Bat命令操作大全：<https://blog.csdn.net/sinat_23338865/article/details/52401317#commentBox>

wmic生成html网页：

wmic /output:a.html logicaldisk where(name='c:') get DeviceID,Size,FreeSpace,Description,FileSystem /format:htable

Python面试题目集合：**<https://github.com/taizilongxu/interview_python>**

**Windows安装Weblogic(7001)**

**https://blog.csdn.net/qq\_36868342/article/details/79967606**

e:

cd E:\Weblogic\Middleware\Oracle\_Home\user\_projects\domains\base\_domain

startWebLogic.cmd

浏览器：<http://localhost:7001/console/login/LoginForm.jsp>

用户名：weblogic；密码：weblogic

**1，23种Python设计模式：https://www.cnblogs.com/Liqiongyu/p/5916710.html**

**Java设计模式：<https://github.com/youlookwhat/DesignPattern>**

**2，Windows安装Redis**

**Windows安装部署：**[**https://blog.csdn.net/u012343297/article/details/78839063**](https://blog.csdn.net/u012343297/article/details/78839063)

开启后台：redis-server.exe redis.windows.conf

启动：redis-cli.exe

pip install redis

python操作redis：

# -\*- coding:utf-8 -\*-

import redis

#创建redis链接对象

r = redis.Redis(host='127.0.0.1',password='123456',port=6379,decode\_responses=True)

#存储键值对

r.set('site','www.qi.cn')

#获取值

print(r.get('site'))

Java操作Redis：（<https://www.cnblogs.com/youcong/p/8098881.html>）

<**dependency**>

<**groupId**>redis.clients</**groupId**>

<**artifactId**>jedis</**artifactId**>

<**version**>2.5.0</**version**>*<!--$NO-MVN-MAN-VER$-->*

</**dependency**>

**3，Windows安装memcache**

schtasks /create /sc onstart /tn memcached /tr "'E:\MyWeb\memcached-1.4.5-amd64\memcached-amd64\memcached.exe' -m 512"

memcached.exe

Windows安装memcache：

<http://www.runoob.com/memcached/window-install-memcached.html>

Python操作Memcache：

pip install pymemcache

**from pymemcache.client.base import Client**

**client = Client(('localhost', 11211))**

**client.set('some\_key', 'some\_value')**

**result = client.get('some\_key')**

**print(result)**

Java操作Memcache：

<**dependency**>

<**groupId**>net.spy</**groupId**>

<**artifactId**>spymemcached</**artifactId**>

<**version**>2.12.3</**version**>

</**dependency**>

@Test

**public** void memcache()

{

**try**{

*// 本地连接 Memcached 服务*

MemcachedClient mcc = **new** MemcachedClient(**new** InetSocketAddress(**"127.0.0.1"**, 11211));

**String** key = **"cacheKey1"**;

mcc.set(key, 0, **"2"**);

**String** value = (**String**) mcc.get(key);

System.out.println(value);

*// 关闭连接*

mcc.shutdown();

}**catch**(Exception ex){

System.out.println( ex.getMessage() );

}

}

**SQL Server(1433)( pip install pymssql)**

SQL Server2008下载地址：<http://www.downza.cn/soft/182269.html>

安装方法：<https://jingyan.baidu.com/article/948f592434b407d80ef5f97d.html>

安装无法重启：<http://www.51testing.com/html/21/318121-229360.html>

SQL Server管理工具下载：<https://www.cnblogs.com/yzl050819/p/8284242.html>

无法SQL登陆：<https://blog.csdn.net/github_35160620/article/details/52676416>

DOS命令：osql -S (local)\SQLEXPRESS -U sa -P a12345678 /q "create login dba with password='abcd1234@', default\_database=master;"

sqlcmd -S"(local)\SQLEXPRESSLYJ" -U"sa" -P"a12345678" -d"master" -Q"select \* from test.dbo.student;" -o sql.txt

create login dba with password='abcd1234@', default\_database=master;

