## 常用Shell命令

### Linux命令信息查看

#### xxx –help

#### man xxx

#### info xxx

### ssh连接远程服务器

ssh asiainfo@42.99.2.60 -p 9800

### vi命令

i 插入字符，进入编辑模式

a 在当前字符后追加一个字符，进入编辑模式

A 在当前行尾追加一个字符，进入编辑模式

o 插入一行，进入编辑模式

x 删除字符

dd 删除一行

h 左移一个字符

l 右移一个字符

j 下移一行

k 上移一行

$ 光标移动到当前行尾

yy 拷贝一行

p 粘贴

G 到文件尾部

/ 查找字符串

:w 保存文件

:q 退出

:x 保存文件，退出

:q! 强行退出，不保存

### linux配置java环境

vi ~/.bashrc

export JAVA\_HOME=/jlp/tools/jdk1.8.0\_144

export JRE\_HOME=${JAVA\_HOME}/jre

export CLASSPATH=.:${JAVA\_HOME}/lib:${JRE\_HOME}/lib

export PATH=${JAVA\_HOME}/bin:$PATH

source ~/.bashrc

java -version

### scp传输文件：

scp WEB\_20170608\_app3.tar.gz [vmuser@192.168.208.103:/tmp/jlp20170608](mailto:vmuser@192.168.208.103:/tmp/jlp20170608)

### tar文件打包和解压：

tar -zcvf UpDPWD\_7to6.tar.gz UpDPWD\_7to6 --exclude=logs --exclude=logs\_bak  --exclude=data

tar -xzvf  UpDPWD\_7to6.tar.gz

### sed 命令批量替换：

sed -i 's/./logs/itms\_oracle\/acs/g'  ./\*.conf

sed -i 's#${basePath}#'$BASE\_PATH'#g' $INSTALL\_PATH/$Mr\_SYSTEM\_DIR/\*/mr\_task/\*.xml

### 清空文件内容：

cat /dev/null >> xxx.txt

echo “” >xxx.txt

### df,du排序目录和文件：

du -h ./| sort -rn

du -sh /export/home/itms/ResourceBind\_LoadData\_2/logs/sql/\* | sort -rn

du -a / | sort -n -r | head -n 10

df   -h

df –hl

### sz,rz上传下载文件命令

sz –bye filename

rz

rz –bye

### curl提交数据post：

curl -**X** POST <http://172.16.158.7:3101/ServOpenCloud/cloud/callback> -d {"msg":"123456"}

curl -i -X POST -H "'Content-type':'application/json', 'charset':'utf-8', 'Accept': 'text/plain'" -d {"msg":"1234556678"} <http://172.16.158.7:3101/ServOpenCloud/cloud/callback>

### java运行jar中class文件：

[java -cp test.jar com.socket.CallWS "QueryGetWay" "http://134.96.131.39:8180/nbitktws/services/CallService?wsdl" "http://vpn.webservice.itms.nokia.com" ""](file:///C:\Users\jlp\AppData\Roaming\Microsoft\Word\java%20-cp%20test.jar%20com.socket.CallWS%20%22QueryGetWay%22%20%22http:\134.96.131.39:8180\nbitktws\services\CallService%3fwsdl%22%20%22http:\vpn.webservice.itms.nokia.com%22%20%22%22)

### 查看linux版本内核：

cat /proc/version

C://Users/jlp/AppData/Local/YNote/data/qq8A98E3F4D2B86891111E3E120B1645DA/744c385fb6ab41a3abcb4a206ee8e96a/ip_image002.jpeg

lsb\_release -a

### centos6.5设置ip地址

40 vi /etc/sysconfig/network-scripts/ifcfg-ens33

41 service network restart

42 systemctl status network.service

43 systemctl status network.service -l

44 systemctl stop NetworkManager

45 systemctl disable NetworkManager

### ll文件排序

ll -Sr 按大小升序排列

ll -Sl 按大小降序排列

### 显示Cpu一行的信息，每隔2秒更新一次，共五次，最后结果写入b.txt

top -d 2 -b -n 5|grep Cpu >> b.txt

top |grep Cpu

top |grep Mem

### 监控启用端口数，文件打开数，cpu和内存使用情况

#!/bin/sh

echo "开始监听进程所有端口 .......";

while [ 1 ]

do

allstatus=`netstat -napl`

SYN\_RCVD=`echo "$allstatus" |grep SYN\_RCVD |wc -l`

SYN\_SENT=`echo "$allstatus" |grep SYN\_SENT |wc -l`

ESTABLISHED=`echo "$allstatus" |grep ESTABLISHED |wc -l`

FIN\_WAIT\_1=`echo "$allstatus" |grep FIN\_WAIT\_1 |wc -l`

TIME\_WAIT=`echo "$allstatus" |grep TIME\_WAIT |wc -l`

CLOSE\_WAIT=`echo "$allstatus" |grep CLOSE\_WAIT |wc -l`

allcount=`echo "$allstatus" |wc -l`

files=`lsof |wc -l`

echo "allcount:$allcount,files:$files,SYN\_RCVD:$SYN\_RCVD,SYN\_SENT:$SYN\_SENT,ESTABLISHED:$ESTABLISHED,FIN\_WAIT\_1:$FIN\_WAIT\_1,TIME\_WAIT:$TIME\_WAIT,CLOSE\_WAIT:$CLOSE\_WAIT"

echo "allcount:$allcount,files:$files,SYN\_RCVD:$SYN\_RCVD,SYN\_SENT:$SYN\_SENT,ESTABLISHED:$ESTABLISHED,FIN\_WAIT\_1:$FIN\_WAIT\_1,TIME\_WAIT:$TIME\_WAIT,CLOSE\_WAIT:$CLOSE\_WAIT" >> stat.log

sleep 1

done;

