

day05-课堂笔记

day05-课堂笔记

1. 爬虫中验证码的识别

1.1 验证码的介绍

1.2 百度AI开发平台

1.2 使用超级鹰破解验证码

2. MongoDB数据库

2.1 MongoDB数据库的介绍

2.2 MongoDB数据库环境搭建

2.3 MongoDB数据库的基本使用

数据库的命令：

集合的命令

2.4 数据操作- 插入（新增数据）

2.5 查询数据

1. 爬虫中验证码的识别

1.1 验证码的介绍

验证码：captcha，全程：全自动的区分人类和机器的图灵测试。可以防止：恶意破解密码、[刷票](#)、论坛灌水，有效防止某个黑客对某一个特定注册用户用特定程序暴力破解方式进行不断的登陆尝试。

爬虫其实就是去访问网址，假设，某一个网址中需要登录，登录的时候需要输入图形验证码的。在爬虫中想要去实现这个登录的功能也是需要去输入验证码。

第三方平台可以去识别验证码

1.2 百度AI开发平台

网址：<http://ai.baidu.com/?track=cp:ainsem|pf:pc|pp:tongyong-kaifangpingtai|pu:kaifangpingtai|ci:|kw:10003799>

想要去使用这个平台需要先去注册成为开发者。

选择 文字识别：<https://ai.baidu.com/tech/ocr/general>



文档：<https://ai.baidu.com/ai-doc/OCR/1k3h7y3db>

获取client_id, client_secret, 进入到个人中心



应用

用量

已建应用: 3 个

管理应用

创建应用

| API | 调用量 |
|-----------------|-----|
| 通用文字识别 (标准版) | 0 |
| 通用文字识别 (标准含位置版) | 0 |
| 通用文字识别 (含生僻字版) | 0 |
| 通用文字识别 (高精度版) | 0 |

创建应用，之后查看详情

| | | | | |
|------------|----------|--------------------------|------------|-------------|
| 应用名称 | AppID | API Key | Secret Key | 包名 |
| spider_spy | 24795179 | 1Nh79RWVgIeAK0FIg07Gf-d9 | ***** 显示 | 文字识别 不需要 |

1.2 使用超级鹰破解验证码

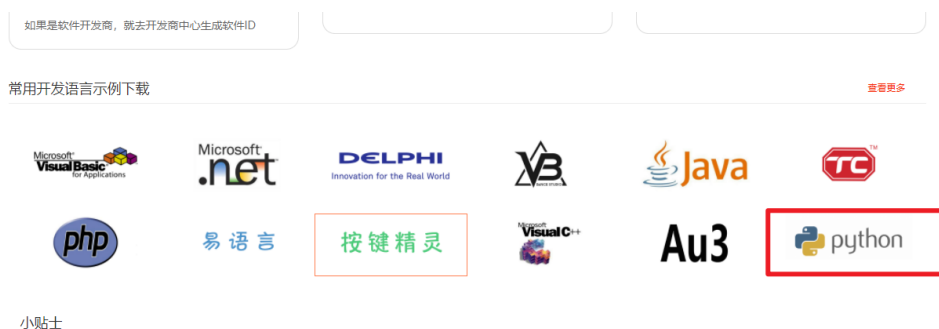
第三方的验证码的识别平台，<http://www.chaojiying.com/>，打开网址登陆到这个网址

- 1 账号: wenbin123
- 2 密码: wenbin123

查看开发文档



选择自己的使用的编程语言 (Python)



下载 超级鹰已经写好的代码



下载完毕之后，得到一个压缩包，将压缩包进行解压。

官方提供的代码

```
1 #!/usr/bin/env python
2 # coding:utf-8
3
4 import requests
```

```

5  from hashlib import md5
6
7
8  class Chaojiying_Client(object):
9
10     def __init__(self, username, password, soft_id):
11         self.username = username
12         password = password.encode('utf8')
13         self.password = md5(password).hexdigest()
14         self.soft_id = soft_id
15         self.base_params = {
16             'user': self.username,
17             'pass2': self.password,
18             'softid': self.soft_id,
19         }
20         self.headers = {
21             'Connection': 'Keep-Alive',
22             'User-Agent': 'Mozilla/4.0 (compatible; MSIE 8.0; Windows NT
3         5.1; Trident/4.0)',
23         }
24
25     def PostPic(self, im, codetype):
26         """
27         im: 图片字节
28         codetype: 题目类型 参考 http://www.chaojiying.com/price.html
29         """
30         params = {
31             'codetype': codetype,
32         }
33         params.update(self.base_params)
34         files = {'userfile': ('ccc.jpg', im)}
35         r =
36         requests.post('http://upload.chaojiying.net/Upload/Processing.php',
37             data=params, files=files,
38             headers=self.headers)
39         return r.json()
40
41     def ReportError(self, im_id):
42         """
43         im_id: 报错题目的图片ID
44         """
45         params = {
46             'id': im_id,
47         }
48         params.update(self.base_params)
49         r =
50         requests.post('http://upload.chaojiying.net/Upload/ReportError.php',
51             data=params, headers=self.headers)
52         return r.json()
53
54 if __name__ == '__main__':
55     chaojiying = Chaojiying_Client('超级鹰用户名', '超级鹰用户名的密码', '96001')
56     # 用户中心>>软件ID 生成一个替换 96001
57     im = open('a.jpg', 'rb').read() # 本地图片文件路径 来替换 a.jpg 有时WIN系统
58     须要//
59     print(chaojiying.PostPic(im, 1902)) # 1902 验证码类型 官方网站>>价格体系
60     3.4+版 print 后要加())