**import pymssql**

**conn = pymssql.connect(host='JCVECB54U9LXMZQ\SQLEXPRESSLYJ',**

**user='sa',**

**password='a12345678',**

**database='test',**

**charset='utf8')**

**#查看连接是否成功**

**cursor = conn.cursor()**

**try:**

**sql = "insert into student values(1, '张三')"**

**cursor.execute(sql)**

**conn.commit()**

**except Exception as ex:**

**conn.rollback()**

**raise ex**

**finally:**

**sql = "select \* from student"**

**cursor.execute(sql)**

**#用一个rs变量获取数据**

**rs = cursor.fetchall()**

**print(rs)**

**postgresql(5433)( pip install psycopg2)**

Windows安装：<https://www.yiibai.com/postgresql/install-postgresql.html>

**import psycopg2**

**# 创建连接对象**

**conn = psycopg2.connect(database="postgres", user="postgres", password="a12345678", host="localhost", port="5433")**

**cur = conn.cursor() # 创建指针对象**

**# 创建表**

**cur.execute("CREATE TABLE student(id integer,name varchar,sex varchar);")**

**# 插入数据**

**cur.execute("INSERT INTO student(id,name,sex)VALUES(%s,%s,%s)", (1, 'Aspirin', 'M'))**

**# 关闭练级**

**conn.commit()**

**# 获取结果**

**cur.execute('SELECT \* FROM student')**

**results = cur.fetchall()**

**print(results)**

**cur.close()**

**conn.close()**

**DB2(50000)(pip install ibm\_db)**

**db2cmd db2**

**CONNECT TO TEST USER db2admin USING a12345678**

**import ibm\_db  
conn = ibm\_db.connect("DATABASE=test;HOSTNAME=localhost;PORT=50000;PROTOCOL=TCPIP;UID=db2admin;PWD=a12345678;", "", "")  
if conn:  
 print('已链接DB2');**

**4，MongoDB(27017)( pip install pymongo)**

Windows安装：<http://www.runoob.com/mongodb/mongodb-window-install.html>

E:\MongoDB\Server\4.0\bin\mongod --dbpath E:\MongoDB\Server\4.0\data\db

E:\MongoDB\Server\4.0\bin\mongo.exe

Python操作mongoDB：(https://www.cnblogs.com/melonjiang/p/6536876.html)

*# -\*- coding:utf-8 -\*-*

**from pymongo import MongoClient  
  
conn = MongoClient('127.0.0.1', 27017)  
db = conn.mydb #连接mydb数据库，没有则自动创建  
my\_set = db.test\_set #使用test\_set集合，没有则自动创建  
  
db.test\_set.remove()  
  
users=[{"name":"zhangsan","age":18},{"name":"lisi","age":20}]  
my\_set.insert(users)  
for i in my\_set.find():  
 print(i)**

Java操作MongoDB（<https://www.cnblogs.com/gaopeng527/p/4399127.html>）：

<dependency>

<groupId>org.mongodb</groupId>

<artifactId>mongo-java-driver</artifactId>

<version>3.2.2</version><!--$NO-MVN-MAN-VER$-->

</dependency>

@Test

**public** void mongodb()

{

Mongo connection = **null**;

**try** {

connection = **new** Mongo(**"localhost:27017"**);

} **catch** (UnknownHostException e) {

*//* ***TODO Auto-generated catch block***

e.printStackTrace();

}

DB db = connection.getDB(**"mylearndb"**);

DBCollection users = db.getCollection(**"users"**);

*/\*\**

*/\*\**

*\* DBObject代表文档，这是一个接口，java中提供了多种实现，最简单的就是BasicDBObject了*

*\*/*

DBObject user = **new** BasicDBObject();

user.put(**"name"**, **"jimmy"**);

user.put(**"age"**, **"34"**);

DBObject address = **new** BasicDBObject();

address.put(**"city"**, **"bj"**);

address.put(**"street"**, **"bq road"**);

address.put(**"mail"**, **"ufpark 68#"**);

*/\*\**

*\* 对于内嵌文档，我们需要先将内嵌文档填充后，再填充到外层文档中！*

*\*/*

user.put(**"address"**, address);

