正则表达式(re=regular expression)

通配符

- . 当前目录
- .. 当前目录的上一级目录
- * 0个或多个字符
- ？ 一个任意字符
- [[:digit:]]
- [[:space:]]

需求

- 找出某个字符串是否以conf结尾?

```
s.endswith("conf")
```

- 找出某个字符串是否以conf结尾， 以数字开头?

特殊的符号

一个完整的正则使用过程

In [11]:

```
import re

# 第一个参数是你正则的规则， 第二个参数是检测的字符串；
# 如果找到匹配， 则返回一个对象；
a =  re.match(r"westos", "westoshello")
print a.group()

# 注意: match方法是从左往右依次匹配的；
a =  re.match(r"westos", "hellowestoshello")
print a


# 如果没有找到匹配， 则返回None；
print re.match(r"hello", "westos")
```

```
westos
None
None
```

\d 单个数字

\D \d的取反 ， 除了数字之外

In [15]:

```
import re


a = re.match(r"\d", "1")
a.group()


a = re.match(r'\d', "w11")
print a


a = re.match(r"\D", "w11")
print a
```

```
None
<_sre.SRE_Match object at 0x2d96b90>
```

\s 匹配空格， \n, \t,\r

\S

In [18]:

```
a = re.match(r"\s", "\tw11")
print a


a = re.match(r"\S", "aa\tw11")
print a
```

```
<_sre.SRE_Match object at 0x2d96cc8>
<_sre.SRE_Match object at 0x2d96d30>
```

\w(word): 匹配字母， 数字或者下划线

\W:

In [3]:

```
import re


print re.match(r"\w", "whl2")


print re.match(r"\w", "_whl2")

print re.match(r"\w", "1whl2")
```

```
<_sre.SRE_Match object at 0x3aec5e0>
<_sre.SRE_Match object at 0x3aec5e0>
<_sre.SRE_Match object at 0x3aec718>
```

[ ]

- [0123456789] = \d = [0-9]
- [^0123456789] = \D = [^0-9]
- [a-zA-Z0-9_] = \w
- [^a-zA-Z0-9_] = \W
- [\n\t\r] = \s
- [^\n\t\r] = \s

表示数量

字符	功能
*	匹配前一个字符出现0次或无限次， 即可有可无, {0,}
+	匹配前一个字符出现1次或无限次，即至少出现一次, {1,}
?	匹配前一个字符出现1次或0次，即前面的字符可省略, {0,1}
{m}	匹配前一个字符出现m次
{m,}	匹配前一个字符至少出现m次
{m,n}	匹配前一个字符出现m次到n次

In [9]:

```
import re

# 代表的是a字符出现0次或者多次；
print re.match(r'a*', "hello")

a =  re.match(r"a*", " ")
a.group()
```

```
<_sre.SRE_Match object at 0x3c35a58>
```

Out[9]:

```
''
```

import re

print re.match(r'a+', "hello")

a = re.match(r'a+', "aaahello") print a.group()

应用1： 匹配电话号

In [15]:

```
import re


reg = r"010-?\d{8}$"
phones = ["010-1234567899", "01012345678", "010123"]


for i in phones:
    a = re.match(reg, i)
    if a:
        print a.group()
    else:
        print "%s 不合法" %(i)
```

```
010-1234567899 不合法
01012345678
010123 不合法
```

应用2：匹配字符串

- 需求: 匹配出， 字符串第一个字母为大写字母， 后面都是小写字母， 并且这些小写字母可有可无；

s = ["hello", "Hello", "hAll"]

In [10]:

```
import re

s = ["hello", "HelloH", "hAll", "A", "a"]
reg = r"[A-Z][a-z]*"

for i in s:
    a = re.match(reg, i)
    if a:
        # group方法只会打印出符和条件的内容;
        print "%s 合法" %(i)
    else:
        print "%s不合法 " %(i)
```

```
hello不合法
HelloH 合法
hAll不合法
A 合法
a不合法
```

表示边界

- ^: 以什么开头
- $: 以什么结尾

应用3： 匹配qq邮箱

- 找出列表中符和条件的邮箱地址, 并存储到/tmp/mail.txt文件中；
- 邮箱地址以@qq.com结尾；
- @qq.com前面的内容由字母，数字或者下划线组成， 但至少4位， 最多20位;

In [1]:

```
import re

mailList =["aa@qq.com", "westos@qq.com", "12fg5@westos.com", "a
           "1westos@qq.com", "1f2fg5@qq.com"]
# 判断邮件地址是否合法；
def ismailOK(mail_name):
    reg = r"\w{4,20}@qq.com$"
    reg = re.compile(reg)
    a = re.match(reg, mail_name)
    if a:
        return True
    else:
        return False


mailOkList = [i for i in mailList if ismailOK(i)]


# 将符和条件的邮件地址写入文件;
with open("/tmp/mail.txt", "a+") as f:
    for i in mailOkList:
        f.write(i+"\n")
        print i
        # f.writelines(li)
```

```
westos@qq.com
1westos@qq.com
1f2fg5@qq.com
```

