正则表达式

就是匹配字符串的规则

没有固定的只需要符合需求就可以

定界符

// ## {}

原子

规则中最小的匹配单位.叫做原子,如果是特殊字符必须加上\转义

转义:相当于取反

分类:

1. 普通字符
2. 特殊字符 需要加上\转义的字符
3. 非打印字符 \n \r \t \f
4. 通用字符匹配

\d 任意一个数字

\D 任意一个非数字

\w 任意一个字 [a-z A-Z 0-9 \_] 下划线算一个字

\W 任意一个非字 ^[a-z A-Z 0-9 \_] ^代表取反

\s 任意一个空白

\S 任意一个非空白

1. 自定义原子表

[13579] 一位数字

[1-9][0-9] 两位数字

1. 点(.) 代表任意字符

元字符

{n} 匹配n位

{n,} 匹配至少n位

{n,m} 至少n位,至多m位

\*至少0至多无限 0-n位{0,}

+ 至少1 至多无限 1-n位 {1,}

? 至少0 至多1 {0,1}

量词默认情况下都是贪婪(尽量的多匹配的)的,如果想取消贪可以在量词之后添加?或者添加模式修正符U来取消贪婪(尽量少匹配)

()

1. 改变优先级
2. 子模式,能获取括号中的每个部分作为数组的其他元素返回
3. 反向引用

每个括号会返回一个子模式,可以通过 \子模式下标来匹配

模式修正符

修正和扩展整个表达式的功能

i 不区分大小写