



MongoDB

讲师：李立超

数据库 (Database)

- 数据库是按照数据结构来组织、存储和管理数据的仓库。
- 我们的程序都是在内存中运行的，一旦程序运行结束或者计算机断电，程序运行中的数据都会丢失。
- 所以我们就需要将一些程序运行的数据持久化到硬盘之中，以确保数据的安全性。而数据库就是数据持久化的最佳选择。
- 说白了，数据库就是存储数据的仓库。

数据库分类

- 数据库主要分成两种：
 - 关系型数据库
 - MySQL、Oracle、DB2、SQL Server
 - 关系数据库中全都是表
 - 非关系型数据库
 - MongoDB、Redis
 - 键值对数据库
 - 文档数据库 MongoDB

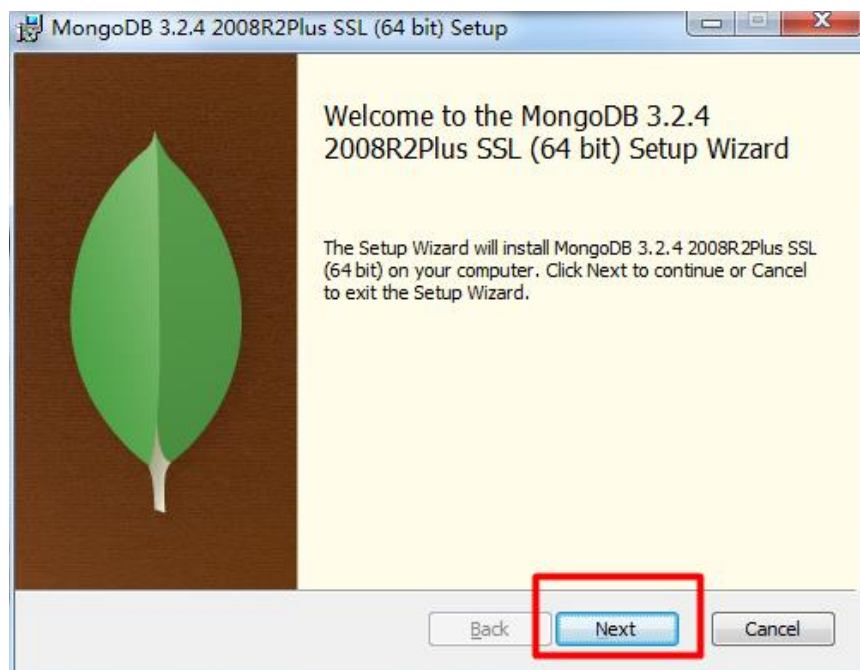
MongoDB简介

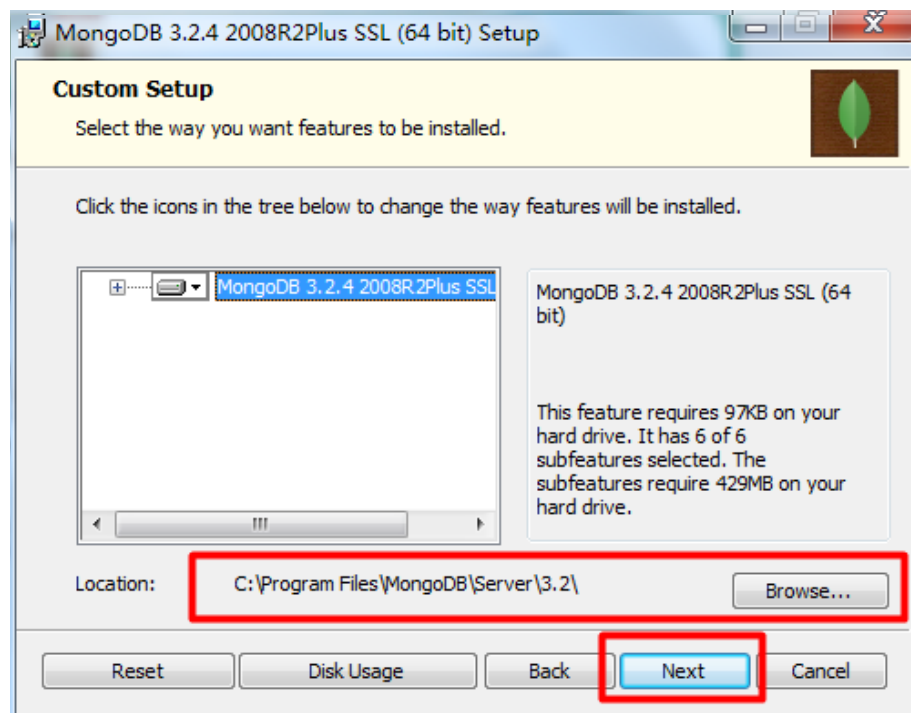
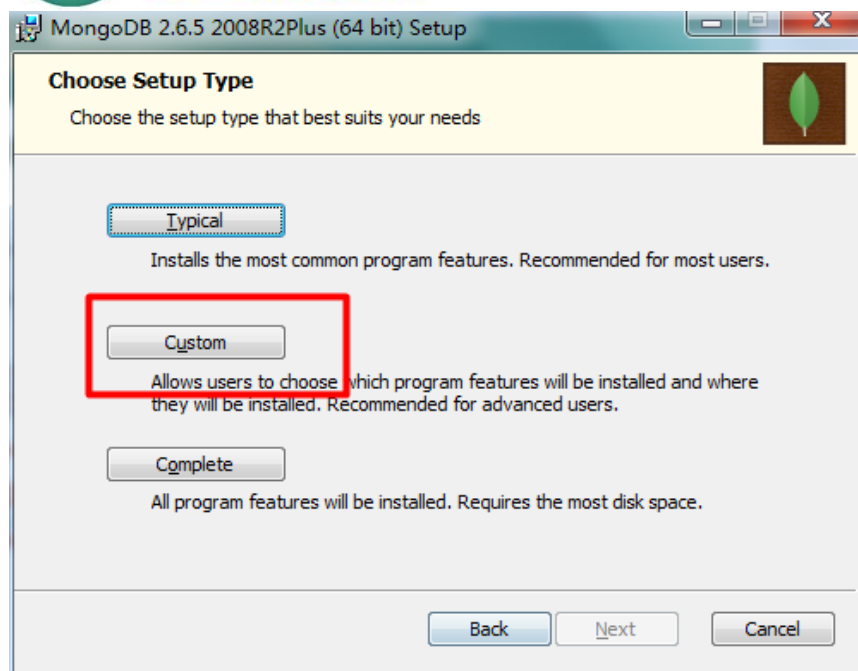
- MongoDB是为快速开发互联网Web应用而设计的数据库系统。
- MongoDB的设计目标是极简、灵活、作为Web应用栈的一部分。
- MongoDB的数据模型是面向文档的，所谓文档是一种类似于JSON的结构，简单理解MongoDB这个数据库中存的是各种各样的JSON。（ BSON ）