```

```
if __name__ == '__main__':
    chaojiying = Chaojiying_Client('wenbin123', 'wenbin123', '96001') # 用户名 用户名 软件ID, 是需要我们去创建一个软件之后才有id
    # 用户中心>>软件ID 生成一个替换 96001
```

创建软件ID

超级鹰首页 > 用户中心 > 软件ID

生成一个软件ID

软件ID列表

| 软件名称 | 软件ID | 软件KEY | 状态 |
|------------|--------|----------------------------------|----|
| python41_2 | 920659 | 69bc4f5af1b7e823c0a4d9d8e2317056 | 编辑 |
| spider_41 | 920657 | 0e5119662e705bc8da73f4c4a2389fdb | 编辑 |
| dasdassd | 920215 | f1cb86d8a375a127439df4cb9e650e20 | 编辑 |
| admin | 920214 | 21ceea3036f9f8197d545adb6042b78c | 编辑 |
| sadasd | 920212 | deec08937f5cb6a6f519d714b4a03822 | 编辑 |
| aaa | 920210 | 20875a238ce9c4af0673b96dd5114feb | 编辑 |
| sdad | 920209 | 2d468cba918465502cd3f8e7097e13af | 编辑 |
| chaojiyou | 920207 | cd11c7eef857d6943b3f26b8a00295a4 | 编辑 |
| fsdfsdf | 920193 | | 编辑 |
| chaoying | | | 编辑 |
| spider25 | | | 编辑 |
| asdsda | | | 编辑 |
| demo | | | 编辑 |
| 123456 | | | 编辑 |
| tiansuo | | | 编辑 |

超级鹰首页 > 用户中心 > 软件ID > 添加软件

软件名称

sgy_spider

软件KEY

b9e12d01dba5930ec844378567ca6ff2

软件说明

测试

提交

软件ID列表

| | | | | |
|-------|------------|--------|----------------------------------|------|
| 软件名称: | sgy_spider | 软件ID: | 922046 | 状态:0 |
| 软件说明: | 测试 | 软件KEY: | b9e12d01dba5930ec844378567ca6ff2 | 编辑 |

```
print(chaojiying.PostPic(im, 1902)) # 1902 验证码类型 官方网站>>价格体系 3.4+版 print 后要加())
```

验证码的类型

修改验证码的类型

超级鹰 专业验证码识别平台

应用案例 开发文档 价格体系 免费测试 关于我们 联系我们

首页 / 超级鹰价格体系

价格体系

英文数字

中文汉字

纯英文

纯数字

任意特殊字符

坐标选择计算等其他类型

标准价格: 1元=1000积分, 根据VIP级别和单次充值金额, 有不同的赠送, 低至五折

可支持长类型验证码, 仅按实际长度计分, 赠送的积分仅为上限部分。

查看充值优惠

| 验证码类型 | 验证码描述 | 官方单价(积分) |
|-------|------------|------------|
| 1902 | 常见4-6位英文数字 | 10, 12, 15 |
| 1101 | 1位英文数字 | 10 |
| 1004 | 1-4位英文数字 | 10 |
| 1005 | 1-5位英文数字 | 12 |
| 1006 | 1-6位英文数字 | 15 |
| 1007 | 1-7位英文数字 | 17.5 |
| 1008 | 1-8位英文数字 | 20 |
| 1009 | 1-9位英文数字 | 22.5 |
| 1010 | 1-10位英文数字 | 25 |
| 1012 | 1-12位英文数字 | 30 |
| 1020 | 1-20位英文数字 | 50 |

中文汉字

```

1  #!/usr/bin/env python
2  # coding:utf-8
3
4  import requests
5  from hashlib import md5
6
7
8  class Chaojiying_Client(object):
9
10     def __init__(self, username, password, soft_id):
11         self.username = username
12         password = password.encode('utf8')
13         self.password = md5(password).hexdigest()
14         self.soft_id = soft_id
15         self.base_params = {
16             'user': self.username,
17             'pass2': self.password,
18             'softid': self.soft_id,
19         }
20         self.headers = {
21             'Connection': 'Keep-Alive',
22             'User-Agent': 'Mozilla/4.0 (compatible; MSIE 8.0; Windows NT
3 5.1; Trident/4.0)',
23         }
24
25     def PostPic(self, im, codetype):
26         """
27         im: 图片字节
28         codetype: 题目类型 参考 http://www.chaojiying.com/price.html
29         """
30         params = {
31             'codetype': codetype,
32         }
33         params.update(self.base_params)
34         files = {'userfile': ('ccc.jpg', im)}
35         r =
36 requests.post('http://upload.chaojiying.net/Upload/Processing.php',
37 data=params, files=files,
38             headers=self.headers)
39         return r.json()
40
41     def ReportError(self, im_id):
42         """
43         im_id: 报错题目的图片ID
44         """
45         params = {
46             'id': im_id,
47         }
48         params.update(self.base_params)
49         r =
50 requests.post('http://upload.chaojiying.net/Upload/ReportError.php',
51 data=params, headers=self.headers)
52         return r.json()
53
54 if __name__ == '__main__':
55     chaojiying = Chaojiying_Client('wenbin123', 'wenbin123', '922046') # 用
56     户中心>>软件ID 生成一个替换 96001

