

## 零宽断言

目前为止，我们学到的正则表达式匹配，都是有“宽度”的，使用 `\w+` 匹配下面文本，会将 `.` 一同匹配：

```
regular. expression.
```

如果不想匹配符号，只匹配一个位置，就要用到“零宽断言”(匹配宽度为零，满足一定的 条件/断言)，零宽断言使用 `(?=表达式)` 的语法，例如 `\w+(?=.)`，其中 `(?=.)` 表示 `.` 前面的位置(先行断言)

```
regular. expression.
```

如果需要匹配后面的位置，如：

```
. regular . expression
```

则要用到后发断言 `(?<=.)`，使用 `(?<=.)\w+` 得到上面的匹配结果

使用 `(?<=<b>).* (?=</b>)` 匹配标签中的内容

```
<b>粗体</b>
```

## 负向零宽断言

负向零宽断言 `(?!表达式)` 也是匹配一个零宽度的位置，不过这个位置的“断言”取表达式的反值，例如 `(?!表达式)` 表示 `表达式` 前面的位置，如果 `表达式` 不成立，匹配这个位置；如果 `表达式` 成立，则不匹配：

```
expression expression, expression; expression.
```

以上为使用 `.+n(?!.)` 的匹配结果。注意与 `.+n[^\.]` 匹配的区别

```
expressionexpression, expression; expression.
```

同样，负向零宽断言也有“先行”和“后发”两种，负向零宽后发断言为 `(?<![表达式])`

使用 `(?<![</])para(?!>)` 匹配下面文本

```
<para>para表示一个段落</para>
```

- `(?<![</])` 表示 `para` 左边不能为 `<` 或 `/`；`(?!>)` 表示 `para` 右边不能为 `>`