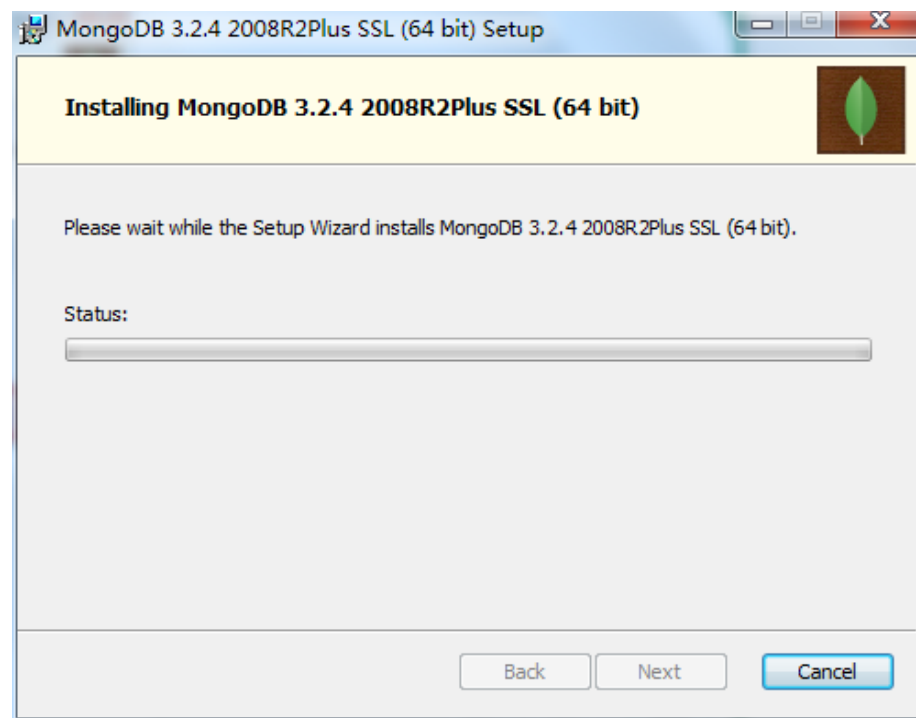
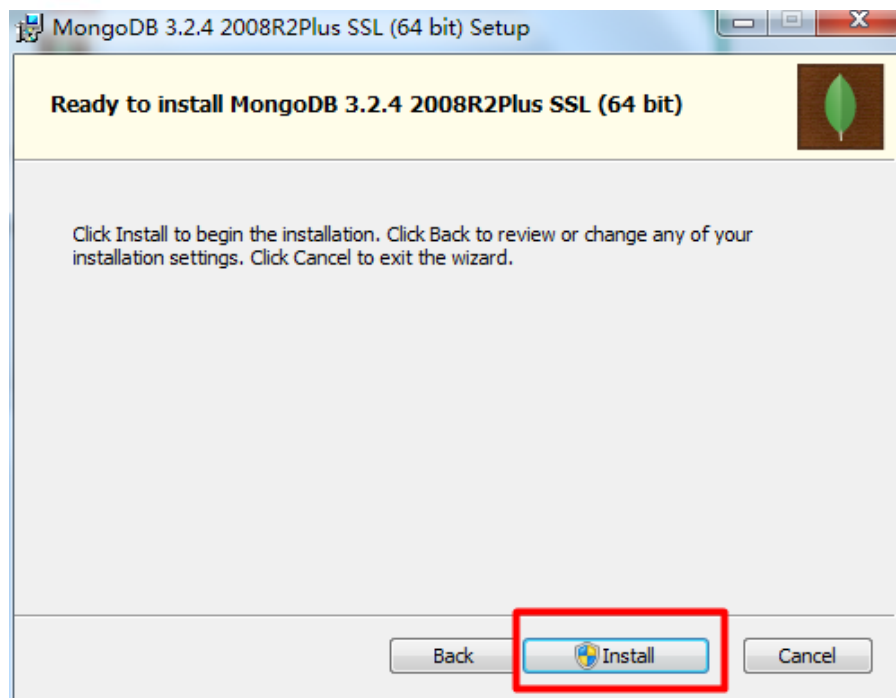
下载MongoDB