```

```

53     im = open('ys.png', 'rb').read() # 本地图片文件路径 来替换 a.jpg 有时WIN系统
    须要//
54     print(chaojiying.PostPic(im, 9103)) # 1902 验证码类型 官方网站>>价格体系
    3.4+版 print 后要加())
55     """
56     验证码: 四位的 数字字母
57     计算 1+1=
58     点击文字: 按循序依次点击 xxxxx
59     12306: 选中下图中的白百何
60     """

```

2. MongoDB数据库

2.1 MongoDB数据库的介绍

MySQL数据库：关系型数据库。字段和字段之间，表与表之间是有关联关系，外键。 3306

学生表

| id | name | age | c_id |
|----|------|-----|------|
| 1 | 张三 | 18 | 2 |
| 2 | 李四 | 19 | 1 |

```
select * from stu inner join classes on c_id = classes.id;
```

班级表

| id | name |
|----|------|
| 1 | 1班 |
| 2 | 2班 |
| 3 | 3班 |

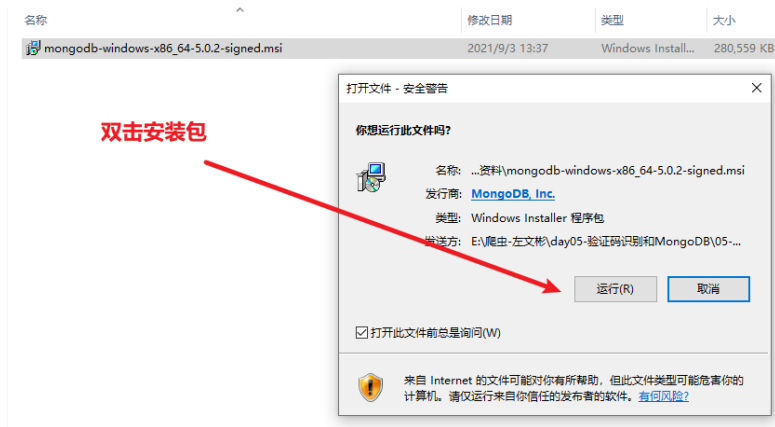
MongoDB数据库：非关系型数据库，字段和字段之间以及表与表之间是没有任何关系的，每条数据相互独立的。 27017

