V1: 6/21/21 🡪 6/25/21

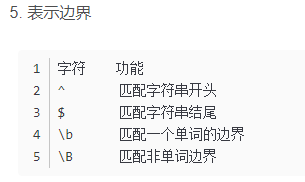
* Re.match vs re.search

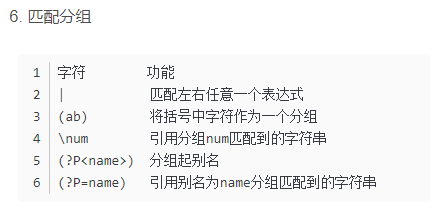


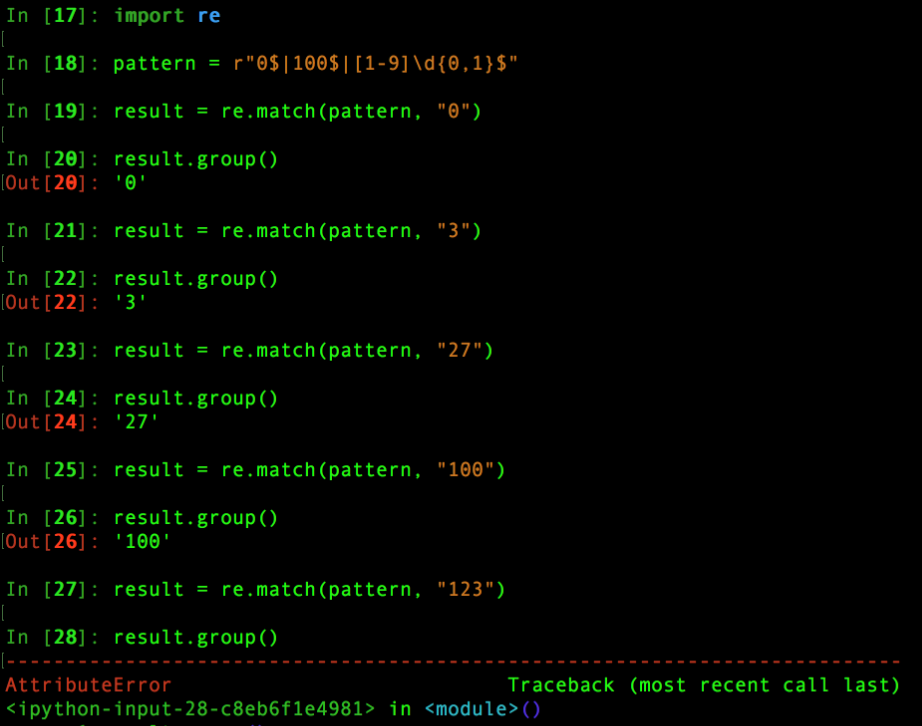