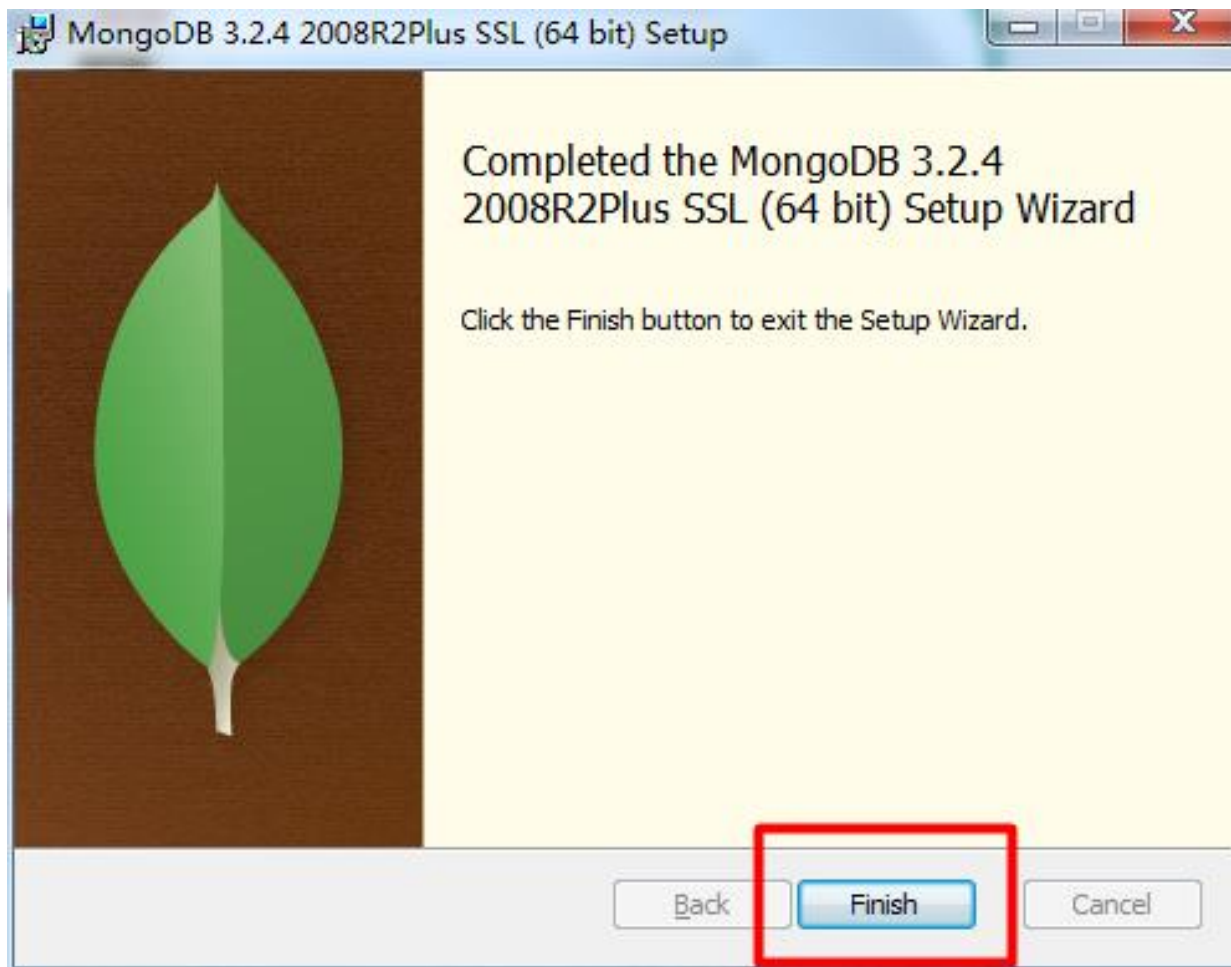
- 下载地址
 - <https://www.mongodb.org/dl/win32/>
- MongoDB的版本偶数版本为稳定版，奇数版本为开发版。
- MongoDB对于32位系统支持不佳，所以3.2版本以后没有再对32位系统的支持。

