2.11 删除字符串中不需要的字符¶

问题¶

你想去掉文本字符串开头，结尾或者中间不想要的字符，比如空白。

解决方案¶

`strip()` 方法能用于删除开始或结尾的字符。 `lstrip()` 和 `rstrip()` 分别从左和从右执行删除操作。 默认情况下，这些方法会去除空白字符，但是你也可以指定其他字符。比如：

```
>>> # Whitespace stripping
>>> s = ' hello world \n'
>>> s.strip()
'hello world'
>>> s.lstrip()
'hello world \n'
>>> s.rstrip()
' hello world'
>>>
>>> # Character stripping
>>> t = '-----hello====='
>>> t.lstrip('-')
'hello====='
>>> t.strip('-=')
'hello'
>>>
```

讨论¶

这些 `strip()` 方法在读取和清理数据以备后续处理的时候是经常会被用到的。 比如，你可以用它们来去掉空格，引号和完成其他任务。

但是需要注意的是去除操作不会对字符串的中间的文本产生任何影响。比如：

```
>>> s = ' hello     world \n'
>>> s = s.strip()
>>> s
'hello     world'
>>>
```

如果你想处理中间的空格，那么你需要求助其他技术。比如使用 `replace()` 方法或者是用正则表达式替换。示例如下：

```
>>> s.replace(' ', '')
'helloworld'
>>> import re
>>> re.sub('\s+', ' ', s)
'hello world'
>>>
```

通常情况下你想将字符串 `strip` 操作和其他迭代操作相结合，比如从文件中读取多行数据。 如果是这样的话，那么生成器表达式就可以大显身手了。比如：

```
with open(filename) as f:
    lines = (line.strip() for line in f)
    for line in lines:
        print(line)
```

在这里，表达式 `lines = (line.strip() for line in f)` 执行数据转换操作。 这种方式非常高效，因为它不需要预先读取所有数据放到一个临时的列表中去。 它仅仅只是创建一个生成器，并且每次返回行之前会先执行 `strip` 操作。

对于更高阶的strip，你可能需要使用 `translate()` 方法。请参阅下一节了解更多关于字符串清理的内容。