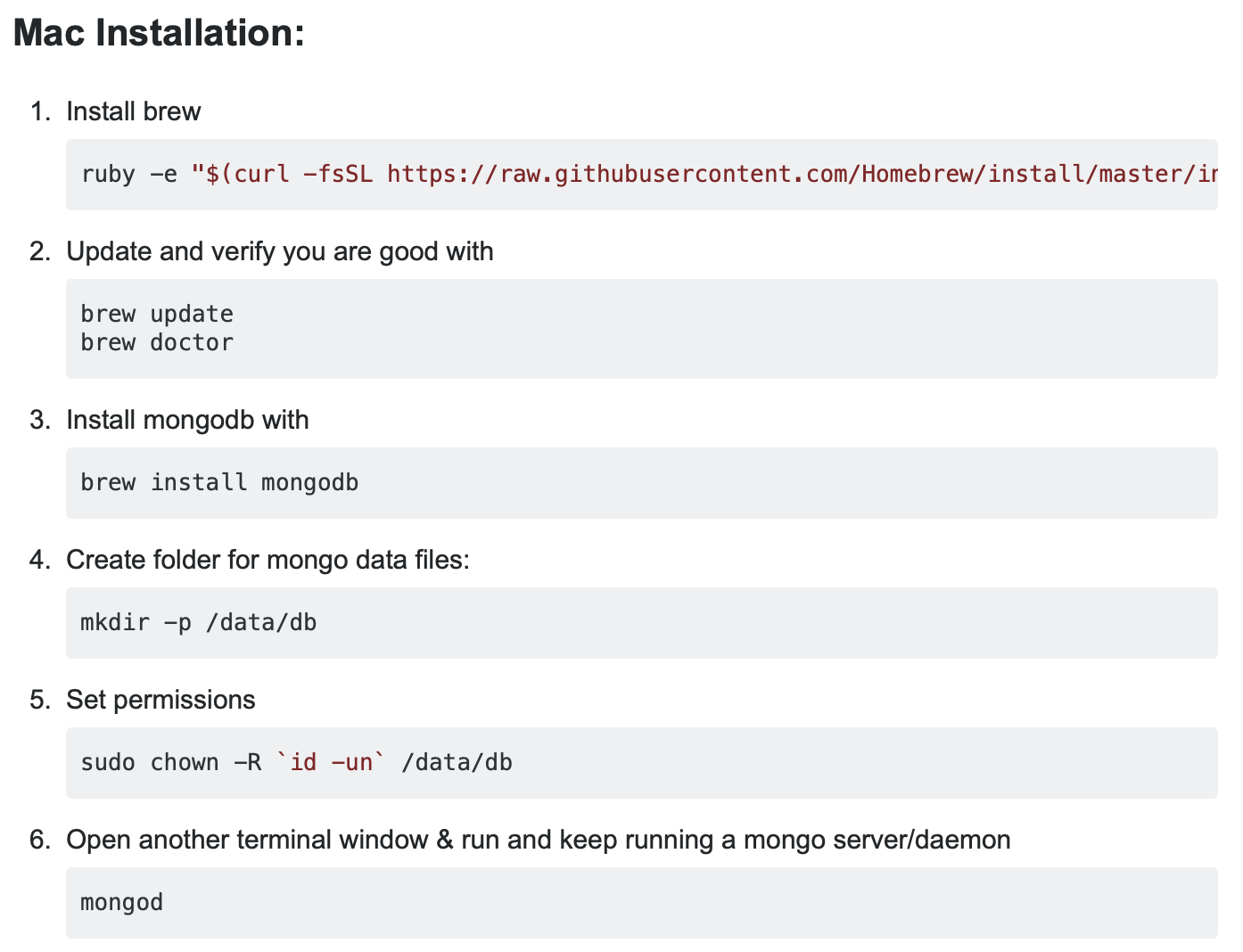
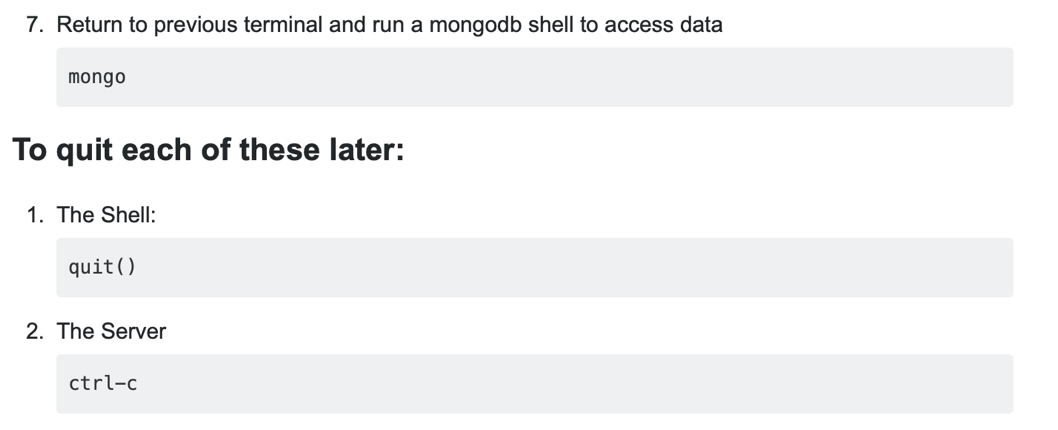
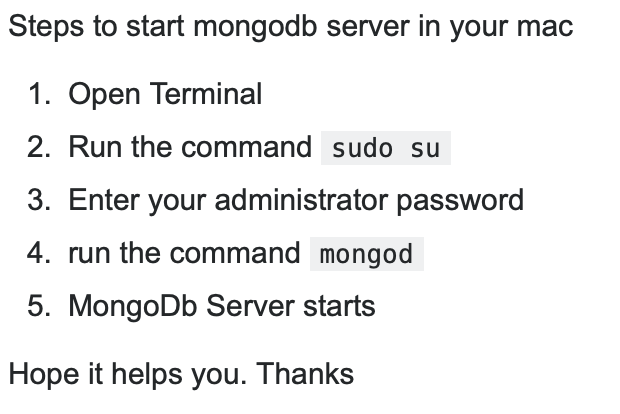
MongoDB on mac

Install:

https://stackoverflow.com/questions/18452023/installing-and-running-mongodb-on-osx

