[cron表达式详解](https://www.cnblogs.com/yanghj010/p/10875151.html)

一、结构

corn从左到右（用空格隔开）：秒 分 小时 月份中的日期 月份 星期中的日期 年份

二、各字段的含义

|  |  |  |
| --- | --- | --- |
| 字段 | 允许值 | 允许的特殊字符 |
| 秒（Seconds） | 0~59的整数 | , - \* /    四个字符 |
| 分（*Minutes*） | 0~59的整数 | , - \* /    四个字符 |
| 小时（*Hours*） | 0~23的整数 | , - \* /    四个字符 |
| 日期（*DayofMonth*） | 1~31的整数（但是你需要考虑你月的天数） | ,- \* ? / L W C     八个字符 |
| 月份（*Month*） | 1~12的整数或者 JAN-DEC | , - \* /    四个字符 |
| 星期（*DayofWeek*） | 1~7的整数或者 SUN-SAT （1=SUN） | , - \* ? / L C #     八个字符 |
| 年(可选，留空)（*Year*） | 1970~2099 | , - \* /    四个字符 |

三、注意事项：

　　每一个域都使用数字，但还可以出现如下特殊字符，它们的含义是：

　　（1）\*：表示匹配该域的任意值。假如在Minutes域使用\*, 即表示每分钟都会触发事件。

　　（2）?：只能用在DayofMonth和DayofWeek两个域。它也匹配域的任意值，但实际不会。因为DayofMonth和DayofWeek会相互影响。例如想在每月的20日触发调度，不管20日到底是星期几，则只能使用如下写法： 13 13 15 20 \* ?, 其中最后一位只能用？，而不能使用\*，如果使用\*表示不管星期几都会触发，实际上并不是这样。

　　（3）-：表示范围。例如在Minutes域使用5-20，表示从5分到20分钟每分钟触发一次

　　（4）/：表示起始时间开始触发，然后每隔固定时间触发一次。例如在Minutes域使用5/20,则意味着5分钟触发一次，而25，45等分别触发一次.

　　（5）,：表示列出枚举值。例如：在Minutes域使用5,20，则意味着在5和20分每分钟触发一次。

　　（6）L：表示最后，只能出现在DayofWeek和DayofMonth域。如果在DayofWeek域使用5L,意味着在最后的一个星期四触发。

　　（7）W:表示有效工作日(周一到周五),只能出现在DayofMonth域，系统将在离指定日期的最近的有效工作日触发事件。例如：在 DayofMonth使用5W，如果5日是星期六，则将在最近的工作日：星期五，即4日触发。如果5日是星期天，则在6日(周一)触发；如果5日在星期一到星期五中的一天，则就在5日触发。另外一点，W的最近寻找不会跨过月份 。

