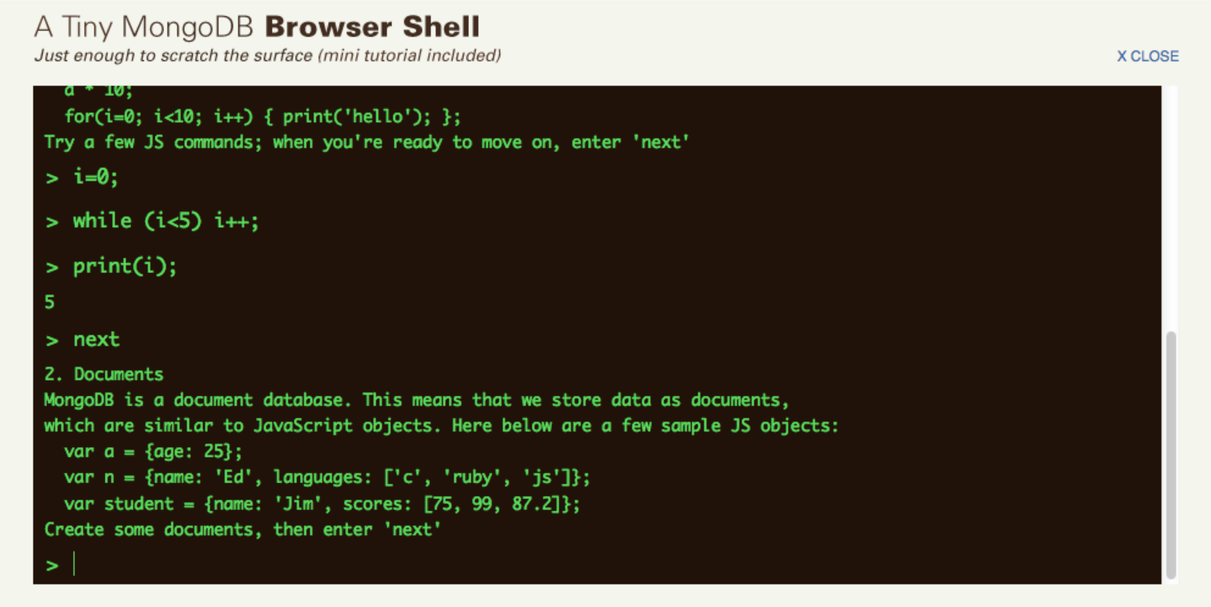
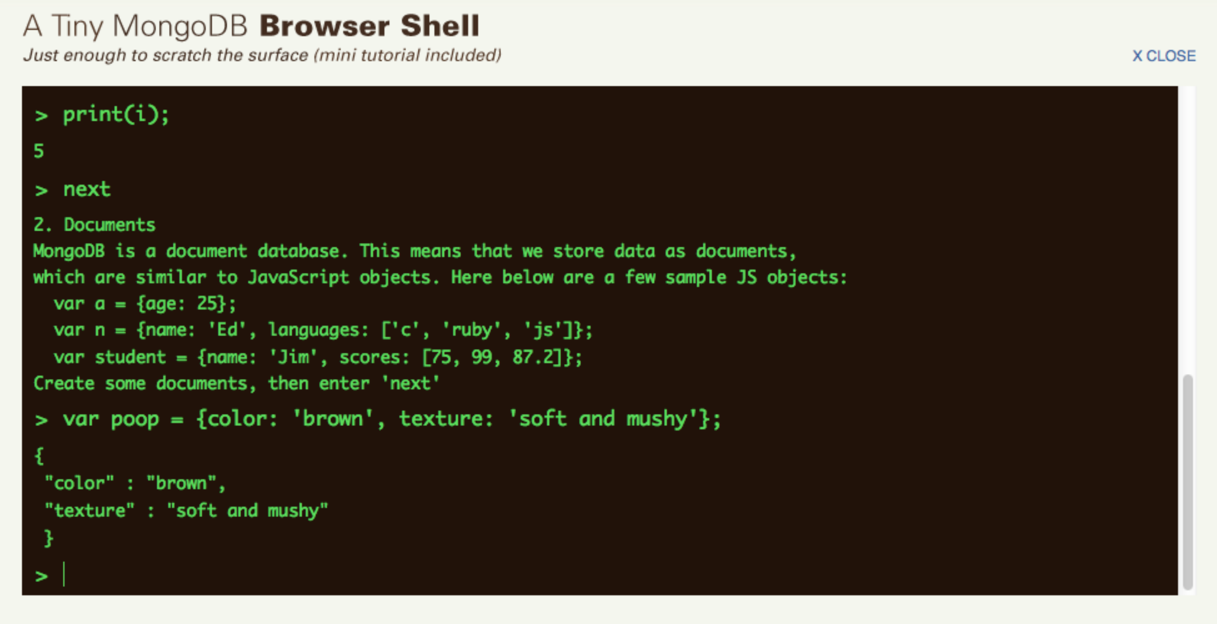
