对于正则表达式，相信很多人都知道，但是很多人的第一感觉就是难学，因为看第一眼时，觉得完全没有规律可寻，而且全是一堆各种各样的特殊符号，完全不知所云。

其实只是对正则不了解而以，了解了你就会发现，原来就这样啊正则所用的相关字符其实不多，

也不难记，更不难懂，唯一难的就是组合起来之后，可读性比较差，而且不容易理解。

今天 Tom 老师带大家一起来领略 Java 正则，从入门到精通，从精通到删除。

一、 什么是正则表达式

正则表达式是一种特殊的字符串模式，用于匹配一组字符串，就好比用模具做产品，而正则就是这个模具，定义一种规则去匹配符合规则的字符。

二、巧用正则解决实际问题

场景一：

在 csdn 复制代码会自带序号，如何快速去掉

场景二：

编辑 word 文档，经常会查找字符串

三、正则表达式在 Java 中的应用

场景一：

HttpServlet 的 url 配置

场景二：

SpringMVC 的 url 配置等框架

场景三：

不规则数据的内容分析（爬虫、文档解析）

四、 正则表达式其实并不是那么难

难懂的主要原因

1、可读性不强

2、一句话包含 N 种逻辑

正则表达式组成

1、边界符

^	开始符
$	结束符
[]	单字符
()	分组
-	区间 a-z、A-Z、0-9

2、转义符

\b	Break、打破，单词边界
\w	Word、单词字母数字组合
\d	Digit、数字
\s	Space、空格
\t	Table、制表符
\n	NewLine、换行
\r	Retun、回车

3、计量符

*	(贪婪) 重复零次或更多(任意),文件选择器*.png
+	(懒惰) 重复一次或更多次(至少重复一次)，网页中的计数 1W+
?	(占位) 重复零次或一次(可有可无)，java 中预编译语句集
{n}	重复 n 次
{n,m}	重复 n 到 m 次(至少重复 n 次，最多重复 m 次)
{n,}	重复 n 次或更多次(大于等于 n 次)

4、逻辑符

|	逻辑或
=	逻辑等于(环视肯定顺序)
!	逻辑非（环视否定顺序）
<=	环视肯定逆序
<!	环视否定逆序

五、 正则表达式引擎的内部工作机制

总是从左往右一次匹配

六、 常用的正则举例

匹配 html 标签 <[^>]+>

匹配中文字符[\u4E00-\u9FA5\uF900-\uFA2D]+

七、 正则表达式的高级用法

肯定顺序常规： [a-z]+(?=;) 字母序列后面跟着;

肯定顺序变种： (?=[a-z]+$).+$ 字母序列

肯定逆序常规： (?<=:)[0-9]+ :后面的数字

肯定逆序变种： \b[0-9]\b(?<=[13579]) 0-9 中的奇数

否定顺序常规： [a-z]+\b(?!;) 不以;结尾的字母序列

否定顺序变种：(?!.*?[1o0])\b[a-z0-9]+\b 不包含 1/o/0 的字母数字系列

否定逆序常规： (?<!age)=([0-9]+) 参数名不为 age 的数据

否定逆序变种： \b[a-z]+(?<!z)\b 不以 z 结尾的单词

一、肯定顺序环视常规用法

源字符串：

notexefile1.txt

exefile1.exe

exefile2.exe

exefile3.exe

notexefile2.php

notexefile3.sh

需求：获取.exe 后缀文件不含后缀的文件名

正则：.+(?=\.exe)

结果：

exefile1

exefile2

exefile3

二、否定顺序环视

源字符串：

notexefile1.txt

exefile1.exe

exefile2.exe

exefile3.exe

notexefile2.php

notexefile3.sh

需求：获取不是.exe 后缀文件不含后缀的文件名

正则：(.+)(?!\.exe)\.[^.]+$

结果：

notexefile1

notexefile2

三、肯定逆序环视

源字符串：

name=Zjmainstay

age=26

需求：获取 name 参数的值

正则：(?<=name=).+

示例很直白，前面必须是 name=，然后获取其后面的数据，由于环视不占位，因此并没有出现在匹配结果中。

四、否逆序环视

源字符串：

name=Zjmainstay

age=26

需求：获取不是 name 参数的值

正则：^[^=]+=(?<!name=)(.+)

跟否定顺序示例一样，我们不能直接用(?<!name=).+进行匹配，正则做法是先把参数部分匹配出来，再用否定逆序环视对它进行限定，限定它不能是 name=，因此实现匹配。

实际应用 innerText

a test

(?<=)[^<]+(?=)