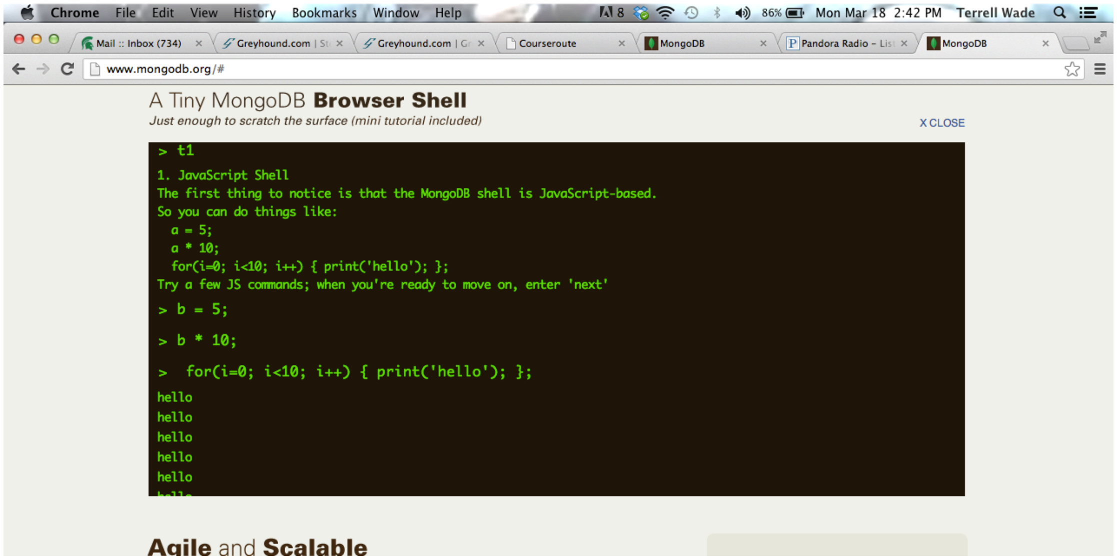
Terrell Wade
Tc 362
Tutorials t1-t5

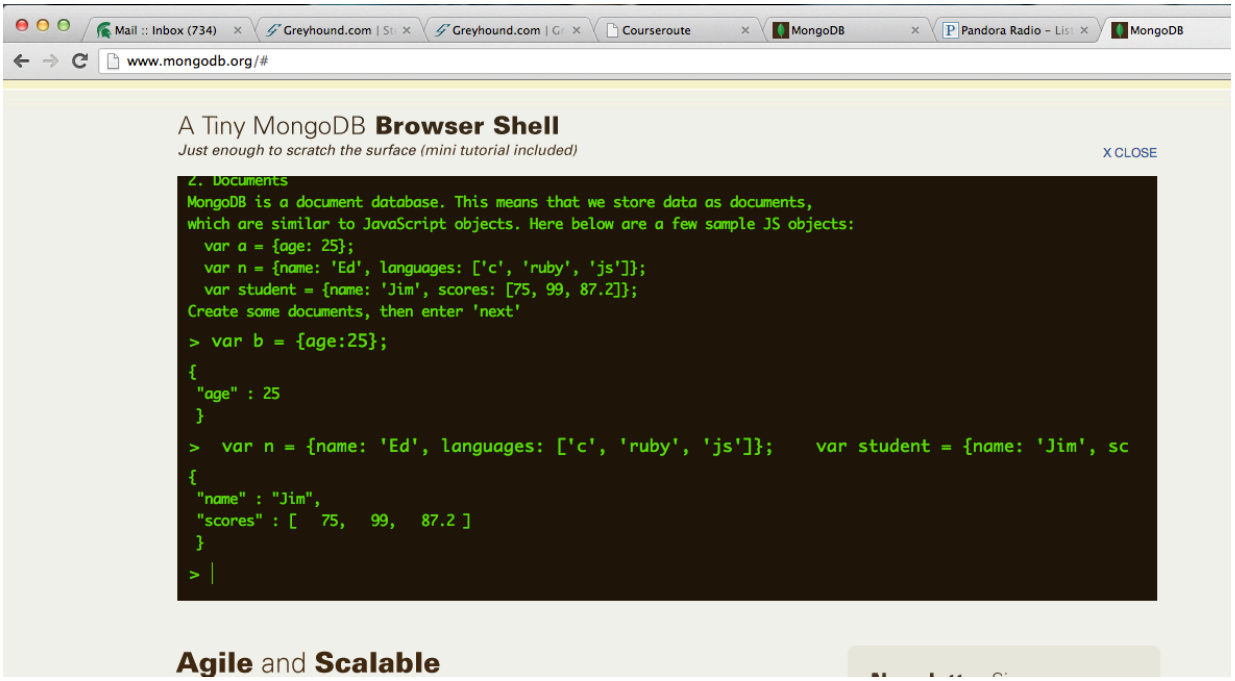


A Tiny MongoDB **Browser Shell**

Just enough to scratch the surface (mini tutorial included)

```
> t1
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'
> b = 5;
> b * 10;
>  for(i=0; i<10; i++) { print('hello'); };
hello
hello
hello
hello
hello
```

Agile and **Scalable**

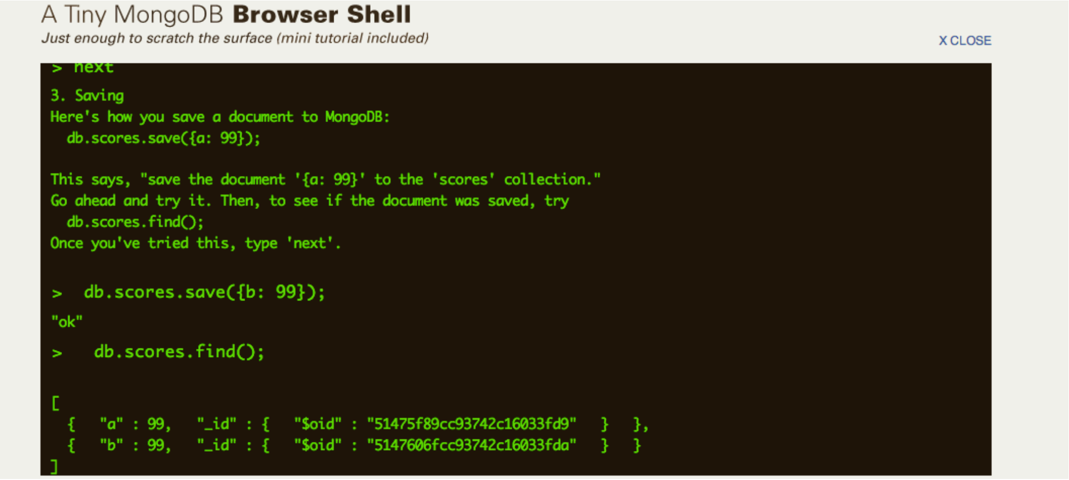


A Tiny MongoDB **Browser Shell**

Just enough to scratch the surface (mini tutorial included)

```
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 b = {age:25};
