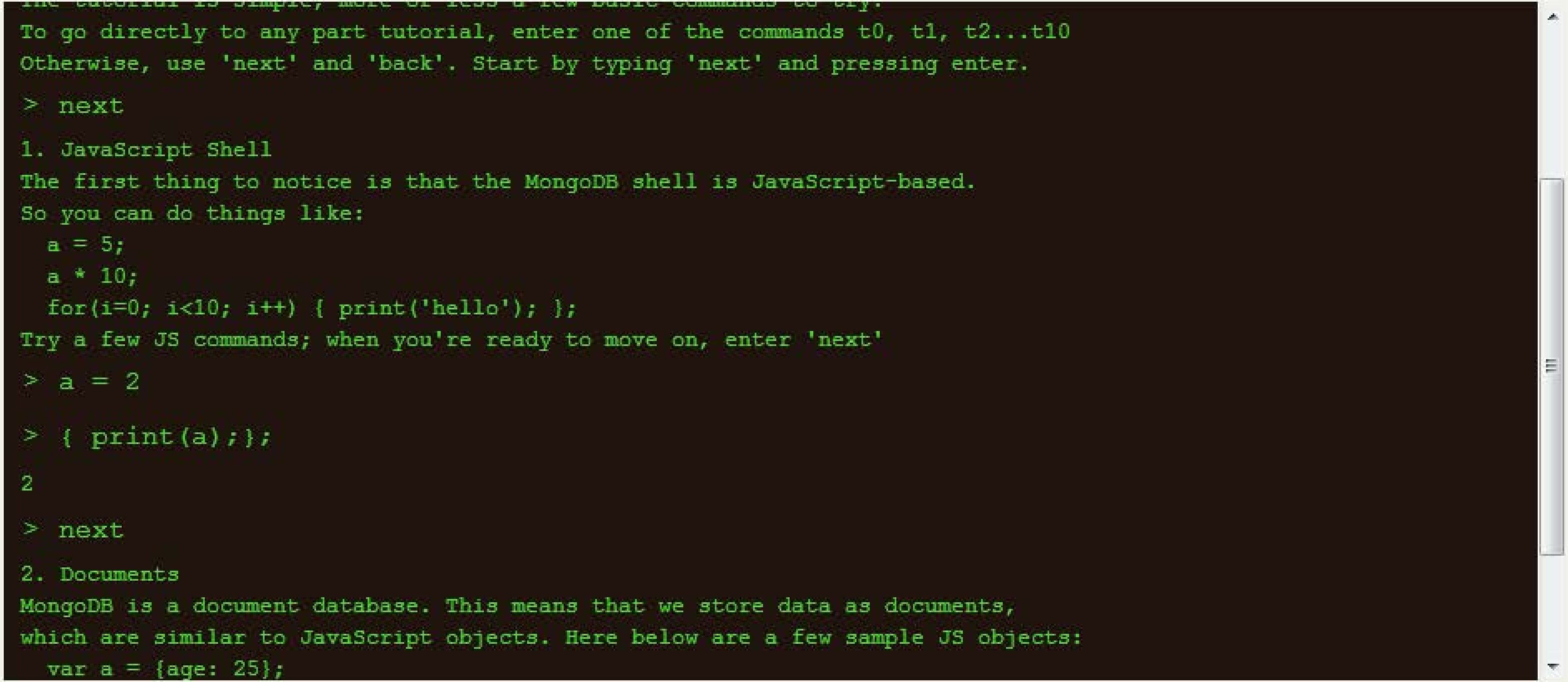
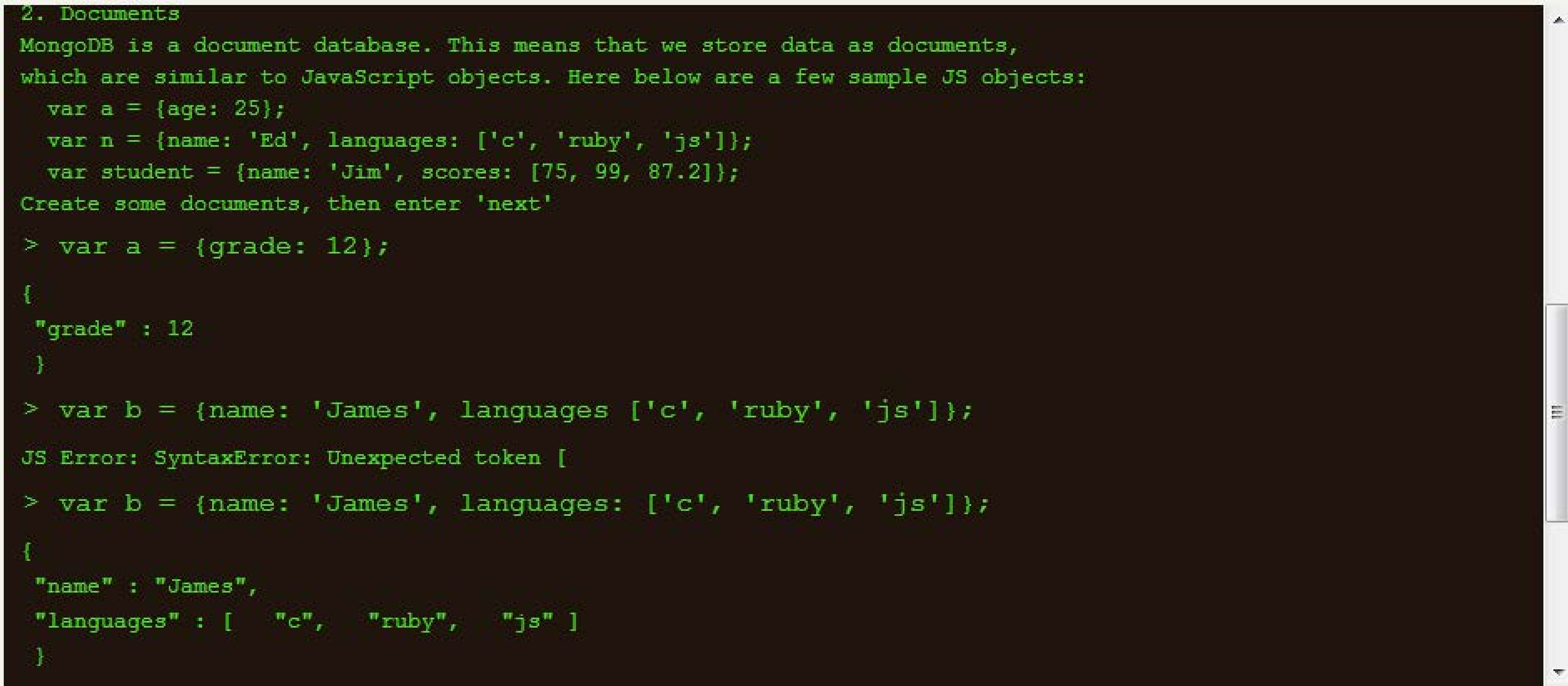
