

QUIZ

Which of the following is a difference between the `.findAndModify()` and `.updateOne()` MongoDB methods?

Only `.findAndModify()` can accept the `upsert` option.

`.findAndModify()` can update multiple documents if they match a criteria, whereas the `.updateOne()` operation cannot.

The `.findAndModify()` method can return the updated document, whereas the `.updateOne()` method cannot.



Correct! When providing the optional parameter of `{ new: true }`, the modified document will be returned.

What option is used with the `.updateOne()` MongoDB command in order to create a new document if no matching document is found based on the query condition?

`{ $push: <document> }`

`{ new: true }`

`{ upsert: true }`



Correct! The `upsert` option is a combination of update and insert and will insert a new document if no matching document is found. It's set false by default.

Replacing the `update` command with `upsert`.

Indicate whether the following statement is true or false:

Once a document is inserted into a collection, we cannot change its `_id` field.

False

True



Correct! Once a document is generated and an `_id` has been assigned, it cannot be changed or updated.

Consider the following documents from a MongoDB collection called `nineties_artists`:

```
{ _id: ObjectId("62a1f23b9ff427af8df91a15"), title: "Waterfalls", artist: "TLC", year: 1994 },
{ _id: ObjectId("62a1f26cd8acb7f172075b97"), title: "Don't Speak", artist: "No Doubt", year: 1984 },
{ _id: ObjectId("62a1f273d9d77ec8c5c4c13f"), title: "Wonderwall", artist: "Oasis", year: 1995 }
```

Which of the following commands correctly updates the `year` field of the document with the title "Don't Speak" to 1995?

```
db.nineties_artists.updateOne( { title: "Don't Speak" } )
```

```
db.nineties_artists.updateOne( { title: "Don't Speak" }, { year: 1995 } )
```

```
db.nineties_artists.modifyOne( { title: "Don't Speak" }, { $set: { year: 1995 } } )
```

```
db.nineties_artists.updateOne( { title: "Don't Speak" }, { $set: { year: 1995 } } )
```



Correct! We can use the `.updateOne()` method to modify the `year` field.

Fill in the MongoDB command below so that it correctly inserts two documents into the `pets` collection: one with the name "Migo" and the other named "Lilou".

```
db.pets.insertMany (
  [ {name: "Migo"}, {name: "Lilou"} ]
)
```



You got it!

Consider the following documents from a MongoDB collection `musicians`:

```
{ _id: ObjectId("62a232193479b228c0cf5058"), name: "Tony Iommi", band: "Black Sabbath", active: true },
{ _id: ObjectId("62a2321f875d1161523484f2"), name: "Bill Ward", band: "Black Sabbath", active: true },
{ _id: ObjectId("62a23224b9f379107d9bf903"), name: "Geezer Butler", band: "Black Sabbath", active: true },
{ _id: ObjectId("62a232298d8a1eccd5abd0ed"), name: "Ozzy Osbourne", band: "Black Sabbath", active: true },
```

Fill in the MongoDB command below so that it correctly updates all of the documents' `active` field to `false`.

```
db.musicians. ({ band: "Black Sabbath"}, {  : { active: false } })
```



You got it!

Fill in the MongoDB command below so that it correctly appends a new value (`91`) into the `scores` field for the following document:

```
{
  _id: ObjectId('62a954f357908dd7af767467'),
  student_id: 34123,
  scores: [77, 89, 91, 82]
}
```

```
db.students. (
  { student_id: 34123 },
  {  : { scores: 91 } }
)
```



You got it!

Which of the following MongoDB commands successfully inserts the following JSON into a collection called `books`?

```
{
  title: "Tale of Two Cities",
  author: "Charles Dickens"
}
```

```
db.insertOne({ title: "Tale of Two Cities", author: "Charles Dickens" });
```

```
db.books.insertOne({title="Tale of Two Cities", author="Charles Dickens"});
```

```
db.books.insertOne({ title: "Tale of Two Cities", author: "Charles Dickens" });
```



That's right! The `.insertOne()` method takes in a document to insert as the first argument.

Fill in the MongoDB command below so that it correctly updates the following document's `name` field from `"CS50"` to `"CS101"`:

```
{
  _id: ObjectId("5349b4ddd2781d08c09890f4"),
  course_id: 4315,
  course_details: {
    name: "CS50",
    branch: "CS",
    capacity: 50
  }
}
```

```
db.courses.updateOne (
  { course_id: 4315 },
  { $set : { "course_details.name" : "CS101" } }
)
```



You got it!