{
 "age" : 25
 }
>  var n = {name: 'Ed', languages: ['c', 'ruby', 'js']};    var student = {name: 'Jim', sc
{
 "name" : "Jim",
 "scores" : [   75,   99,   87.2 ]
 }
> |
```

Agile and **Scalable**



A Tiny MongoDB **Browser Shell**

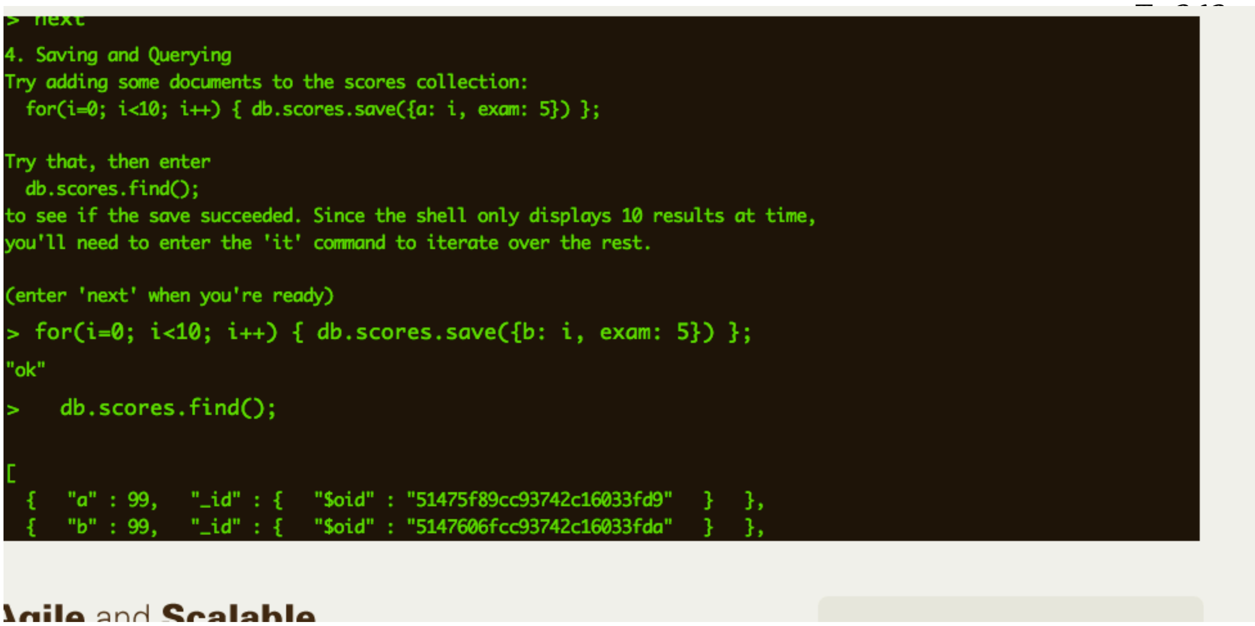
Just enough to scratch the surface (mini tutorial included)

```
> 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({b: 99});
"ok"
>   db.scores.find();

[
  {   "a" : 99,   "_id" : {   "$oid" : "51475f89cc93742c16033fd9"   }   },
  {   "b" : 99,   "_id" : {   "$oid" : "5147606fcc93742c16033fda"   }   }
]
```