### 根据关键字搜索并杀进程

kill -9 ·ps -ef|grep xxx |awk "{print $1}"

### fdisk,磁盘目录挂载

df –h

将磁盘/dev/sda3挂载到/data目录：mount -o noatime /dev/sda3 /data

### 输出格式化时间：

date "+%Y%m%d%H%M%S"

### 文件判断

-e filename 如果 filename存在，则为真 [ -e /var/log/syslog ]

-d filename 如果 filename为目录，则为真 [ -d /tmp/mydir ]

-f filename 如果 filename为常规文件，则为真 [ -f /usr/bin/grep ]

-L filename 如果 filename为符号链接，则为真 [ -L /usr/bin/grep ]

-r filename 如果 filename可读，则为真 [ -r /var/log/syslog ]

-w filename 如果 filename可写，则为真 [ -w /var/mytmp.txt ]

-x filename 如果 filename可执行，则为真 [ -L /usr/bin/grep ]

filename1-nt filename2 如果 filename1比 filename2新，则为真 [ /tmp/install/etc/services -nt /etc/services ]

filename1-ot filename2 如果 filename1比 filename2旧，则为真 [ /boot/bzImage -ot arch/i386/boot/bzImage ]

### 关闭防火墙

service iptables stop

### Alias使用

Vi /etc/bashrc

Alias vertica=””

Source /etc/bashrc

### 输出目录下所有文件名：

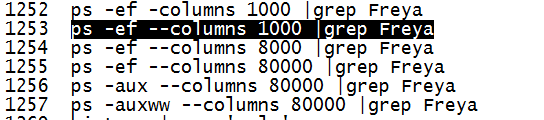
|  |
| --- |
| tasks=` ls -l ./sss/ns3.0/JOB\_UNITAG\_TAOBAO\_GMWP/conf/task/ |awk '{print $9}'`  for taskname in $tasks  do  echo "$taskname";  done |

### 用户权限管理

### 查看java进程的系统参数（比如classpath）

C:\Users\Administrator\Documents\fhnw Files\X4945@fhkj\images\bb5aa869-cf41-4dc8-96fc-92ba2c0fcdab.png

### Ps –ef的时候，控制长度



### 日志分组统计

|  |
| --- |
| grep 'UNITAG\_BASE' ./\*/Loki\_unitag.log\*| grep '\[com.fiberhome.unitag.client.msg' |awk '{print $1 "," $8}' | awk '{a[$1]++}END{for(i in a){print i,a[i]}}' |sort -ir |

### 变量值作为变量名

|  |
| --- |
| tempvname=`eval echo '$'"${Params[$i]}"` |

### 去重统计

cat ID\_NUM-\*.nb | awk '{print $4}' |sort -u |wc -l

### -Dlog4j.debug显示log4j配置文件路径

### 将目录关联到对应盘

mount /dev/vdb /pgsql

unmount /dev/vdb

fdisk –l

### 格式化某盘

mkfs.ext4 /dev/vdb

### 看占用磁盘的进程

fuser -m /dev/vdb

### sed查出文件中的第n行

|  |  |
| --- | --- |
| 第5行 | sed -n '5p' ID\_NUM-20190730-1564569407\_14.nb |
| 第1到30行 | sed -n '1,30p' ID\_NUM-20190730-1564569407\_14.nb |

### 性能分析

测试磁盘读写速度：

|  |
| --- |
| hdparm -Tt /dev/sdb  time dd if=/dev/sdb bs=1024 count=1000000 of=/1GB.file |

### 查找文件并删除

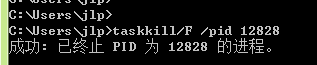
|  |
| --- |
| find ./mr -name task\_lib | xargs rm -rfv |

## Cmd命令

### 根据端口号查看进程

C://Users/jlp/AppData/Local/YNote/data/qq8A98E3F4D2B86891111E3E120B1645DA/d9ff66c836664e8086cd3057a4ce43d0/clipboard.png

### 强制杀进程



### 查看连接数

netstat –ano

netstat –ano|findstr “8080”

netstat -an | find /i /c "TIME\_WAIT"

这里/i参数是忽略大小写，/c参数是统计包含"TIME\_WAIT"字符串的行数或记录数。

### 杀死进程：

1>首先找到进程号对应的进程名称

tasklist|findstr 进程号

如:tasklist|findstr 3112

2>然后根据进程名称杀死进程

taskkill /f /t /im 进程名称

如:taskkill /f /t /im /javaw.exe

### 文件排序

dir/oS

### 创建递归目录

md \test\test1\test2…..\

### 查看内网所有ip和mac

arp –a

## 大数据hadoop

Hdfs和yarn命令行

### 问题记录

1.

