

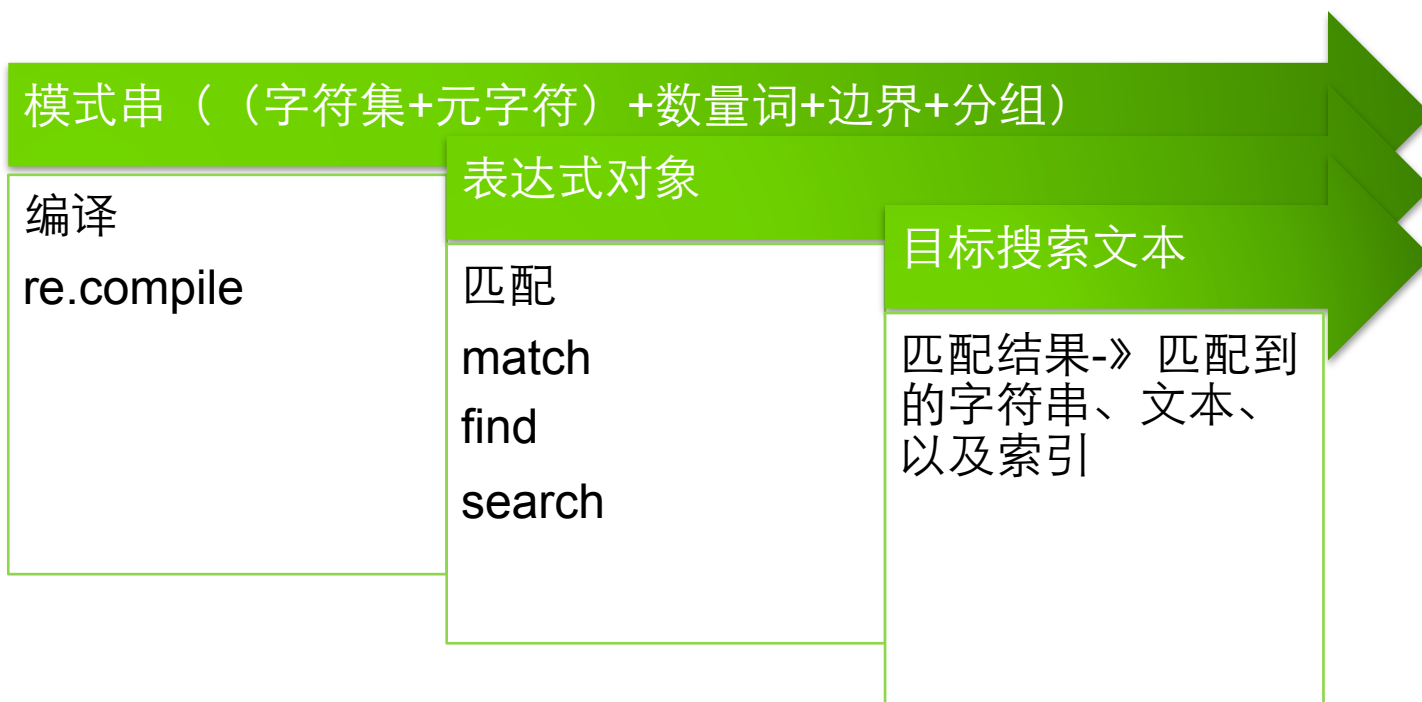
【声明】 本视频和幻灯片为炼数成金网络课程的教学资料，所有资料只能在课程内使用，不得在课程以外范围散播，违者将可能被追究法律和经济责任。

课程详情访问炼数成金培训网站

<http://edu.dataguru.cn>

第六周 python正则表达式

- 正则表达式的使用
- `re.compile()`
- 正则表达式介绍
- 贪婪模式和非贪婪模式
- `re`模块的其他工具



一般字符	
.	除换行符的任意字符 DOTALL
\	转义字符
[...]	字符集合
\d	数字: [0-9]
\D	非数字[^0-9]
\s	空白字符[<空格>\t\r\n\f\n]
\S	非空白字符[^\s]
\w	单词字符[A-Za-z0-9_]
\W	非单词字符[^\w]

*	匹配前一个字符0或者多次
+	匹配前一个字符1次或者多次
?	匹配前一个字符0次或者1次
{m}	匹配前一个字符m次
{m,n}	匹配前一个字符m至n次
数量词?	变成非贪婪模式

^	匹配字符串开头，多行匹配每一行开头
\$	匹配字符串末尾，多行匹配每一行末尾
\A	仅匹配字符串开头
\Z	仅匹配字符串末尾
\b	匹配\w 和 \W 之间

	左右表达式任意匹配一个 先匹配左边一旦成功则跳过匹配右边 如果 没有包含在()中，匹配整个正则表达式
(...)	分组匹配，从左到右，每遇到一个 (编号+1 分组后面可加数量词
(?P<name>...)	除了分组序号外，指定一个 name的别名
\<number>	引用编号为<number>的分组匹配到的字符串
(?P=name)	引用别名为<name>的分组匹配到的串

(?:...)	(...)不分组版本，用于使用 或者后接数量词
(?iLmsux)	iLmsux的每个字符代表一个匹配模式，只能用在正则表达式的开头，可选多个
(?#...)	#号后的内容将作为注释
(?=...)	之后的字符串内容需要匹配表达式才能成功匹配
(?!...)	之后的字符串不匹配表达式才能成功
(?<=...)	之前的字符串需要匹配表达式才能成功
(?<!...)	之前的字符串需要不匹配表达式才能成功
(?(id/name) yes no)	如果编号为id/名字为name的组匹配到字符串，则需要匹配yes，否则匹配no，no可以省略

I	re.I	忽略大小写
L	re.L	使用预定字符类 <code>\w \W \b \B \s \S</code> 取决当前区域设定
m	re.M	多行模式改变 <code>^</code> 和 <code>\$</code> 的行为
s	re.S	. 任意匹配模式
u	re.U	使用预定字符类 <code>\w \W \b \B \s \S \d \D</code> 取决unicode定义的字符属性
<u>x</u>	<u>re.X</u>	详细模式，可以多行，忽略空白字符，并且可以加入注释

■ 数量词的贪婪模式与非贪婪模式

正则表达式通常用于在文本中查找匹配的字符串。Python里数量词默认是贪婪的（在少数语言里也可能是默认非贪婪），总是尝试匹配尽可能多的字符；非贪婪的则相反，总是尝试匹配尽可能少的字符。例如：正则表达式`"ab*"`如果用于查找`"abbbbc"`，将找到`"abbbb"`。而如果使用非贪婪的数量词`"ab*?"`，将找到`"a"`。

- **re.compile(strPattern[, flag])**
- pattern
- match
- search
- **split**
- **findall**
- **finditer**
- **sub**

【声明】 本视频和幻灯片为炼数成金网络课程的教学资料，所有资料只能在课程内使用，不得在课程以外范围散播，违者将可能被追究法律和经济责任。

课程详情访问炼数成金培训网站

<http://edu.dataguru.cn>