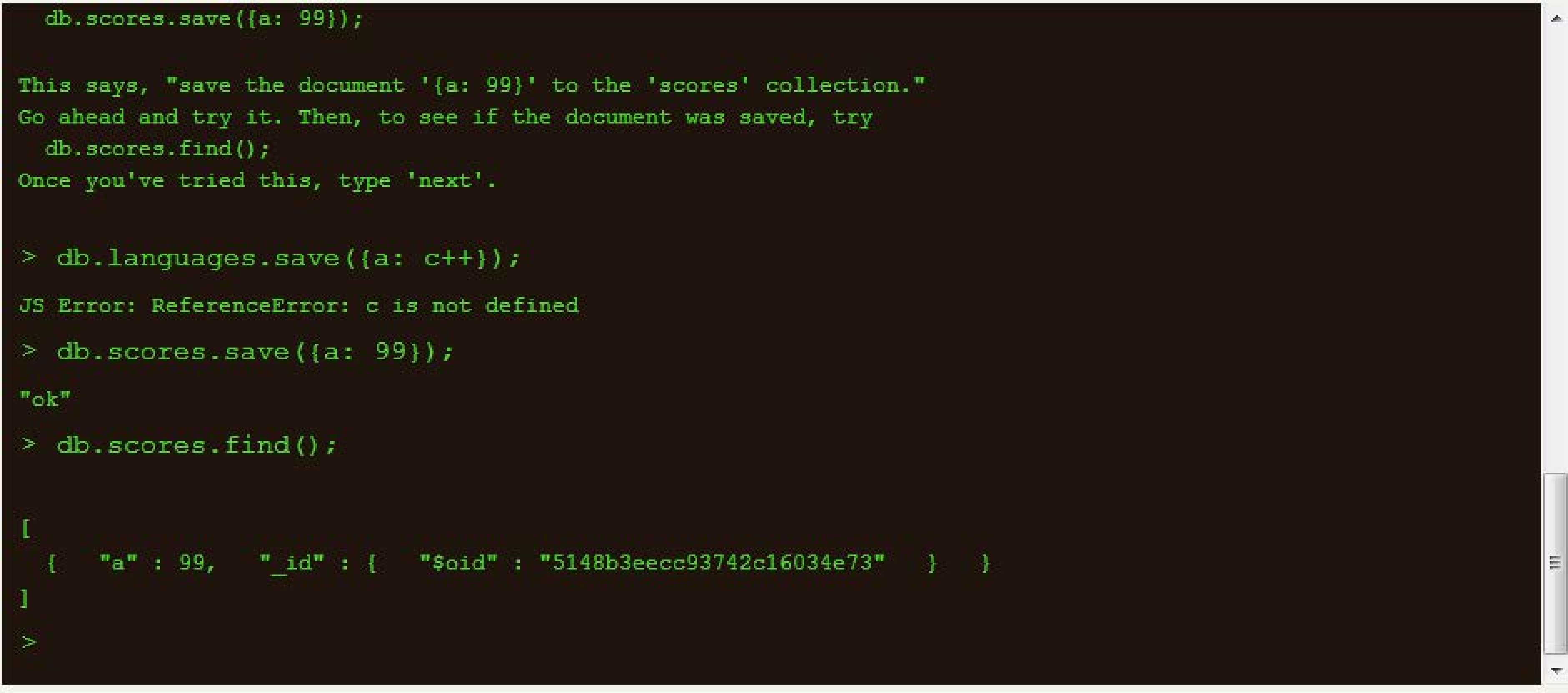