EXECPTION:class org.apache.hadoop.ipc.RemoteException:File /tmp/hadoop-yarn/staging/fenghuo/.staging/job\_1556592094139\_1848/job.jar could only be replicated to 0 nodes instead of minReplication (=1). There are 3 datanode(s) running and no node(s) are excluded in this operation.

## ORACLE

### 对已有的表添加字段或更新字段：

alter table tab\_devicetype\_info add IS\_QOE NUMBER(2);

alter table tab\_devicetype\_info modify IS\_QOE NUMBER(2);

### 查看执行计划

set linesize 3000;

SET AUTOT ON;

set timing on;

SET AUTOTRACE TRACEONLY EXPLAIN;

### 查看索引状态

select INDEX\_NAME,STATUS  from user\_indexes where table\_name=upper('TAB\_VOIP\_SERV\_PARAM');

select INDEX\_NAME,TABLE\_NAME,STATUS  from user\_indexes where STATUS<>'VALID';

select index\_name,table\_name,tablespace\_name,status from dba\_indexes where status<>'VALID';

select COLUMN\_NAME from user\_ind\_columns  where index\_name=upper('mac\_tab\_gw\_device\_stbmac');

### 重建索引

alter index index\_name rebuild

### 查询CPU使用率高的SQL

select \* from (select sql\_text,sql\_id,cpu\_time from v$sql order by cpu\_time desc) where rownum<=10 order by  rownum asc ;

查看正在执行的SQL

select a.username, a.sid,b.SQL\_TEXT, b.SQL\_FULLTEXT from v$session a, v$sqlarea b where a.sql\_address = b.address;

select count(1) from v$session a, v$sqlarea b where a.sql\_address = b.address;

### 根据sid与serial# kill 需要中止的sql语句

select sql\_text, b.sid,b.serial#, 'alter system kill session'''||b.SID||','||b.serial#||''';',b.username

from v$process a, v$session b, v$sqlarea c

where a.addr = b.paddr

and b.sql\_hash\_value = c.hash\_value ;

### 查看被锁对象的序列号、sid

SELECT o.owner,o.object\_name,o.object\_type,s.sid,s.serial# FROM v$locked\_object l,dba\_objects o,v$session s WHERE l.object\_id=o.object\_id AND l.session\_id=s.sid ORDER BY o.object\_id,xidusn DESC

### 利用sid 和序列号删除

alter system kill session '3535,10855';

### sqlplus命令

sqlplus [gtmsmanager/mypassword123@//172.16.7.110:1521/gtms](mailto:gtmsmanager/mypassword123@//172.16.7.110:1521/gtms)

### user\_tables表，查询表信息

select table\_name from user\_tables where user\_name = 'GTMSMANAGER'

### exp，imp导入导出：

exp  用户名/密码@localhost:1521/orcl  file=D:/a.dmp

imp  用户名/密码@localhost:1521/orcl  file=D:/a.dmp

如果是sid那就

exp  用户名/密码@sid file=D:/a.dmp

imp  用户名/密码@sid  file=D:/a.dmp

## MySql

创建序号

|  |
| --- |
| select @rownumxxx:=@rownumxxx+1 as rowindex,a.\* from buslib\_bk\_info\_keyword\_sort a ,(select @rownumxxx:=0 ) b |

分组拼接

|  |
| --- |
| select labelcode ,labelname,GROUP\_CONCAT(DISTINCT inputpath ORDER BY inputpath SEPARATOR '\r\n') as inputpath from label\_combing GROUP BY labelcode |

### 角色创建和授权

#### 查询用户权限：

show grants for testjlp;

#### 用户分配权限：

GRANT ALL PRIVILEGES ON testjlp.\* to 'testjlp'@'%' identified by '1234';

#### 新建用户：

create user 'testjlp'@'localhost' identified by '1234';

#### 设置密码：

set password for 'root'@'localhost' = password('123456');

### 数据库备份：

备份：mysqldump -uroot -p123456 --database testjlp > testjlp.sql

导入：mysql -uroot -p123456 <testjlp.sql

备份（不带创建数据库）：mysqldump -uroot -p123456 testjlp > testjlp.sql

导入（不带创建数据库）：mysql -uroot -p123456 METADATE\_MM <testdatabase.sql

还原：mysql -uroot -p123456 <testjlp.sql

直接执行sql：mysql -uroot -p123456 -e "create database METADATE\_MM"

