

正则表达式 - 简介

除非您以前使用过正则表达式，否则您可能不熟悉一些术语。但是，毫无疑问，您已经使用过不涉及脚本的某些正则表达式概念。

例如，您很可能使用 `?` 和 `*` 通配符来查找硬盘上的文件。`?` 通配符匹配文件名中的 0 个或 1 个字符，而 `*` 通配符匹配零个或多个字符。像 `data(\w)?\.dat` 这样的模式将查找下列文件：

```
data.dat
data1.dat
data2.dat
datax.dat
dataN.dat
```

使用 `*` 字符代替 `?` 字符扩大了找到的文件的数量。`data.*\.dat` 匹配下列所有文件：

```
data.dat
data1.dat
data2.dat
data12.dat
datax.dat
dataXYZ.dat
```

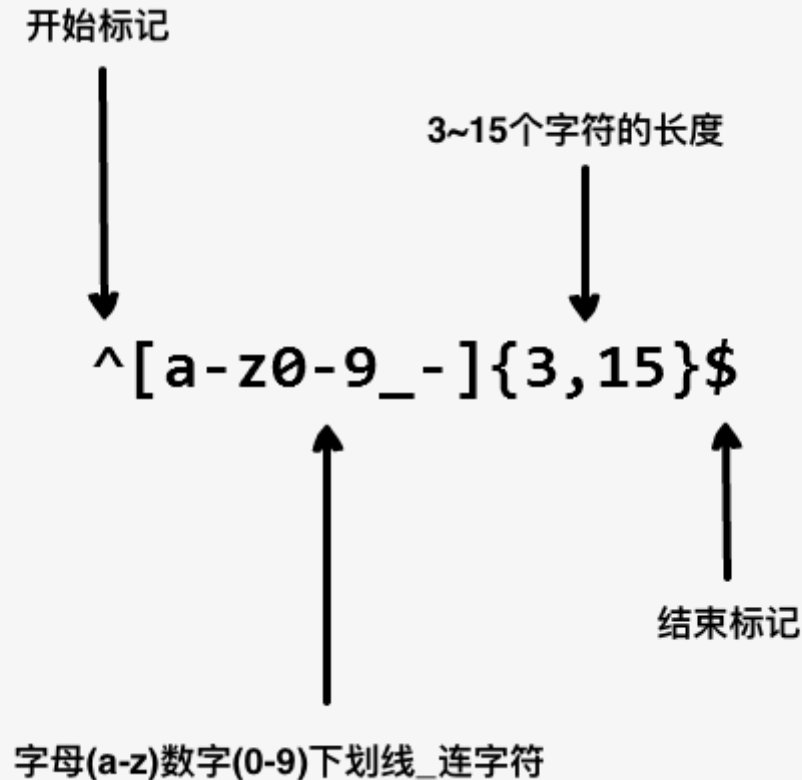
尽管这种搜索方法很有用，但它还是有限的。通过理解 `*` 通配符的工作原理，引入了正则表达式所依赖的概念，但正则表达式功能更强大，而且更加灵活。

正则表达式的使用，可以通过简单的办法来实现强大的功能。下面先给出一个简单的示例：

```
^[0-9]+abc$
```

- `^` 为匹配输入字符串的开始位置。
- `[0-9]+` 匹配多个数字，`[0-9]` 匹配单个数字，`+` 匹配一个或者多个。
- `abc$` 匹配字母 `abc` 并以 `abc` 结尾，`$` 为匹配输入字符串的结束位置。

我们在写用户注册表单时，只允许用户名包含字符、数字、下划线和连接字符(-)，并设置用户名的长度，我们就可以使用以下正则表达式来设定。



以上的正则表达式可以匹配 `runoob`、`runoob1`、`run-oob`、`run_oob`，但不匹配 `ru`，因为它包含了小写的字母而且太短了，也不匹配 `runoob$`，因为它包含特殊字符。

实例

匹配以数字开头，并以 `abc` 结尾的字符串。：

```
var str = "123abc";
var patt1 = /^[0-9]+abc$/;
document.write(str.match(patt1));
```

以下标记的文本是获得的匹配的表达式：

123abc

尝试一下 »