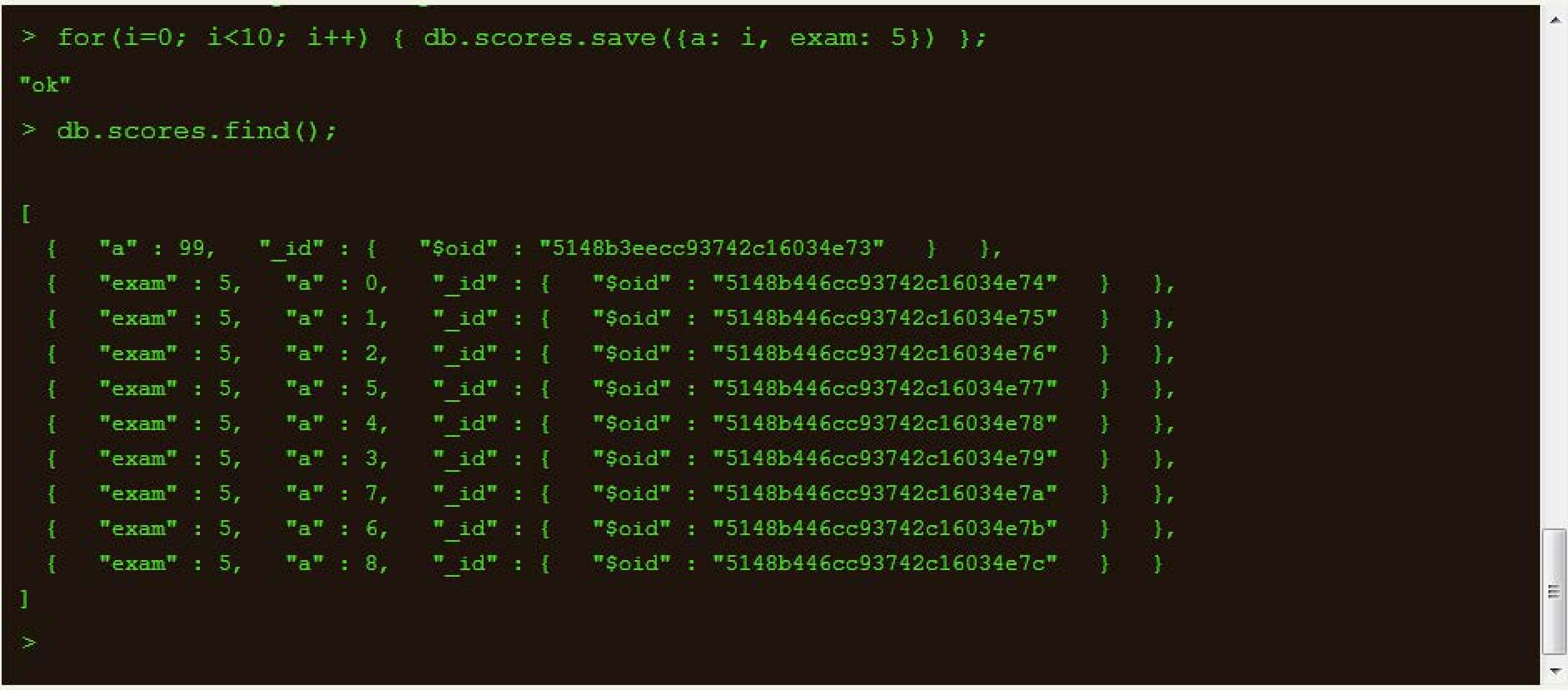
```
To go directly to any part tutorial, enter one of the commands t0, t1, t2...t10
Otherwise, use 'next' and 'back'. Start by typing 'next' and pressing enter.

> next

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 = 2

> { print(a);};

2

> 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};
```