导入：mysql -uroot -p123456 METADATE\_MM <testdatabase.sql

### mysql连接MySQL数据库：

mysql -umysql -pailk#QAZ -h[172.16.158.54](qq://txfile/) -P8066

mysql –umysql –pailk#QAZ –h172.16.158.56 –P9066

### mysqldump 命令备份mysql表结构：

mysqldump --opt　-d　test　-uroot　-p　>　xxx.sql

mysqldump -uroot -pmysql -d clo\_msginfo> /home/devops/ai

mysqldump -uroot -pmysql -d > /home/devops/ai

### linux安装mysql

groupadd mysql

#添加用户mysql 到用户组mysql

useradd -g mysql mysql

apt-get install libaio1

apt-get install libaio-dev

cd /usr/local/mysql/<br>mkdir ./data/mysql

chown -R mysql:mysql ./

./scripts/mysql\_install\_db --user=mysql --datadir=/usr/local/mysql/data/mysql

cp support-files/mysql.server /etc/init.d/mysqld

chmod 755 /etc/init.d/mysqld

cp support-files/my-default.cnf /etc/my.cnf

#修改启动脚本

vi /etc/init.d/mysqld

#修改项：

basedir=/usr/local/mysql/

datadir=/usr/local/mysql/data/mysql

#启动服务

service mysqld start

#测试连接

./mysql/bin/mysql -uroot

#加入环境变量，编辑 /etc/profile，这样可以在任何地方用mysql命令了

export PATH=$PATH:/usr/local/mysql//bin<br>source /etc/profile

#启动mysql

service mysqld start

#关闭mysql

service mysqld stop

#查看运行状态

service mysqld status

### 远程下载mysql数据

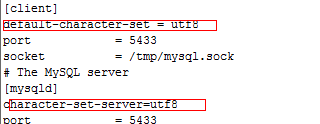
|  |
| --- |
| mysql -h$UNITAG\_IP -uUNITAG -pUNITAG --default-character-set=utf8 -e 'use UNITAG;SELECT LABELCODE ,LABELNAME FROM LABEL\_BASE\_INFO;' >LABELS.TXT |

### Msyql绝对秒转时间

***SELECT FROM\_UNIXTIME(1558152000,'%Y%m%d-%h:%i:%S') from dual***

### 中文乱码终极解决方案

vi my.cnf



show variables like ‘chara%’

### 解决root无法登陆问题

Vi /etc/my.cnf

在【mysqld】下添加一行

skip-grant-tables

然后重启mysqld服务

## Postgresql

|  |
| --- |
| SELECT \* from pg\_indexes where tablename like '%label\_dime\_attr%' ;  select \* from pg\_statio\_all\_indexes  select \* from pg\_stat\_user\_indexes;  alter table label\_dime\_attr\_xxx rebuild index index\_label\_dime\_attr\_xxx;  set session enable\_seqscan=false; |

## Sybase数据库

### Isql连接数据库

isql -Uitvtestmanager -Sldims -H192.168.4.74:4100 -Pmypassword123

## Vertica数据库

重启：su - dbadmin

Admintools -t start\_db -d zdrcenter -p vertica

登录：vsql -q -Udbadmin -h172.16.40.80 -dunitag –wvertica

重命名表：ALTER TABLE LABELATT\_DIME\_PERSON\_0802 RENAME TO LABELATT\_DIME\_PERSON\_08021

设置数据类型：ALTER TABLE LABELATT\_DIME\_PERSON ALTER COLUMN P60ADL00 set data type varchar(100);

## HBase Shell

## 前端

### Js获取当前根路径

var getRootPath = function getRootPath() {

        // 获取当前网址，如： <http://localhost:8083/uimcardprj/share/meun.jsp>

        var curWwwPath = window.document.location.href;

        // 获取主机地址之后的目录，如： uimcardprj/share/meun.jsp

        var pathName = window.document.location.pathname;

        var pos = curWwwPath.indexOf(pathName);

        // 获取主机地址，如： <http://localhost:8083>

        var localhostPaht = curWwwPath.substring(0, pos);

        // 获取带"/"的项目名，如：/uimcardprj

        var projectName = pathName.substring(0,

                pathName.substr(1).indexOf('/') + 1);

        return (localhostPaht + projectName);

    };

    var rootPath = getRootPath();

## 几个常常遇到的问题

### 中文乱码

### 跨域

### Session，cookie

### 浏览器缓存

### 正则

#### Eclipse下替换所有注释的步骤：

1."//      替换为    tihuanzhushi

2.://      替换为        lllkkkkjkjkjkjkjk

3.//.\*?$                  替换为空格

4./\\*{1,2}[\s\S]\*?\\*/     替换为空格

5.tihuanzhushi      替换为    "//

6.lllkkkkjkjkjkjkjk   替换为        ://

#### Eclipse下替换所有System.out.println();的步骤：

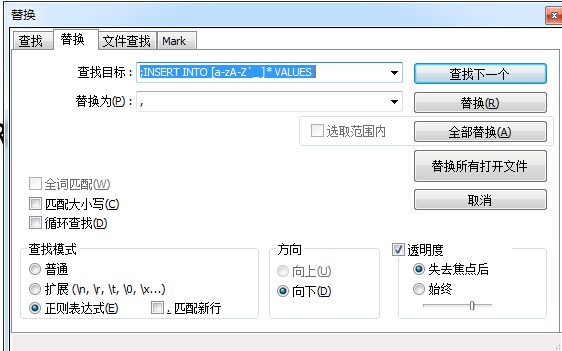
正则全局替换System.out.println([\u4E00-\u9FA5A-Za-z0-9\_"()+-].\*);为空

#### 分开的insert替换成合并的逗号隔开的insert

;