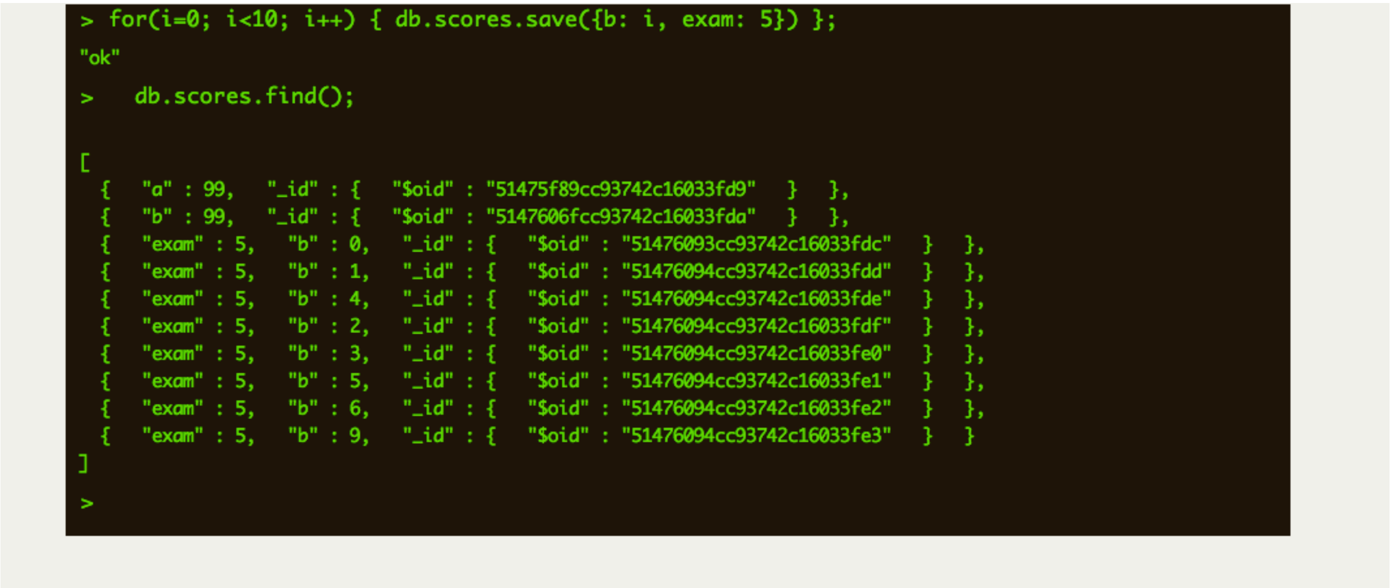
```
> next
4. Saving and Querying
Try adding some documents to the scores collection:
  for(i=0; i<10; i++) { db.scores.save({a: i, exam: 5}) };

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({b: i, exam: 5}) };
"ok"
>   db.scores.find();

[
  {   "a" : 99,   "_id" : {   "$oid" : "51475f89cc93742c16033fd9"   }   },
  {   "b" : 99,   "_id" : {   "$oid" : "5147606fcc93742c16033fda"   }   },
```

