```
   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'); };

 hello
 hello
 hello
 hello
 hello
 hello
 hello
 hello
 hello
 hello
```

```
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: 25};

{
 "age" : 25
 }

> var n {name: 'chris', languages: ['a', 'gold', 'js']};

JS Error: SyntaxError: missing ; before statement

> var n = {name: 'chris', languages: ['a', 'gold', 'js']};

{
 "name" : "chris",
 "languages" : [   "a",   "gold",   "js" ]
 }
```

Agile and **Scalable**

```
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: 99});

"ok"

> db.scores.find();


[
  {   "a" : 99,   "_id" : {   "$oid" : "5148f352cc93742c16035383"   }   }
]

>
```

```
(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" : {    "$oid" : "5148f352cc93742c16035383"    }   },
  {   "exam" : 5,    "a" : 1,    "_id" : {    "$oid" : "5148f3a5cc93742c16035384"    }   },
  {   "exam" : 5,    "a" : 0,    "_id" : {    "$oid" : "5148f3a5cc93742c16035385"    }   },
  {   "exam" : 5,    "a" : 5,    "_id" : {    "$oid" : "5148f3a5cc93742c16035386"    }   },
  {   "exam" : 5,    "a" : 4,    "_id" : {    "$oid" : "5148f3a5cc93742c16035387"    }   },
  {   "exam" : 5,    "a" : 3,    "_id" : {    "$oid" : "5148f3a5cc93742c16035388"    }   },
  {   "exam" : 5,    "a" : 2,    "_id" : {    "$oid" : "5148f3a5cc93742c16035389"    }   },
  {   "exam" : 5,    "a" : 9,    "_id" : {    "$oid" : "5148f3a5cc93742c1603538a"    }   },
  {   "exam" : 5,    "a" : 6,    "_id" : {    "$oid" : "5148f3a5cc93742c1603538b"    }   },
  {   "exam" : 5,    "a" : 8,    "_id" : {    "$oid" : "5148f3a5cc93742c1603538c"    }   }
]
```

```
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" : {    "$oid" : "5148f3a5cc93742c16035389"    }   }
]
> db.scores.find({a: {'$gt': 15}});


[
  {   "a" : 99,    "_id" : {    "$oid" : "5148f352cc93742c16035383"    }   }
]

>
```