

《Python 核心编程（第三版）》读书笔记

作者: [Leif160519](#)

原文链接: <https://ld246.com/article/1571283397023>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

<p>
</p>

<p>下载链接: https://leif.fun/downloads/books/python/Python 核心编程 (第 3 版) PDF 高清晰完整中文版.pdf</p>

<p>百度云: 链接:https://pan.baidu.com/s/1L3ij8eefhZW0XqnRSvztQA 密码:ypyh</p>

<p>花了 2.36 软妹币买的。。。</p>

<h2 id="2019年10月17日">2019 年 10 月 17 日</h2>

<h2 id="第一部分-通用应用主题">第一部分 通用应用主题</h2>

<h2 id="第1章-正则表达式">第 1 章 正则表达式</h2>

<h3 id="1-2-特殊符号和字符">1.2 特殊符号和字符</h3>

表示法	描述	正则表达式示例
符号		
literal	匹配文本字符串的字面值 literal	foo
re1 re2	匹配正则表达式 re1 或者 re2	foo bar
.	匹配任何字符(除了\n 之外)	b.b
^	匹配字符串起始部分	Dear
\$	匹配字符串终止部分	/bin/*sh\$

```
<td>*</td>
<td>匹配 0 次或者多次前面出现的正则表达式</td>
<td>[A-Za-z0-9]*</td>
</tr>
<tr>
<td>+</td>
<td>匹配 1 次或者多次前面出现的正则表达式</td>
<td>[a-z]+\.\com</td>
</tr>
<tr>
<td>?</td>
<td>匹配 0 次或者 1 次前面出现的正则表达式</td>
<td>goo?</td>
</tr>
<tr>
<td>{N}</td>
<td>匹配 N 次前面出现的正则表达式</td>
<td>[0-9]{3}</td>
</tr>
<tr>
<td>{M,N}</td>
<td>匹配 M~N 次前面出现的正则表达式</td>
<td>[0-9]{5,9}</td>
</tr>
<tr>
<td>[...]</td>
<td>匹配来自字符集的任意单一字符</td>
<td>[aeiou]</td>
</tr>
<tr>
<td>[..x-y..]</td>
<td>匹配 x~y 范围中的任意单一字符</td>
<td>[0-9], [A-Za-z]</td>
</tr>
<tr>
<td>[^...]</td>
<td>不匹配此字符集中出现的任何一个字符，包括某一范围的字符(如果在此字符集中出现)</td>
<td>[^aeiou],[^A-Za-z0-9]</td>
</tr>
<tr>
<td>(*|+|?|{})?</td>
<td>用于匹配上面频繁出现/重复出现符号的非贪婪版本(*、+、?、{})</td>
<td>.*?[a-z]</td>
</tr>
<tr>
<td>(..)</td>
<td>匹配封闭的正则表达式，然后另存为子组</td>
<td>([0-9]{3})?,f(oo|u)bar</td>
</tr>
<tr>
<td>特殊字符</td>
<td></td>
<td></td>
</tr>
```

```
<tr>
<td>\d</td>
<td>匹配任何十进制数字，与[0-9]一致(\D 与\d 相反，不匹配任何非数值型的数字)</td>
<td>data\d+.txt</td>
</tr>
<tr>
<td>\w</td>
<td>匹配任何字母数字字符，与[A-Za-z0-9_]相同(\W 与之相反)</td>
<td>[A-Za-z_]\w+</td>
</tr>
<tr>
<td>\s</td>
<td>匹配任何空格字符，与[\n\t\r\v\f]相同(\S 与之相反)</td>
<td>of\sthe</td>
</tr>
<tr>
<td>\b</td>
<td>匹配任何单词边界(\B 与之相反)</td>
<td>\bThe\b</td>
</tr>
<tr>
<td>\N</td>
<td>匹配已保存的子组 N(参见上面的(...))</td>
<td>price: \16</td>
</tr>
<tr>
<td>\c</td>
<td>逐字匹配任何特殊字符 c(即，仅按照字面意义匹配，不匹配特殊含义)</td>
<td>\., \, \*</td>
</tr>
<tr>
<td>\A(\Z)</td>
<td>匹配字符串的起始(结束)(另见上面介绍的 ^ 和 $)</td>
<td>\ADear</td>
</tr>
<tr>
<td>扩展表示法</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(?!Lmsux)</td>
<td>在正则表达式中嵌入一个或者多个特殊“标记”参数(或者通过函数/方法)</td>
<td>(?x), (? im)</td>
</tr>
<tr>
<td>(?:...)</td>
<td>表示一个匹配不用保存的分组</td>
<td>(?:\w+\.)*</td>
</tr>
<tr>
<td>(?P<name>...)</td>
<td>像一个仅由 name 标识而不是数字 ID 标识的正则分组匹配</td>
<td>(?P<data>)</td>
```

```
</tr>
<tr>
<td>(?P=name)</td>
<td>在同一字符串中匹配由(?P<<name>>)分组的之前文本</td>
<td>(?P=data)</td>
</tr>
<tr>
<td>(?#...)</td>
<td>表示注释，所有内容都被忽略</td>
<td>(?#comment)</td>
</tr>
<tr>
<td>(?=...)</td>
<td>匹配条件是如果...出现在之后的位置，而不使用输入字符串;称作正向前视断言</td>
<td>(?=.com)</td>
</tr>
<tr>
<td>(?!=...)</td>
<td>匹配条件是如果...不出现在之后的位置，而不使用输入字符串;称作负向前视断言</td>
<td>(?!.net)</td>
</tr>
<tr>
<td>(?&lt;=...)</td>
<td>匹配条件是如果...出现在之前的位置，而不使用输入字符串;称作正向后视断言</td>
<td>(?&lt;=800-)</td>
</tr>
<tr>
<td>(?&lt;!...)</td>
<td>匹配条件是如果...不出现在之前的位置，而不使用输入字符串;称作负向后视断言</td>
<td>(?&lt;!192\.168\.)</td>
</tr>
<tr>
<td>(?(id/name)Y|N )</td>
<td>如果分组所提供的 id 或者 name(名称)存在，就返回正则表达式的条件匹配 Y，如果不存在，返回 N;N 是可选项</td>
<td>(?(1)y|x)</td>
</tr>
</tbody>
</table>
```