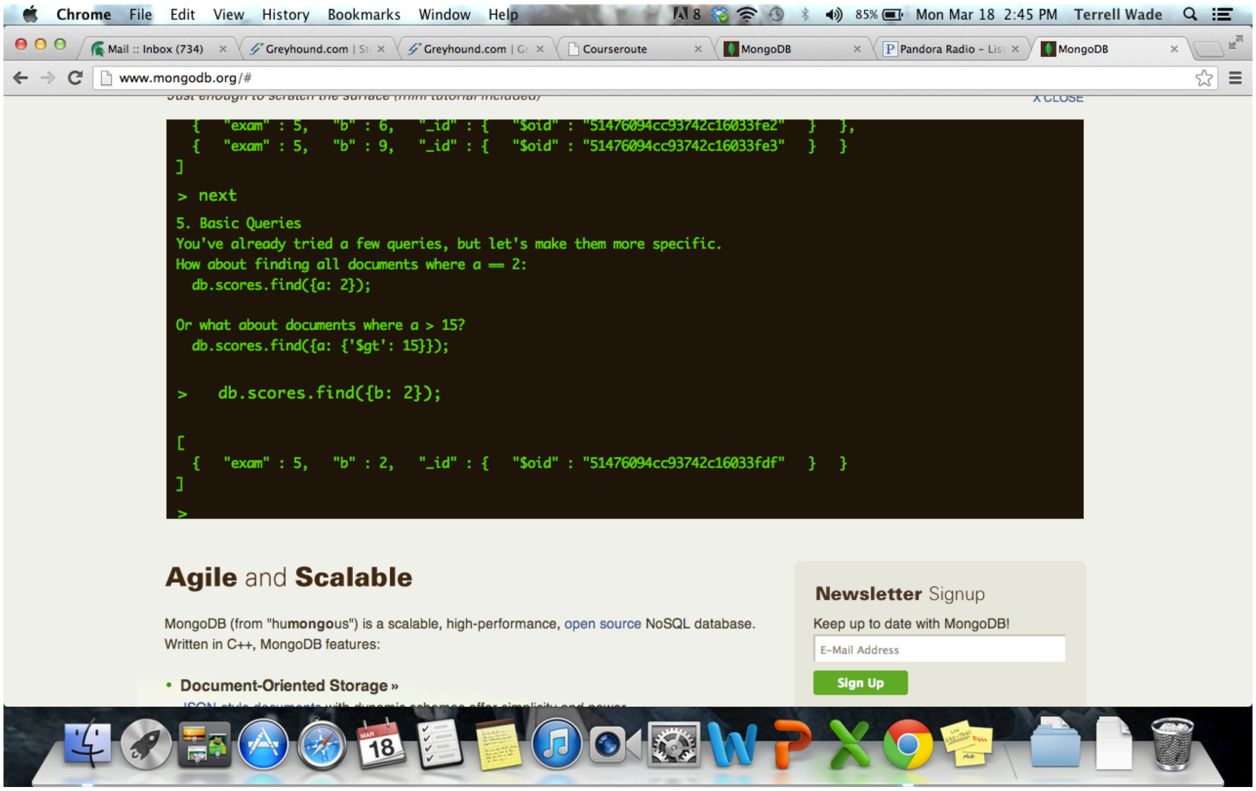
Agile and Scalable



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

[
  {   "a" : 99,   "_id" : {   "$oid" : "51475f89cc93742c16033fd9"   }   },
  {   "b" : 99,   "_id" : {   "$oid" : "5147606fcc93742c16033fda"   }   },
  {   "exam" : 5,   "b" : 0,   "_id" : {   "$oid" : "51476093cc93742c16033fdc"   }   },
  {   "exam" : 5,   "b" : 1,   "_id" : {   "$oid" : "51476094cc93742c16033fdd"   }   },
  {   "exam" : 5,   "b" : 4,   "_id" : {   "$oid" : "51476094cc93742c16033fde"   }   },
  {   "exam" : 5,   "b" : 2,   "_id" : {   "$oid" : "51476094cc93742c16033fdf"   }   },
  {   "exam" : 5,   "b" : 3,   "_id" : {   "$oid" : "51476094cc93742c16033fe0"   }   },
  {   "exam" : 5,   "b" : 5,   "_id" : {   "$oid" : "51476094cc93742c16033fe1"   }   },
  {   "exam" : 5,   "b" : 6,   "_id" : {   "$oid" : "51476094cc93742c16033fe2"   }   },
  {   "exam" : 5,   "b" : 9,   "_id" : {   "$oid" : "51476094cc93742c16033fe3"   }   }
]
>
```

Terrell Wade
Tc 362
Tutorials t1-t5



Just enough to scratch the surface (mini tutorial included)