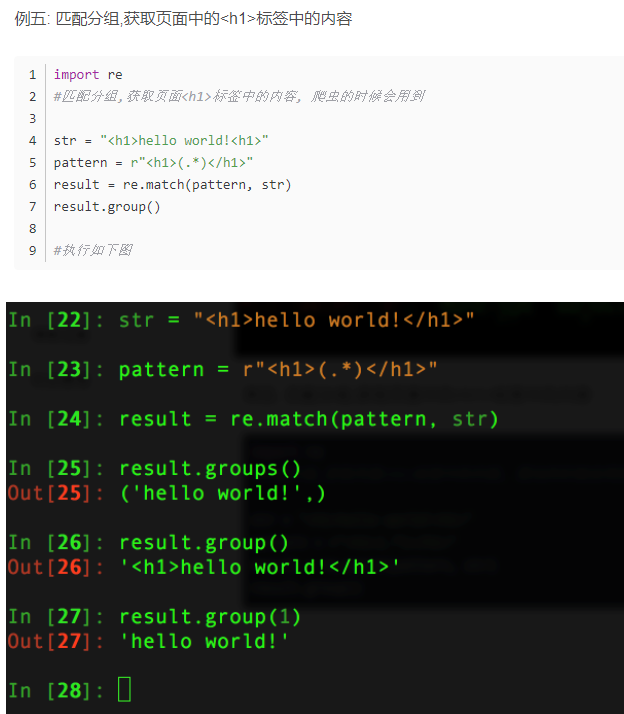












#引用分组,精确获取多个标签内的内容

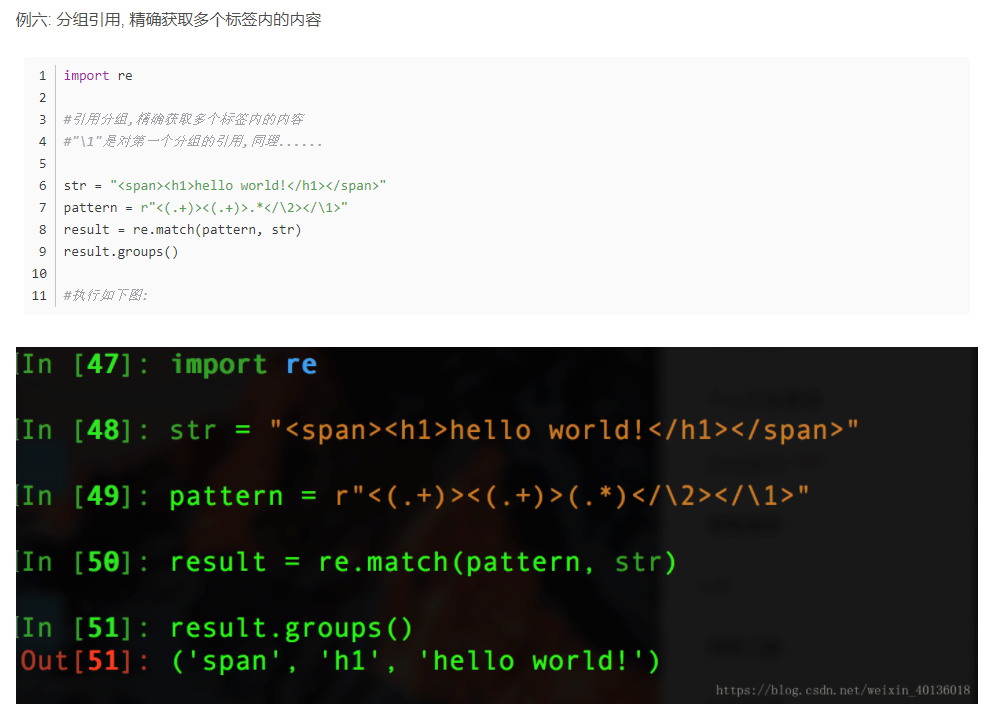
#"\1"是对第一个分组的引用,同理......