关系型数据库MySQL：数据库 表 字段数据

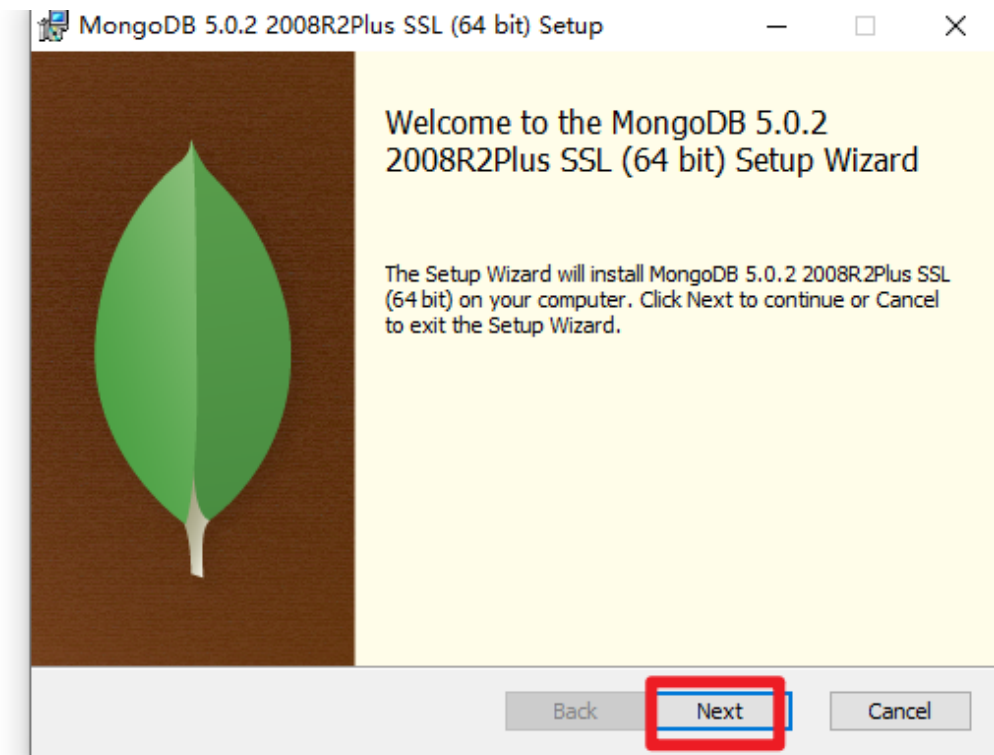
非关系型数据库MongoDB：数据库 集合 文档（一个文档就是一条数据）

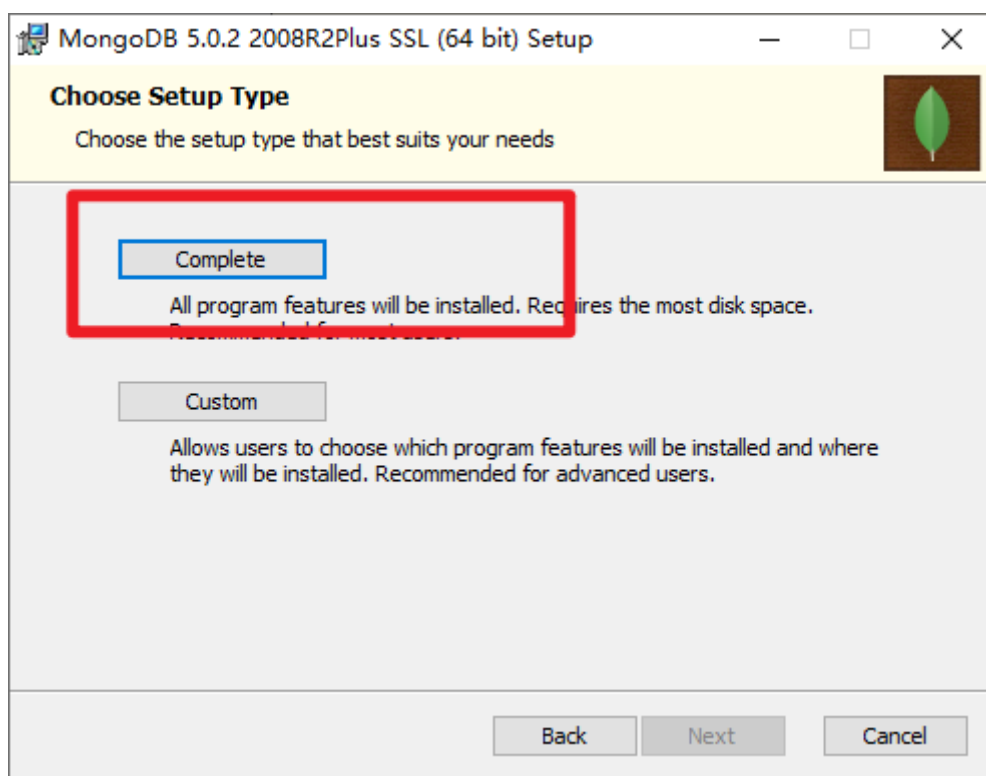
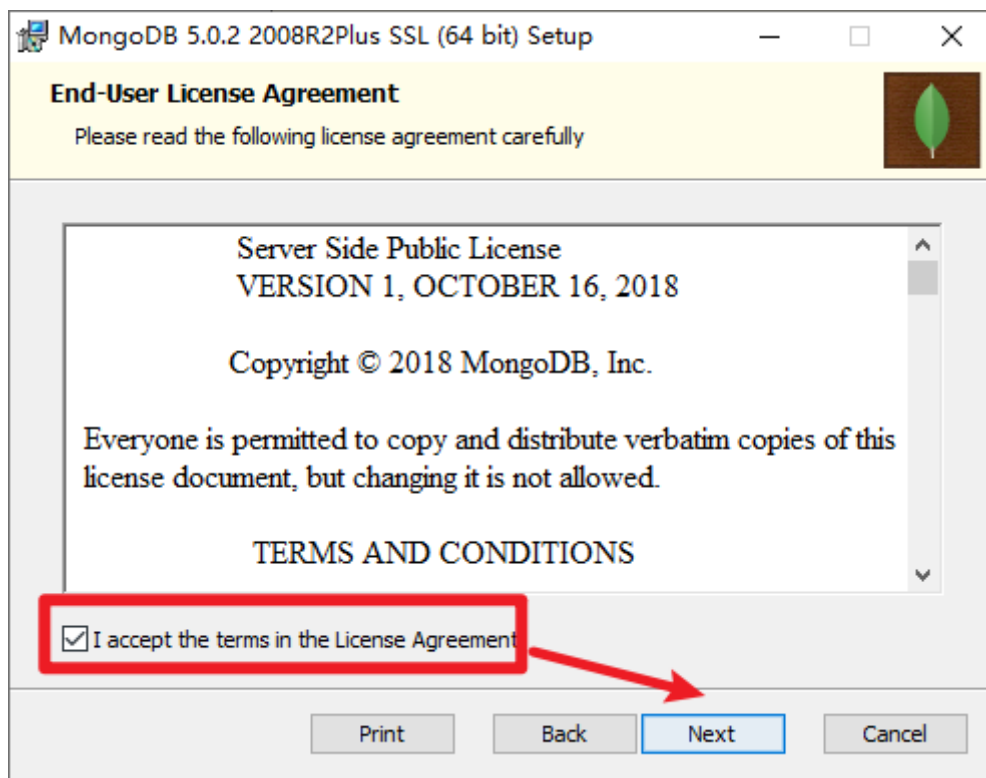
2.2 MongoDB数据库环境搭建

① 第一步



② 点击下一步进行安装





MongoDB 5.0.2 2008R2Plus SSL (64 bit) Service Custo...

Service Configuration

Specify optional settings to configure MongoDB as a service.

☒ Install MongoDB as a Service

☒ Run service as Network Service user

☐ Run service as a local or domain user:

Account Domain:

Account Name:

Account Password:

Service Name:

Data Directory:

Log Directory:

< Back Next > Cancel

MongoDB Compass

Install MongoDB Compass

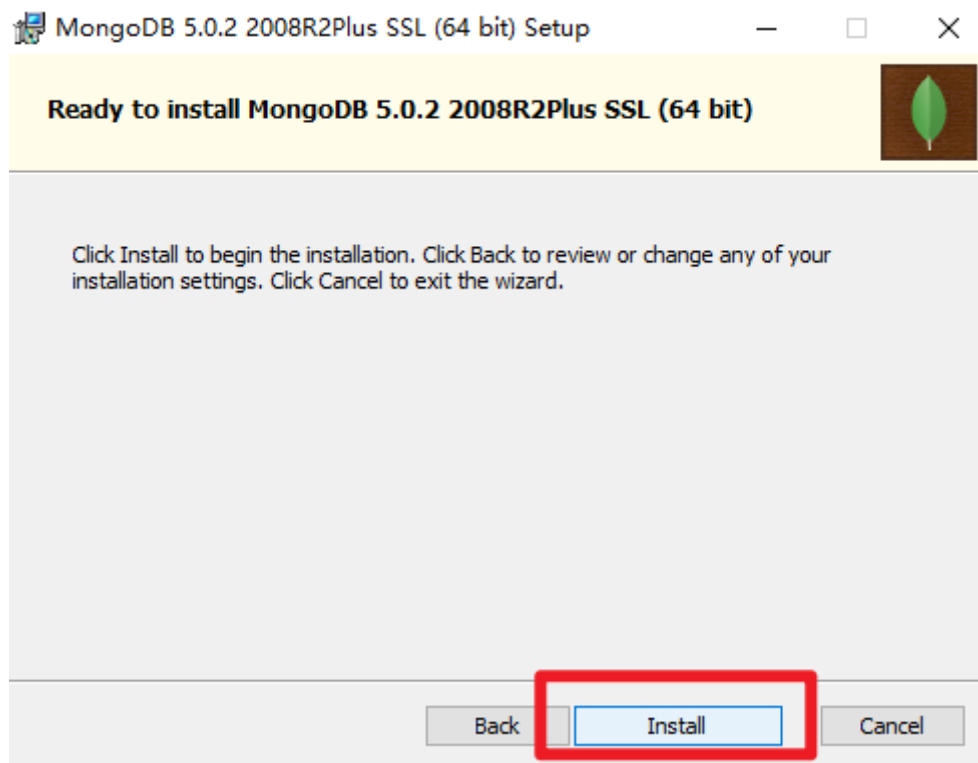
MongoDB Compass is the official graphical user interface for MongoDB.

By checking below this installer will automatically download and install the latest version of MongoDB Compass on this machine. You can learn more about MongoDB Compass here: <https://www.mongodb.com/products/comp...>

