2.8 多行匹配模式¶

问题¶

你正在试着使用正则表达式去匹配一大块的文本，而你需要跨越多行去匹配。

解决方案¶

这个问题很典型的出现在当你用点(.)去匹配任意字符的时候，忘记了点(.)不能匹配换行符的事实。 比如，假设你想试着去匹配C语言分割的注释：

```
>>> comment = re.compile(r'/\*(.*?)\*/')
>>> text1 = '/* this is a comment */'
>>> text2 = '''/* this is a
... multiline comment */
... '''
>>>
>>> comment.findall(text1)
[' this is a comment ']
>>> comment.findall(text2)
[]
>>>
```

为了修正这个问题，你可以修改模式字符串，增加对换行的支持。比如：

```
>>> comment = re.compile(r'/\*((?:.|\n)*?)\*/')
>>> comment.findall(text2)
[' this is a\n multiline comment ']
>>>
```

在这个模式中， `(?:.|\n)` 指定了一个非捕获组 (也就是它定义了一个仅仅用来做匹配，而不能通过单独捕获或者编号的组)。

讨论¶

`re.compile()` 函数接受一个标志参数叫 `re.DOTALL` ，在这里非常有用。 它可以让正则表达式中的点(.)匹配包括换行符在内的任意字符。比如：

```
>>> comment = re.compile(r'/\*(.*?)\*/', re.DOTALL)
>>> comment.findall(text2)
[' this is a\n multiline comment ']
```

对于简单的情况使用 `re.DOTALL` 标记参数工作的很好， 但是如果模式非常复杂或者是为了构造字符串令牌而将多个模式合并起来(2.18节有详细描述)， 这时候使用这个标记参数就可能出现一些问题。 如果让你选择的话，最好还是定义自己的正则表达式模式，这样它可以在不需要额外的标记参数下也能工作的很好。