*// 将该文档插入到集合中*

users.**insert**(user);

*// 从集合中查询数据，我们就查询一条，调用findOne即可*

DBObject dbUser = users.findOne();

System.out.println(**"name"** + **" : "** + dbUser.get(**"name"**) );

System.out.println(**"age"** + **" : "** + dbUser.get(**"age"**) );

DBObject dbAddress = (DBObject)user.get(**"address"**);

System.out.println(**"city"** + **" : "** + dbAddress.get(**"city"**) );

System.out.println(**"street"** + **" : "** + dbAddress.get(**"street"**) );

System.out.println(**"mail"** + **" : "** + dbAddress.get(**"mail"**) );

}

Django（pip install django）：

教程：<https://code.ziqiangxuetang.com/django/django-tutorial.html>

django-admin.py startproject HelloWorld

python manage.py startapp learn # 创建learn的app

python36 manage.py runserver 127.0.0.1:8000 #重启django服务器

import django

django.VERSION #查看版本

**Flask（pip install flask）**

教程1：<http://docs.jinkan.org/docs/flask/>

教程2：<http://www.pythondoc.com/flask-mega-tutorial/index.html>

**from** flask **import** Flask  
app = Flask(\_\_name\_\_)  
  
@app.route(**'/'**)  
**def** hello\_world():  
 **return 'Hello World!'**@app.route(**'/appmy'**)  
**def** appmy():  
 **return 'appmy'  
  
if** \_\_name\_\_ == **'\_\_main\_\_'**:  
 app.run(debug=**True**)

**Hadoop**

Windows安装部署：<https://blog.csdn.net/xiligey1/article/details/79728152>

e:

cd E:\MyWeb\hadoop-2.7.5\hadoop-2.7.5\sbin

start-all.cmd  
<http://localhost:8088/cluster>

<http://localhost:50070/dfshealth.html#tab-overview>

**Spark**

Windows安装部署：<https://blog.csdn.net/xiligey1/article/details/79728987>

安装Python库：

e:

cd E:\MyWeb\spark-2.3.2-bin-hadoop2.7\spark-2.3.2-bin-hadoop2.7\python

启动：spark-shell或pyspark

from pyspark import SparkContext

**RabbitMQ(pip install pika)**

Windows安装部署：<https://www.cnblogs.com/miketwais/p/RabbitMQ.html>

e:

cd E:\RabbitMQ Server\rabbitmq\_server-3.3.4\sbin

rabbitmq-plugins.bat enable rabbitmq\_management

rabbitmq-service.bat install

net start "RabbitMQ"

**mysql(3306)(pip install pymysql)**

Windows安装部署：

**import** pymysql  
conn = pymysql.connect(host=**'127.0.0.1'**, user=**'root'**, passwd=**'123456'**, db=**'test'**)  
cur = conn.cursor()  
*# 查询*sql = **"select** *\** **from info"**reCount = cur.execute(sql) *# 返回受影响的行数*print(reCount)  
data = cur.fetchall() *# 返回数据,返回的是tuple类型*print(data)  
cur.close()  
conn.close()

**Oracle(1521)(pip install cx\_Oracle)**

Windows安装部署：

**import** cx\_Oracle  
conn=cx\_Oracle.connect(**'luyanjie2/a12345678@orcl'**)  
c=conn.cursor()  
c.execute(**"create table python\_curd(id number, name varchar2(50),password varchar(50),primary key(id))"**)  
conn.commit();  
c.close()  
conn.close()