　　（8）LW:这两个字符可以连用，表示在某个月最后一个工作日，即最后一个星期五。

　　（9）#:用于确定每个月第几个星期几，只能出现在DayofMonth域。例如在4#2，表示某月的第二个星期三。

## 表达式全集

|  |  |
| --- | --- |
| **字符** | **描述** |
| **\** | 将下一个字符标记为一个特殊字符、或一个原义字符、或一个向后引用、或一个八进制转义符。例如，“n”匹配字符“n”。“\n”匹配一个换行符。串行“\\”匹配“\”而“\(”则匹配“(”。 |
| **^** | 匹配输入字符串的开始位置。 |
| **$** | 匹配输入字符串的结束位置。 |
| **\*** | 匹配前面的子表达式零次或多次。例如，zo\*能匹配“z”以及“zoo”。\*等价于{0,}。 |
| **+** | 匹配前面的子表达式一次或多次。例如，“zo+”能匹配“zo”以及“zoo”，但不能匹配“z”。+等价于{1,}。 |
| **?** | 匹配前面的子表达式零次或一次。例如，“do(es)?”可以匹配“does”或“does”中的“do”。?等价于{0,1}。 |
| **{*n*}** | *n*是一个非负整数。匹配确定的*n*次。例如，“o{2}”不能匹配“Bob”中的“o”，但是能匹配“food”中的两个o。 |
| **{*n*,}** | *n*是一个非负整数。至少匹配*n*次。例如，“o{2,}”不能匹配“Bob”中的“o”，但能匹配“foooood”中的所有o。“o{1,}”等价于“o+”。“o{0,}”则等价于“o\*”。 |
| **{*n*,*m*}** | *m*和*n*均为非负整数，其中*n*<=*m*。最少匹配*n*次且最多匹配*m*次。例如，“o{1,3}”将匹配“fooooood”中的前三个o。“o{0,1}”等价于“o?”。 |
| **?** | 当该字符紧跟在任何一个其他限制符（\*,+,?，{*n*}，{*n*,}，{*n*,*m*}）后面时，匹配模式是非贪婪的。非贪婪模式尽可能少的匹配所搜索的字符串，而默认的贪婪模式则尽可能多的匹配所搜索的字符串。 |
| **.** | 匹配除“\*n*”之外的任何单个字符。要匹配包括“\*n*”在内的任何字符，请使用像“(.|\n)”的模式。 |
| **(pattern)** | 匹配pattern并获取这一匹配。所获取的匹配可以从产生的Matches集合得到，在VBScript中使用SubMatches集合，在JScript中则使用$0…$9属性。要匹配圆括号字符，请使用“\(”或“\)”。 |
| **(?:pattern)** | 匹配pattern但不获取匹配结果，也就是说这是一个非获取匹配，不进行存储供以后使用。这在使用或字符“(|)”来组合一个模式的各个部分是很有用。例如“industr(?:y|ies)”就是一个比“industry|industries”更简略的表达式。 |
| **(?=pattern)** | 正向肯定预查，在任何匹配pattern的字符串开始处匹配查找字符串。这是一个非获取匹配，也就是说，该匹配不需要获取供以后使用。 |
| **(?!pattern)** | 正向否定预查，在任何不匹配pattern的字符串开始处匹配查找字符串。这是一个非获取匹配，也就是说，该匹配不需要获取供以后使用。 |
| **(?<=pattern)** | 反向肯定预查，与正向肯定预查类拟，只是方向相反。 |
| **(?<!pattern)** | 反向否定预查，与正向否定预查类拟，只是方向相反。 |
| **x|y** | 匹配x或y。例如，“z|food”能匹配“z”或“food”。 |
| **[xyz]** | 字符集合。匹配所包含的任意一个字符。例如，“[abc]”可以匹配“plain”中的“a”。 |
| **[^xyz]** | 负值字符集合。匹配未包含的任意字符。例如，“[^abc]”可以匹配“plain”中的“p”。 |
| **[a-z]** | 字符范围。匹配指定范围内的任意字符。 |
| **[^a-z]** | 负值字符范围。匹配任何不在指定范围内的任意字符。 |
| **\b** | 匹配一个单词边界，也就是指单词和空格间的位置。例如，“er\b”可以匹配“never”中的“er”，但不能匹配“verb”中的“er”。 |
| **\B** | 匹配非单词边界。“er\B”能匹配“verb”中的“er”，但不能匹配“never”中的“er”。 |
| **\cx** | 匹配由x指明的控制字符。例如，\cM匹配一个Control-M或回车符。x的值必须为A-Z或a-z之一。否则，将c视为一个原义的“c”字符。 |
| **\d** | 匹配一个数字字符。等价于[0-9]。 |
| **\D** | 匹配一个非数字字符。等价于[^0-9]。 |
| **\f** | 匹配一个换页符。等价于\x0c和\cL。 |
| **\n** | 匹配一个换行符。等价于\x0a和\cJ。 |
| **\r** | 匹配一个回车符。等价于\x0d和\cM。 |
| **\s** | 匹配任何空白字符，包括空格、制表符、换页符等等。等价于[ \f\n\r\t\v]。 |
| **\S** | 匹配任何非空白字符。等价于[^ \f\n\r\t\v]。 |
| **\t** | 匹配一个制表符。等价于\x09和\cI。 |
| **\v** | 匹配一个垂直制表符。等价于\x0b和\cK。 |
| **\w** | 匹配包括下划线的任何单词字符。等价于“[A-Za-z0-9\_]”。 |
| **\W** | 匹配任何非单词字符。等价于“[^A-Za-z0-9\_]”。 |
| **\x*n*** | 匹配*n*，其中*n*为十六进制转义值。十六进制转义值必须为确定的两个数字长。例如，“\x41”匹配“A”。“\x041”则等价于“\x04&1”。正则表达式中可以使用ASCII编码。. |
| **\*num*** | 匹配*num*，其中*num*是一个正整数。对所获取的匹配的引用。例如，“(.)\1”匹配两个连续的相同字符。 |
| **\*n*** | 标识一个八进制转义值或一个向后引用。如果\*n*之前至少*n*个获取的子表达式，则*n*为向后引用。否则，如果*n*为八进制数字（0-7），则*n*为一个八进制转义值。 |
| **\*nm*** | 标识一个八进制转义值或一个向后引用。如果\*nm*之前至少有*nm*个获得子表达式，则*nm*为向后引用。如果\*nm*之前至少有*n*个获取，则*n*为一个后跟文字*m*的向后引用。如果前面的条件都不满足，若*n*和*m*均为八进制数字（0-7），则\*nm*将匹配八进制转义值*nm*。 |
| **\*nml*** | 如果*n*为八进制数字（0-3），且*m和l*均为八进制数字（0-7），则匹配八进制转义值*nm*l。 |
| **\u*n*** | 匹配*n*，其中*n*是一个用四个十六进制数字表示的Unicode字符。例如，\u00A9匹配版权符号（©）。 |