

## Cheat Sheet

Command	Description
<code>db.version()</code>	This code returns the version of MongoDB that is being used.
<code>show dbs</code>	This code displays a list of all databases in the MongoDB server.
<code>use training</code>	This code switches to the "training" database or creates it if it doesn't exist.
<code>db.createCollection("mycollection")</code>	This code creates a new collection named "mycollection" within the current database.
<code>show collections</code>	This code lists all the collections in the current database.
<code>db.mycollection.insert({"color":"white," "example":"milk"})</code>	This code inserts a new document into the "mycollection" collection.
<code>exit</code>	This code is used to exit the MongoDB shell or terminate the current session.
<code>start_mongo</code>	A custom or external command used to start the MongoDB server.
<code>mongosh -u root -p NTc0My1yc2FubmFy --authenticationDatabase admin local</code>	This command starts the MongoDB shell (mongosh) with the username "root" and the password "NTc0My1yc2FubmFy" for authentication, connecting to the "local" database.
<code>db.languages.count()</code>	This code returns the number of documents in the "languages" collection.
<code>db.languages.findOne()</code>	This code retrieves a single document from the "languages" collection.
<code>db.languages.find()</code>	This code retrieves all documents from the "languages" collection.
<code>db.languages.find().limit(3)</code>	This code retrieves the first three documents from the "languages" collection.
<code>db.languages.find({"name":"python"})</code>	This code retrieves documents from the "languages" collection where the "name" field is equal to "python."
<code>db.languages.find({},{"name":1})</code>	This code retrieves all documents from the "languages" collection but only includes the "name" field in the result.
<code>db.languages.find({},{"name":0})</code>	This code retrieves all documents from the "languages" collection but excludes the "name" field in the result.
<code>db.languages.find({"type":"object oriented"}, {"name":1})</code>	This code retrieves documents from the "languages" collection where the "type" field is equal to "object oriented" and only includes the "name" field in the result.
<code>db.collection.updateMany([what documents to find], {\$set:[what fields to set]})</code>	This code updates multiple documents in the "collection" by specifying the criteria for matching documents and the fields to update.
<code>db.languages.updateMany({},{\$set: {"description":"programming language"}})</code>	This code updates all documents in the "languages" collection by setting the "description" field to "programming language."
<code>db.languages.updateMany({"name":"python"},{\$set: {"creator":"Guido van Rossum"}})</code>	This code updates documents in the "languages" collection where the "name" field is equal to "python" by setting the "creator" field to "Guido van Rossum."
<code>db.languages.updateMany({"type":"object oriented"},{\$set:{"compiled":true}})</code>	This code updates documents in the "languages" collection where the "type" field is equal to "object oriented" by setting the "compiled" field to "true."
<code>db.languages.remove({"name":"scala"})</code>	This code removes documents from the "languages" collection where the "name" field is equal to "scala."
<code>db.languages.remove({"type":"object oriented"})</code>	This code removes documents from the "languages" collection where the "type" field is equal to "object oriented."
<code>db.languages.remove({})</code>	This code removes all documents from the "languages" collection, effectively clearing the collection.