**postgresql**

Windows安装部署：<https://blog.csdn.net/sunxiaopengsun/article/details/77034134>

**HBase**

Windows安装部署：<https://blog.csdn.net/wm6752062/article/details/80381704>

**Hive**

Windows安装部署：<https://blog.csdn.net/f7anty/article/details/72629622>

**Zookeeper(pip install kazoo)**

Windows安装部署：<https://blog.csdn.net/qiunian144084/article/details/79192819>

CentOS安装部署：<https://blog.csdn.net/yanyan42/article/details/81667335>

e:

cd E:\MyWeb\zookeeper-3.4.13\bin

zkServer.cmd

zkCli.cmd

*# -\*- coding: utf-8 -\*-***from** kazoo.client **import** KazooClient  
zk = KazooClient(hosts=**'127.0.0.1:2181'**)  
zk.start()  
zk.ensure\_path(**'/mysql/test'**)  
*##zk.create('/mysql/test/node',b'node value')***if** zk.exists(**"/mysql/test"**):  
 print(zk.get\_children(**'/mysql/test'**))  
 print(zk.get(**'/mysql/test/node'**))

**Jenkis**

Windows安装部署：<https://blog.csdn.net/ronnyjiang/article/details/51208009>

java -jar jenkins.war

**运维监控工具Zabbix**

开启：systemctl start zabbix-server

停止：systemctl stop zabbix-server

*#Admin , zabbix*

浏览器访问：http://192.168.56.1/zabbix/

服务器监控工具架：Zabbix

导入Zabbix数据库：<https://blog.csdn.net/purplegalaxy/article/details/37819899>

lamp安装Zabbix：

bvg<https://www.centos.bz/2017/11/centos-7%E5%AE%89%E8%A3%85zabbix-3-4/>

centos下的lnmp环境安装Zabbix：<https://www.cnblogs.com/zhnhelloworld/p/5864788.html>

Zabbix错误配置修改：<https://www.cnblogs.com/rusking/p/4441362.html>

重置密码：<https://www.centos.bz/2017/08/zabbix-forget-password-reset/>

**运维监控工具GoAccess(仅支持[linux系统](https://www.baidu.com/s?wd=linux%E7%B3%BB%E7%BB%9F&tn=24004469_oem_dg&rsv_dl=gh_pl_sl_csd" \t "_blank))**

Nginx监控日志工具GoAccess的CentOS安装：

<https://www.cnblogs.com/yjlch1016/p/9102894.html>

生成web报表：./goaccess /usr/local/nginx/logs/access.log -o /var/www/html/nginx\_log/index.html --real-time-html --time-format='%H:%M:%S' --date-format='%d/%b/%Y' --log-format=COMBINED

**运维监控工具Open-Falcon(仅支持**[**linux系统**](https://www.baidu.com/s?wd=linux%E7%B3%BB%E7%BB%9F&tn=24004469_oem_dg&rsv_dl=gh_pl_sl_csd)**)**

服务器监控软件Open-Falcon：<http://blog.51cto.com/zhanx/2059434>

**自动化运维工具Saltstack**

多台机器远程执行命令工具Saltstack安装：

<https://www.centos.bz/2017/07/centos-6-7-saltstack-install/>

**自动化运维工具Ansible**

自动化应用部署工Ansible具安装：https://www.cnblogs.com/gzxbkk/p/7515634.html

**Pyhton2 Windows安装（与Python3兼容）**

pip2 报错解决方案：<https://www.cnblogs.com/wuzhiyi/p/6051311.html>

安装Jetty：<https://blog.csdn.net/u010824252/article/details/46609339>

安装Netty: <http://www.cnblogs.com/coderJiebao/p/Netty01.html>

消息队列产品：Kafka、ActiveMQ、ZeroMQ、RabbitMQ

缓存：memcached、Redis

会话：Session、Cookies

分布式：Zookeeper+dubbo(java)+Spring

分布式计算：Hadoop、Spark

大数据数据库：HBase、Hive

Nginx请求负载分布式

<https://www.cnblogs.com/lilongsheng1125/p/4978485.html>

MySQL分布式数据库

<https://www.cnblogs.com/telwanggs/p/7448242.html>

常用小技巧：

array\_unique(,SORT\_REGULAR)

array\_merge()

number\_format(数据,位数)

//日期转换成时间戳

$oldtime = '2019-01-02 10:30:21';//严格按照此格式

$catime = strtotime($oldtime);

//时间戳转换成日期

$timespacehe=time();

$mydate=date("Y-m-d H:i:s",$timespacehe);