

# Reg Exp Talk 1



这个指引文档在 [知识共享署名-相同方式共享 3.0 协议](#) 之条款下提供

Available under [Creative Commons Attribution-ShareAlike License](#)

# 介绍

正则表达式，又称正规表示式、正规表示法、正规运算式、规则运算式、常规表示法（英语：Regular Expression，在代码中常简写为 regex、regexp 或 RE），是计算机科学的一个概念。正则表达式使用单个字符串来描述、匹配一系列符合某个句法规则的字符串。在很多文本编辑器里，正则表达式通常被用来检索、替换那些符合某个模式的文本。

来源：[维基百科](#)



# Python 中的使用方法

```
>>> import re
>>> re.findall('[0-9]', '1234')
['1', '2', '3', '4']
>>> help(re)      # 探索其他函数，如 search, match
>>>
```



# Bash 中的使用方法

```
$ grep '[0-9]' << FOE # grep 命令, FOE 表示结束
> 1234                 # stdin 输入内容
> HCC                  # 第二行
> py2exe               # 第三行
> FOE                  # stdin 结束
1234                   # grep 命令的输出
py2exe                 # 第二行
$
```



# Gedit 中的使用方法

- ▶ 用 `Ctrl-F` 打开查找
- ▶ 点击弹出框体左边的放大镜，选择“用正则表达式匹配”
- ▶ 输入表达式



# 中括号

- ▶ 中括号内一般可以表示一些字符区间，匹配一个字符
  - ▶ `[0-9a-zA-Z-]` 分别代表数字、小写字母、A 和-
- ▶ 例如

```
>>> re.findall('[0-9A-Z]', '12abAB')
```

```
['1', '2', 'A', 'B']
```

```
>>> re.findall('[1-9a]', '12abAB')
```

```
['1', '2', 'a']
```

```
>>>
```



## 加号和点

- ▶ 加号代表将前一个规则匹配多次，点表示任何字符

```
>>> re.findall('[1-9]+', '12ab45AB')  
['12', '45']  
>>> re.findall('.+', '1ab\na2\n43\n')  
['1ab', 'a2', '43']  
>>>
```



# 小括号

- ▶ 小括号在findall时代表需要得到的结果
  - ▶ 尝试：有多个小括号时会怎样？

```
>>> re.findall(r'http://[a-z\.]+/',  
... 'http://shiyiquan.net/club/hcc/')  
['http://shiyiquan.net/']  
>>> re.findall(r'http://([a-z\.]+)/',  
... 'http://shiyiquan.net/club/hcc/')  
['shiyiquan.net']  
>>>
```

- ▶ 也可以指定加号运算的优先级

```
>>> re.findall('((ab)+)', 'bababababa')  
[('abababab', 'ab')]  
>>> re.findall('(a(b)+)', 'babbbbbba')  
[('abbbbb', 'b')]  
>>>
```





# 大括号

## ► 大括号定义重复次数

```
>>> re.findall(r'[0-9]{3,4}',  
...           '123a1234a12345')  
['123', '1234', '1234']  
>>>
```



## ^ 和 \$

- ▶ ^表示匹配字符串的开始，\$表示匹配结束

```
>>> re.findall(r'^[0-9]', '123')
```

```
['1']
```

```
>>> re.findall(r'[0-9]$', '123')
```

```
['3']
```



## 实例 - urls.py

- 节选自 `shierquan` 项目的 `quan_account/urls.py`

```
urlpatterns = [  
    url(r'^signup/$', user_signup),  
    url(r'^login/$', user_login),  
    url(r'^club/([A-Za-z\-]+)/follow/$', follow),  
    url(r'^logout/$', user_logout),  
    url(r'^create/$', club_create),  
    url(r'^search/(user)/$', search),  
]
```



# 练习

- ▶ 打开附带的[RegExpTalk01.html](#)，完成练习。
- ▶ Exercise 19 提示：研究[确定有限状态自动机（DFA）](#)。编写题目要求的相应 DFA，并将其转换为正则表达式。



# 感谢参加此次活动