MongoDB安装



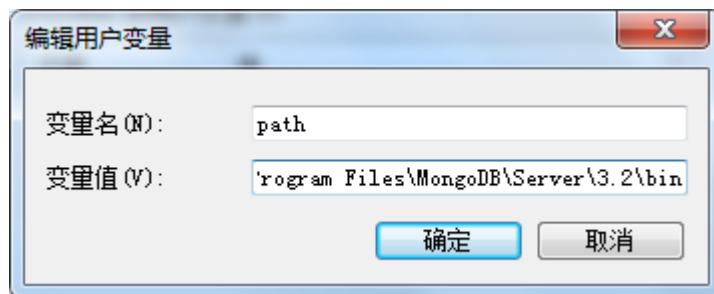
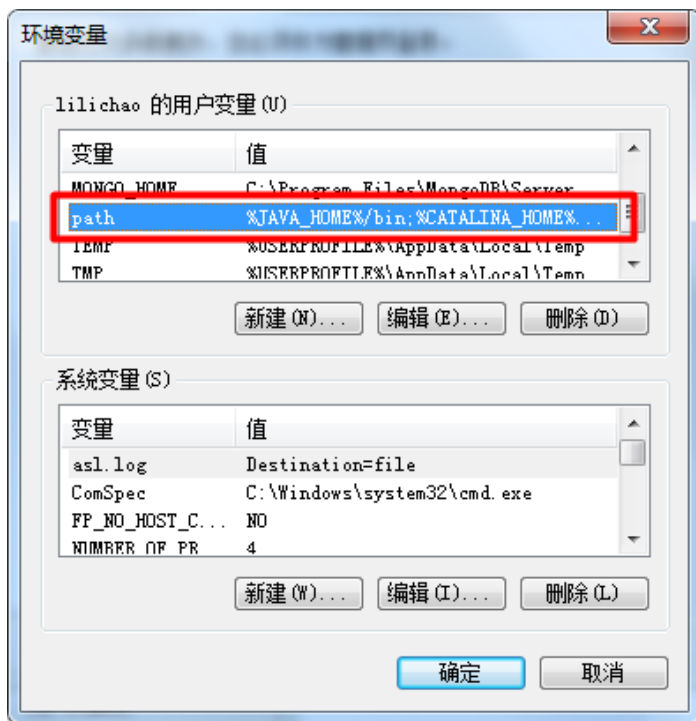






启动MongoDB

- 将MongoDB的bin目录添加到path下



启动MongoDB

- 在C盘根目录下创建data文件夹，在data下创建db文件夹
- 打开CMD命令行窗口，输入mongod

```
C:\Users\lilichao>mongod
```

- 32位系统第一次启动：
 - mongod --storageEngine=mmapv1
- 出现如下输出

```
2017-07-05T11:52:16.032+0800 I FTDC [initandlisten] Initializing full time diagnostic data capture w
2017-07-05T11:52:16.034+0800 I NETWORK [initandlisten] waiting for connections on port 27017
```

指定端口和路径

- 在控制台启动MongoDB
 - mongod --dbpath 路径 --port 端口号
 - 示例：
 - mongod --dbpath C:\Users\lilichao\Desktop\mongo\data\db --port 123
- 注意：打开的命令行窗口不能关闭

配置mongo的windows服务

- 在mongo安装目录的server下3.x目录下创建mongod.cfg文件，并添加如下内容

systemLog:

destination: file

path: c:\data\log\mongod.log

storage:

dbPath: c:\data\db

- 管理员模式打开控制台，并输入如下指令

```
sc.exe create MongoDB binPath= "\"mongo bin路径\mongod.exe\"" --service --  
config= "\"mongo路径\mongod.cfg\"" DisplayName= "MongoDB" start= "auto"
```

- 删除服务

```
sc delete MongoDB
```

关闭MongoDB

- 打开新的命令行窗口
- 登录服务器
 - mongo
- 切换管理员用户
 - admin
- 关闭数据库
 - db.shutdownServer()