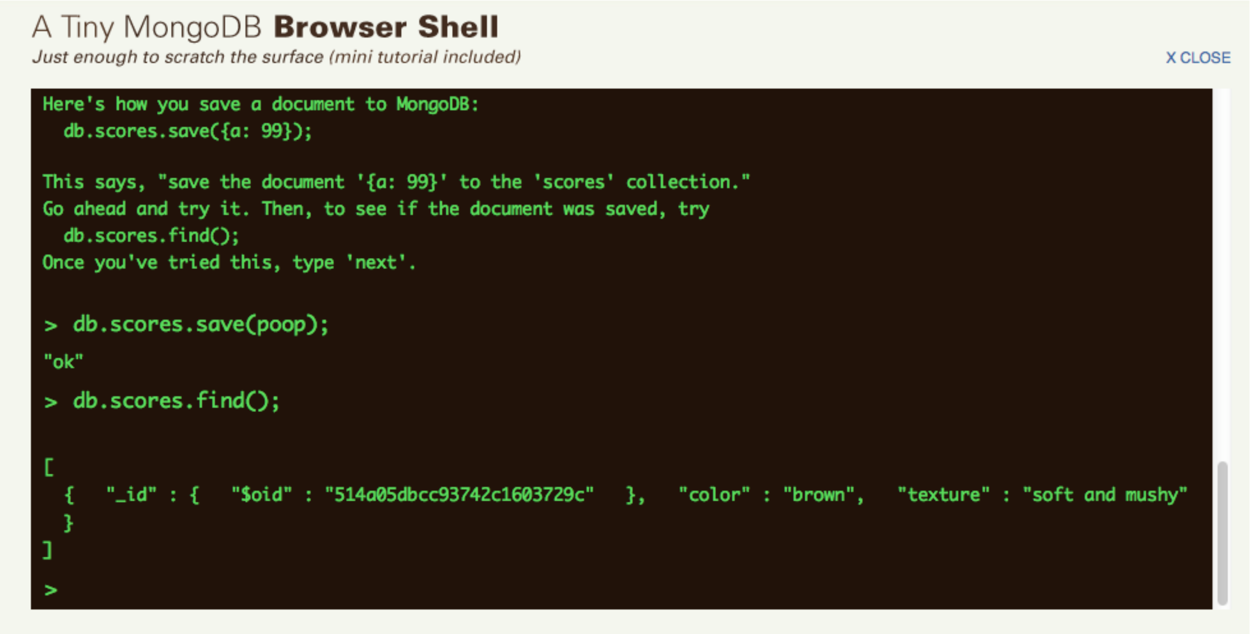