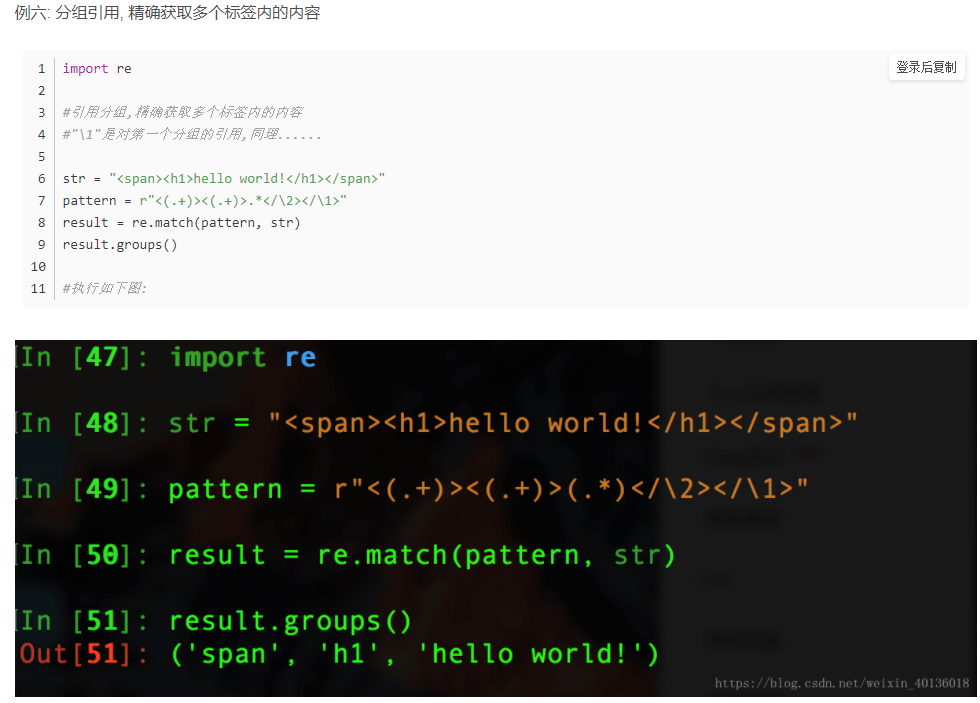
str = "<span><h1>hello world!</h1></span>"

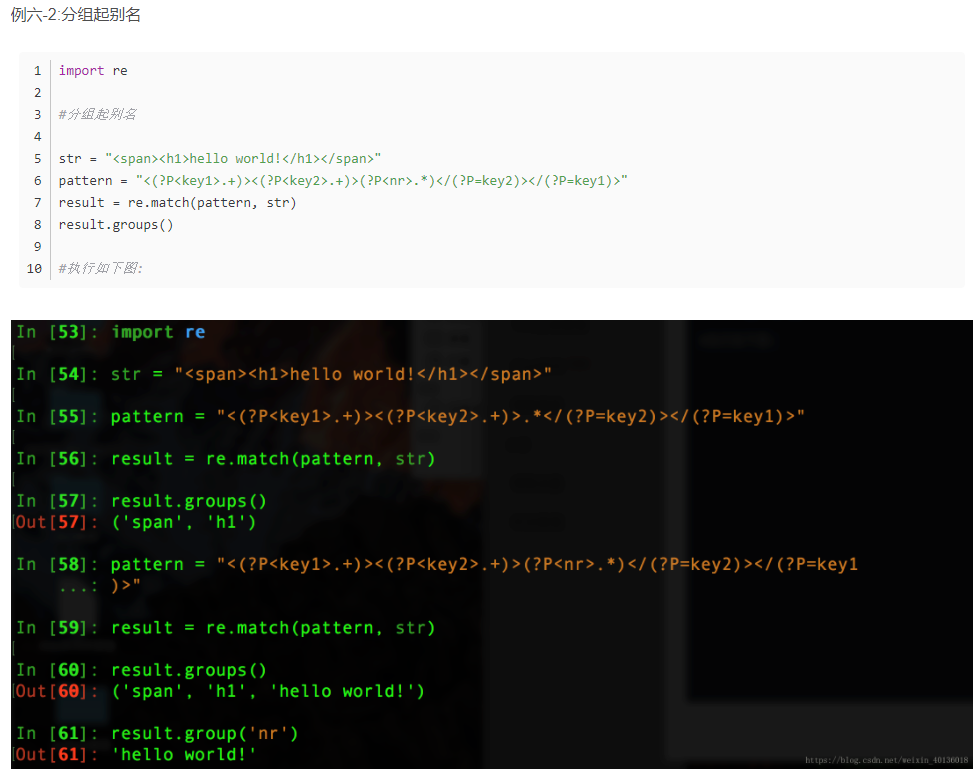
pattern = r"<(.+)><(.+)>.\*</\2></\1>"