参数	说明
--help -h	返回基本帮助和用法文本
--version	返回MongoDB的版本
--config<文件名> -f<文件名>	指定包含运行时配置的配置文件
--verbose -v	增加发送到控制台日志的数量
--quiet	减少发送到控制台日志的数量
--port<端口>	指定mongod的端口，默认27017
--bind_ip<端口>	指定ip地址
--maxConns<编号>	指定链接的最大数
--logpath<路径>	指定日志文件的路径
--auth	启用远程主机的身份验证
--dbpath<路径>	指定数据库实例的路径
--nohttpinterface	禁用HTTP接口
--nojournal	禁用日志
--noprealloc	禁止预分配数据文件
--repair	在所有数据库上运行修复程序

Mongo Shell

- 登录mongo shell
 - mongo
- 命令
 - help <option> 语法帮助
 - use <database> 更改当前操作的数据库
 - show <option> 根据参数显示列表
 - dbs 显示数据库列表
 - collections 显示当前数据库的集合
 - profile 显示时间超过1毫秒的system.profile条目
 - log[name] 显示登录记忆的最后一段
 - exit 退出数据库
 - load(script) 加载js文件db.auth(username , password)
在当前数据库做身份验证

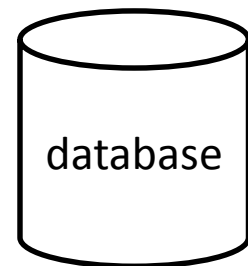
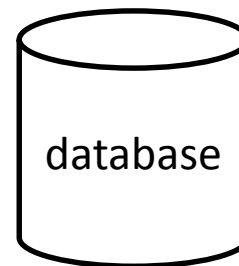
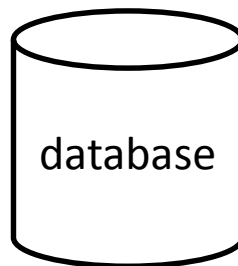
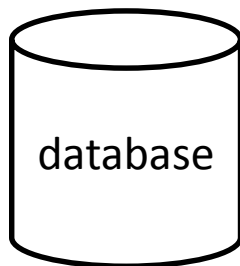
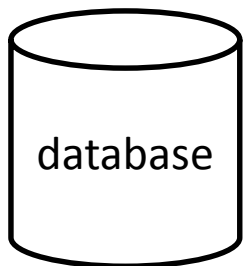
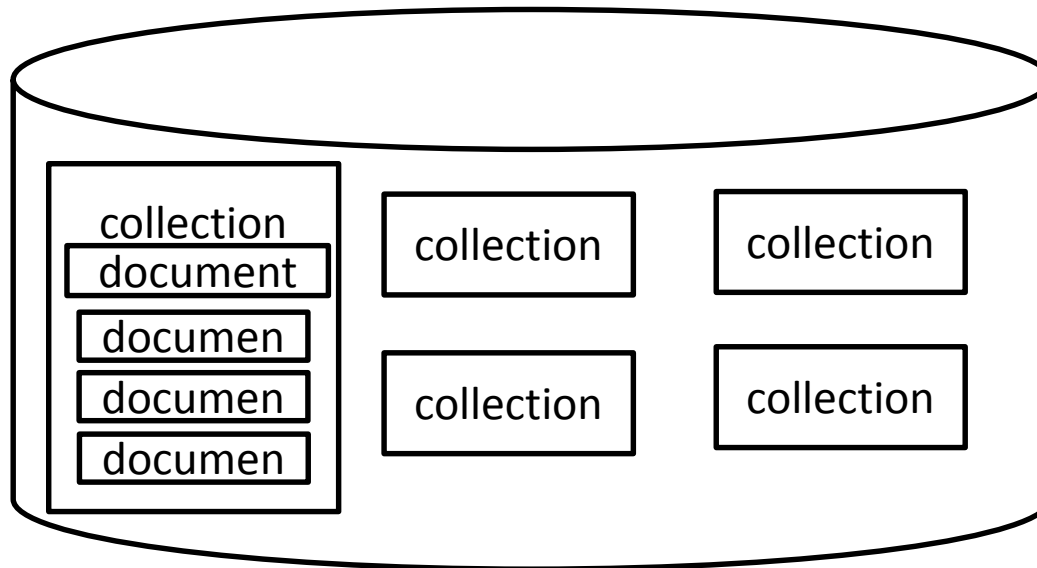
三个概念