A Tiny MongoDB **Browser Shell**
Just enough to scratch the surface (mini tutorial included)
X CLOSE

```
  a * 10;
  for(i=0; i<10; i++) { print('hello'); };
Try a few JS commands; when you're ready to move on, enter 'next'
> i=0;
> while (i<5) i++;
> print(i);
5
> 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'
> |
```


A Tiny MongoDB **Browser Shell**
Just enough to scratch the surface (mini tutorial included)
X CLOSE

```
> print(i);
5
> 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 poop = {color: 'brown', texture: 'soft and mushy'};
{
 "color" : "brown",
 "texture" : "soft and mushy"
 }
> |
```



A Tiny MongoDB **Browser Shell**

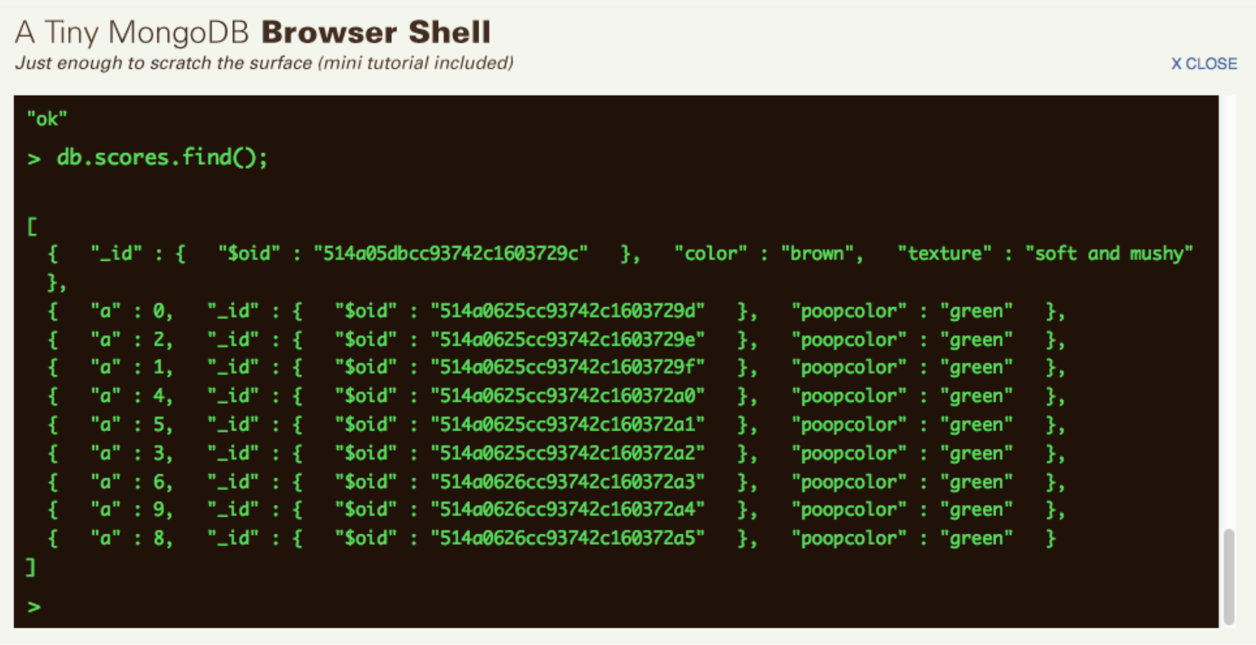
Just enough to scratch the surface (mini tutorial included)

```
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(poop);
"ok"
> db.scores.find();


[
  {   "_id" : {   "$oid" : "514a05dbcc93742c1603729c"   },   "color" : "brown",   "texture" : "soft and mushy"
  }
]
>
```