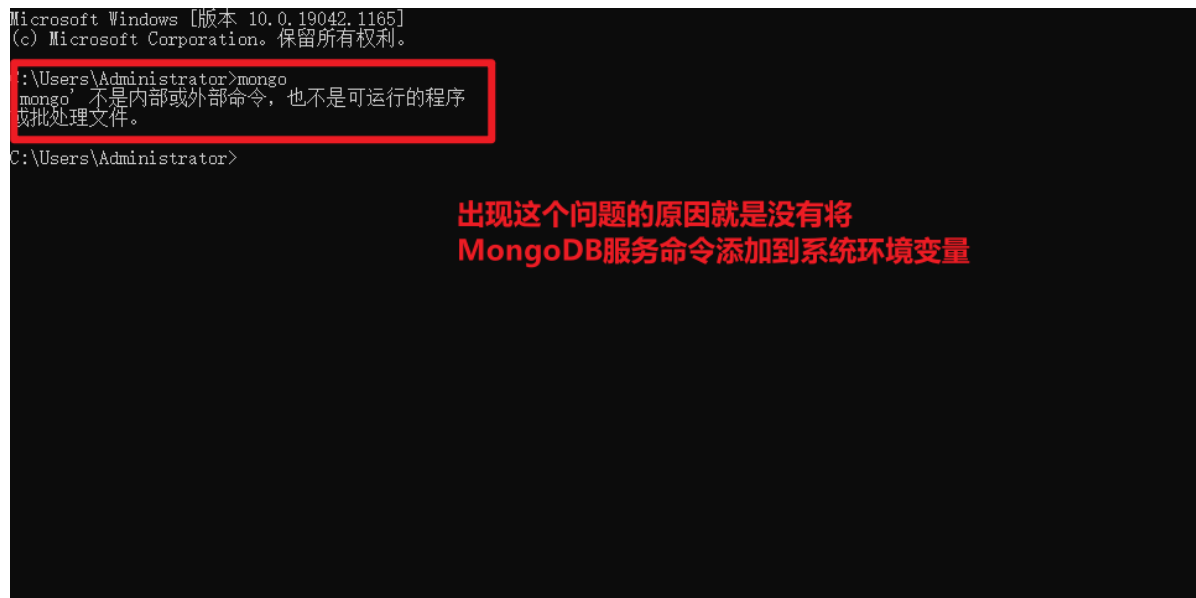
建议去掉 勾选

☐ Install MongoDB Compass

Back Next Cancel

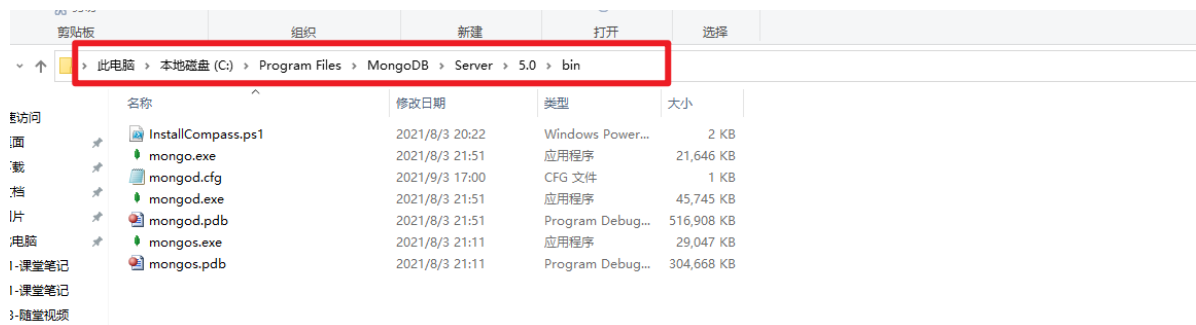


安装完毕之后，win+R 输入cmd 打开 命令行窗口



还需要将MongoDB命令添加到环境变量中，

找到MongoDB的安装路径下，有一个 bin 的文件夹

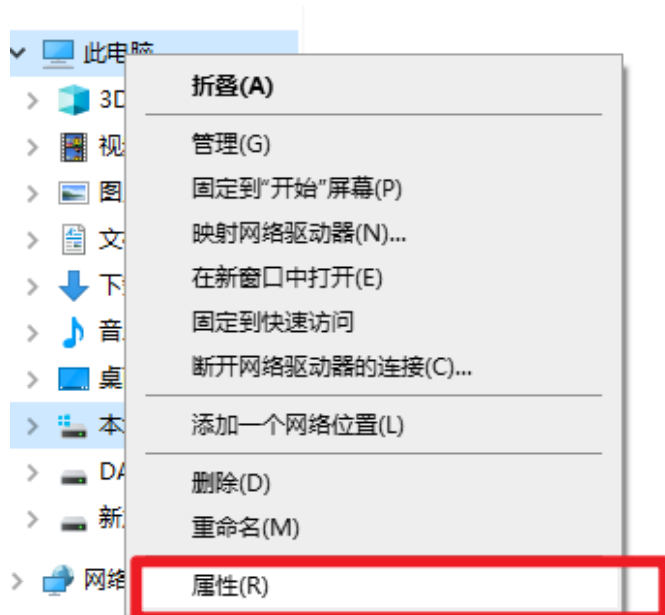


复制这个路径



将此路径添加到系统环境变量中

右键我的电脑

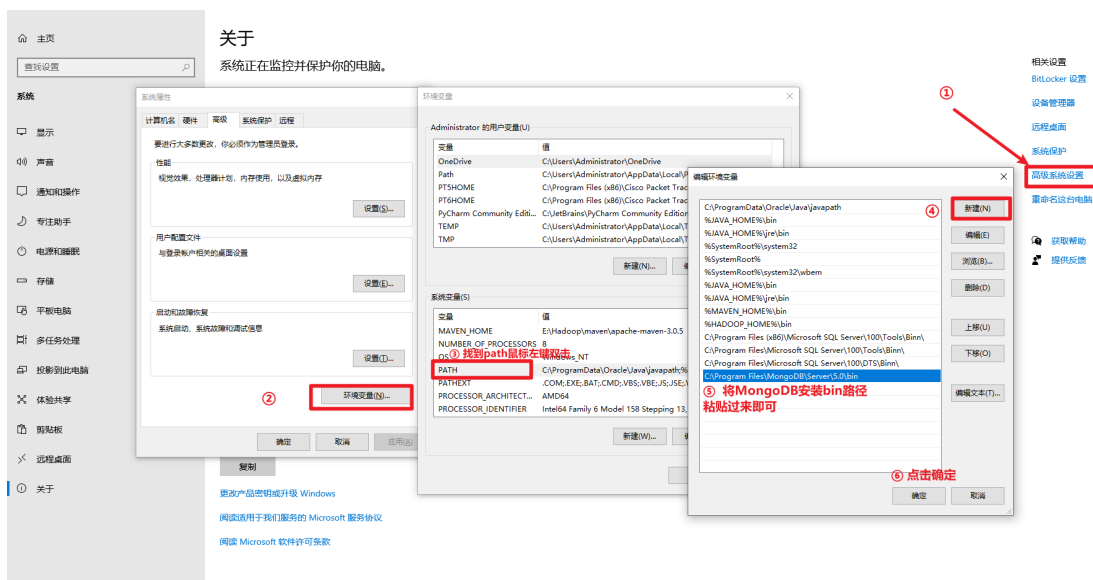


鼠标右键 此电脑

选择高级系统设置



再选择环境变量



一路确定完毕之后，记得将之前打开的那个cmd关闭掉，再次重新打开cmd

再次 输入 mongo 如果出现下图所示表示环境以及配置成功

```
Microsoft Windows [版本 10.0.19042.1165]
(c) Microsoft Corporation. 保留所有权利。

C:\Users\Administrator>mongo
MongoDB shell version v5.0.2
connecting to: mongodb://127.0.0.1:27017/?compressors=disabled&gssapiServiceName=mongodb
Implicit session: session { "id" : UUID("bfd67c9c-e000-4dac-874f-708d80954eae") }
MongoDB server version: 5.0.2
=====
Warning: the "mongo" shell has been superseded by "mongosh",
which delivers improved usability and compatibility. The "mongo" shell has been deprecated and will be removed in
an upcoming release.
We recommend you begin using "mongosh".
For installation instructions, see
https://docs.mongodb.com/mongodb-shell/install/
=====
The server generated these startup warnings when booting:
  2021-09-03T17:00:07.486+08:00: Access control is not enabled for the database. Read and write access to data and
configuration is unrestricted
-----

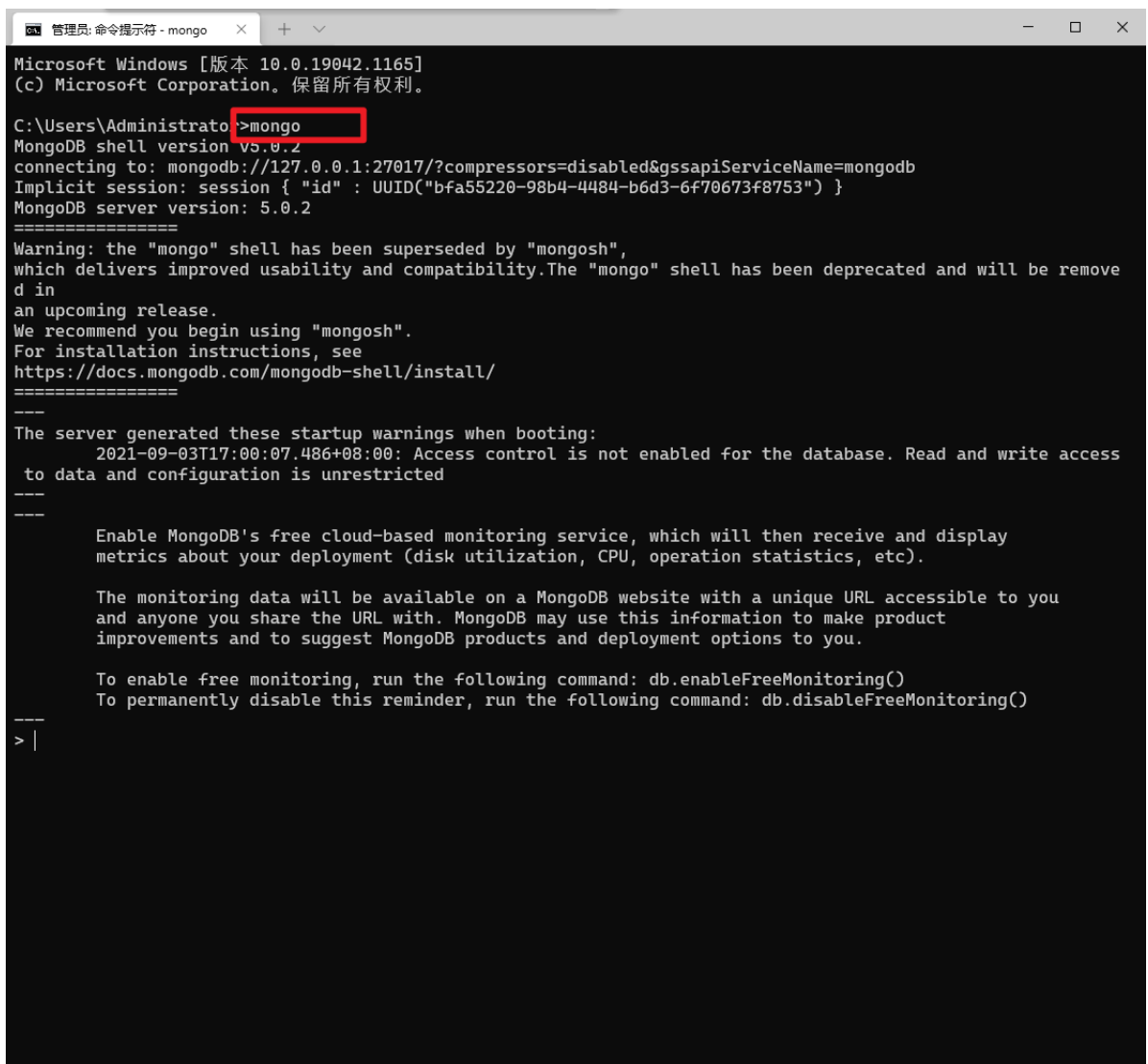
  Enable MongoDB's free cloud-based monitoring service, which will then receive and display
metrics about your deployment (disk utilization, CPU, operation statistics, etc).

  The monitoring data will be available on a MongoDB website with a unique URL accessible to you
and anyone you share the URL with. MongoDB may use this information to make product
improvements and to suggest MongoDB products and deployment options to you.