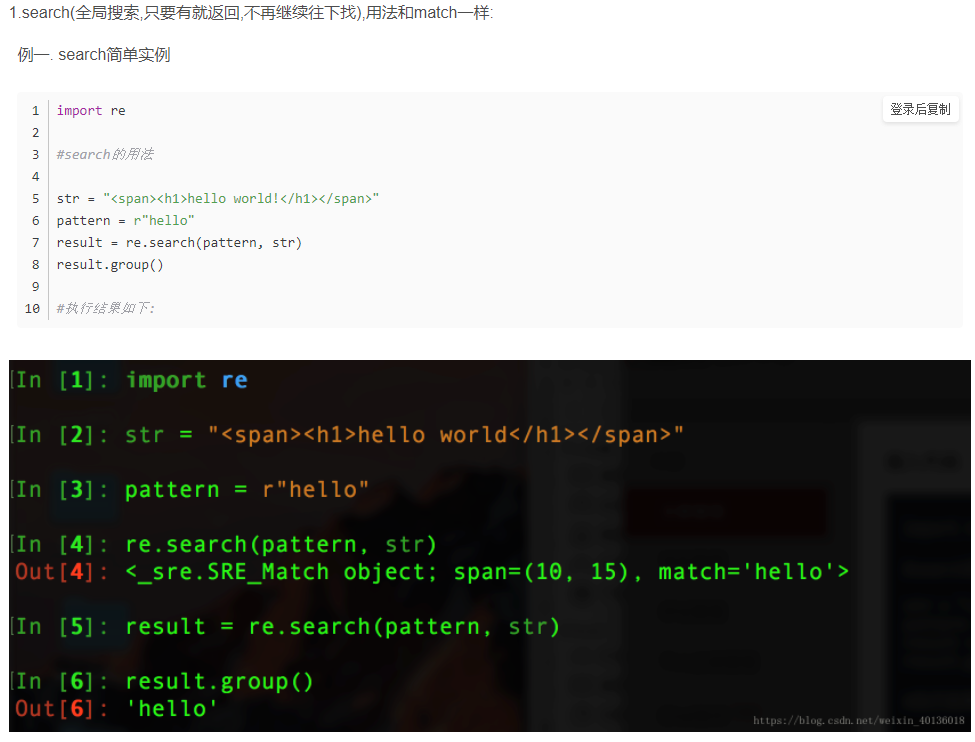
result = re.match(pattern, str)

result.groups()