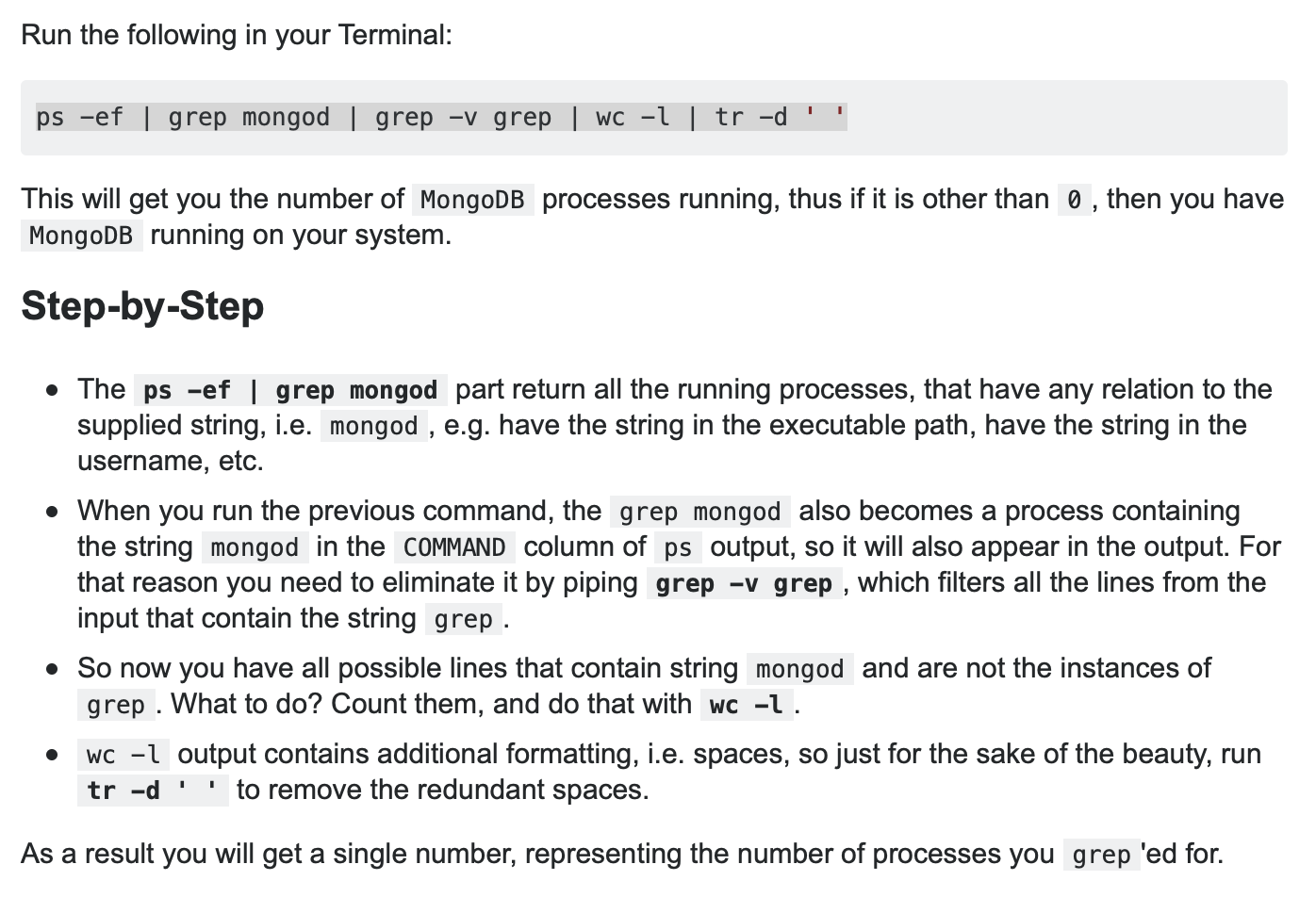


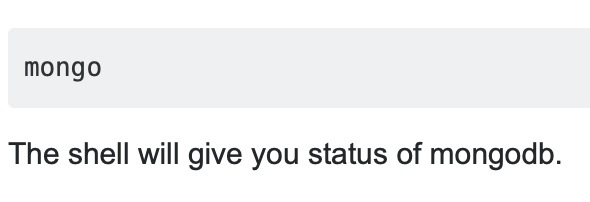


****

[**How to check if mongo db is running on Mac?**](https://stackoverflow.com/questions/31561098/how-to-check-if-mongo-db-is-running-on-mac)

ps -ef | grep mongod | grep -v grep | wc -l | tr -d ' '





MongoDB folder location in mac ( server )

‎⁨Macintosh HD⁩ ▸ ⁨usr⁩ ▸ ⁨local⁩ ▸ ⁨var⁩ ▸ ⁨mongodb⁩

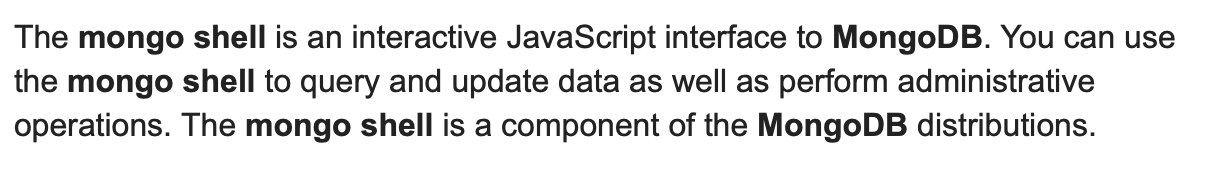
MongoDB Collections location in mac ( data )

‎ ‎⁨Macintosh HD⁩ ▸ ⁨data⁩ ▸ ⁨db⁩

collection-0--8069778669997908300.wt

collection-2--8069778669997908300.wt

what is the mongo shell



To start mongoDB server

New Terminal:

mongod

To start mongoDB shell

New Terminal:

Mongo

On shell terminal

> show dbs

admin 0.000GB

config 0.000GB

local 0.000GB

> use classes

switched to db classes

> db.createCollection('class')

{ "ok" : 1 }

> db.createCollection('student')

{ "ok" : 1 }

> db.createCollection('students')

{ "ok" : 1 }

> db student.drop()

> db.class.insert({code:'FBW1', mamber:12, isRunning:true})

> db.class.find().pretty()

{

"\_id" : ObjectId("5cb457a1c1d9bcafc2e3d87b"),

"code" : "FBW1",

"mamber" : 12,

"isRunning" : true

}

>

> db.teacher.insert({firstName:'kostas', lastNmame:'Diakogiannis', age:31, nickName:'Jake'})

WriteResult({ "nInserted" : 1 })

> db.student.insert({firstName:'Mauro', lastName:'Navaroo', favTopics:['swimming', 'runnung'], scores:[{html:78,css:88,js:77}]})

WriteResult({ "nInserted" : 1 })

> db.student.find().pretty()

{

"\_id" : ObjectId("5cb45bd4c1d9bcafc2e3d87d"),

"firstName" : "Mauro",

"lastName" : "Navaroo",

"favTopics" : [

"swimming",

"runnung"

],

"scores" : [

{

"html" : 78,

"css" : 88,

"js" : 77

}

]

}

>

> db.student.find({firstName:'mohammed'})

> db.student.find({firstName:'mohammed'}).pretty()

{

"\_id" : ObjectId("5cb45d90c1d9bcafc2e3d87e"),

"firstName" : "mohammed",

"lastName" : "wahba",

"favTopics" : [

"swimming2",

"runnung2"

],

"scores" : [

{

"html" : 44,

"css" : 48,

"js" : 33

}

]

}

>

> db.student.find({firstName:'mohammed'},{lastName: 1}).pretty()

{ "\_id" : ObjectId("5cb45d90c1d9bcafc2e3d87e"), "lastName" : "wahba" }

>

> db.student.find({firstName:'mohammed'},{lastName: 1, \_id:0}).pretty()

{ "lastName" : "wahba" }

>

> db.student.update({firstName:'mohammed'},{$set:{firstName:'Mmohammed', lastName:'wahbee'}})

WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })

>

> db.student.update({},{$set:{instute:'DCI'}}, {multi:true})

WriteResult({ "nMatched" : 3, "nUpserted" : 0, "nModified" : 2 })

