## 问题

给你一个英文单词,将这个单词的中文意思翻译给我?比如:



我完全可以做得更加仔细、也更加方便。

本质上来说,这个问题的核心部分---语言转换---已经由百度等解决,而我只是调用他们的一个功能而已---实现的方式就是http请求和正则表达式.

## 基本知识

**http请求**

在java中,要获取一个网络上的资源文件,使用socket来工作.

需要注意的是,socket用于连接服务器而不是直接获取网页,获取网页是通过发送特定的http请求实现的.所以,总结如下:

java的socket(任何语言的socket)唯一的作用是连接服务器,而不是完成具体的功能(比如获取网页);

socket的单一作用性也确保了其扩展性;

连接socket时并不是使用直接的网页网址,虽然这样更加直接,但根本上将应当先连接服务器的ip

然后使用BufferedReader(获取文件)和PrintWriter(发送请求)同服务器进行交流

任何资源性的代码都应当符合模式:start do end;end部分是必不可少但也是容易忘记的

**基本结构**

**Socket s=new Socket(**www.baidu.com , 80);// connected **and ready**

**BufferedReader in….**

**PrintWriter out..**

**out.print(“GET / HTPP/1.1/r/nHost:** [www.baidu.com/r/n/r/n](http://www.baidu.com/r/n/r/n)**”);**

**out.flush();**

**while(in.readLine()!=null)print\_it;**

**close(in,out,socket);**

**正则表达式**

为了理解正则表达式的工作原理,首先明白一点,正则表达式是为了两个字符串之间的匹配,一个叫做模式串,一个叫做被匹配串;一个模式串可以匹配一批潜在的母串,但是需要预先的处理.

java区分了处理pattern和pattern匹配两个过程.首先,pattern被构造出来然后编译,然后,如果需要进行一次匹配,通过pattern.getMatcher来进行下一步.

上面只是步骤性的知识,并未触及正则表达式本身.实际上,由于正则表达式是如此的复杂因此在长时间的不用之后任何人都有可能将其遗忘.下面是对java正则表达式的一个简单总结:

1. 最简单的模式串莫过于一个平凡的串,比如使用ab来匹配”abort in abortion”
2. 然后如果比较 ababortion 和abababout tabagainst 会发现是对ab的重复.由此引出正则表达式对重复的处理以及组的概念:

(ab){n,m}重复次数 ()表明ab作为一组,{n,m}表明重复次数

如果匹配 ababababhano abab也许还能引出贪婪与否的概念.

{n,m}有以下特殊情况 \*任意多个 +一到多个 ?0或者1个 如果在重复的后面加上?就构成非贪婪

1. 组Group的概念,这有助于理解Matcher.group函数.
2. 开头和末尾 ^ $
3. 其他是一些暂时没有想到会自然引出的概念

z|food 匹配z或者food

[abc] 匹配a或b或c [^abc]不匹配abc \d \D

(?:pattern) 非捕获组

下面是一些例子:

整数或者小数：^[0-9]+\.{0,1}[0-9]{0,2}$

3 只能输入数字："^[0-9]\*$"。

4 只能输入n位的数字："^\d{n}$"。

5 只能输入至少n位的数字："^\d{n,}$"。

6 只能输入m~n位的数字：。"^\d{m,n}$"

7 只能输入零和非零开头的数字："^(0|[1-9][0-9]\*)$"。

8 只能输入有两位小数的正实数："^[0-9]+(.[0-9]{2})?$"。

9 只能输入有1~3位小数的正实数："^[0-9]+(.[0-9]{1,3})?$"。

10 只能输入非零的正整数："^\+?[1-9][0-9]\*$"。

11 只能输入非零的负整数："^\-[1-9][]0-9"\*$。

12 只能输入长度为3的字符："^.{3}$"。

13 只能输入由26个英文字母组成的字符串："^[A-Za-z]+$"。

14 只能输入由26个大写英文字母组成的字符串："^[A-Z]+$"。

15 只能输入由26个小写英文字母组成的字符串："^[a-z]+$"。

16 只能输入由数字和26个英文字母组成的字符串："^[A-Za-z0-9]+$"。

17 只能输入由数字、26个英文字母或者下划线组成的字符串："^\w+$"。

18 验证用户密码："^[a-zA-Z]\w{5,17}$"正确格式为：以字母开头，长度在6~18之间，只能包含字符、数字和下划线。

19 验证是否含有^%&',;=?$\"等字符："[^%&',;=?$\x22]+"。

20 只能输入汉字："^[\u4e00-\u9fa5]{0,}$"

