修饰符

在正则表达式中，修饰符是用来修改搜索模式的标志，添加在正则表达式的末尾，以控制匹配的方式。格式为：`/pattern/flags`。`pattern` 为正则表达式，`flags` 为修饰符。

正则表达式的修饰符可以单独使用，也可以组合使用，它们提供了更灵活的匹配选项，适应不同的需求。

1. g - 全局搜索（Global）

- 示例：`/abc/g`
- 匹配：`"abc"`, `"abcabc"`, `"abcxyzabc"`
- 不匹配： `"ac"`。

2. i - 不区分大小写（Case Insensitive）

- 示例：`/abc/i`
- 匹配：`"abc"`, `"AbC"`, `"ABC"`
- 不匹配：`"ac"`

3. m - 多行匹配（Multiline）

- 示例：`/^abc/m`
- 匹配：`"abc"`（字符串的开头），`"xyz\nabc"`（字符串的第二行）
- 不匹配：`"xyz\nabc"`（字符串的开头）。

4. s - 单行匹配（Single line）

- 示例：`/abc/s`
- 匹配：`"abc"`（字符串中的任何位置，包括换行符）
- 不匹配：`"ab\nc"`, `"a\nb\nc"`

5. u - Unicode 匹配模式

- 示例：/[\u4e00-\u9fa5]+/u
- 匹配：匹配中文字符
- 不匹配：字母、数字

6. y - 粘附匹配

`y` 修饰符是 ECMAScript 6 中引入的，它使得正则表达式的匹配从字符串的当前位置开始，而不是从上次匹配的位置开始，这种方式被称为粘附匹配（sticky matching）。

- 示例：`/abc/y`
- 匹配：`"abc"`（字符串的开头）
- 不匹配：`"xyz\nabc"`（字符串的开头之后）

7. x - 忽略空白字符（Whitespace）

x 修饰符用于忽略正则表达式中的空白字符（除了在字符类中的空白字符），这样可以使正则表达式更易读，可以添加注释和格式化。

- 示例：`/a b c/x`
- 匹配：`"abc"`
- 不匹配：`"a b c"`

`x` 和 `y` 这两个修饰符都提供了更多的灵活性和可读性，但需要注意的是，它们可能在某些环境中不被完全支持，使用时，最好检查目标环境的正则表达式引擎的兼容性。