>

// db.collection.deleteOne() for Delelt the first one in  matches the criteria

> db.student.deleteOne({firsName:'ali'})

{ "acknowledged" : true, "deletedCount" : 1 }

>

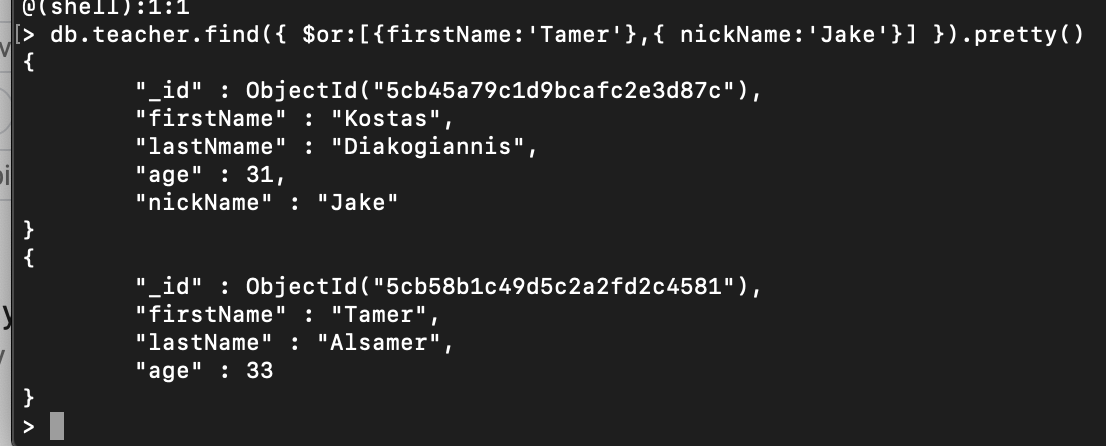
// db.collection.deleteMany() for method deletes all documents that match the criteria.

> db.student.deleteMany({firsName:'ali'})

{ "acknowledged" : true, "deletedCount" : 3 }

>

**$or:[{option1}, {option2} ]**



**$gt greater than**

**$gte greater than or equal**

**$eg equal**

**$lt. lower than**

**$lte. Lower than or equal**

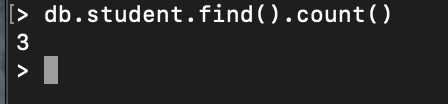
**$neq not equal**





**.count**

db.student.find().count()

****

**Import json file to MngogDB. Via my mac**

**1**

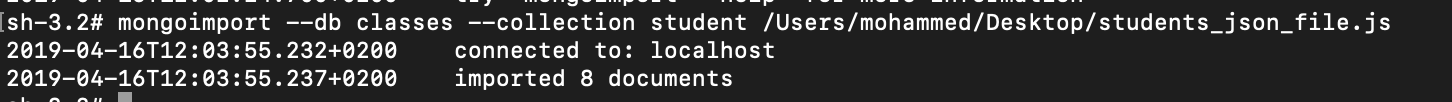
New terminal

**2**

/usr/local/mongodb/bin

**3**

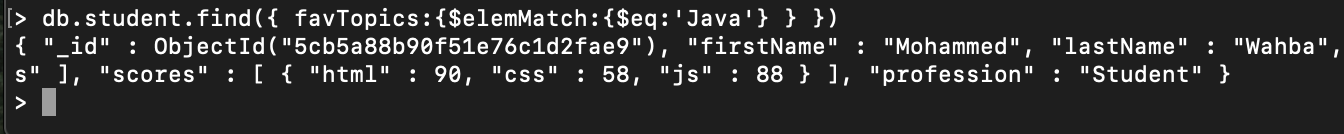
mongoimport --db classes --collection student –file /Users/mohammed/Desktop/students\_json\_file.json

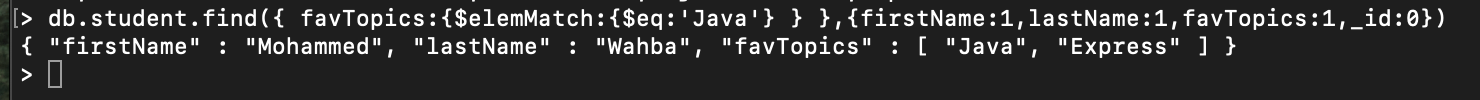
****

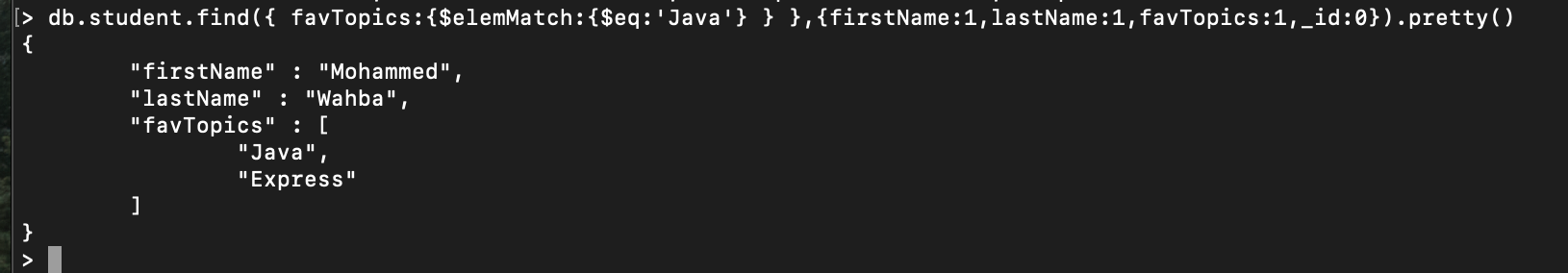
**MongoDB deals with array {$elemMatch: }**

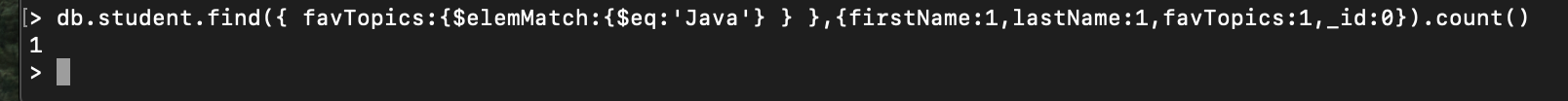
**https://docs.mongodb.com/manual/reference/operator/update/each/**

****



****

****

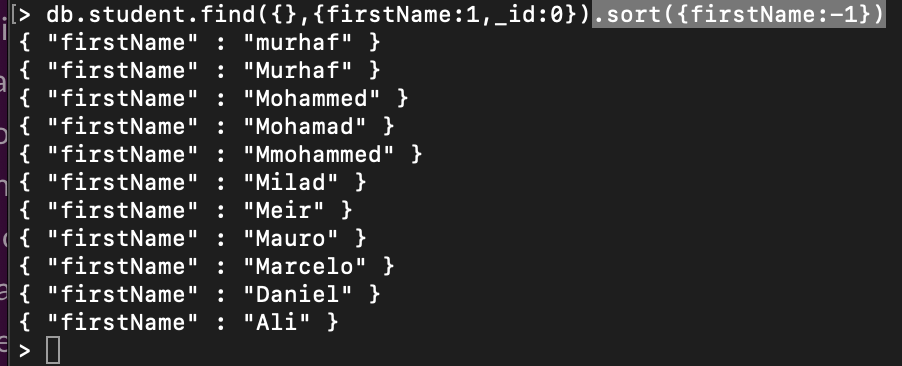
****

**Sort () 1 creasing -1 decreasing**

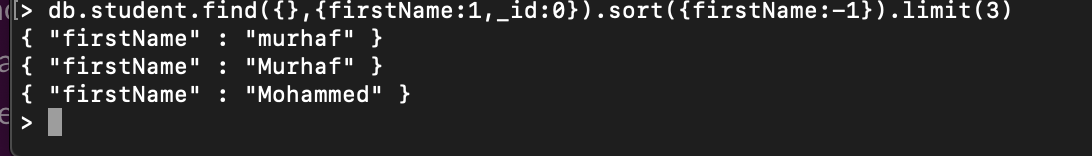
.sort({firstName:1})

