

正则表达式 --- 修饰符 (Flag) “m” — 多行模式匹配介绍 (五)

原创 wincheshe 于 2022-03-12 17:41:08 发布 578 收藏 2

版权

分类专栏: 正则表达式 文章标签: 正则表达式 perl 开发语言 javascript 学习

文章目录

Flag "m" — 多行模式
行的开头 ^
行的结尾 \$
锚符 ^\$ 对比 \n

Flag “m” — 多行模式

通过 flag `/.../m` 可以开启多行模式。

这仅仅会影响 `^` 和 `$` 锚符的行为。

在多行模式下，它们不仅仅匹配文本的开始与结束，还匹配每一行的开始与结束。

行的开头 ^

在这个有多行文本的例子中，正则表达式 `/^\d+/gm` 将匹配每一行的开头数字：

```
1 | let str = `1st place: Winnie
2 | 2nd place: Piglet
3 | 33rd place: Eeyore`;
4 |
5 | alert( str.match(/^\d+/gm) ); // 1, 2, 33
```

没有 flag `/.../m` 时，仅仅是第一个数字被匹配到：

```
1 let str = `1st place: Winnie
2 2nd place: Piglet
3 33rd place: Eeyore`;
4
5 alert( str.match(/^\d+/g) ); // 1
```

这是因为默认情况下，锚符 `^` 仅仅匹配文本的开头，在多行模式下，它匹配行的开头。

正则表达式引擎将会在文本中查找以锚符 `^` 开始的字符串，我们找到之后继续匹配 `\d+` 模式。

行的结尾 `$`

美元符 `$` 行为也相似。

正则表达式 `\w+$` 会找到每一行的最后一个单词：

```
1 let str = `1st place: Winnie
2 2nd place: Piglet
3 33rd place: Eeyore`;
4
5 alert( str.match(/\w+$/gim) ); // Winnie,Piglet,Eeyore
```

没有 `/.../m` flag 的话，美元符 `$` 将会仅仅匹配整个文本的结尾，所以只有最后的一个单词会被找到。

锚符 `^$` 对比 `\n`

要寻找新的一行的话，我们不仅可以使⽤锚符 `^` 和 `$`，也可以使用换行符 `\n`。

它和锚符 `^` 和 `$` 的第一个不同点是它不像锚符那样，它会“消耗”掉 `\n` 并且将其（`\n`）加入到匹配结果中。

举个例子，我们在下面的代码中用它来替代 `$`：

```
1 let str = `1st place: Winnie
2 2nd place: Piglet
3 33rd place: Eeyore`;
4
```

```
alert( str.match(/\w+\n/gim) ); // Winnie\n,Piglet\n
```

这里，我们每次匹配到的时候都会被添加一个换行符。

还有一个不同点——换行符 `\n` 不会匹配字符串结尾。这就是为什么在上面的例子中 `Eeyore` 没有匹配到。

所以，通常情况下使用锚符更棒，用它匹配出来的结果更加接近我们想要的结果。

正则前面的 (?i) (?s) (?m) (?is) (?im)

Q:经常看见的正则前面的 (?i) (?s) (?m) (?is) (?im) 是什么意思?

A: 称为内联匹配模式，通常用内联匹配模式代替使用枚举值RegexOptions指定的全局匹配模式，写起来更简洁。

(?i) 表示所在位置右侧的表达式开启忽略大小写模式

(?s) 表示所在位置右侧的表达式开启单行模式。

更改句点字符 (.) 的含义，以使它与每个字符（而不是除 \n 之外的所有字符）匹配。

注意：(?s)通常在匹配有换行的文本时使用

(?m) 表示所在位置右侧的表示式开启指定多行模式。

更改 ^ 和 \$ 的含义，以使它们分别与任何行的开头和结尾匹配，

而不只是与整个字符串的开头和结尾匹配。

注意：(?m)只有在正则表达式中涉及到多行的 “^” 和 “\$” 的匹配时，才使用Multiline模式。

上面的匹配模式可以组合使用，比如(?is),(?im)。

另外，还可以用(?i:exp)或者(?i)exp(?-i)来指定匹配的有效范围。

附：

.表示除\n之外的任意字符

*表示匹配0-无穷

+表示匹配1-无穷