

mongodb和python交互

安装环境

```
1 pip install pymongo
```

官方文档: <https://pymongo.readthedocs.io/en/stable/>

使用样例:

1. 引入包pymongo

```
1 import pymongo
```

2. 连接, 创建客户端

```
1 client = pymongo.MongoClient("localhost", 27017)
2 client = pymongo.MongoClient('mongodb://localhost:27017/')
```

3. 获得数据库test1

```
1 db = client.test1
2 db = client['test']
```

4. 获得集合movie

```
1 collection = db.movie
2 collection = db['movie']
```

5. 添加数据

```

1  #增加一条
2  m1={name:'300集',actor:'高总',level:10}
3  m1_id = movie.insert_one(s1).inserted_id
4  #增加多条
5  mids = movie.insert_many([movie1,movie2])

```

注意： 原insert方法也可以实现上面的功能，但是在PyMongo 3.x的版本已经不推荐使用了

6. 查找数据

- find() 返回一个生成器对象
- find_one() 返回一条数据

```

1  result = movie.find_one()
2  result = movie.find_one({'name':'300集'})
3  result = movie.find_one({'_id':ObjectId('5932a80115c2606a59e8a049')})
4  result = movie.find_one({'level':{'$gt':1}})
5  results = movie.find()

```

比较符号

符号	含义	示例
\$lt 小于 {'age': {'\$lt': 20}}		
\$gt 大于 {'age': {'\$gt': 20}}		
\$lte 小于等于 {'age': {'\$lte': 20}}		
\$gte 大于等于 {'age': {'\$gte': 20}}		
\$ne 不等于 {'age': {'\$ne': 20}}		
\$in 在范围内 {'age': {'\$in': [20, 23]}}		
\$nin 不在范围内 {'age': {'\$nin': [20, 23]}}		

功能符号

符号	含义	示例	示例含义
\$regex 匹配正则表达式 {'name': {'\$regex': '^M.*'}}	name以M开头		
\$exists 属性是否存在 {'name': {'\$exists': True}}	name属性存在		
\$type 类型判断 {'age': {'\$type': 'int'}}	age的类型为int		
\$mod 数字模操作 {'age': {'\$mod': [5, 0]}}	年龄模5余0		
\$text 文本查询 {'\$text': {'\$search': 'Mike'}}	text类型的属性中包含Mike字符串		
\$where 高级条件查询 {'\$where': 'obj.fans_count == obj.follows_count'}	自身粉丝数等于关注数		

7. 获取文档个数

```
1 count = movie.count()
2 count = collection.find({'level': 10}).count()
```

8. 排序

```
1 results = collection.find().sort('name', pymongo.ASCENDING)
```

9. 偏移

```
1 collection.find().sort('name', pymongo.ASCENDING).skip(2).limit(2)
```

10. 更新

```
1 args = {"name": "曹操"}
2 student.update_one(args, {'$set': {'field': value}})
3 result = collection.update_many(query, {'$set': {'field': value}})
```

注意： 也可以实现上面的功能，但是在PyMongo 3.x的版本已经不推荐使用了

11. 删除

```
1 result = collection.remove({'name': '300集'}) #符合条件的所有数据均会被删除
2 result = collection.delete_one({'name': '300集'})
3 result = collection.delete_many({'age': {'$lt': 25}})
```