



## 基本字符匹配 (Basic)

- 匹配除了换行符之外的任意单个字符。
- 匹配方括号内的任意一个字符。例如, `[abc]` 会匹配 "a", "b", 或 "c"。可以使用 `-` 表示范围, 例如, `[a-z]` 就表示所有小写字母。
- 匹配不在方括号中的任意字符。例如, `[^abc]` 表示非 "a", "b", "c" 的任意字符。
- 逻辑或操作。例如, `cat|dog` 匹配 "cat" 或 "dog"。

## 字符类 (Character Class)

- `\d` 匹配任意数字, 等价于 `[0-9]`。
- `\D` 匹配任意非数字字符, 等价于 `[^0-9]`。
- `\w` 匹配任意字母数字字符及下划线, 等价于 `[a-zA-Z0-9_]`。
- `\W` 匹配任意非字母数字字符, 等价于 `[^a-zA-Z0-9_]`。
- `\s` 匹配任意空白字符 (空格、Tab、换行符等)。
- `\S` 匹配任意非空白字符。

## 定位符 (Anchors)

- `^` 匹配每行的开头。例如: `^Hello` 匹配以 **Hello** 开始的行。
- `$` 匹配每行的结尾。例如: `World$` 匹配以 **World** 结尾的行。
- `\b` 匹配一个单词的边界。例如, `\bin\b` 就会精确匹配单词 **in**, 而不会匹配单词 **find** 中的 **in**。
- `\B` 匹配非单词边界, 与 `\b` 相反。

## 旗标 (Flags)

- `i` case **i**nsensitive, 表示忽略大小写。
- `m` **m**ulti-line, 多行模式
- `s` 允许 `.` 匹配包括换行符在内的所有字符。
- `g` **g**lobal, 全局搜索, 匹配文本中的所有实例。

## 前瞻 (Lookahead)

- `(?=...)` 正向前瞻 (Positive Lookahead)  
只有当后面分组中的匹配成立时, 才会匹配到它前面的指定内容。  
例如: `cat(?=dog)` 会匹配 `catdog` 中的 `cat`, 但是不会匹配 `catcow` 或者 `catpig` 中的 `cat`。(找到狗前面的猫)
- `(?!...)` 负向前瞻 (Negative Lookahead)  
只有当后面分组中的匹配不成立时, 才会匹配到它前面的指定内容。  
例如: `cat(?!dog)` 不会匹配到 `catdog` 中的 `cat`, 但是会匹配 `catcow` 中的 `cat`。(找到不在狗前面的猫)

## 量词 (Quantifiers)

- `*` 匹配0次或者多次前面的元素。
- `+` 匹配1次或者多次前面的元素。
- `?` 匹配0次或者1次前面的元素。
- `{n}` 确切匹配n次前面的元素。
- `{n,}` 匹配n次或更多次前面的元素。
- `{n,m}` 匹配n次到m次前面的元素。

## 贪婪匹配和非贪婪匹配

- `ab{3,}` 默认贪婪匹配, 只要后面的字符依然满足条件, 就会继续匹配。  
`abbb`、`abbbb`、`abbbbbb`、`abbbbbbb` 都满足要求。
- `ab{3,}?` 在量词后面加上问号就表示非贪婪匹配, 满足条件之后, 即使后面依然符合条件也不会再继续匹配了。只有 `abbb` 满足要求, `abbbb`、`abbbbbb`、`abbbbbbb` 就只有前面的部分满足要求。

## 分组和引用 (Groups)

- `(abc)` 捕获分组。可以将多个字符当做一个整体来处理, 也可以在后面引用捕获到的分组内容。  
例如: 想要匹配 2024-05-01 这样的日期格式, 就可以使用:  
`(\d{4})-(\d{1,2})-(\d{1-2})`  
来分别捕获年、月和日的值, 后面也可以使用 `\1`、`\2` 和 `\3` 或者 `$1`、`$2`、`$3` 来引用它们。
- `(?:abc)` 非捕获分组, 仅使用分组功能, 而不捕获。后面也不可以使用 `\1` 或者 `$1` 来引用。

## 后顾 (Lookbehind)

- `(?<=...)` 正向后顾 (Positive Lookbehind)  
只有当前面分组中的匹配成立时, 才会匹配到它后面的指定内容。  
例如: `(?<=cat)dog` 会匹配 `catalog` 中的 `dog`, 但是不会匹配 `cowdog` 或者 `pigdog` 中的 `dog`。(找到猫后面的狗)
- `(?<!=...)` 负向后顾 (Negative Lookbehind)  
只有当前面分组中的匹配不成立时, 才会匹配到它后面的指定内容。  
例如: `(?<!=cat)dog` 不会匹配到 `catdog` 中的 `dog`, 但是会匹配到 `cowdog` 和 `pigdog` 中的 `dog`。(找到不在猫后面的狗)