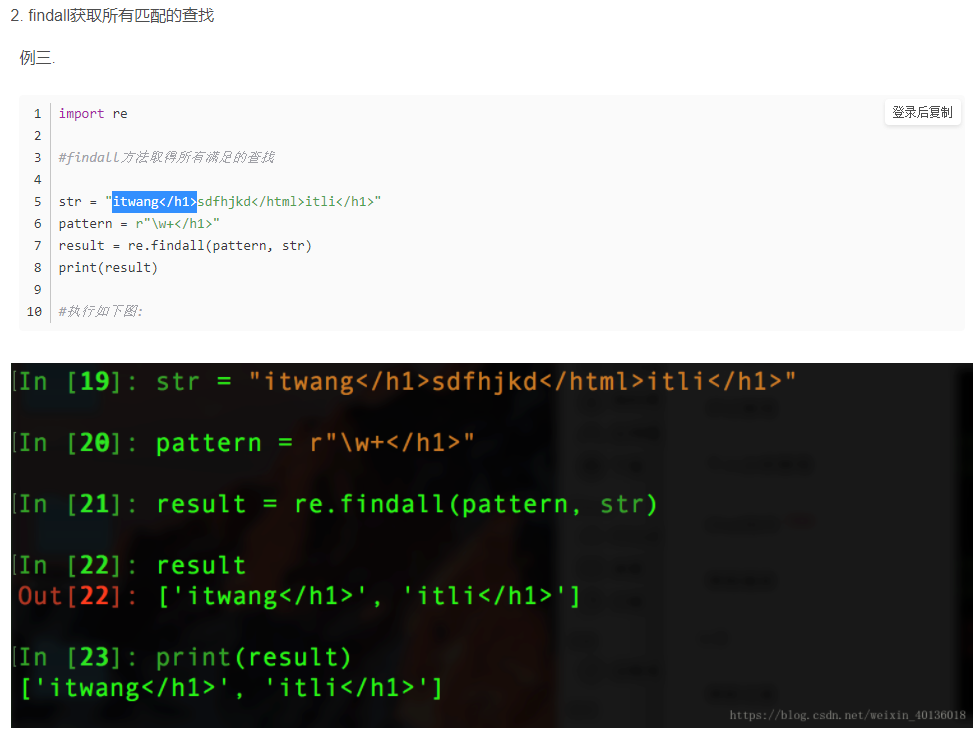


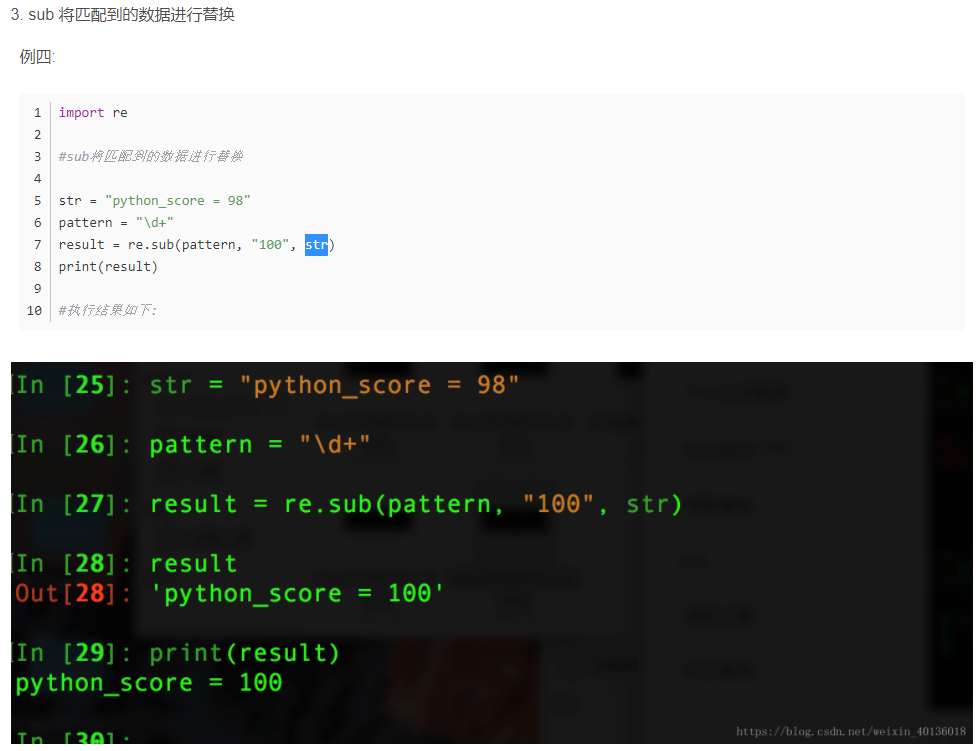


















# python使用正则表达式代码\_python使用正则表达式（Regular Expression）方法超详细

**https://blog.csdn.net/weixin\_39904116/article/details/114952547?utm\_medium=distribute.pc\_relevant.none-task-blog-baidujs\_title-9&spm=1001.2101.3001.4242**

1．点号“.”

**一个点号可以代替除了换行符(\n)以外的任何一个字符**，包括但不限于英文字母、数字、汉字、英文标点符号和中文标点符号。

2．星号“\*”

一个星号可以表示它前面的一个子表达式(普通字符、另一个或几个正则表达式符号)**0次到无限次**。

3．问号“?”