  To enable free monitoring, run the following command: db.enableFreeMonitoring()
  To permanently disable this reminder, run the following command: db.disableFreeMonitoring()
-----
>
```

2.3 MongoDB数据库的基本使用

执行这些语句，先打开cmd，在cmd中输入 mongo 回车。



```
管理员: 命令提示符 - mongo
Microsoft Windows [版本 10.0.19042.1165]
(c) Microsoft Corporation. 保留所有权利。

C:\Users\Administrator>mongo
MongoDB shell version v5.0.2
connecting to: mongodb://127.0.0.1:27017/?compressors=disabled&gssapiServiceName=mongodb
Implicit session: session { "id" : UUID("bfa55220-98b4-4484-b6d3-6f70673f8753") }
MongoDB server version: 5.0.2
=====
Warning: the "mongo" shell has been superseded by "mongosh",
which delivers improved usability and compatibility. The "mongo" shell has been deprecated and will be removed in
an upcoming release.
We recommend you begin using "mongosh".
For installation instructions, see
https://docs.mongodb.com/mongodb-shell/install/
=====
The server generated these startup warnings when booting:
  2021-09-03T17:00:07.486+08:00: Access control is not enabled for the database. Read and write access
to data and configuration is unrestricted
-----

  Enable MongoDB's free cloud-based monitoring service, which will then receive and display
metrics about your deployment (disk utilization, CPU, operation statistics, etc).

  The monitoring data will be available on a MongoDB website with a unique URL accessible to you
and anyone you share the URL with. MongoDB may use this information to make product
improvements and to suggest MongoDB products and deployment options to you.

  To enable free monitoring, run the following command: db.enableFreeMonitoring()
  To permanently disable this reminder, run the following command: db.disableFreeMonitoring()
-----
> |
```

