

# MongoDB 数据库

非关系型数据库；数据库存储方式为：文档，数值为：键值对形式

mongodb文档类似于json对象，字段的值可以包含其他的文档，数组，以及数组文档

## MongoDB概念解析

SQL术语/概念	MongoDB术语/概念	解释说明
database	database	数据库
table	collection	表/集合
row	document	行/文档
column	field	数据字段/域
index	index	索引
table joins		表关联/MongoDB不支持
primary key	primary key	主键索引/mongodb自动将_id设置为主键索引

## 一、MongoDB的服务启动，创建数据库

### 1、服务开启

- (1)找到你的MongoDB的bin目录
- (2)在终端输入：cd MongoDB的bin目录
- (3)mongod.exe --dbpath=你要选择创建数据库的路径（如果路径里面有空格，要用"" 引起来）

### 2、进入MongoDB数据库

- (1)cd MongoDB的bin目录
- (2)mongo.exe
- (3)show dbs

## 二、对于库的操作database

### 1、创建库

use 库名

注意：

- （1）当use一个不存在的库时，其实这个库已经被创建，但是没有数据，用show dbs 不会显示此库

#用use 库名，库名不存在时，该库已经被创建，但是没有数据，用show dbs也不会显示此库

```
> show dbs
admin  0.000GB
local  0.000GB
test   0.000GB
> use python
switched to db python
> show dbs
admin  0.000GB
local  0.000GB
test   0.000GB
```

#只有使用该库，且插入数据后，使用show dbs才会显示该库

```
> use python
switched to db python
> db.createCollection('goods')
{ "ok" : 1 }
> show dbs
admin  0.000GB
local  0.000GB
python 0.000GB
test   0.000GB
```

(2) db.getName() 获取当前所在的库

```
> use zm
switched to db zm
> show dbs
admin  0.000GB
local  0.000GB
python 0.000GB
test   0.000GB
> db.getName()
zm
```

(3) mongodb严格区分大小写

```
> db.getName()
python
> db.getname()
2017-11-22T17:30:52.926+0800 E QUERY [thread1] TypeError: db.getname is not a function :
@(shell):1:1
```

(4) 在MongoDB里面，不管什么操作，都是db。。。

## 2、删除数据库

(1) 删除数据库之前最好use一下，确定自己在哪个库下

(2) db.dropDatabase()

```
> show dbs
admin    0.000GB
local    0.000GB
python   0.000GB
> use python
switched to db python
> db.getName()
python
> db.dropDatabase()
{ "dropped" : "python", "ok" : 1 }
> show dbs
admin    0.000GB
local    0.000GB
> db.dropDatabase()
{ "dropped" : "test", "ok" : 1 }
```

### 3、查看所有的库

```
show dbs
```

## 三、对于集合的操作collection

---

### 1、创建集合（也就是创建表）

```
db.createCollection('集合名称')
```

如：db.createCollection('user') #在当前库里面创建一个user集合

### 2、删除集合

```
db.集合名.drop()
```

如：db.user.drop() #删除user集合

### 3、显示所有集合

```
show collections
```

```
> db.createCollection('user')
{ "ok" : 1 }
> show collections
user
> db.user.drop()
true
> show collections
```

## 四、INSERT/SAVE 文档的插入

---

### 1、insert的插入一条文档

```
db.集合名.insert(文档)
```

```
> db.user.insert({name:"李四",hobby:"写代码"})
WriteResult({ "nInserted" : 1 })
```

## 2、insert 插入多条文档

db.集合名.insert([文档1,[文档2]])

注意：如果没有中括号，那么只会把第一条数据插入成功

```
> db.user.find()
{ "_id" : ObjectId("5a154692fd3ebfeb6bda1c1f"), "name" : "张三", "age" : 18 }
> db.user.insert({name:"王五"},{name:"赵六"})
WriteResult({ "nInserted" : 1 })
> db.user.find()
{ "_id" : ObjectId("5a154692fd3ebfeb6bda1c1f"), "name" : "张三", "age" : 18 }
{ "_id" : ObjectId("5a1547d9fd3ebfeb6bda1c23"), "name" : "王五" }
```

## 3、save（更多的是作为修改，但是可以作为文档的插入）

db.集合名.save(文档): 插入一条文档

如：db.集合名.save({"\_id":ObjectId("5a152a67fa08a5e7ad2ad092"),zzz:'vvv'}) #将id为92的文档进行覆盖

```
#用save插入一条文档
> db.user.save({xxx:"zzz"})
WriteResult({ "nInserted" : 1 })
> db.user.find()
{ "_id" : ObjectId("5a154915fd3ebfeb6bda1c24"), "xxx" : "zzz" }

#将原来存在的id重新赋值，原有值被覆盖掉
> db.user.save({"_id":ObjectId("5a154915fd3ebfeb6bda1c24"),xxx:"vvv"})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.user.find()
{ "_id" : ObjectId("5a154915fd3ebfeb6bda1c24"), "xxx" : "vvv" }
```

## 4、3.2版本以后，建议使用

db.collection.insertOne()

```
> db.user.insertOne({name:"张三"})
{
  "acknowledged" : true,
  "insertedId" : ObjectId("5a1556e3fd3ebfeb6bda1c26")
}
> db.user.find()
{ "_id" : ObjectId("5a1556e3fd3ebfeb6bda1c26"), "name" : "张三" }
```

db.collection.insertMany()

```
> db.user.insertMany([{name:"李四"},{name:"王五"}])
{
  "acknowledged" : true,
  "insertedIds" : [
    ObjectId("5a15575bfd3ebfeb6bda1c27"),
    ObjectId("5a15575bfd3ebfeb6bda1c28")
  ]
}
> db.user.find()
{ "_id" : ObjectId("5a1556e3fd3ebfeb6bda1c26"), "name" : "张三" }
{ "_id" : ObjectId("5a15575bfd3ebfeb6bda1c27"), "name" : "李四" }
{ "_id" : ObjectId("5a15575bfd3ebfeb6bda1c28"), "name" : "王五" }
```

## 五、文档的删除remove

---

### 1、只删除一条数据

`db.collection.remove(条件,1)`

如：`db.user.remove({name:"张三"},1)` #删除第一个匹配到的文档

### 2、删除多条数据

`db.collection.remove(条件)`

如：`db.user.remove({name:"张三"})` #删除多条文档

### 3、全部删除

`db.collection.remove({})` #删除当前集合的全部文档

### 4、3.2版本以后，建议使用

`db.collection.deleteOne()` #删除一条文档

`db.collection.deleteMany()` #删除多条文档