# MongoDB limit与skip方法

【<https://www.runoob.com/mongodb/mongodb-limit-skip.html>】

**前言：**

如果你需要在MongoDB中读取指定数量的数据记录，可以使用MongoDB的limit方法，limit()方法接收一个数字参数，该参数指定从MongoDB中读取的记录条数。

## 一、limit()方法：

limit()方法基本语法如下所示：

|  |
| --- |
| >db.COLLECTION\_NAME.find().limit(NUMBER) |

【查询文档中的两条记录】

|  |
| --- |
| > db.col.find()  { "\_id" : ObjectId("5ee4f5ed43833a9e2bc40a3f"), "title" : "PHP 教程", "description" : "PHP 是一种创建动态交互性站点的强 有力的服务器端脚本语言。", "by" : "菜鸟教程", "url" : "http://www.runoob.com", "tags" : [ "php" ], "likes" : 200 }  { "\_id" : ObjectId("5ee4f61943833a9e2bc40a40"), "title" : "Java 教程", "description" : "Java 是由Sun Microsystems公司于1995年5月推出的高级程序设计语言。", "by" : "菜鸟教程", "url" : "http://www.runoob.com", "tags" : [ "java" ], "likes" : 150 }  { "\_id" : ObjectId("5ee4f6ff43833a9e2bc40a42"), "title" : 123, "description" : "helloworld", "by" : "chenming", "url" : "www.chenming.com" }  > db.col.find({},{\_id:1,title:1}).limit(2)  { "\_id" : ObjectId("5ee4f5ed43833a9e2bc40a3f"), "title" : "PHP 教程" }  { "\_id" : ObjectId("5ee4f61943833a9e2bc40a40"), "title" : "Java 教程" } |

## skip()方法：

我们除了可以使用limit()方法来读取指定数量的数据外，还可以使用skip()方法来跳过指定数量的数据，skip()方法同样接受一个数字参数作为跳过的记录条数。

skip()方法语法格式如下：

|  |
| --- |
| >db.COLLECTION\_NAME.find().limit(NUMBER).skip(NUMBER) |

以下实例只会显示第二条文档数据

|  |
| --- |
| > db.col.find({},{\_id:1,title:1}).limit(2)  { "\_id" : ObjectId("5ee4f5ed43833a9e2bc40a3f"), "title" : "PHP 教程" }  { "\_id" : ObjectId("5ee4f61943833a9e2bc40a40"), "title" : "Java 教程" }  > db.col.find()  { "\_id" : ObjectId("5ee4f5ed43833a9e2bc40a3f"), "title" : "PHP 教程", "description" : "PHP 是一种创建动态交互性站点的强 有力的服务器端脚本语言。", "by" : "菜鸟教程", "url" : "http://www.runoob.com", "tags" : [ "php" ], "likes" : 200 }  { "\_id" : ObjectId("5ee4f61943833a9e2bc40a40"), "title" : "Java 教程", "description" : "Java 是由Sun Microsystems公司于1995年5月推出的高级程序设计语言。", "by" : "菜鸟教程", "url" : "http://www.runoob.com", "tags" : [ "java" ], "likes" : 150 }  { "\_id" : ObjectId("5ee4f6ff43833a9e2bc40a42"), "title" : 123, "description" : "helloworld", "by" : "chenming", "url" : "www.chenming.com" }  > db.col.find({},{\_id:1,title:1}).limit(1).skip(1);  { "\_id" : ObjectId("5ee4f61943833a9e2bc40a40"), "title" : "Java 教程" } |

## 注意点：

skip()和limit()方法只适合小数据量分页，如果是百万级效率就会非常低，因为skip()方法是一条条数据数过去的，建议使用where\_limit。

如何避免使用skip呢？首先来回顾SQL分页的后一种时间戳分页方案，这种利用字段的有序性质，利用查询来取数据的方式，可以直接避免掉了大量的数数。也就是说，如果能附带上这样的查询条件那查询效率就会提高，事实上是这样的么？我们来验证一下：

这里我们假设查询100001条数据，这条数据的Amount值是：2399927，我们来写两条语句分别如下：

|  |
| --- |
| db.test.sort({"amount":1}).skip(100000).limit(10) //183ms  db.test.find({amount:{$gt:2399927}}).sort({"amount":1}).limit(10) //53ms |

很明显后者的性能是前者的三分之一，差距是非常大的，也印证了skip()效率差的理论。