****

.sort({firstName:-1})

****

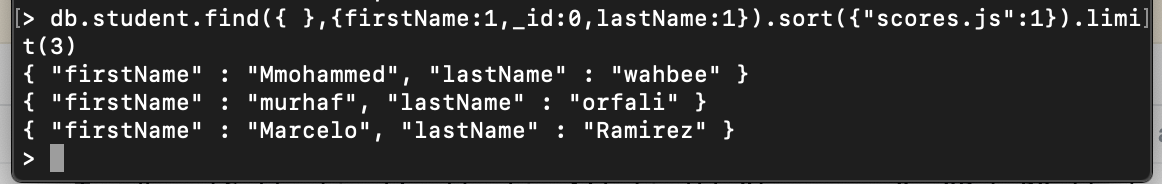
**.limit(3) to show the first 3**

****

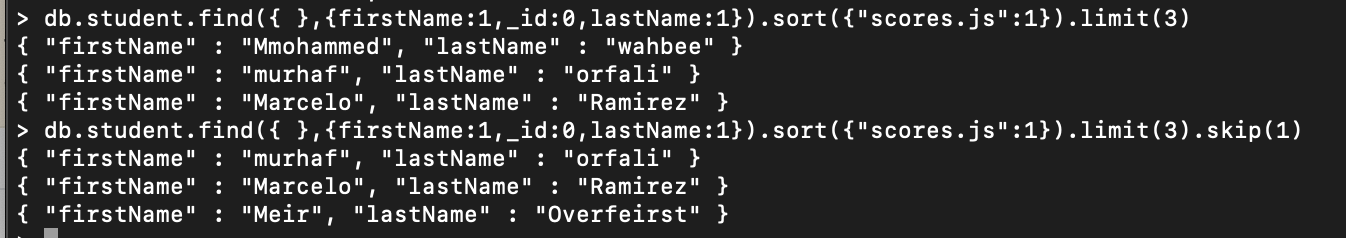
**Ex:**

**Sort, find and return the top 3 JS scores, return the firstname and lastanme and score**

db.student.find({ },{firstName:1,\_id:0,lastName:1}).sort({"scores.js":1}).limit(3)

****

**Skip()**

****

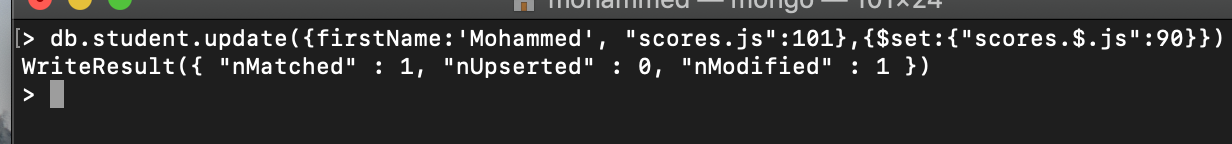
[Mongo update query to find field within array of objects](https://stackoverflow.com/questions/52631006/mongo-query-to-find-field-within-array-of-objects)

db.student.update(

{firstName:'Mohammed', **"array.object":Curretvalue**},

{$set:{"array.**$.**object":NewValue}}

)

****

Why to define **"array.object":Curretvalue**, only when I need update the object inside array. But not use to update only simple array without object.

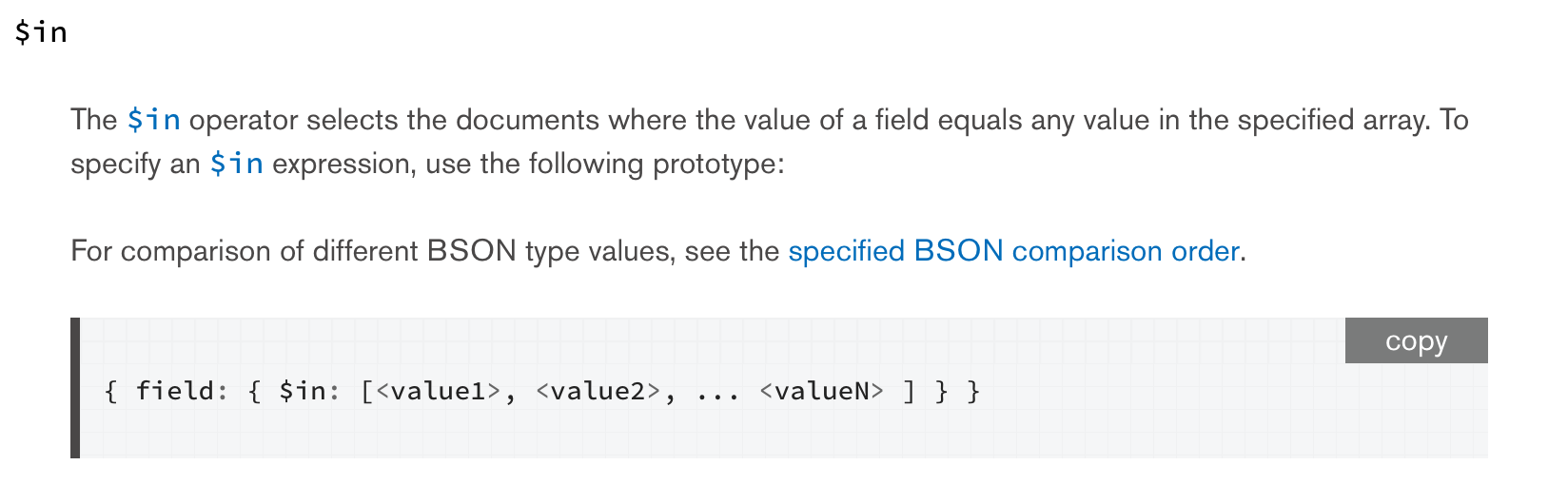
Why to write .**$,** to find **automatically** the element index in array the match the needed field (**"**array.object":Curretvalue **).**

So we can replace $ by index 0 or 1 2 3 4 … :

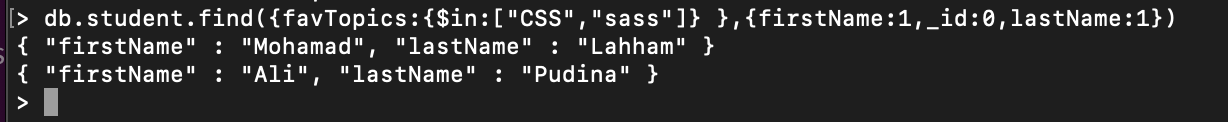
{firstName:'Mohammed', **"array.object":Curretvalue**},