数据库的命令：

- MongoDB中是不需要提前创建数据库的
- 查看所有数据库： `show dbs / show databases`

```
> show dbs
admin    0.000GB
config  0.000GB
local    0.000GB
> show databases
admin    0.000GB
config  0.000GB
local    0.000GB
> |
```

查看所有的数据库

- 切换要使用的数据库： `use 数据库名`
- 查看当前使用的数据库： `db`

```
> show databases
admin    0.000GB
config  0.000GB
local    0.000GB
> use aaa
switched to db aaa
> db
aaa
> |
```

- 删除数据库： 先切换到要删除的数据库 `use 数据库名` 再去执行 `db.Dropdatabase()`

```
> db.dropDatabase()
{ "ok" : 1 }
> |
```

集合的命令

- 集合可以理解为等同于 MySQL数据库中的 数据表
- 并且 集合 是不需要提前去创建的
- 查看当前数据库所有的集合： `show collections` （等同于MySQL中： `show tables`）

```
> show dbs
admin    0.000GB
config  0.000GB
local    0.000GB
> use admin
switched to db admin
> db
admin
> show collections
system.version
> |
```

查看数据库中所有的集合

- 删除集合： `db.集合名.drop()`


```

> show dbs  查看所有的数据库
admin      0.000GB
config     0.000GB
local      0.000GB
students   0.000GB
> use students  切换数据库
switched to db students
> show collections  显示数据库中所有的集合（表）
stu
> db.stu.drop()  删除集合（表）
true
> show collections
> db.dropDatabase()  删除数据库
{ "ok" : 1 }
> show dbs
admin      0.000GB
config     0.000GB
local      0.000GB
> |

```

在MongoDB数据库中，数据库以及数据集合是不需要提前创建的，当我们向一个不存在的数据库以及集合中插入数据的时候，数据库和数据集合会自动的创建出来

2.4 数据操作- 插入（新增数据）

语法

```

1  插入数据的时候，接收的是一个文档（一条数据）文档的格式一个json字符串，（Python字典）
2  db.集合名.insert({key:value})
3
4
5  zhangsan 18 true 女
6  db.stu.insert({"name": "zhangsan", "age": 18, "gender": true, "like":
   "meinv"})
7
8
9  如果想要同时插入多条数据，
10 db.集合名.insert([{}, {}, {}, {}, .....])
11
12 db.stu.insert([
13     {"name": "lisi", "age": 19, "gender": true, "like": "chi"},
14     {"name": "wangwu", "age": 20, "gender": true, "like": "waner"},
15     {"name": "xiaoli", "age": 18, "gender": false, "like": "帅哥"}
16 ])

```

```

> db.stu.find()
{ "_id" : ObjectId("6131f24567059b3cd6281417"), "name" : "zhangsan", "age" : 18, "gender" : true, "like" : "meinv" }
> db.stu.insert([
...   {"name": "lisi", "age": 19, "gender": true, "like": "chi"},
...   {"name": "wangwu", "age": 20, "gender": true, "like": "waner"},
...   {"name": "xiaoli", "age": 18, "gender": false, "like": "帅哥"}
... ])
BulkWriteResult({
  "writeErrors" : [ ],
  "writeConcernErrors" : [ ],
  "nInserted" : 3,
  "nUpserted" : 0,
  "nMatched" : 0,
  "nModified" : 0,
  "nRemoved" : 0,
  "upserted" : [ ]
})
> db.stu.find()
{ "_id" : ObjectId("6131f24567059b3cd6281417"), "name" : "zhangsan", "age" : 18, "gender" : true, "like" : "meinv" }
{ "_id" : ObjectId("6131f30067059b3cd6281418"), "name" : "lisi", "age" : 19, "gender" : true, "like" : "chi" }
{ "_id" : ObjectId("6131f30067059b3cd6281419"), "name" : "wangwu", "age" : 20, "gender" : true, "like" : "waner" }
{ "_id" : ObjectId("6131f30067059b3cd628141a"), "name" : "xiaoli", "age" : 18, "gender" : false, "like" : "帅哥" }
帅哥" }
>

```

2.5 查询数据

语法

```
1  查询全部数据      select * from stu;
2  db.集合名.find()
3
4  查询 stu集合中所有的学生信息
5  db.stu.find()
6
7  查询一条数据
8  db.集合名.findOne()
9
10 db.stu.findOne()
11
12 美化输出
13 db.集合名.find().pretty()
14
15 db.stu.find().pretty()
```

```
> db.stu.find().pretty()
{
  "_id" : ObjectId("6131f24567059b3cd6281417"),
  "name" : "zhangsan",
  "age" : 18,
  "gender" : true,
  "like" : "meinv"
}
{
  "_id" : ObjectId("6131f30067059b3cd6281418"),
  "name" : "lisi",
  "age" : 19,
  "gender" : true,
  "like" : "chi"
}
{
  "_id" : ObjectId("6131f30067059b3cd6281419"),
  "name" : "wangwu",
  "age" : 20,
  "gender" : true,
  "like" : "waner"
}
{
  "_id" : ObjectId("6131f30067059b3cd628141a"),
  "name" : "xiaoli",
  "age" : 18,
  "gender" : false,
  "like" : "帅哥"
}
> |
```

最终的目的是要在Python代码中去操作MongoDB数据库。

