

运算符优先级

正则表达式从左到右进行计算，并遵循优先级顺序，这与算术表达式非常类似。

相同优先级的从左到右进行运算，不同优先级的运算先高后低。各种正则表达式运算符的优先级从最高到最低顺序：

序号	字符	含义
1	<code>\</code>	转义符
2	<code>()</code> , <code>(?:)</code> , <code>(?=)</code> , <code>[]</code>	圆括号和方括号
3	<code>*</code> , <code>+</code> , <code>?</code> , <code>{n}</code> , <code>{n,}</code> , <code>{n,m}</code>	限定符（量词）
4	<code>^</code> , <code>\$</code> , <code>\任何元字符</code> 、 <code>任何字符</code>	定位点和序列（即：位置和顺序）
5	<code> </code>	替换，"或"操作

字符具有高于替换运算符的优先级，使得 `"m|food"` 匹配 `"m"` 或 `"food"`。若要匹配 `"mood"` 或 `"food"`，请使用括号创建子表达式，从而产生 `"(m|f)ood"`。

一些常见正则表达式运算符按照优先级从高到低的顺序：

- 转义符号： `\` 是用于转义其他特殊字符的转义符号。它具有最高的优先级。

示例： `\d`、`\.` 等，其中 `\d` 匹配数字，`\.` 匹配点号。

- 括号： 圆括号 `()` 用于创建子表达式，具有高于其他运算符的优先级。

示例： `(abc)+` 匹配 `"abc"` 一次或多次。

- 量词： 量词指定前面的元素可以重复的次数。

示例： `a*` 匹配零个或多个 `"a"`。

- 字符类： 字符类使用方括号 `[]` 表示，用于匹配括号内的任意字符。

示例： `[aeiou]` 匹配任何一个元音字母。

- 断言： 断言是用于检查字符串中特定位置的条件的元素。

示例： `^` 表示行的开头，`$` 表示行的结尾。

- 连接： 连接在没有其他运算符的情况下表示字符之间的简单连接。

示例： `abc` 匹配 `"abc"`。

- 管道： 管道符号 `|` 表示"或"关系，用于在多个模式之间选择一个。

示例： `cat|dog` 匹配 `"cat"` 或 `"dog"`。