问号表示它前面的子表达式**0次或者1次**。注意，这里的问号是英文问号。

4．反斜杠“\”

**反斜杠在正则表达式里面不能单独使用**，甚至在整个Python里都不能单独使用。反斜杠需要和其他的字符配合使用来把特殊符号变成普通符号，把普通符号变成特殊符号。如：“\n”。

5．数字“\d”

正则表达式里面使用“\d”来表示一位数字。再次强调一下，“\d”虽然是由反斜杠和字母d构成的，但是要把“\d”看成一个正则表达式符号整体。

6．**小括号“()”**

小括号可以把括号里面的内容提取出来。

四、常见的正则表达式举例

1.  **.\*?**(匹配所有内容)

例如：'(.\*?)'   将网页的标题爬取下来。

2、\w 单词字符[A-Za-z0-9\_]， "+" 匹配前一个字符1次或无限次 例如：一个人的邮箱是这样的lixiaomei@qq.com，那么我们如何从一大堆的字符串把它提取出来呢？

pattern: \w+@\w+\.com

思考：若邮箱为hello123@heuet.edu.com,如何匹配？

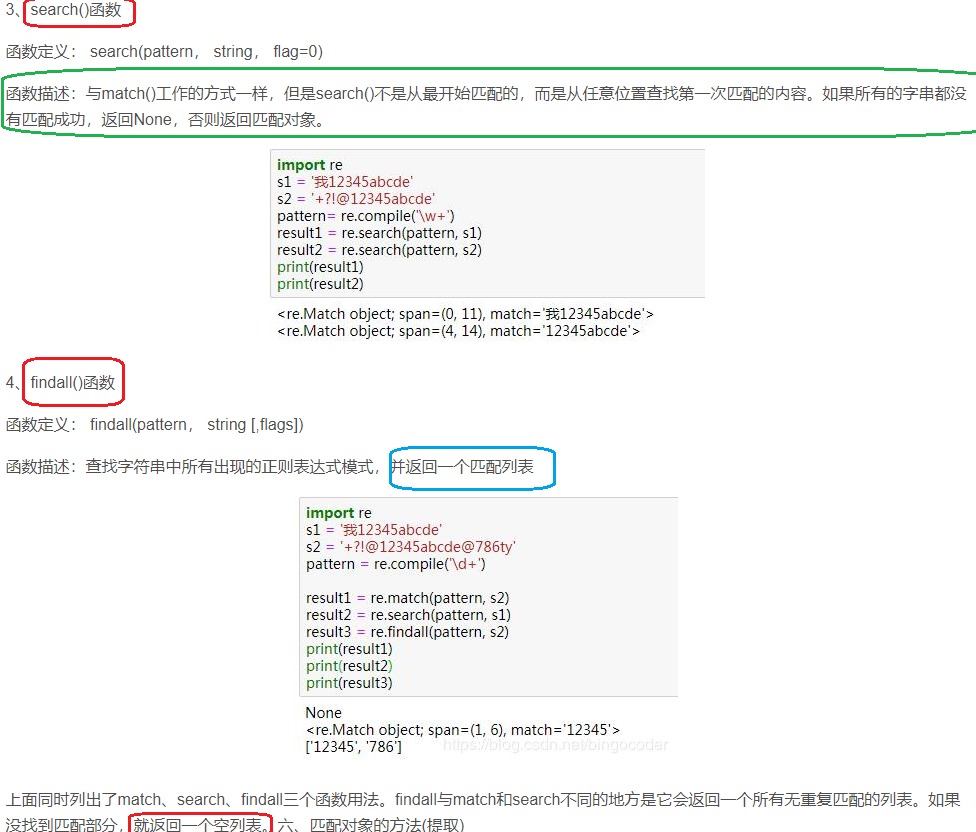
pattern:  **\w+**@**(\w+\.)?\w+\.com**

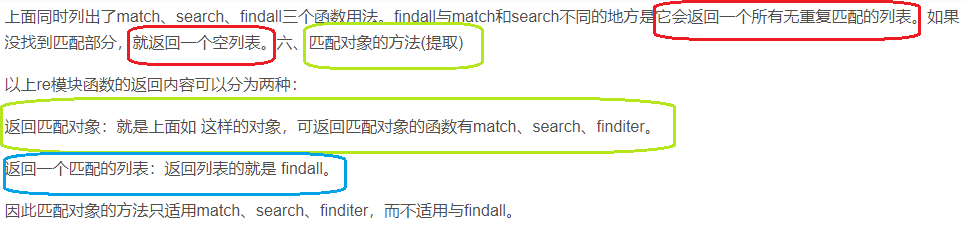