INSERT INTO [a-zA-Z`\_ ]\* VALUES

替换为逗号，如图



#### 常用正则

##### 一、校验数字的表达式

1 数字：^[0-9]\*$

2 n位的数字：^\d{n}$

3 至少n位的数字：^\d{n,}$

4 m-n位的数字：^\d{m,n}$

5 零和非零开头的数字：^(0|[1-9][0-9]\*)$

6 非零开头的最多带两位小数的数字：^([1-9][0-9]\*)+(.[0-9]{1,2})?$

7 带1-2位小数的正数或负数：^(\-)?\d+(\.\d{1,2})?$

8 正数、负数、和小数：^(\-|\+)?\d+(\.\d+)?$

9 有两位小数的正实数：^[0-9]+(.[0-9]{2})?$

10 有1~3位小数的正实数：^[0-9]+(.[0-9]{1,3})?$

11 非零的正整数：^[1-9]\d\*$ 或 ^([1-9][0-9]\*){1,3}$ 或 ^\+?[1-9][0-9]\*$

12 非零的负整数：^\-[1-9][]0-9"\*$ 或 ^-[1-9]\d\*$

13 非负整数：^\d+$ 或 ^[1-9]\d\*|0$

14 非正整数：^-[1-9]\d\*|0$ 或 ^((-\d+)|(0+))$

15 非负浮点数：^\d+(\.\d+)?$ 或 ^[1-9]\d\*\.\d\*|0\.\d\*[1-9]\d\*|0?\.0+|0$

16 非正浮点数：^((-\d+(\.\d+)?)|(0+(\.0+)?))$ 或 ^(-([1-9]\d\*\.\d\*|0\.\d\*[1-9]\d\*))|0?\.0+|0$

17 正浮点数：^[1-9]\d\*\.\d\*|0\.\d\*[1-9]\d\*$ 或 ^(([0-9]+\.[0-9]\*[1-9][0-9]\*)|([0-9]\*[1-9][0-9]\*\.[0-9]+)|([0-9]\*[1-9][0-9]\*))$

18 负浮点数：^-([1-9]\d\*\.\d\*|0\.\d\*[1-9]\d\*)$ 或 ^(-(([0-9]+\.[0-9]\*[1-9][0-9]\*)|([0-9]\*[1-9][0-9]\*\.[0-9]+)|([0-9]\*[1-9][0-9]\*)))$

19 浮点数：^(-?\d+)(\.\d+)?$ 或 ^-?([1-9]\d\*\.\d\*|0\.\d\*[1-9]\d\*|0?\.0+|0)$

##### 二、校验字符的表达式

1 汉字：^[\u4e00-\u9fa5]{0,}$

2 英文和数字：^[A-Za-z0-9]+$ 或 ^[A-Za-z0-9]{4,40}$

3 长度为3-20的所有字符：^.{3,20}$

4 由26个英文字母组成的字符串：^[A-Za-z]+$

5 由26个大写英文字母组成的字符串：^[A-Z]+$

6 由26个小写英文字母组成的字符串：^[a-z]+$

7 由数字和26个英文字母组成的字符串：^[A-Za-z0-9]+$

8 由数字、26个英文字母或者下划线组成的字符串：^\w+$ 或 ^\w{3,20}$

9 中文、英文、数字包括下划线：^[\u4E00-\u9FA5A-Za-z0-9\_]+$

10 中文、英文、数字但不包括下划线等符号：^[\u4E00-\u9FA5A-Za-z0-9]+$ 或 ^[\u4E00-\u9FA5A-Za-z0-9]{2,20}$

11 可以输入含有^%&',;=?$\"等字符：[^%&',;=?$\x22]+

12 禁止输入含有~的字符：[^~\x22]+

##### 三、特殊需求表达式

1 Email地址：^\w+([-+.]\w+)\*@\w+([-.]\w+)\*\.\w+([-.]\w+)\*$

2 域名：[a-zA-Z0-9][-a-zA-Z0-9]{0,62}(/.[a-zA-Z0-9][-a-zA-Z0-9]{0,62})+/.?

3 InternetURL：[a-zA-z]+://[^\s]\* 或 ^http://([\w-]+\.)+[\w-]+(/[\w-./?%&=]\*)?$

4 手机号码：^(13[0-9]|14[5|7]|15[0|1|2|3|5|6|7|8|9]|18[0|1|2|3|5|6|7|8|9])\d{8}$

5 电话号码("XXX-XXXXXXX"、"XXXX-XXXXXXXX"、"XXX-XXXXXXX"、"XXX-XXXXXXXX"、"XXXXXXX"和"XXXXXXXX)：^(\(\d{3,4}-)|\d{3.4}-)?\d{7,8}$

6 国内电话号码(0511-4405222、021-87888822)：\d{3}-\d{8}|\d{4}-\d{7}

7 身份证号(15位、18位数字)：^\d{15}|\d{18}$

8 短身份证号码(数字、字母x结尾)：^([0-9]){7,18}(x|X)?$ 或 ^\d{8,18}|[0-9x]{8,18}|[0-9X]{8,18}?$

9 帐号是否合法(字母开头，允许5-16字节，允许字母数字下划线)：^[a-zA-Z][a-zA-Z0-9\_]{4,15}$

10 密码(以字母开头，长度在6~18之间，只能包含字母、数字和下划线)：^[a-zA-Z]\w{5,17}$

11 强密码(必须包含大小写字母和数字的组合，不能使用特殊字符，长度在8-10之间)：^(?=.\*\d)(?=.\*[a-z])(?=.\*[A-Z]).{8,10}$