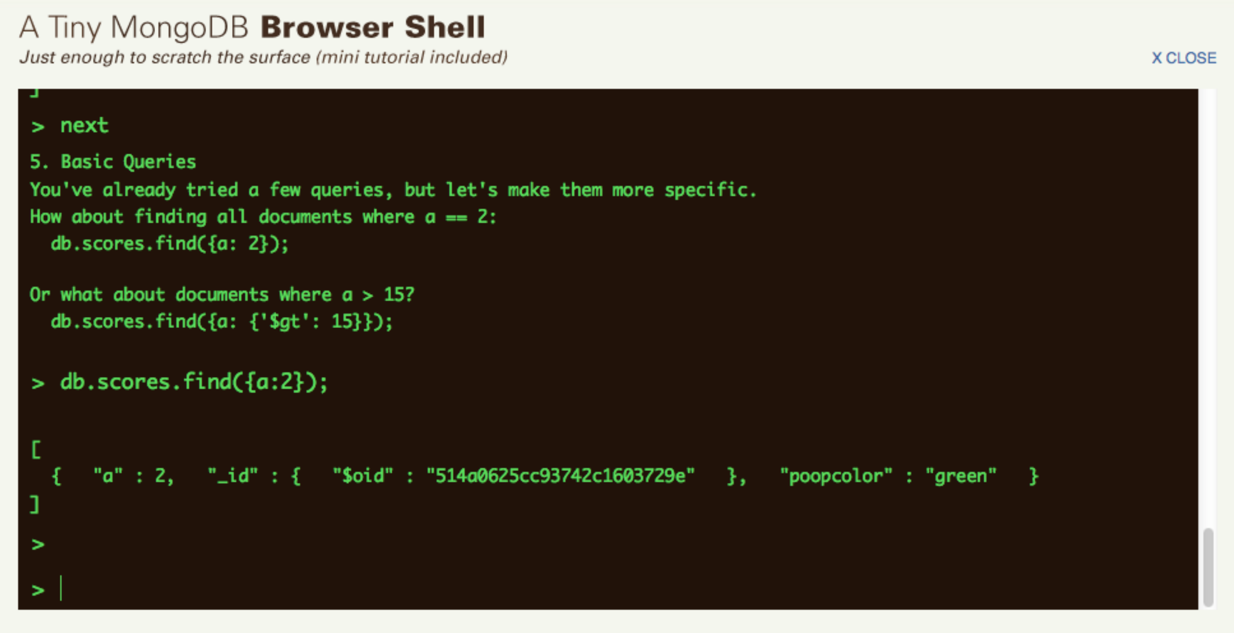


A Tiny MongoDB **Browser Shell**

Just enough to scratch the surface (mini tutorial included)

```
"ok"
> db.scores.find();


[
  {   "_id" : {   "$oid" : "514a05dbcc93742c1603729c"   },   "color" : "brown",   "texture" : "soft and mushy"
  },
  {   "a" : 0,   "_id" : {   "$oid" : "514a0625cc93742c1603729d"   },   "poopcolor" : "green"   },
  {   "a" : 2,   "_id" : {   "$oid" : "514a0625cc93742c1603729e"   },   "poopcolor" : "green"   },
  {   "a" : 1,   "_id" : {   "$oid" : "514a0625cc93742c1603729f"   },   "poopcolor" : "green"   },
  {   "a" : 4,   "_id" : {   "$oid" : "514a0625cc93742c160372a0"   },   "poopcolor" : "green"   },
  {   "a" : 5,   "_id" : {   "$oid" : "514a0625cc93742c160372a1"   },   "poopcolor" : "green"   },
  {   "a" : 3,   "_id" : {   "$oid" : "514a0625cc93742c160372a2"   },   "poopcolor" : "green"   },
  {   "a" : 6,   "_id" : {   "$oid" : "514a0626cc93742c160372a3"   },   "poopcolor" : "green"   },
  {   "a" : 9,   "_id" : {   "$oid" : "514a0626cc93742c160372a4"   },   "poopcolor" : "green"   },
  {   "a" : 8,   "_id" : {   "$oid" : "514a0626cc93742c160372a5"   },   "poopcolor" : "green"   }
]
>
```


A Tiny MongoDB Browser Shell
Just enough to scratch the surface (mini tutorial included)
X CLOSE

```
]
> 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});


[
  {   "a" : 2,   "_id" : {   "$oid" : "514a0625cc93742c1603729e"   },   "poopcolor" : "green"   }
]
>

> |
```