# 正则表达式

1. **标准字符集合 （注意大小写，大写是相反的意思）**
   1. **\d** 任意一个数字
   2. **\w** 任意一个字母数字或者下划线
   3. **\s** 包括空格、制表符、换行符等空白字符的任意一个
   4. **.** 小数点可以匹配任意一个字符（除了换行符**）**
      1. 如果要匹配包括”\n”在内的所有字符一般用[\s\S]
2. **自定义字符集合**
   1. **[ab5@]** 匹配a、b、5、@中任意一个字符
   2. **[^369]** 匹配除3、6、9之外的任意一个字符
   3. **[f-k]** 匹配f到k之间的任意一个字符
   4. **[^A-F0-3]** 匹配A-F，0-3 之外的任意一个字符
      1. 正则表达式的特殊符号，被包含到中括号中，则失去特殊意义，除了^,-之外。
      2. 标准字符集合，除小数点外，如果被包含于中括号，自定义字符集合将包含改集合 比如：
         1. [\d.\-+] 匹配：数字、小数点、+、-
3. **量词** 
   1. **{n}** 表示重复n次
   2. **{m,n}** 匹配最少m次最多n次
      1. 贪婪模式：优先匹配符合要求的最多个字符（默认）
      2. 非贪婪模式：在表达式后加个 ？ 匹配最少个字符
   3. **{m,}** 匹配最少
   4. **？** 匹配出现0-1次 相当于{0,1}
   5. **+** 表达式至少出现1次，相当于{1，}
   6. **\*** 表达式不出现或出现任意次，相当于{0，}
4. **字符边界（匹配符合哪种条件的位置）**
   1. **^** 匹配字符串开始位置
   2. $ 匹配字符串结束的位置
   3. \b 匹配一个单词边界
      1. \b匹配这样一个位置：前面的字符和后面的字符不全是\w
5. **匹配模式**
   1. **忽略大小写模式**
      1. 默认情况下是要区分大小写的
   2. **单行模式**
      1. 整个文本看做一个字符串，只有一个开头，一个结尾
      2. 此时小数点可以匹配包含换行符（\n）在内的任意字符
   3. **多行模式**
      1. 每一行都是一个字符串，都有开头和结尾
      2. 如果需要仅匹配字符串开始和结尾位置，可以使用\A和\Z代替^和$。
6. **选择符合分组**
   1. **|** 分支表达式 左右两边表达式之间或的关系，匹配左边或者右边
   2. **（）**捕获组 会把捕获到的字符保存在程序（占用内存）
      1. 每一对捕获组根据左括号的顺序从一开始自动开始编号
      2. 通过反向引用，可以对分组已经捕获的字符串进行引用
      3. （\w{2}）\1 等于 （\w{2}）(\w{2})
   3. **（？：）**非捕获组 获得组织结构，不捕获内容 （不保存捕获到的字符）
7. **预搜索（处理的表达式不计入结果）**
   1. **（？=exp）** 断言自身出现的位置的后面能匹配的表达式exp
   2. **（？<=exp）** 断言自身出现的位置的前面能匹配的表达式exp
   3. **（？！exp）** 断言此位置后面不能匹配表达式exp
   4. **（？<!exp）** 断言此位置前面不能匹配表达式exp
8. 学习实践
   1. 电话号码验证
      1. 电话号码由数字和-构成
      2. 电话号码为7到8位
      3. 如果电话号码中包含有区号，那么区号为三位到四位首位是0
      4. 区号用-和其他部分隔开
      5. 移动电话号码为11位
      6. 11位移动电话的第一位和第二位为13 15 18
   2. 电话号码匹配正则：(0\d{2,3}-\d{7,8}）|（1[35789]\d{9}）
   3. 电子邮箱地址验证
      1. 用户名：字母、数字、中划线、下划线组成
      2. @
      3. 网址：字母、数字组成
      4. 小数点：.
      5. 组织域名：2-4位字母组成
      6. 不区分大小写