

# Python MongoDB

MongoDB 是目前最流行的 NoSQL 数据库之一，使用的数据类型 BSON（类似 JSON）。

MongoDB 数据库安装与介绍可以查看我们的 [MongoDB 教程](#)。

## PyMongo

Python 要连接 MongoDB 需要 MongoDB 驱动，这里我们使用 PyMongo 驱动来连接。

### pip 安装

pip 是一个通用的 Python 包管理工具，提供了对 Python 包的查找、下载、安装、卸载的功能。

安装 pymongo:

```
$ python3 -m pip3 install pymongo
```

也可以指定安装的版本:

```
$ python3 -m pip3 install pymongo==3.5.1
```

更新 pymongo 命令：

```
$ python3 -m pip3 install --upgrade pymongo
```

### easy\_install 安装

旧版的 Python 可以使用 easy\_install 来安装，easy\_install 也是 Python 包管理工具。

```
$ python -m easy_install pymongo
```

更新 pymongo 命令：

```
$ python -m easy_install -U pymongo
```

## 测试 PyMongo

接下来我们可以创建一个测试文件 demo\_test\_mongodb.py，代码如下：

### demo\_test\_mongodb.py 文件代码：

```
#!/usr/bin/python3
import pymongo
```

执行以上代码文件，如果没有出现错误，表示安装成功。

## 创建数据库

### 创建一个数据库

创建数据库需要使用 MongoClient 对象，并且指定连接的 URL 地址和要创建的数据库名。

如下实例中，我们创建的数据库 runoobdb：

#### 实例

```
#!/usr/bin/python3
import pymongo
myclient = pymongo.MongoClient("mongodb://localhost:27017/")
mydb = myclient["runoobdb"]
```

**注意：**在 MongoDB 中，数据库只有在内容插入后才会创建！就是说，数据库创建后要创建集合(数据表)并插入一个文档(记录)，数据库才会真正创建。

### 判断数据库是否已存在

我们可以读取 MongoDB 中的所有数据库，并判断指定的数据库是否存在：

#### 实例

```
#!/usr/bin/python3
import pymongo
myclient = pymongo.MongoClient('mongodb://localhost:27017/')
dblist = myclient.list_database_names()
# dblist = myclient.database_names()
if "runoobdb" in dblist:
    print("数据库已存在！")
```

**注意：**database\_names 在最新版本的 Python 中已废弃，Python3.7+ 之后的版本改为了 list\_database\_names()。

## 创建集合

MongoDB 中的集合类似 SQL 的表。

### 创建一个集合

MongoDB 使用数据库对象来创建集合，实例如下：

#### 实例

```
#!/usr/bin/python3
import pymongo
myclient = pymongo.MongoClient("mongodb://localhost:27017/")
mydb = myclient["runoobdb"]
mycol = mydb["sites"]
```

**注意:** 在 MongoDB 中，集合只有在内容插入后才会创建! 就是说，创建集合(数据表)后要再插入一个文档(记录)，集合才会真正创建。

### 判断集合是否已存在

我们可以读取 MongoDB 数据库中的所有集合，并判断指定的集合是否存在：

#### 实例

```
#!/usr/bin/python3
import pymongo
myclient = pymongo.MongoClient('mongodb://localhost:27017/')
mydb = myclient['runoobdb']
collist = mydb.list_collection_names()
# collist = mydb.collection_names()
if "sites" in collist: # 判断 sites 集合是否存在
print("集合已存在! ")
```

**注意：**collection\_names 在最新版本的 Python 中已废弃，Python3.7+ 之后的版本改为了list\_collection\_names()。

### 增、删、改、查等操作

下表列出了 MongoDB 的更多操作，详情可点击具体链接：

序号	功能
1	<a href="#">添加数据</a>
2	<a href="#">查询数据</a>
3	<a href="#">修改数据</a>
4	<a href="#">数据排序</a>
5	<a href="#">删除数据</a>

[← Python3 range\(\) 函数用法](#)

[Python Mongodb 插入文档 →](#)

[✎ 点我分享笔记](#)