12 日期格式：^\d{4}-\d{1,2}-\d{1,2}

13 一年的12个月(01～09和1～12)：^(0?[1-9]|1[0-2])$

14 一个月的31天(01～09和1～31)：^((0?[1-9])|((1|2)[0-9])|30|31)$

15 钱的输入格式：

16 1.有四种钱的表示形式我们可以接受:"10000.00" 和 "10,000.00", 和没有 "分" 的 "10000" 和 "10,000"：^[1-9][0-9]\*$

17 2.这表示任意一个不以0开头的数字,但是,这也意味着一个字符"0"不通过,所以我们采用下面的形式：^(0|[1-9][0-9]\*)$

18 3.一个0或者一个不以0开头的数字.我们还可以允许开头有一个负号：^(0|-?[1-9][0-9]\*)$

19 4.这表示一个0或者一个可能为负的开头不为0的数字.让用户以0开头好了.把负号的也去掉,因为钱总不能是负的吧.下面我们要加的是说明可能的小数部分：^[0-9]+(.[0-9]+)?$

20 5.必须说明的是,小数点后面至少应该有1位数,所以"10."是不通过的,但是 "10" 和 "10.2" 是通过的：^[0-9]+(.[0-9]{2})?$

21 6.这样我们规定小数点后面必须有两位,如果你认为太苛刻了,可以这样：^[0-9]+(.[0-9]{1,2})?$

22 7.这样就允许用户只写一位小数.下面我们该考虑数字中的逗号了,我们可以这样：^[0-9]{1,3}(,[0-9]{3})\*(.[0-9]{1,2})?$

23 8.1到3个数字,后面跟着任意个 逗号+3个数字,逗号成为可选,而不是必须：^([0-9]+|[0-9]{1,3}(,[0-9]{3})\*)(.[0-9]{1,2})?$

24 备注：这就是最终结果了,别忘了"+"可以用"\*"替代如果你觉得空字符串也可以接受的话(奇怪,为什么?)最后,别忘了在用函数时去掉去掉那个反斜杠,一般的错误都在这里

25 xml文件：^([a-zA-Z]+-?)+[a-zA-Z0-9]+\\.[x|X][m|M][l|L]$

26 中文字符的正则表达式：[\u4e00-\u9fa5]

27 双字节字符：[^\x00-\xff] (包括汉字在内，可以用来计算字符串的长度(一个双字节字符长度计2，ASCII字符计1))

28 空白行的正则表达式：\n\s\*\r (可以用来删除空白行)

29 HTML标记的正则表达式：<(\S\*?)[^>]\*>.\*?</\1>|<.\*? /> (网上流传的版本太糟糕，上面这个也仅仅能部分，对于复杂的嵌套标记依旧无能为力)

30 首尾空白字符的正则表达式：^\s\*|\s\*$或(^\s\*)|(\s\*$) (可以用来删除行首行尾的空白字符(包括空格、制表符、换页符等等)，非常有用的表达式)

31 腾讯QQ号：[1-9][0-9]{4,} (腾讯QQ号从10000开始)

32 中国邮政编码：[1-9]\d{5}(?!\d) (中国邮政编码为6位数字) 33 IP地址：\d+\.\d+\.\d+\.\d+ (提取IP地址时有用) 34 IP地址：((?:(?:25[0-5]|2[0-4]\\d|[01]?\\d?\\d)\\.){3}(?:25[0-5]|2[0-4]\\d|[01]?\\d?\\d))

