembly ”}	selects by field value
new	boolean	true	When true, returns the modified document rather than original, default, false
projection	{<field1> : 0 ,<field2> : 1 ... }	{school : 0 , program: 1}	the subset of fields to return, where 0 means do not return the field, and 1 means return it
query	(<field> : <value>)	{ school: “General Assembly ”}	selects by field value
remove	boolean	true	When true, remove this document. When false, do not remove - usually either update or remove is used when these are options
sort	(<field> : <value>)	{ school: “General	sorts by field value

Option	type	example	what it does
		Assembly ”}	
update	({$set: {<field> : <value>}})	{$set: { school: “General Assembly ”}}	sets a new value. If you do not use $set, the whole document will be destroyed, can also use other ones like `$inc` in a similar fashion (see docs - link below for more). When using `modify` must use either `update` or `remove`
upsert	boolean	true	When true, if does not exist, create it. When false, if does not exists, do not create it

---

Update Operators

When using update, you must use at least one of these operators, or else your document will be destroyed

Mongo docs for update (fields, arrays and modifiers)

- A few common update operators:
- `$inc` : increment the value
- `$rename` : renames a field
- `$set` : set the value of a field
- `$unset` : remove the specified field from a document

- **Note** : failing to use an operator, will cause the document to be overwritten and 'destroyed'

- **Note** : Modifying arrays for updates use their own operators like `$pull`, `$pop`, `$push` (scroll down in the above doc to see)

Example:

```
db.students.findOneAndUpdate({ name: 'Karolin'}, {$set:{graduated: true}})
```

Query Selectors

When searching for specific docs, these selectors can help you find what you want

[Mongo docs for query selectors]

- You can use query selectors like
- `$eq` : matches values that are equal to a specified value
- `$ne` : matches all values that are not equal to the specified value
- `$gt` : matches values that are greater than a specified value
- `$gte` : matches values that are greater than or equal to a specified value
- `$in` : matches any of the values specified in an array
- `$nin` : matches none of the values specified in an array
- `$lt` : matches values that are less than a specified value
- `$lte` : matches values that are less than or equal to a specified value
- You can use logical operators like `$and`, `$not`, `$nor` and `$or`
- You can select if a field exists using `$exists`

Example:

```
db.students.find({state:{$ne:'NY'}})
```

This will find all the students who do NOT live in NY state.