2.1 使用多个界定符分割字符串¶

问题¶

你需要将一个字符串分割为多个字段，但是分隔符(还有周围的空格)并不是固定的。

解决方案¶

`string` 对象的 `split()` 方法只适应于非常简单的字符串分割情形， 它并不允许有多个分隔符或者是分隔符周围不确定的空格。 当你需要更加灵活的切割字符串的时候，最好使用 `re.split()` 方法：

```
>>> line = 'asdf fjdk; afed, fjek,asdf, foo'
>>> import re
>>> re.split(r'[;,\s]\s*', line)
['asdf', 'fjdk', 'afed', 'fjek', 'asdf', 'foo']
```

讨论¶

函数 `re.split()` 是非常实用的，因为它允许你为分隔符指定多个正则模式。 比如，在上面的例子中，分隔符可以是逗号，分号或者是空格，并且后面紧跟着任意个的空格。 只要这个模式被找到，那么匹配的分隔符两边的实体都会被当成是结果中的元素返回。 返回结果为一个字段列表，这个跟 `str.split()` 返回值类型是一样的。

当你使用 `re.split()` 函数时候，需要特别注意的是正则表达式中是否包含一个括号捕获分组。 如果使用了捕获分组，那么被匹配的文本也将出现在结果列表中。比如，观察一下这段代码运行后的结果：

```
>>> fields = re.split(r'(;|,|\s)\s*', line)
>>> fields
['asdf', ' ', 'fjdk', ';', 'afed', ',', 'fjek', ',', 'asdf', ',', 'foo']
>>>
```

获取分割字符在某些情况下也是有用的。 比如，你可能想保留分割字符串，用来在后面重新构造一个新的输出字符串：

```
>>> values = fields[::2]
>>> delimiters = fields[1::2] + ['']
>>> values
['asdf', 'fjdk', 'afed', 'fjek', 'asdf', 'foo']
>>> delimiters
[' ', ';', ',', ',', ',', '']
>>> # Reform the line using the same delimiters
>>> ''.join(v+d for v,d in zip(values, delimiters))
'asdf fjdk;afed,fjek,asdf,foo'
>>>
```

如果你不想保留分割字符串到结果列表中去，但仍然需要使用到括号来分组正则表达式的话， 确保你的分组是非捕获分组，形如 `(?:...)` 。比如：

```
>>> re.split(r'(?:,|;|\s)\s*', line)
['asdf', 'fjdk', 'afed', 'fjek', 'asdf', 'foo']
>>>
```