### 浏览器兼容性

ForEach不要使用，旧的浏览器不支持，改成for循环

## 常用工具表

### 正则字符含义表

|  |  |
| --- | --- |
| **字符** | **描述** |
| \ | 将下一个字符标记为一个特殊字符、或一个原义字符、或一个向后引用、或一个八进制转义符。例如，’n’ 匹配字符 “n”。’\n’ 匹配一个换行符。序列 ‘\\’ 匹配 “\” 而 “\(” 则匹配 “(”。 |
| ^ | 匹配输入字符串的开始位置。如果设置了 RegExp 对象的 Multiline 属性，^ 也匹配 ‘\n’ 或 ‘\r’ 之后的位置。 |
| $ | 匹配输入字符串的结束位置。如果设置了RegExp 对象的 Multiline 属性，$ 也匹配 ‘\n’ 或 ‘\r’ 之前的位置。 |
| \* | 匹配前面的子表达式零次或多次。例如，zo\* 能匹配 “z” 以及 “zoo”。\* 等价于{0,}。 |
| + | 匹配前面的子表达式一次或多次。例如，’zo+’ 能匹配 “zo” 以及 “zoo”，但不能匹配 “z”。+ 等价于 {1,}。 |
| ? | 匹配前面的子表达式零次或一次。例如，”do(es)?” 可以匹配 “do” 或 “does” 中的”do” 。? 等价于 {0,1}。 |
| {n} | n 是一个非负整数。匹配确定的 n 次。例如，’o{2}’ 不能匹配 “Bob” 中的 ‘o’，但是能匹配 “food” 中的两个 o。 |
| {n,} | n 是一个非负整数。至少匹配n 次。例如，’o{2,}’ 不能匹配 “Bob” 中的 ‘o’，但能匹配 “foooood” 中的所有 o。’o{1,}’ 等价于 ‘o+’。’o{0,}’ 则等价于 ‘o\*’。 |
| {n,m} | m 和 n 均为非负整数，其中n <= m。最少匹配 n 次且最多匹配 m 次。例如，”o{1,3}” 将匹配 “fooooood” 中的前三个 o。’o{0,1}’ 等价于 ‘o?’。请注意在逗号和两个数之间不能有空格。 |
| ? | 当该字符紧跟在任何一个其他限制符 (\*, +, ?, {n}, {n,}, {n,m}) 后面时，匹配模式是非贪婪的。非贪婪模式尽可能少的匹配所搜索的字符串，而默认的贪婪模式则尽可能多的匹配所搜索的字符串。例如，对于字符串 “oooo”，’o+?’ 将匹配单个 “o”，而 ‘o+’ 将匹配所有 ‘o’。 |
| . | 匹配除 “\n” 之外的任何单个字符。要匹配包括 ‘\n’ 在内的任何字符，请使用象 ‘[.\n]’ 的模式。 |
| (pattern) | 匹配 pattern 并获取这一匹配。所获取的匹配可以从产生的 Matches 集合得到，在VBScript 中使用 SubMatches 集合，在JScript 中则使用 $0…$9 属性。要匹配圆括号字符，请使用 ‘\(’ 或 ‘\)’。 |
| (?:pattern) | 匹配 pattern 但不获取匹配结果，也就是说这是一个非获取匹配，不进行存储供以后使用。这在使用 “或” 字符 (|) 来组合一个模式的各个部分是很有用。例如， ‘industr(?:y|ies) 就是一个比 ‘industry|industries’ 更简略的表达式。 |
| (?=pattern) | 正向预查，在任何匹配 pattern 的字符串开始处匹配查找字符串。这是一个非获取匹配，也就是说，该匹配不需要获取供以后使用。例如，’Windows (?=95|98|NT|2000)’ 能匹配 “Windows 2000″ 中的 “Windows” ，但不能匹配 “Windows 3.1″ 中的 “Windows”。预查不消耗字符，也就是说，在一个匹配发生后，在最后一次匹配之后立即开始下一次匹配的搜索，而不是从包含预查的字符之后开始。 |
| (?!pattern) | 负向预查，在任何不匹配 pattern 的字符串开始处匹配查找字符串。这是一个非获取匹配，也就是说，该匹配不需要获取供以后使用。例如’Windows (?!95|98|NT|2000)’ 能匹配 “Windows 3.1″ 中的 “Windows”，但不能匹配 “Windows 2000″ 中的 “Windows”。预查不消耗字符，也就是说，在一个匹配发生后，在最后一次匹配之后立即开始下一次匹配的搜索，而不是从包含预查的字符之后开始 |
| x|y | 匹配 x 或 y。例如，’z|food’ 能匹配 “z” 或 “food”。’(z|f)ood’ 则匹配 “zood” 或 “food”。 |
| [xyz] | 字符集合。匹配所包含的任意一个字符。例如， ‘[abc]’ 可以匹配 “plain” 中的 ‘a’。 |
| [^xyz] | 负值字符集合。匹配未包含的任意字符。例如， ‘[^abc]’ 可以匹配 “plain” 中的’p'。 |
| [a-z] | 字符范围。匹配指定范围内的任意字符。例如，’[a-z]’ 可以匹配 ‘a’ 到 ‘z’ 范围内的任意小写字母字符。 |
| [^a-z] | 负值字符范围。匹配任何不在指定范围内的任意字符。例如，’[^a-z]’ 可以匹配任何不在 ‘a’ 到 ‘z’ 范围内的任意字符。 |
| \b | 匹配一个单词边界，也就是指单词和空格间的位置。例如， ‘er\b’ 可以匹配”never” 中的 ‘er’，但不能匹配 “verb” 中的 ‘er’。 |
| \B | 匹配非单词边界。’er\B’ 能匹配 “verb” 中的 ‘er’，但不能匹配 “never” 中的 ‘er’。 |
| \cx | 匹配由 x 指明的控制字符。例如， \cM 匹配一个 Control-M 或回车符。x 的值必须为 A-Z 或 a-z 之一。否则，将 c 视为一个原义的 ‘c’ 字符。 |
| \d | 匹配一个数字字符。等价于 [0-9]。 |
| \D | 匹配一个非数字字符。等价于 [^0-9]。 |
| \f | 匹配一个换页符。等价于 \x0c 和 \cL。 |
| \n | 匹配一个换行符。等价于 \x0a 和 \cJ。 |
| \r | 匹配一个回车符。等价于 \x0d 和 \cM。 |
| \s | 匹配任何空白字符，包括空格、制表符、换页符等等。等价于 [ \f\n\r\t\v]。 |
| \S | 匹配任何非空白字符。等价于 [^ \f\n\r\t\v]。 |
| \t | 匹配一个制表符。等价于 \x09 和 \cI。 |
| \v | 匹配一个垂直制表符。等价于 \x0b 和 \cK。 |
| \w | 匹配包括下划线的任何单词字符。等价于’[A-Za-z0-9\_]’。 |
| \W | 匹配任何非单词字符。等价于 ‘[^A-Za-z0-9\_]’。 |
| \xn | 匹配 n，其中 n 为十六进制转义值。十六进制转义值必须为确定的两个数字长。例如，’\x41′ 匹配 “A”。’\x041′ 则等价于 ‘\x04′ & “1″。正则表达式中可以使用 ASCII 编码。. |
| \num | 匹配 num，其中 num 是一个正整数。对所获取的匹配的引用。例如，’(.)\1′ 匹配两个连续的相同字符。 |
| \n | 标识一个八进制转义值或一个向后引用。如果 \n 之前至少 n 个获取的子表达式，则 n 为向后引用。否则，如果 n 为八进制数字 (0-7)，则 n 为一个八进制转义值。 |
| \nm | 标识一个八进制转义值或一个向后引用。如果 \nm 之前至少有 nm 个获得子表达式，则 nm 为向后引用。如果 \nm 之前至少有 n 个获取，则 n 为一个后跟文字 m 的向后引用。如果前面的条件都不满足，若 n 和 m 均为八进制数字 (0-7)，则 \nm 将匹配八进制转义值 nm。 |
| \nml | 如果 n 为八进制数字 (0-3)，且 m 和 l 均为八进制数字 (0-7)，则匹配八进制转义值 nml。 |
| \un | 匹配 n，其中 n 是一个用四个十六进制数字表示的 Unicode 字符。例如， \u00A9 匹配版权符号 (?)。 |