- 数据库 (database)
 - 数据库是一个仓库，在仓库中可以存放集合。
- 集合 (collection)
 - 集合类似于数组，在集合中可以存放文档。
- 文档 (document)
 - 文档数据库中的最小单位，我们存储和操作的内容都是文档。

基本概念

- 文档 (document)
 - 类似于JS中的对象，在MongoDB中每一条数据都是一个文档
- 集合 (collection)
 - 集合就是一组文档，也就是集合是用来存放文档的
 - 集合中存储的文档可以是各种各样的，没有格式要求
- 多个文档组成集合，多个集合组成数据库

mongoDB



创建数据库

- use 数据库名
 - 使用use时，如果数据库存在则会进入到相应的数据库，如果不存在则会自动创建
 - 一旦进入数据库，则可以使用db来引用当前库
- db.collection.insert(文档)
 - 向集合中插入文档，如果集合不存在则创建
- db.createCollection()
 - 创建一个新的集合
- db.collection.drop()
 - 删除集合

文档的增删改查

- 插入文档
 - `db.collection.insert ()`
- 查询文档
 - `db.collection.find ()`
- 删除文档
 - `db.collection.remove()`
- 修改文档
 - `db.collection.update()`

添加文档

- `db.collection.insert` (文档对象)
 - `insert()`可以用于向集合中添加一个或多个文档，可以传递一个对象，或一个数组。
 - 可以将对象或数组中的对象添加进集合中
 - 添加时如果集合或数据库不存在，会自动创建
 - 插入的文档对象会默认添加`_id`属性，这个属性对应一个唯一的id，是文档的唯一标识

删除文档

- `db.collection.remove()`
 - `remove()`可以用来移除指定文档对象
 - 方法接收一个查询文档作为参数，只有符合条件的文档才会被删除
 - 删除数据是永久的，不能撤销
- `db.collection.drop()`
 - 删除集合

修改文档

- `db.collection.update()`
- 替换文档
 - 可以在`update()`中传递两个参数，一个是查询文档，一个是新的文档，这样符合条件的文档将会被新文档所替换
 - `update()`的第三个参数，用来指定是否使用`upsert`，默认为`false`
 - `update()`的第四个参数，用来指定是否同时修改多个文档，默认为`false`

修改器

- 使用update会将整个文档替换，但是大部分情况下我们是不需要这么做的
- 如果只需要对文档中的一部分进行更新时，可以使用更新修改器来进行。
- 我们将要学习以下几个修改器
 - \$set、 \$unset、 \$inc、 \$push、 \$addToSet

\$set、\$unset

- \$set用来指定一个字段的值，如果这个字段不存在，则创建它。
- 语法：
 - `db.test_coll.update(查询对象, {$set:更新对象});`
- \$unset可以用来删除文档中一个不需要的字段，用法和set类似。

\$inc

- \$inc用来增加已有键的值，或者该键不存在那就创建一个
- \$inc只能用于Number类型的值

查询文档

- find()、findOne()
 - MongoDB使用find()来对文档进行查询
 - find()需要一个查询文档作为参数，如果不传该参数，则会返回集合中的所有元素。
 - 可以将查询条件以键值对的形式添加到查询文档中
 - 查询条件
 - \$lt、\$lte、\$gt、\$gte、\$ne、\$or、\$in、\$nin、\$not、\$exists、\$and