表示分组

- | ： 匹配| 左右任意一个表达式即可；
- (ab): 将括号中的字符作为一个分组
- \num: 引用分组第num个匹配到的字符串
- (?P): 分组起别名
- (?P=name) : 引用分组的别名

In [70]:

```
import re

# westos或者hello
print re.match(r"westos|hello", "helloaaa")
print re.match(r"westos|hello", "westoshelloaaa")
```

```
<_sre.SRE_Match object at 0x34cad98>
<_sre.SRE_Match object at 0x34cad98>
```

应用4： 匹配出0-100之间的数字， 包括1和100

In [77]:

```
import re



# 1
# 2 3 4 5 6 7
# 11 12 13     20
# 21 22 23     30
# 100


reg = r"^0$|^100$|[1-9]\d?$"

re.match(reg, "0")
```

Out[77]:

```
<_sre.SRE_Match at 0x35332a0>
```

groups以元组方式返回符和条件的分组

In [105]:

```
import re

mailList =["aa@qq.com", "westos@qq.com", "12fg5@westos.com", "a
           "1westos@qq.com", "1f2fg5@qq.com"]
# 判断邮件地址是否合法；
def mail_user_name(mail_name):
    reg = r"(\w{4,20})@(qq.com)$"
    a = re.match(reg, mail_name)
    print a.groups()
    return a.group()


print mail_user_name("westos@qq.com")
```

```
('westos', 'qq.com')
westos@qq.com
```

In [111]:

```
# s = "<html><h1>westos</h1></html>"

# reg = r"<\w+><\w+>\w+</\w+></\w+>"

# print re.match(reg, s)



s = "<html><h1>westos</h1></html>"

reg = r"<(\w+)><(\w+)>(\w+)</\2></\1>"

a = re.match(reg, s)
a.group()
a.groups()
```

Out[111]:

```
('html', 'h1', 'westos')
```

In [158]:

```
s = "http://www.westos.org/jishu/book/helkl/westos"

reg = r'http://.+/(?P<first_content>\w+)/(?P<second_content>\w+

a=  re.match(reg, s)
print a
# a.group()
# a.group()
# a.groups()
# a.groupdict()
```

None

re高级用法

- search()方法: 只找到符和条件的第一个并返回；
- findall()方法： 返回符合条件的所有内容；
- sub()方法： 对符合正则的内容进行替换；
- split()方法： 指定多个分隔符进行分割；

In [171]:

```
import re


s = "阅读次数为1000, 转发次数为100"

reg = r"\d+"

# a =  re.search(reg, s)
# a.group()


# re.findall(reg, s)




# print re.sub(reg, '0' , s)


# #
# def addNum(x):
#     # a是整形
#     a = int(x.group()) + 1
#     return str(a)




# print re.sub(reg, addNum, s)
```

阅读次数为1001, 转发次数为101

In [172]:

```
s = "fentiao 18:18811112222"

print re.split(r":| ", s)
```

['fentiao', '18', '18811112222']

python贪婪和非贪婪

- 非贪婪模式， 总是匹配尽可能少的字符；
- *， ？，+， {m,n}后面加上?, 使得贪婪模式编程非贪婪模式;

In [181]:

```
s = "This is a number 111-234-22-456"


# 默认情况下python正则是贪婪模式的；
r = re.match(r".+(\d+-\d+-\d+-\d+)", s)
r.group(1)
```

Out[181]:

```
'1-234-22-456'
```

In [182]:

```
s = "This is a number 111-234-22-456"


# 默认情况下python正则是贪婪模式的；
r = re.match(r".+?(\d+-\d+-\d+-\d+)", s)
r.group(1)
```

Out[182]:

```
'111-234-22-456'
```

In [ ]:

```

```

JSON

json（javascript object）

In [194]:

```
import json

dic = {
    "service": "ftp",
    "port": 22,
    "service1": "ftp",
    "port1": 22
}

in_json =  json.dumps(dic ,indent=4)
print type(in_json)


print in_json

out_json = json.loads(in_json)
print type(out_json)
```

```
<type 'str'>
{
    "service1": "ftp",
    "port": 22,
    "service": "ftp",
    "port1": 22
}
<type 'dict'>
```

json应用案例: 获取你输入IP的所在地理位置；

In [ ]:

```

```

In [211]:

```
import urllib2
import json

ipadd = raw_input("IP:")

url = "http://freegeoip.net/json/%s"%(ipadd)

# 模拟浏览器访问指定链接
urlres = urllib2.urlopen(url)

res = urlres.read()
# print res


res_dic = json.loads(res)
# print res_dic
# print res_dic['ip']

print """
        查询结果显示:
    IP: {}
    Country: {}
    longitude:{}
    latitude: {}

""".format(res_dic['ip'], res_dic['country_name'], res_dic['lo
```

```
IP:106.45.6.7

        查询结果显示:
    IP: 106.45.6.7
    Country: China
    longitude:106.2731
    latitude: 38.4681
```

In [213]:

```
'hello {}   {}'.format(2,(1,2,3,4,5))
```

Out[213]:

```
'hello 2   (1, 2, 3, 4, 5)'
```