继续阅读本教程将让您也可以自由应用这样的代码。

为什么使用正则表达式？

典型的搜索和替换操作要求您提供与预期的搜索结果匹配的确切文本。虽然这种技术对于对静态文本执行简单搜索和替换任务可能已经足够了，但它缺乏灵活性，若采用这种方法搜索动态文本，即使不是不可能，至少也会变得很困难。

通过使用正则表达式，可以：

- 测试字符串内的模式。
例如，可以测试输入字符串，以查看字符串内是否出现电话号码模式或信用卡号码模式。这称为数据验证。
- 替换文本。
可以使用正则表达式来识别文档中的特定文本，完全删除该文本或者用其他文本替换它。

- 基于模式匹配从字符串中提取子字符串。
可以查找文档内或输入域内特定的文本。

例如，您可能需要搜索整个网站，删除过时的材料，以及替换某些 HTML 格式标记。在这种情况下，可以使用正则表达式来确定在每个文件中是否出现该材料或该 HTML 格式标记。此过程将受影响的文件列表缩小到包含需要删除或更改的材料的那些文件。然后可以使用正则表达式来删除过时的材料。最后，可以使用正则表达式来搜索和替换标记。

发展历史

正则表达式的"祖先"可以一直上溯至对人类神经系统如何工作的早期研究。Warren McCulloch 和 Walter Pitts 这两位神经生理学家研究出一种数学方式来描述这些神经网络。

1956 年, 一位叫 Stephen Kleene 的数学家在 McCulloch 和 Pitts 早期工作的基础上，发表了一篇标题为"神经网络事件的表示法"的论文，引入了正则表达式的概念。正则表达式就是用来描述他称为"正则集的代数"的表达式，因此采用"正则表达式"这个术语。

随后，发现可以将这一工作应用于使用 Ken Thompson 的计算搜索算法的一些早期研究，Ken Thompson 是 Unix 的主要发明人。正则表达式的第一个实用应用程序就是 Unix 中的 qed 编辑器。

如他们所说，剩下的就是众所周知的历史了。从那时起直至现在正则表达式都是基于文本的编辑器和搜索工具中的一个重要部分。

应用领域

目前，正则表达式已经在很多软件中得到广泛的应用，包括 *nix (Linux, Unix等)、HP 等操作系统，PHP、C#、Java 等开发环境，以及很多的应用软件中，都可以看到正则表达式的影子。

C# 正则表达式

在我们的 C# 教程中，[C# 正则表达式](#) 这一章节专门介绍了有关 C# 正则表达式的知识。

Java 正则表达式

在我们的 Java 教程中，[Java 正则表达式](#) 这一章节专门介绍了有关 Java 正则表达式的知识。

JavaScript 正则表达式

在我们的 JavaScript 教程中，[JavaScript RegExp 对象](#) 这一章节专门介绍了有关 JavaScript 正则表达式的知识，同时我们还提供了完整的 [JavaScript RegExp 对象参考手册](#)。

Python 正则表达式

在我们的 Python 基础教程中，[Python 正则表达式](#) 这一章节专门介绍了有关 Python 正则表达式的知识。

Ruby 正则表达式

在我们的 Ruby 教程中，[Ruby 正则表达式](#) 这一章节专门介绍了有关 Ruby 正则表达式的知识。

命令或环境	.	[]	^	\$	\(\)	\{ \}	?	+		()
vi	√	√	√	√	√					

Visual C++	√	√	√	√	√					
awk	√	√	√	√		awk是支持该语法的，只是要在命令 行加入 --posix or --re-interval参数即可，可见 man awk中的interval expression	√	√	√	√
sed	√	√	√	√	√	√				
delphi	√	√	√	√	√		√	√	√	√
python	√	√	√	√	√	√	√	√	√	√
java	√	√	√	√	√	√	√	√	√	√
javascript	√	√	√	√	√		√	√	√	√
php	√	√	√	√	√					
perl	√	√	√	√	√		√	√	√	√
C#	√	√	√	√			√	√	√	√

[← 正则表达式 - 教程](#)
[正则表达式 - 语法 →](#)
[✎ 点我分享笔记](#)