**一、MongoDB简介及shell操作**

MongoDB是一个面向文档(document-oriented)的数据库，不是关系型数据库。

与关系型数据库相比，面向文档的数据库没有"行"的概念，取而代之的是"文档"模型。

基本概念：

文档：MongoDB中数据的基本单元，类似于关系型数据库中的行。

集合：集合就是一组文档，一个集合相当于一张表。

MongoDB的一个实例可以拥有多个相互独立的数据库，每个数据库都拥有自己的集合。

1 启动MongoDB

在命令提示符中输入 mongod命令

MongoDB作为网络服务来运行，客户端可连接到该服务器并执行操作。

2 MongoDB shell

MongoDB自带JavaScript shell,可在shell中使用命令行与MongoDB交互。

shell是一个功能完备的JavaScript解释器，可运行任意JavaScript程序。

运行shell：

在命令提示符中输入mongo命令启动shell,shell会自动连接到MongoDB服务器。

>x = 200

200

>x / 5

40

可充分利用JavaScript的标准库

>Math.sin(Math.PI / 2)

1

>"Hello, World!".replace("World", "MongoDB")

Hello, MongoDB!

可定义并调用JavaScript函数

>function factorial(n){

...if(n <= 1) return

...return n \* factoial(n - 1)

...}

>factoial(5)

120

shell是一个独立的MongoDB客户端，启动时，shell会连接到MongoDB服务器的test数据库，并将数据库连接赋值给全局变量db

这个变量是通过shell访问MongoDB的主要入口点

如果要查看db当前指向哪个数据库，可以使用db命令：

>db

test

选择其它数据库：

>use demo

switched to db demo

>db

demo

**二、shell中的基本操作**

在shell中查看或操作数据会用到4个基本操作：创建、读取、更新和删除(即CRUD操作)

1 创建

创建一个名为post的局部变量，它是一个JavaScript对象，有三个键："title" "content" "date"

>post = {"title" : "我的备忘录",

..."content" : "下午3点有会。",

..."date" : new Date()}

{

"title" : "我的备忘录",

"content" : "下午3点有会，谢谢合作",

"date" : ISODate("2016-09-18T13:51:53.003Z")

}

我们可以用insert方法将其保存到blog集合中

>db.blog.insert(post)

用以上这条命令可将post保存到数据库中，查看它可调用find方法

>db.blog.find()

2 读取

find和findOne方法可以用于查询集合里的文档，如果只想查看一个文档，可用findOne

>db.blog.findOne()

3 更新

使用update修改备忘录内容。

update接受(至少)两个参数：

第一个是限定条件

第二个是新的文档

例如：要为上面的备忘录添加作者，就需要增加一个新的键，用于保存作者。

首先，修改变量past，增加"author"键：

>past.author = []

[]

然后执行update操作，增加作者的名字。

>db.blog.update({title : "我的备忘录"}, past)

现在，文档已经有了"author"键，再用find()查看一下，可以看到新的键被添加了：

>db.blog.find()

4 删除

使用remove方法可将文档从数据库中永久删除。

如果没有使用任何参数，会将集合内的所有文档全部删除。

也可能使用参数来指定要删除的文档。例如：

>db.blog.remove({title : "我的备忘录"})

以上这条命令会把刚刚创建的文档删除。