断言

正则表达式的断言是一种特殊的模式匹配技术，用于在匹配时对字符串进行条件性的预测。断言不会消耗输入字符串，仅仅是在匹配的位置上进行条件判断。断言分为正向断言和负向断言，分别用于描述匹配位置前面或后面的条件。

正则表达式的断言有 4 种形式：

- `(?=pattern)` -- 零宽正向先行断言：匹配位置之前有指定的条件。
- `(?!pattern)` -- 零宽负向先行断言：匹配位置之前没有指定的条件。
- `(?<=pattern)` -- 零宽正向后行断言：匹配位置之后有指定的条件。
- `(?<!pattern)` -- 零宽负向后行断言：匹配位置之后没有指定的条件。

几个名字概念说明:

> - 零宽：只匹配位置，零宽意味着断言在匹配时不会"消耗"字符串，它只是对位置进行条件判断，不包括匹配位置之前或之后的字符在匹配结果中。
> - 先行：表示断言发生在匹配位置之前。
> - 后行：表示断言发生在匹配位置之后。
> - 正向：匹配括号中的表达式，即断言所作的条件判断是肯定的，即只有当条件成立时，匹配才成功。
> - 负向：不匹配括号中的表达式，即断言所作的条件判断是否定的，即只有当条件不成立时，匹配才成功。

1. 零宽正向先行断言

零宽正向先行断言也称正向向前查找，模式：`(?=pattern)`。

这个断言用于在匹配位置之前添加一个条件，只有当这个条件匹配成功时，整个模式才会成功匹配，但匹配位置之前的内容并不包括在匹配结果中。

实例：匹配包含 `"mozilla"` 后面跟着 `"org"` 的字符串。

```
/mozilla(?=org)/
```

在上述正则表达式中，`(?=org)` 表示在 `"mozilla"` 之后必须紧跟着 `"org"` 才算匹配成功。

2. 零宽负向先行断言

零宽负向先行断言也称负向向前查找，模式：`(?!pattern)`。

这个断言用于在匹配位置之前添加一个条件，只有当这个条件不匹配时，整个模式才会成功匹配。

实例：匹配包含 `"mozilla"` 后面不跟着 `"org"` 的字符串。

```
/mozilla(?!org)/
```

在上述正则表达式中，`(?!org)` 表示在 `"mozilla"` 之后不能跟着 `"org"` 才算匹配成功。

3. 零宽正向后行断言

零宽正向后行断言，又称正向向后查找，模式：`(?<=pattern)`。

这个断言用于在匹配位置之后添加一个条件，只有当这个条件匹配成功时，整个模式才会成功匹配。同样，匹配位置之后的内容并不包括在匹配结果中。

实例：匹配前面跟着 `"mozilla"` 的单词。

```
/(?<=mozilla)\w+/
```

在上述正则表达式中，`(?<=mozilla)` 表示在匹配位置之前必须有 `"mozilla"` 才算匹配成功。

4. 零宽负向后行断言

零宽负向后行断言，又称负向向后查找，模式：`(?<!pattern)`。

这个断言用于在匹配位置之后添加一个条件，只有当这个条件不匹配时，整个模式才会成功匹配。

实例：匹配前面不跟着 "mozilla" 的单词。

```
/(?<![0-9]+)mozilla/
```

在上述正则表达式中，`(?<![0-9]+)` 表示在匹配的字符串 `"mozilla"` 之前不能有数字才算匹配成功。