21 验证Email地址："^\w+([-+.]\w+)\*@\w+([-.]\w+)\*\.\w+([-.]\w+)\*$"。

22 验证InternetURL："^http://([\w-]+\.)+[\w-]+(/[\w-./?%&=]\*)?$"。

23 验证电话号码："^(\(\d{3,4}-)|\d{3.4}-)?\d{7,8}$"正确格式为："XXX-XXXXXXX"、"XXXX-XXXXXXXX"、"XXX-XXXXXXX"、"XXX-XXXXXXXX"、"XXXXXXX"和"XXXXXXXX"。

24 验证身份证号（15位或18位数字）："^\d{15}|\d{18}$"。

25 验证一年的12个月："^(0?[1-9]|1[0-2])$"正确格式为："01"～"09"和"1"～"12"。

26 验证一个月的31天："^((0?[1-9])|((1|2)[0-9])|30|31)$"正确格式为；"01"～"09"和"1"～"31"。

27 匹配中文字符的正则表达式： [\u4e00-\u9fa5]

28 匹配双字节字符(包括汉字在内)：[^\x00-\xff]

29 应用：计算字符串的长度（一个双字节字符长度计2，ASCII字符计1）

30 String.prototype.len=function(){return this.replace(/[^\x00-\xff]/g,"aa").length;}

31 匹配空行的正则表达式：\n[\s| ]\*\r

32 匹配html标签的正则表达式：<(.\*)>(.\*)<\/(.\*)>|<(.\*)\/>

33 匹配首尾空格的正则表达式：(^\s\*)|(\s\*$)

# grep的问题和正则表达式分组

### grep的有效工具：使用—color选项

grep的问题在于不能处理多行数据，我们可以考虑把多行数据转化为单行数据再处理；

但是通常这涉及复杂的分组问题

## 正则表达式符号特例

# Bre的所有特殊字符都是用\转义，而Ere和Pre按正常使用

建议grep按照-P选项来使用

# 使用curl搜索百度的一个实例

1. www.baidu.com/s?wd=我的不是你的
2. <http://fanyi.baidu.com/transpage?query=http://www.eclipse.org/downloads&source=url&ie=utf8&from=en&to=zh&render=1> 这是网页翻译
3. curl --get 'http://m.youdao.com/translate?inputtext=egg&type=EN2ZH\_CN' -H 'User-Agent: Mozilla/5.0 (Linux; U; Android 4.4.4; Nexus 5 Build/KTU84P) AppleWebkit/534.30 (KHTML, like Gecko) Version/4.0 Mobile Safari/534.30'

曲线政策，使用手机版即可

query=what;

curl --get 'http://m.youdao.com/translate?inputtext='$query'&type=EN2ZH\_CN' -H 'User-Agent: Mozilla/5.0 (Linux; U; Android 4.4.4; Nexus 5 Build/KTU84P) AppleWebkit/534.30 (KHTML, like Gecko) Version/4.0 Mobile Safari/534.30' -o youdaoyes;cat youdaoyes|grep '<ul id="translateResult">' -C2|grep '<li>'

# 使用手机uA的程序

#!/bin/bash

query=$1;

curl -s --get 'http://m.youdao.com/translate?inputtext='$query'&type=EN2ZH\_CN' -H 'User-Agent: Mozilla/5.0 (Linux; U; Android 4.4.4; Nexus 5 Build/KTU84P) AppleWebkit/534.30 (KHTML, like Gecko) Version/4.0 Mobile Safari/534.30' -o ~/temp/youdaoyes;cat ~/temp/youdaoyes|grep '<ul id="translateResult">' -C2|grep '<li>'|cut -d'>' -f2|cut -d'<' -f1