{$set:{"array.**0.**object":NewValue}}

**$in**

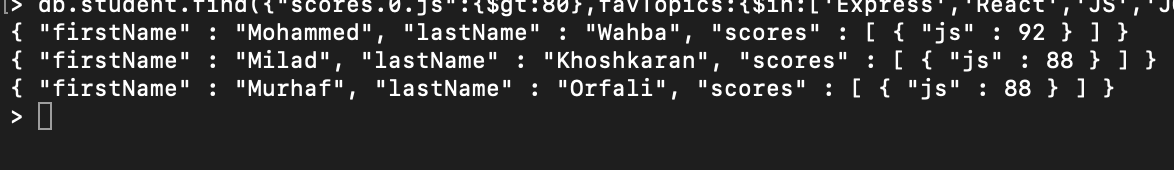
****

Ex. Find all student that has CSS or sass

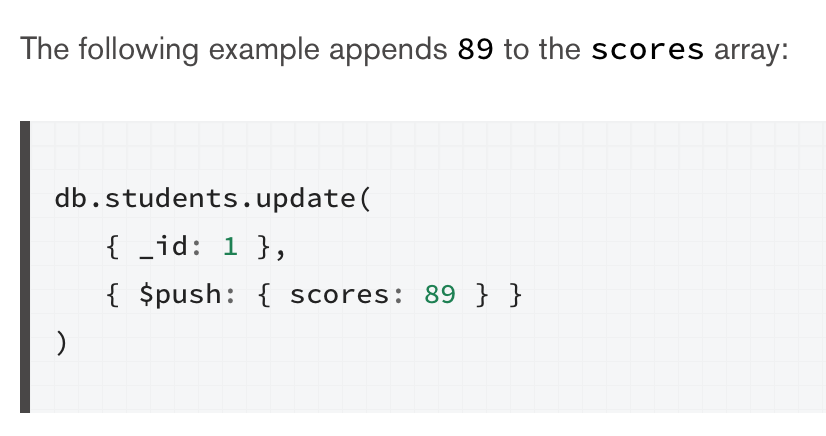


Complex ex. Return the firstName and the lastName of every student whose JS score is more than 80, and it's favorite topic is one of : 'JS', 'JQuery', 'React' or 'Express'. Sort the results by the JS score.

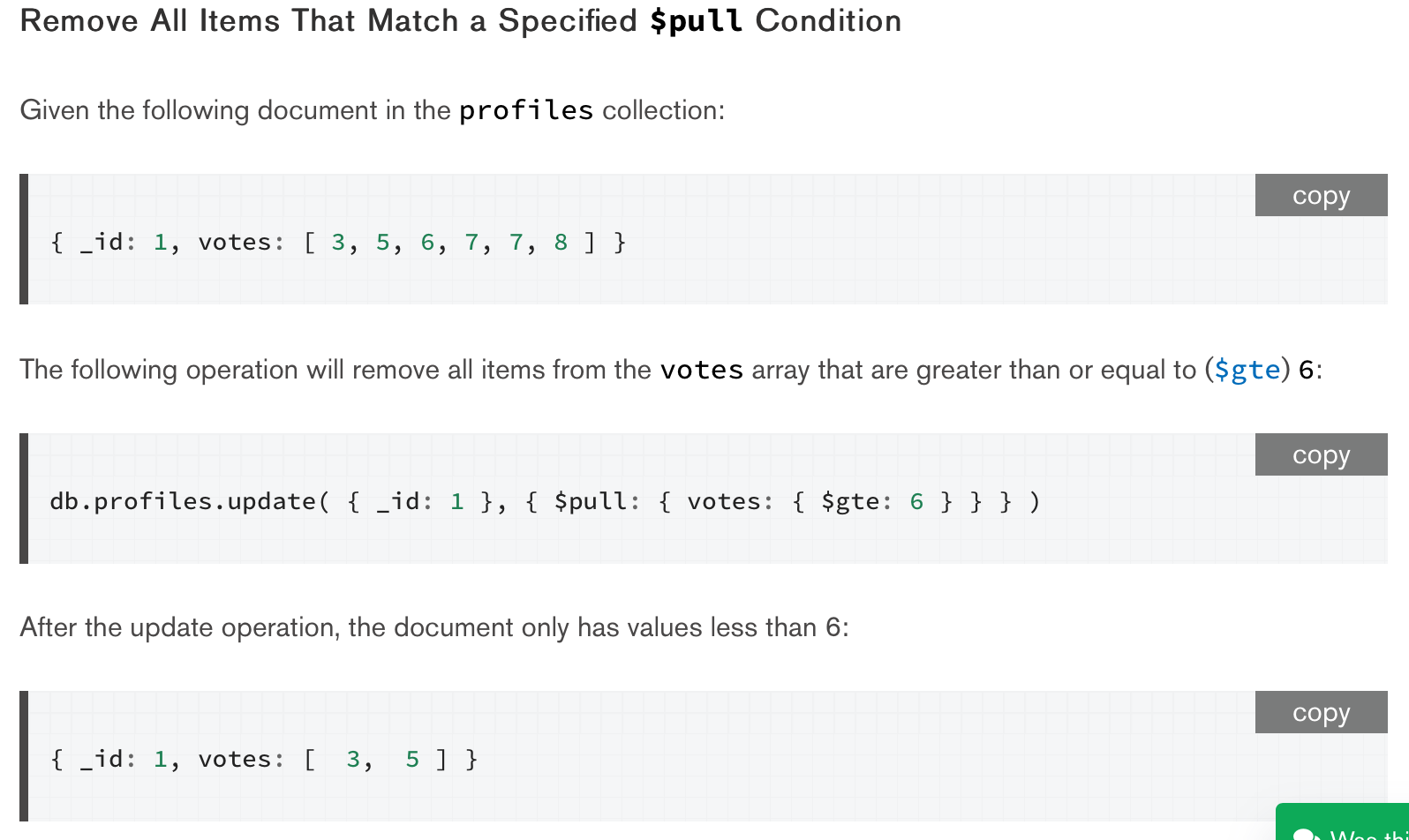
**db.student.find({"scores.0.js":{$gt:80},favTopics:{$in:['Express','React','JS','JQuery']}},{"scores.js":1,firstName:1,lastName:1,\_id:0}).sort({"scores.js":-1})**