```
  {   "exam" : 5,   "b" : 6,   "_id" : {   "$oid" : "51476094cc93742c16033fe2"   }  },
  {   "exam" : 5,   "b" : 9,   "_id" : {   "$oid" : "51476094cc93742c16033fe3"   }  }
]
> 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({b: 2});

[
  {   "exam" : 5,   "b" : 2,   "_id" : {   "$oid" : "51476094cc93742c16033fdf"   }  }
]
>
```

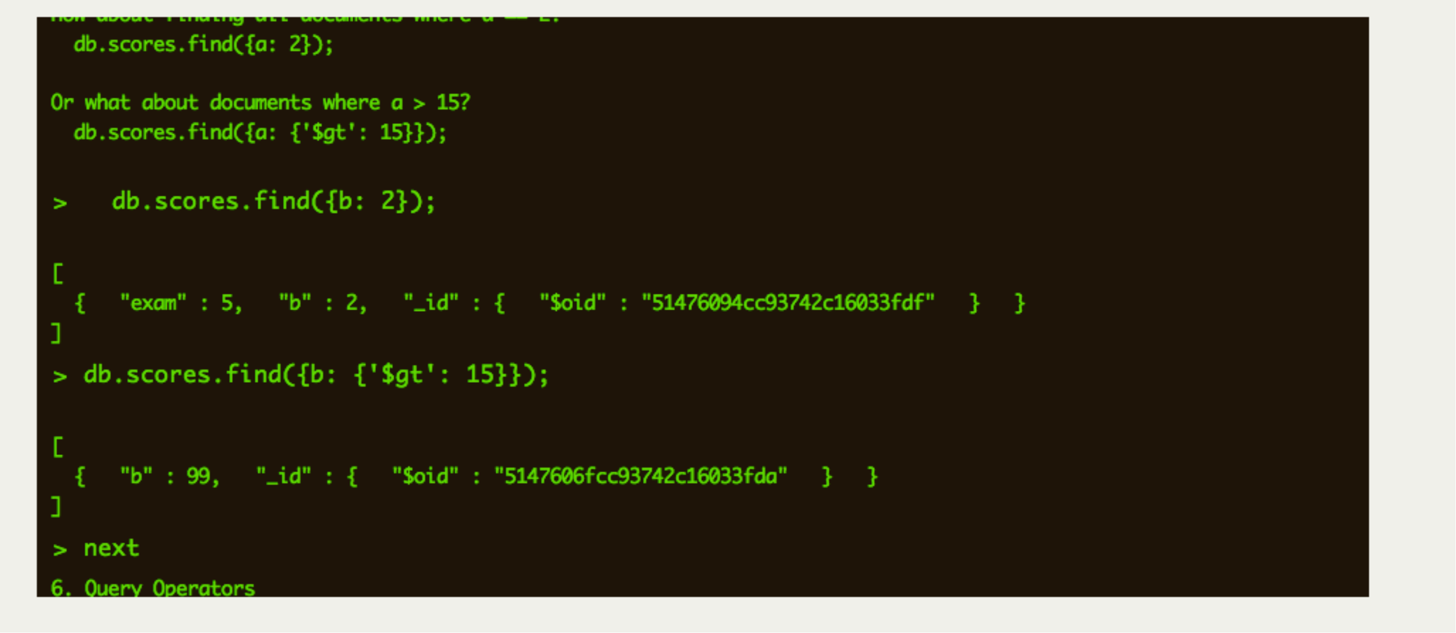
Agile and **Scalable**

MongoDB (from "humongous") is a scalable, high-performance, open source NoSQL database. Written in C++, MongoDB features:

- **Document-Oriented Storage »**
  JSON-style documents with dynamic schemas offer simplicity and power

Newsletter Signup

Keep up to date with MongoDB!



```
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({b: 2});

[
  {   "exam" : 5,   "b" : 2,   "_id" : {   "$oid" : "51476094cc93742c16033fdf"   }   }
]
> db.scores.find({b: {'$gt': 15}});

[
  {   "b" : 99,   "_id" : {   "$oid" : "5147606fcc93742c16033fda"   }   }
]
> next
6. Query Operators
```