**6. 部分例子**

|  |  |
| --- | --- |
| **正则表达式** | **说明** |
| /\b([a-z]+) \1\b/gi | 一个单词连续出现的位置 |
| /(\w+):\/\/([^/:]+)(:\d\*)?([^# ]\*)/ | 将一个URL解析为协议、域、端口及相对路径 |
| /^(?:Chapter|Section) [1-9][0-9]{0,1}$/ | 定位章节的位置 |
| /[-a-z]/ | A至z共26个字母再加一个-号。 |
| /ter\b/ | 可匹配chapter，而不能terminal |
| /\Bapt/ | 可匹配chapter，而不能aptitude |
| /Windows(?=95 |98 |NT )/ | 可匹配Windows95或Windows98或WindowsNT,当找到一个匹配后，从Windows后面开始进行下一次 |

### ASCII码表



### 逻辑运算

|  |
| --- |
| 与：and-> 有0出0，全1出1 例如：1 ，1-->1 1 ，0-->0 0 ，1-->0 0 ，0-->0  或：or-> 有1出1，全0出0 例如：1 ，1-->1 1 ，0-->1 0 ，1-->1 0 ，0-->0  非：not->有1出0，有0出1 例如：1 -->0 0-->1  与非：nand->先按与的操作，然后结果取反 例如：1 ，1-->0 1 ，0-->1 0 ，1-->1 0 ，0-->1  或非：nor->先按或的操作，然后结果取反 例如：1 ，1-->0 1 ，0-->0 0 ，1-->0 0 ，0-->1  异或：xor->相异为1，相同为0 例如：1 ，1-->0 1 ，0-->1 0 ，1-->1 0 ，0-->0  同或：xnor->相同为1，相异为0 例如：1 ，1-->1 1 ，0-->0 0 ，1-->0 0 ，0-->1 |

## Log4j配置项

配置样例

|  |
| --- |
| log4j.rootLogger=debug,cons,Console  log4j.logger.com.nebula.freya=info,LOGFILE  log4j.logger.com.fiberhome=info,JOB  log4j.logger.com.fh=info,JOB  #JOB任务日志  log4j.appender.JOB=org.apache.log4j.DailyRollingFileAppender  log4j.appender.JOB.file=./log/job.log  log4j.appender.JOB.layout=org.apache.log4j.PatternLayout  log4j.appender.JOB.layout.ConversionPattern=%d{yyyy-MM-dd HH\:mm\:ss} [%p] [%t] %l - %m%n  log4j.appender.JOB.level=debug  log4j.appender.JOB.MaxFileSize=100MB  log4j.appender.JOB.MaxBackupIndex=10  #freya框架日志  log4j.appender.LOGFILE=org.apache.log4j.DailyRollingFileAppender  log4j.appender.LOGFILE.file=./log/freya.log  log4j.appender.LOGFILE.layout=org.apache.log4j.PatternLayout  log4j.appender.LOGFILE.layout.ConversionPattern=%d{yyyy-MM-dd HH\:mm\:ss} [%p] [%t] %l - %m%n  log4j.appender.LOGFILE.level=debug  log4j.appender.LOGFILE.MaxFileSize=100MB  log4j.appender.LOGFILE.MaxBackupIndex=1  #root日志，看调试日志  log4j.appender.cons=org.apache.log4j.RollingFileAppender  log4j.appender.cons.file=./log/console.log  log4j.appender.cons.layout=org.apache.log4j.PatternLayout  log4j.appender.cons.layout.ConversionPattern=%d{yyyy-MM-dd HH\:mm\:ss} [%p] [%t] %l - %m%n  log4j.appender.cons.level=debug  log4j.appender.cons.MaxFileSize=50MB  log4j.appender.cons.MaxBackupIndex=1  #console  log4j.appender.Console=org.apache.log4j.ConsoleAppender  log4j.appender.Console.layout=org.apache.log4j.PatternLayout  log4j.appender.Console.layout.ConversionPattern=%-5p -%t - %c %m%n  log4j.appender.Console.Threshold=info |