链滴

# 开发笔记：使用 Python 正则表达式模块时遇到的一个小问题

作者：StephenZhang

原文链接：https://ld246.com/article/1607599557212

来源网站：链滴

## Preface

今天在使用Python3内置的正则表达式模块，即re时遇到了一点小问题。因此做个记录分享出来。

## 正文

今天在处理某个配置文件时，需要对文件中一个不定长数组进行解析，提取部分参数。这个数组由n不定长的小数组组合而来，但它仍是一维的；而小数组的前三项无需关心，后面的奇数项不相同而偶项相同，而偶数项就是我需要的数据，并统计它们在小数组中的长度。示例如下：

```
arr1 = [1, 22, 56, 122, 1289, 34, 1289, 45, 1289] # (122, 34, 45), len = 3
arr2 = [2, 20, 0, 0, 1289,] # (0), len = 1
arr_to_handle = arr1 + arr2
```

其中小数组中的偶数项（n \ge 3)会随着文件不同而不同。

现在要提取我需要的数据，首先就是找到数据在哪。由于最后处理的数组的长度不定，因此不太方便靠索引处理，于是我打算将其转换成字符串之后，使用正则表达式匹配我需要的区间。

经过观察得知，由各个小数组组成的大数组总是具有这样的规律：

- 长度不确定，但所有元素都是十进制数字
- 数组的最后一项总是一个固定不变的数据
- 存在至少一个含有两个元素的子数组，子数组的偶次项是我需要的数据，奇次项是上一规律中不变数据

那么将所有的元素，使用一个独特的符号（例如/）连接起来，得到一个巨大的字符串，就可以使用则搜索了：

```
arr_to_handle = [1, 22, 56, 122, 1289, 34, 1289, 45, 1289, 2, 20, 0, 0, 1289]
```

arr2str = '/1/22/56/122/1289/34/1289/45/1289/2/20//0/0/1289'

根据arr2str的内容，可以轻松写出正则表达式r'/[+-]?\d+/1289/'，但是该式只能匹配到某一个子数，即便是使用re.findall()函数，也会丢失原本在小数组中的长度信息。我期望的匹配到的是/122/1289/34/1289/45/1289和/0/1289，这样字符串稍加处理即可还原长度信息。

那么对正则表达式进行改进，需要刚才示例中的pattern出现至少一次，那么新的正则表达式可以写作r'(/[+-]?\d+/1289/)+'，现在把新的pattern传递给re.findall()时，神奇的事情发生了，该函数只返回了['/45/1289', '/0/1289']，显然这不是我所期待的结果。

经过查询资料，最后发现这并不是re.findall()函数的错误，而我的表达式也基本正确——但是括号内group被过早的捕获了，以至于无法返回正确结果，但是它的原理是正确的。详情参见issue 42448。

因此只需在匹配过程中，禁止提前捕获即可，所以只需将正则表达式修改为r'(?:/[+-]?\d+/1289/)+'可，其中?:表示在当前圆括号的作用域中禁用捕获。

## 总结

写代码也不能望文生义呀，今天被这个bug折腾了一天，最后终于圆满解决，唉！