```
1. JavaScript Shell
The first thing to notice is that the MongoDB shell is JavaScript-based.
So you can do things like:
  a = 5;
  a * 10;
  for(i=0; i<10; i++) { print('hello'); };
Try a few JS commands; when you're ready to move on, enter 'next'
> a=5; a*10; for(i=0; i<10; i++) { print('hello
JS Error: TypeError: Object # has no method 'error'
> a=5; a*10; for(i=0; i<10; i++) { print('hello');};
hello
hello
hello
hello
hello
hello
hello
```

```
> next
2. Documents
MongoDB is a document database. This means that we store data as documents,
which are similar to JavaScript objects. Here below are a few sample JS objects:
  var a = {age: 25};
  var n = {name: 'Ed', languages: ['c', 'ruby', 'js']};
  var student = {name: 'Jim', scores: [75, 99, 87.2]};
Create some documents, then enter 'next'
> var a = {age: 21; var n = {name: "Ash', scores: ['c', 'ruby', 'js']}; var student = {nam
JS Error: TypeError: Object # has no method 'error'
> var a = {age: 21}; var n = {name: "Ash', languages: ['c', 'ruby', 'js']}; var student =
JS Error: TypeError: Object # has no method 'error'
> var a = {age: 21}; var n = {name: "Ash', languages: ['c', 'ruby', 'js']}; var student =
JS Error: TypeError: Object # has no method 'error'
> var a = {age: 21};
{
```

```
JS Error: TypeError: Object # has no method 'error'
> var a = {age: 21};
{
 "age" : 21
 }
> var n = {name: 'Ash', languages: ['c', 'ruby', 'js']};
{
 "name" : "Ash",
 "languages" : [   "c",   "ruby",   "js" ]
 }
> var student = {name: 'Boyd', scores: [100, 95, 90]};
{
 "name" : "Boyd",
 "scores" : [   100,   95,   90 ]
 }
> next
```

```
3. Saving
Here's how you save a document to MongoDB:
  db.scores.save({a: 99});

This says, "save the document '{a: 99}' to the 'scores' collection."
Go ahead and try it. Then, to see if the document was saved, try
  db.scores.find();
Once you've tried this, type 'next'.

> db.scores.save({a: 100});
"ok"
> db.scores.find();

[
  {   "a" : 99,   "_id" : {   "$oid" : "5145e979cc93742c16032e31"   }   },
  {   "a" : 100,   "_id" : {   "$oid" : "5145eb82cc93742c16032e44"   }   }
]
```

```
> next
4. Saving and Querying
Try adding some documents to the scores collection:
  for(i=0; i<10; i++) { db.scores.save({a: i, exam: 5}) };

Try that, then enter
  db.scores.find();
to see if the save succeeded. Since the shell only displays 10 results at time,
you'll need to enter the 'it' command to iterate over the rest.

(enter 'next' when you're ready)
> for(i=0; i<10; i++) { db.scores.save({a: i, exam: 5}) };
"ok"
> db.scores.find();

[
  {   "a" : 99,   "_id" : {   "$oid" : "5145e979cc93742c16032e31"   }   },
```

```
> for(i=0; i<10; i++) { db.scores.save({a: i, exam: 5}) };
"ok"
> db.scores.find();

[
  {   "a" : 99,   "_id" : {   "$oid" : "5145e979cc93742c16032e31"   }   },
  {   "a" : 100,   "_id" : {   "$oid" : "5145eb82cc93742c16032e44"   }   },
  {   "exam" : 5,   "a" : 0,   "_id" : {   "$oid" : "5145ebb7cc93742c16032e45"   }   },
  {   "exam" : 5,   "a" : 2,   "_id" : {   "$oid" : "5145ebb7cc93742c16032e46"   }   },
  {   "exam" : 5,   "a" : 1,   "_id" : {   "$oid" : "5145ebb7cc93742c16032e47"   }   },
  {   "exam" : 5,   "a" : 6,   "_id" : {   "$oid" : "5145ebb8cc93742c16032e48"   }   },
  {   "exam" : 5,   "a" : 7,   "_id" : {   "$oid" : "5145ebb8cc93742c16032e49"   }   },
  {   "exam" : 5,   "a" : 8,   "_id" : {   "$oid" : "5145ebb8cc93742c16032e4a"   }   },
  {   "exam" : 5,   "a" : 9,   "_id" : {   "$oid" : "5145ebb8cc93742c16032e4b"   }   },
  {   "exam" : 5,   "a" : 4,   "_id" : {   "$oid" : "5145ebb8cc93742c16032e4c"   }   }
]
> next
```

5. Basic Queries
You've already tried a few queries, but let's make them more specific.
How about finding all documents where a == 2:

```
db.scores.find({a: 2});
```

Or what about documents where a > 15?

```
db.scores.find({a: {'$gt': 15}});
```

```
> db.scores.find({a: 2});

[
  {   "exam" : 5,   "a" : 2,   "_id" : {   "$oid" : "5145ebb7cc93742c16032e46"   }   }
]
> db.scores.find({a: {'$gt': 15}});

[
  {   "a" : 99,   "_id" : {   "$oid" : "5145e979cc93742c16032e31"   }   },
  {   "a" : 100,   "_id" : {   "$oid" : "5145eb82cc93742c16032e44"   }   }
```

```
  db.scores.find({a: 2});

Or what about documents where a > 15?
  db.scores.find({a: {'$gt': 15}});

> db.scores.find({a: 2});

[
  {   "exam" : 5,   "a" : 2,   "_id" : {   "$oid" : "5145ebb7cc93742c16032e46"   }   }
]
> db.scores.find({a: {'$gt': 15}});

[
  {   "a" : 99,   "_id" : {   "$oid" : "5145e979cc93742c16032e31"   }   },
  {   "a" : 100,   "_id" : {   "$oid" : "5145eb82cc93742c16032e44"   }   }
]
> next
```