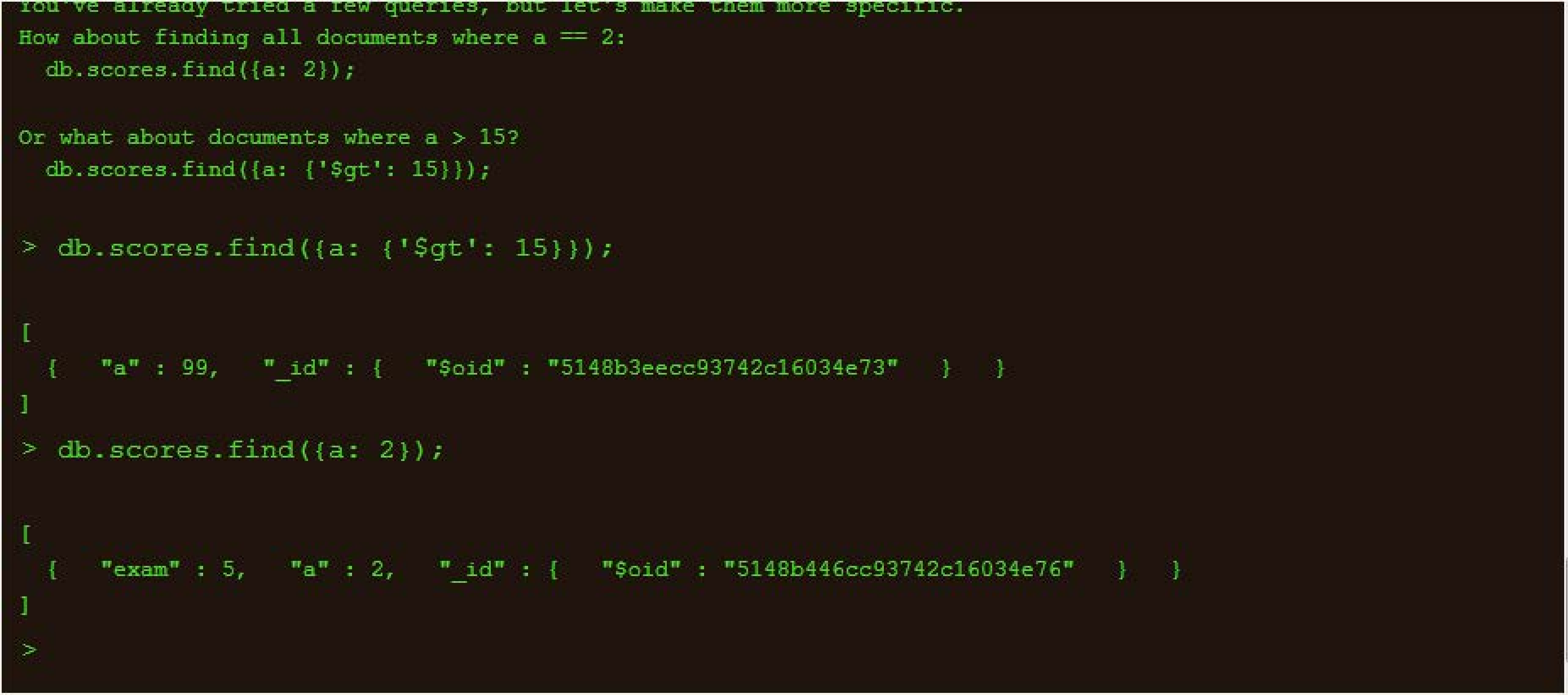
```
2. Documents
MongoDB is a document database. This means that we store data as documents,
which are similar to JavaScript objects. Here below are a few sample JS objects:
  var a = {age: 25};
  var n = {name: 'Ed', languages: ['c', 'ruby', 'js']};
  var student = {name: 'Jim', scores: [75, 99, 87.2]};
Create some documents, then enter 'next'
> var a = {grade: 12};
{
 "grade" : 12
 }
> var b = {name: 'James', languages ['c', 'ruby', 'js']};
JS Error: SyntaxError: Unexpected token [
> var b = {name: 'James', languages: ['c', 'ruby', 'js']};
{
 "name" : "James",
 "languages" : [   "c",   "ruby",   "js" ]
 }
```



```
  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.languages.save({a: c++});

JS Error: ReferenceError: c is not defined

> db.scores.save({a: 99});

"ok"

> db.scores.find();


[
  {   "a" : 99,   "_id" : {   "$oid" : "5148b3eecc93742c16034e73"   }   }
]

>
```



