

# 1 Mongo数据库的操作

## 1.1 查看数据库

列出所有在物理上存在的数据库

```
1 show dbs
```

## 1.2 切换数据库/创建数据库

如果数据库不存在，则指向数据库，但不创建，直到插入数据或创建集合时数据库才被创建

```
1 use 数据库名
```

## 1.3 删除数据库

删除当前指向的数据库 如果数据库不存在，则什么也不做

```
1 use 数据库名
2 db.dropDatabase()
```

# 2 集合的操作

## 2.1 创建集合

无需手动创建集合：向不存在的集合中第一次添加数据时，集合会自动被创建出来

```
1 db.createCollection(集合名,{capped:true,size:num})
```

- capped：默认值为false表示不设置上限，值为true表示设置上限
- size：当capped值为true时，需要指定此参数，表示上限大小，当文档达到上限时，会将

之前的数据覆盖，单位为字节

## 2.2 删除集合

```
1 db.集合名称.drop()
```

## 2.3 查看集合

```
1 show tables
2 show collections
```

# 3 数据的操作

## 3.1 数据的增加

注意：插入文档时，如果不指定\_id参数，MongoDB会为文档分配一个唯一的ObjectId

### 3.1.1 insert

```
1 db.集合名称.insert(document)
```

### 3.1.2 save

使用save函数里，如果原来的对象不存在，那他们都可以向collection里插入数据，如果已经存在，save会调用update更新里面的记录

```
1 db.集合名称.save(document)
```

### 3.1.3 举例

已存在数据： `{_id : 'abc123', " name " : " 小王 " }`

再次进行插入操作时 `insert({_id : 'abc123', " name " : " 小李 " })` 会报主键重复的错误提示

`save({ _id : 'abc123', " name " : " 小李 " })` 会把 小王 修改为 小李

如果集合中不存在 `_id : 'abc123'`,

`insert({_id : 'abc123', " name " : " 小李 " })` 增加一条数据

`save({ _id : 'abc123', " name " : " 小李 " })` 增加一条数据

### 3.1.4 增加多条

```
1 db.集合名.insert([{k1:v1},{k2:v2}])
```

## 3.2 数据的修改

### 3.2.1 格式

```
1 db.集合名称.update(  
2     <query>,  
3     <update>,  
4     {multi: <boolean>}  
5 )
```

- 参数query:查询条件，类似sql语句update中where部分
- 参数update:更新操作符，类似sql语句update中set部分
- 参数multi:可选，默认是false，表示只更新找到的第一条记录，值为true表示把满足条件的文档全部更新

### 1.2.2 举例

```
1 # 只更新找到的第一条，并且会修改结构
2 db.person.update({name:"zs"},{age:16})
3 # 只更新数据，为不更新文档结构
4 db.person.update({name:"zs"},{$set:{age:123}})
5 # 更新所有找到匹配的数据
6 db.person.update({name:"zs"},{$set:{age:123}}, {multi: true})
```

### 1.3 数据的删除

```
1 # 删除所有匹配数据
2 db.person.remove({name:"zs"})
3 # 只更新一条
4 db.person.remove({name:"zss"},{justOne:true})
```

### 1.4 数据的查找

```
1 db.集合名.find()
```