****

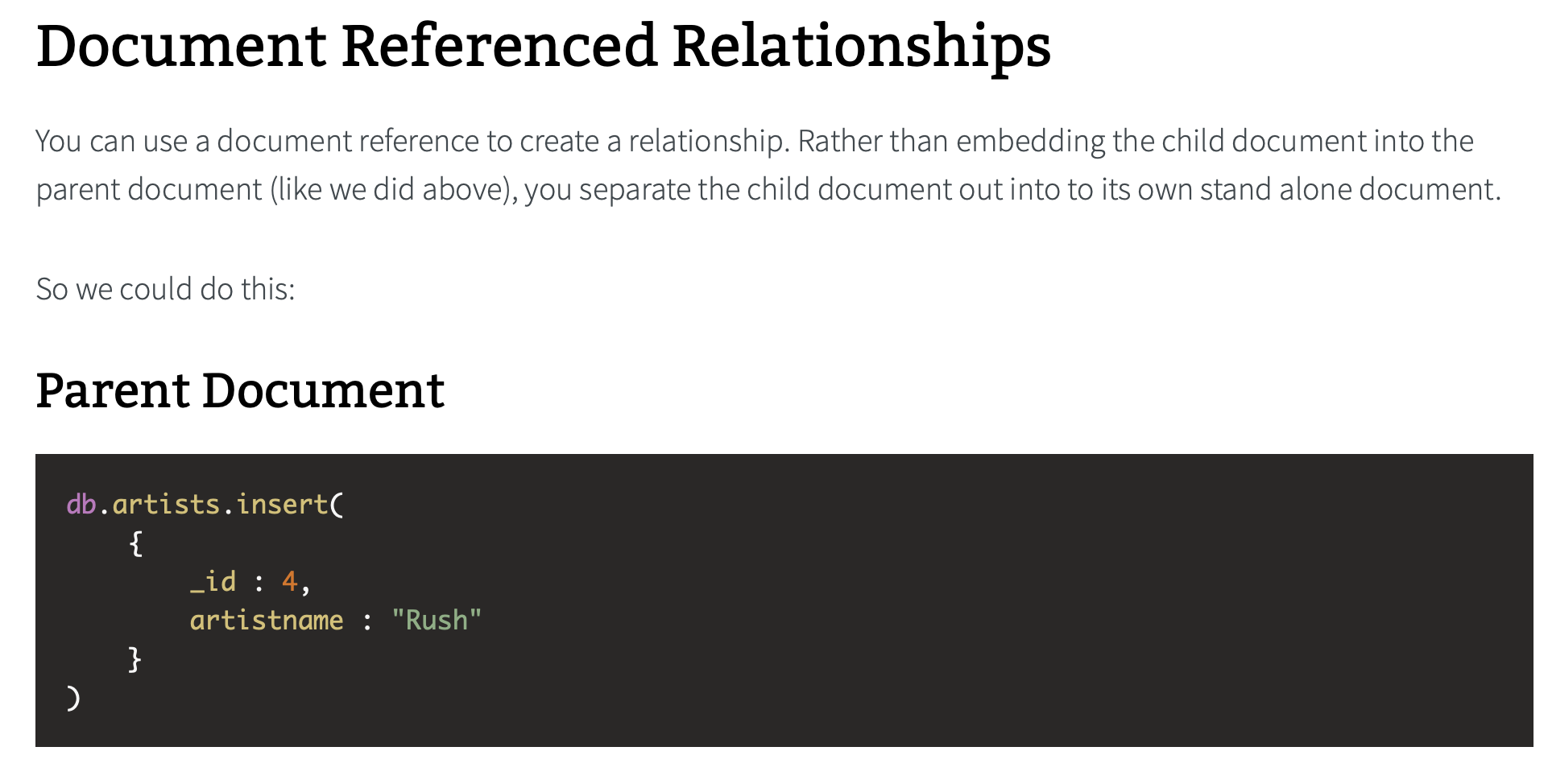
**Push to array $push**

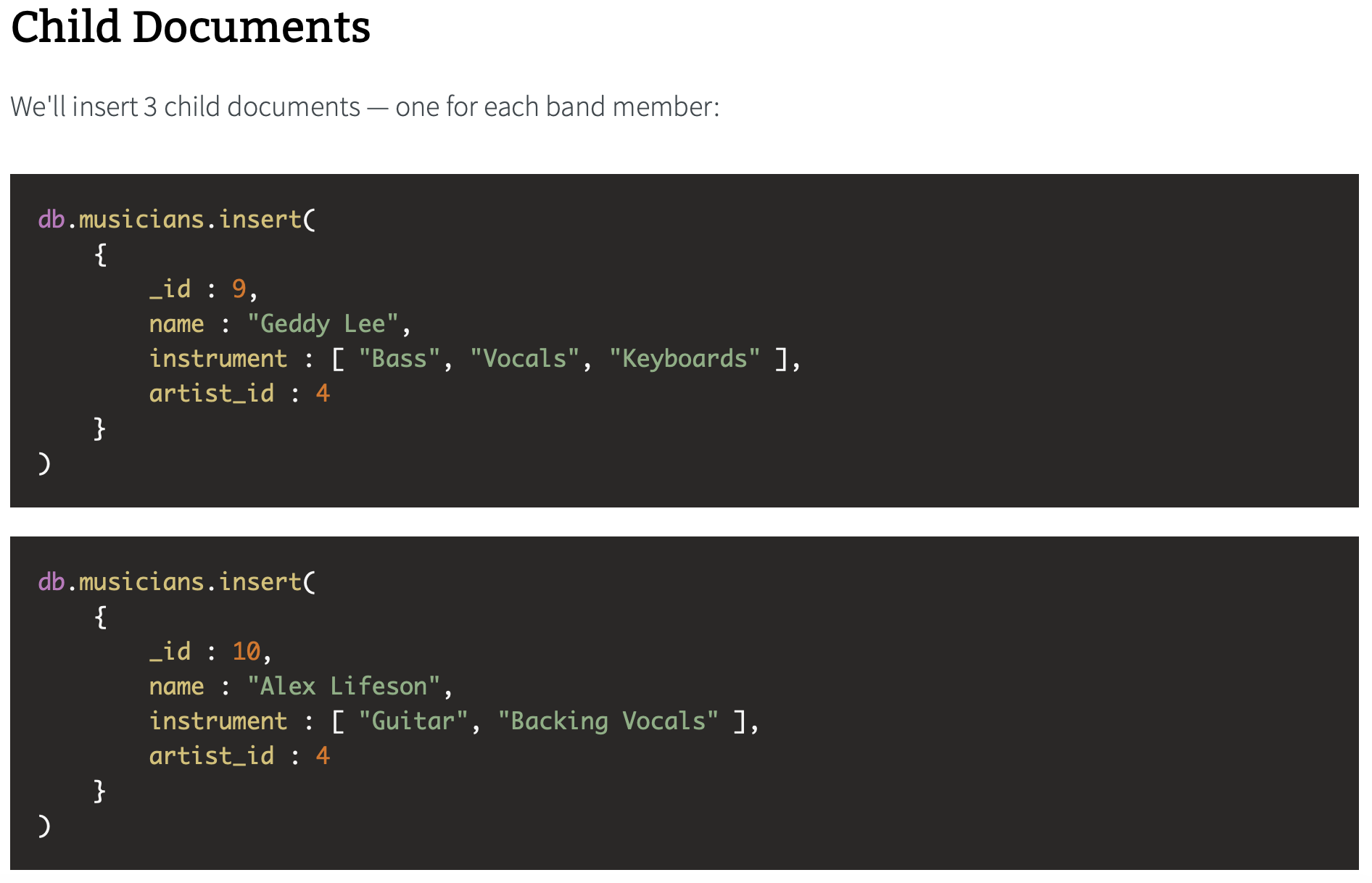
****

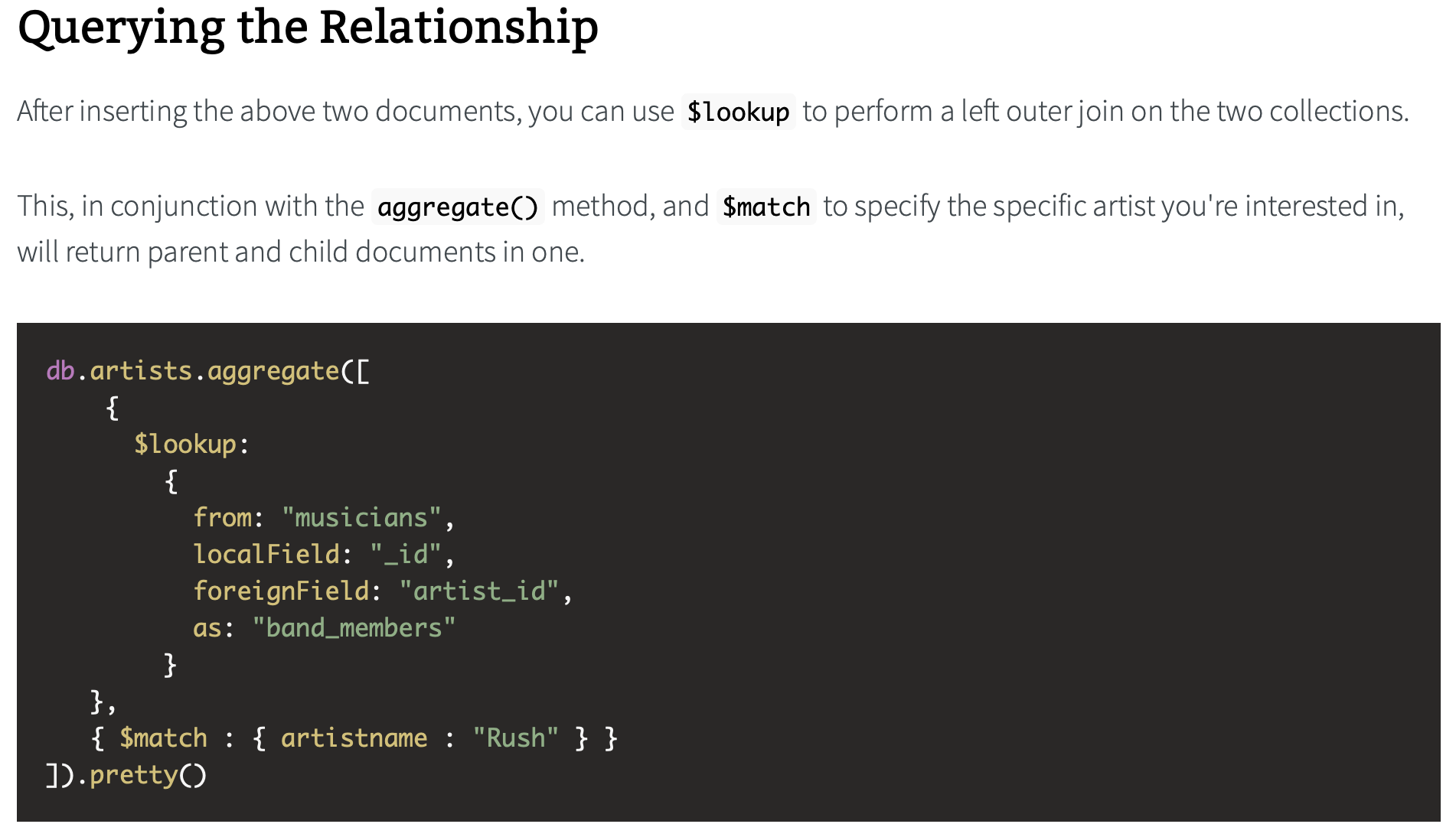
**Push to array $push**

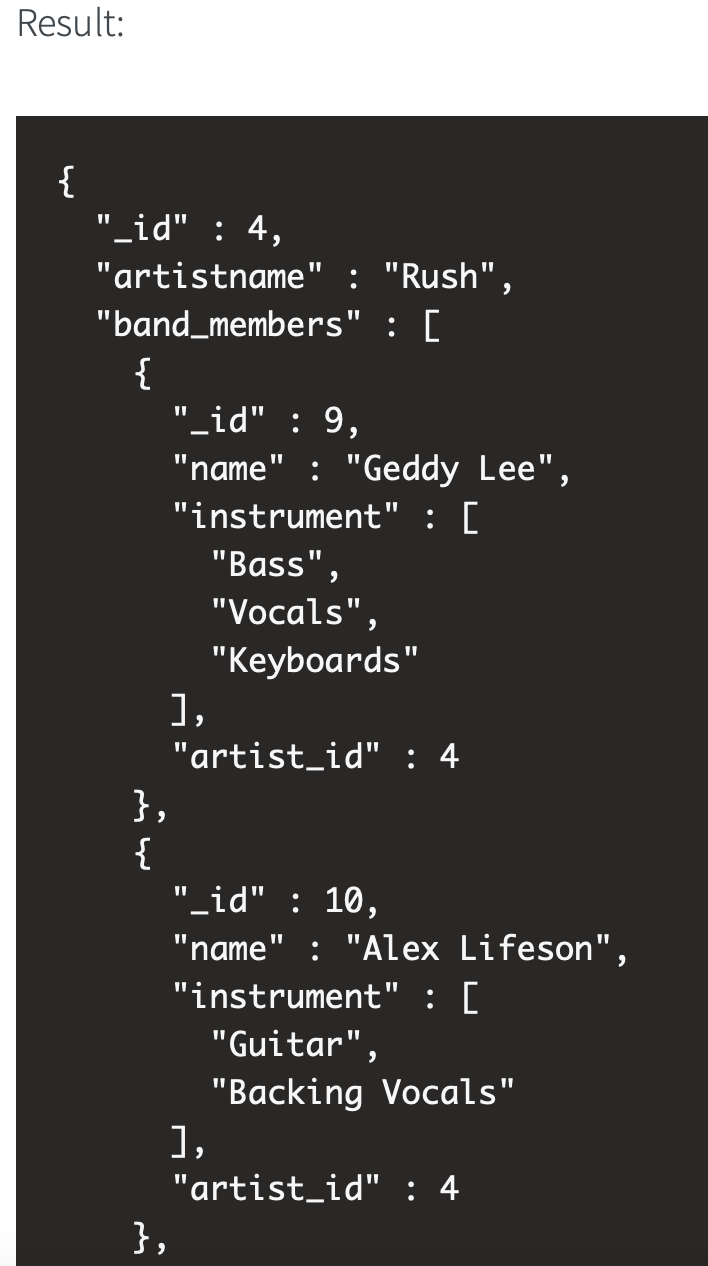