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

"ok"

> db.scores.find();


[
  {   "a" : 99,   "_id" : {   "$oid" : "5148b3eecc93742c16034e73"   }   },
  {   "exam" : 5,   "a" : 0,   "_id" : {   "$oid" : "5148b446cc93742c16034e74"   }   },
  {   "exam" : 5,   "a" : 1,   "_id" : {   "$oid" : "5148b446cc93742c16034e75"   }   },
  {   "exam" : 5,   "a" : 2,   "_id" : {   "$oid" : "5148b446cc93742c16034e76"   }   },
  {   "exam" : 5,   "a" : 5,   "_id" : {   "$oid" : "5148b446cc93742c16034e77"   }   },
  {   "exam" : 5,   "a" : 4,   "_id" : {   "$oid" : "5148b446cc93742c16034e78"   }   },
  {   "exam" : 5,   "a" : 3,   "_id" : {   "$oid" : "5148b446cc93742c16034e79"   }   },
  {   "exam" : 5,   "a" : 7,   "_id" : {   "$oid" : "5148b446cc93742c16034e7a"   }   },
  {   "exam" : 5,   "a" : 6,   "_id" : {   "$oid" : "5148b446cc93742c16034e7b"   }   },
  {   "exam" : 5,   "a" : 8,   "_id" : {   "$oid" : "5148b446cc93742c16034e7c"   }   }
]

>
```



```
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: {'$gt': 15}});


[
  {   "a" : 99,   "_id" : {   "$oid" : "5148b3eecc93742c16034e73"   }   }
]
> db.scores.find({a: 2});


[
  {   "exam" : 5,   "a" : 2,   "_id" : {   "$oid" : "5148b446cc93742c16034e76"   }   }
]
>
```