?代表了匹配0次或者1次括号分组内的匹配内容，**"()"则表示被括内容是一个分组，分组序号从pattern字符串起始往后依次排列**。因为是匹配0次或1次，那么就意味着括号内的部分是可有可无的，**所以这个pattern就可能匹配以上两种邮箱格式**。

扩展： **\w+**@ **(\w+\.)\*** \w+\.com 模式就更厉害了，" \* " 可以匹配0次或无限次。

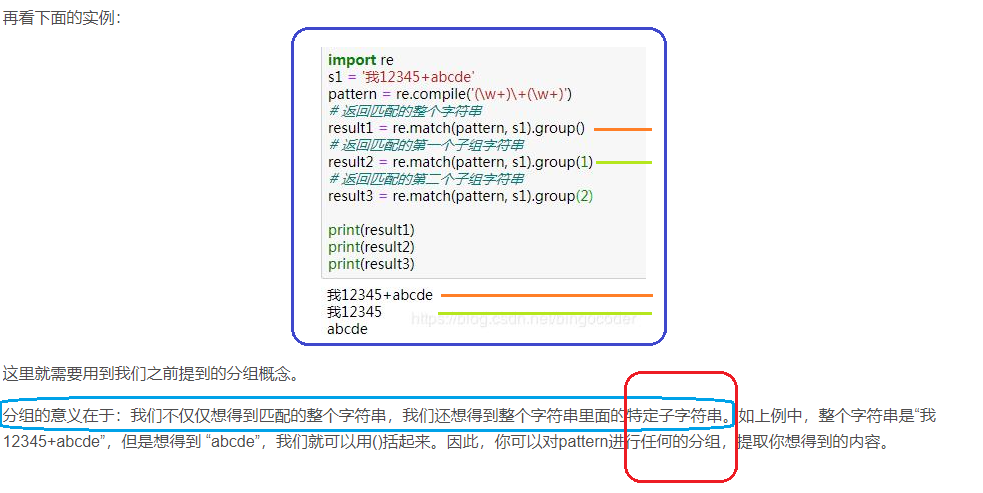


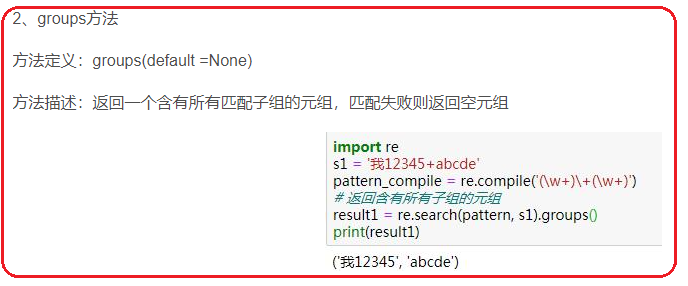








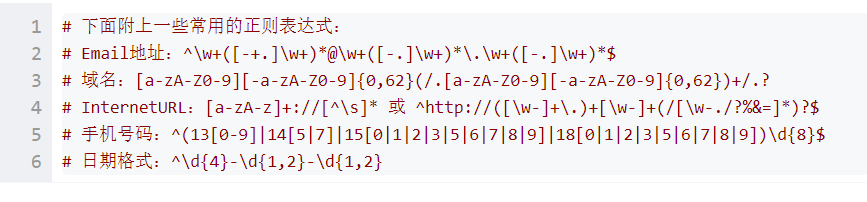












# 下面附上一些常用的正则表达式：

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

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

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

# 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}$

# 日期格式：^\d{4}-\d{1,2}-\d{1,2}