****

**To create a document referenced relationship in MongoDB**

****

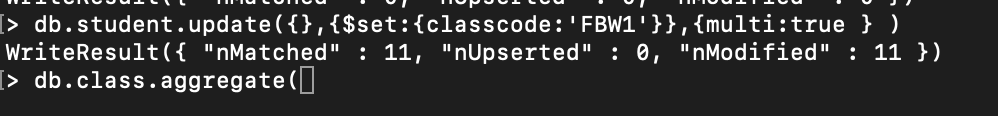
****

****

****

****

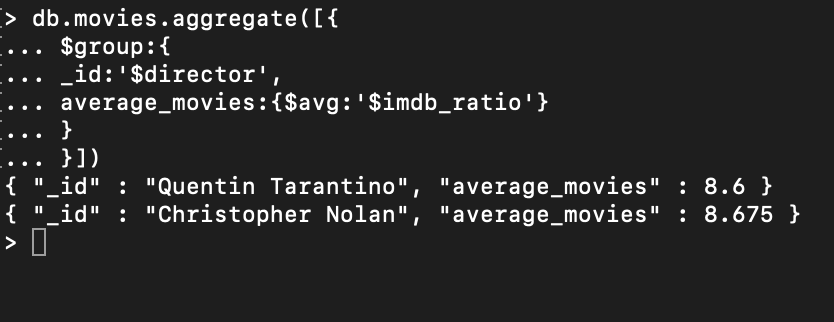
To apply on our example:







**Aggregate group**

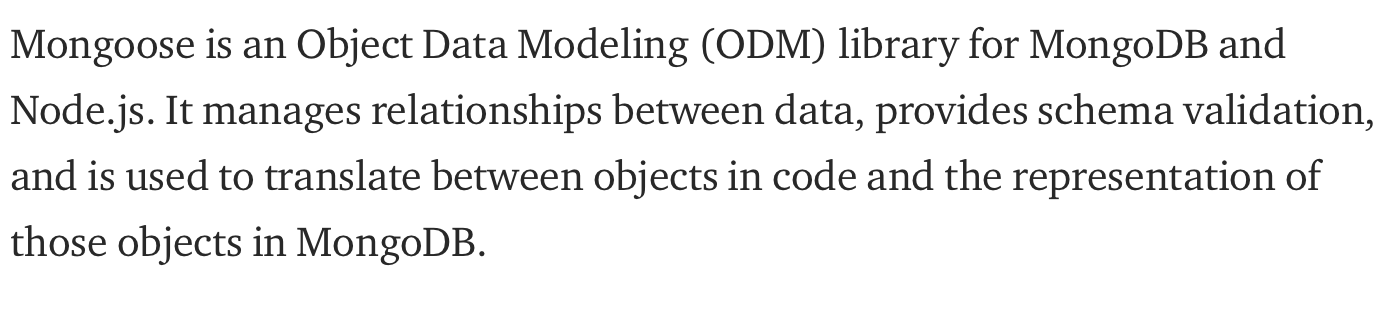


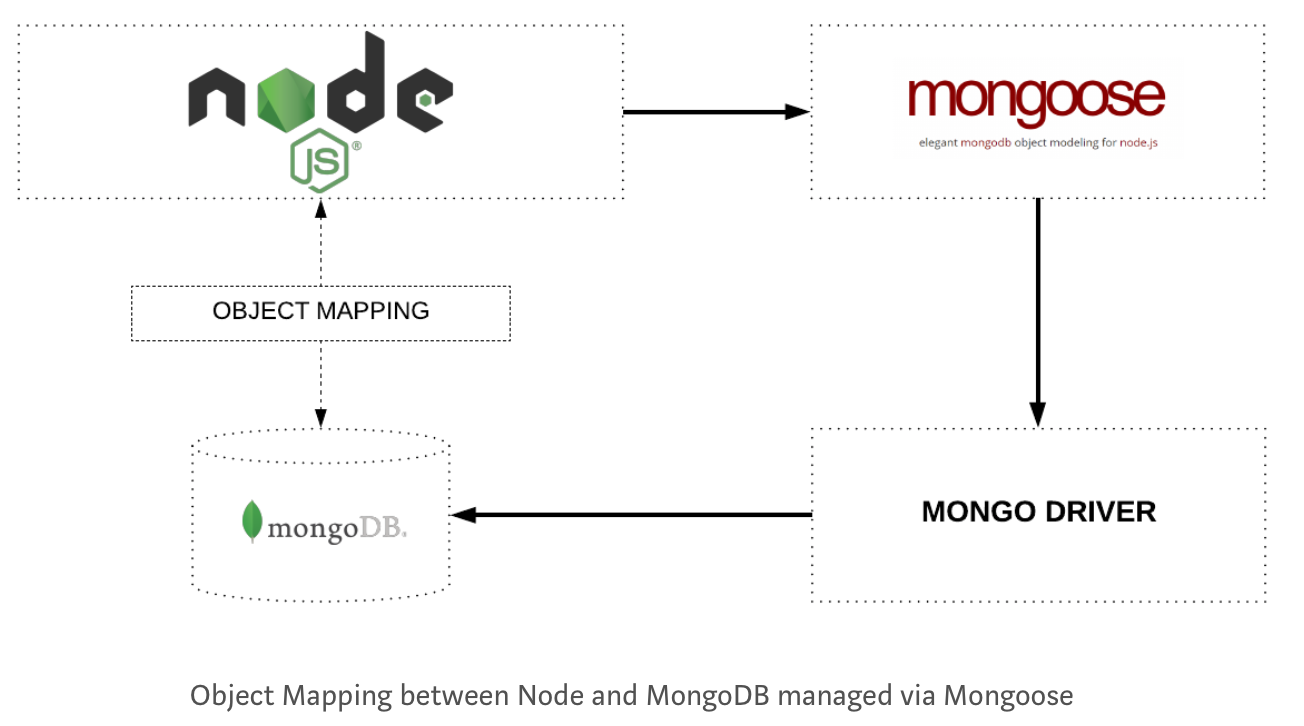
**MongoDB Compass**

**Trello**

Trello is a task management app that gives you a visual **overview** of what is being worked on and who is working on it. It used the Kanban system, which was developed in Toyota as a system to keep production levels high and maintain flexibility.

**mongoose**

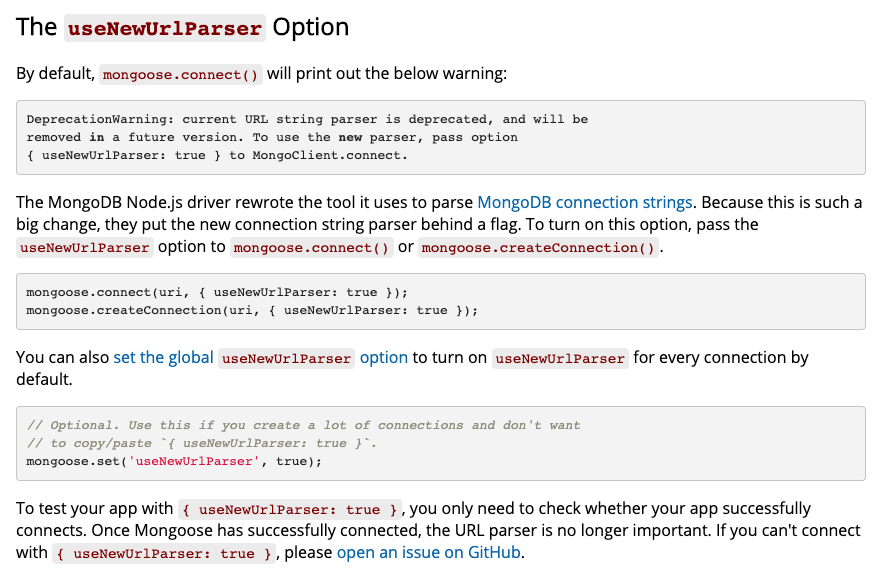
****

****

**Install this in your project:**

npm install --save mongoose

**mongoose.set('useNewUrlParser', true);**

****

<https://mongoosejs.com/docs/deprecations.html>

**css Example**

Css Example

**css Example**

Css Example