1. 行定位符：
   1. ^ 行首匹配 $ 行末匹配 都没有 任意位置匹配
2. 元字符：
   1. . 除换行符任意字符
   2. \w 字母、数字、下划线、汉字
   3. \W 除上述外所有字符
   4. \s 单个空白符，包括tab、换行
   5. \S 除上述外所有字符
   6. \b 单词开始或结束，分界符一般为空格、标点、换行
   7. \d 数字
   8. \D 非数字
3. 限定符：
   1. ? 匹配前面的字符0次或一次。
   2. + 匹配前面的字符1次或多次。
   3. \* 匹配前面的字符0次或多次。
   4. {n} 匹配前面的字符n次。
   5. {n,} 匹配前面的字符最少n次。
   6. {n, m} 匹配前面的字符最少n次最多m次。
4. 字符类：
   1. [aeiou]、[a-z]、[0-9] 匹配方括号内所有字符。
   2. [^aeiou]、[^a-z0-9] 排除方括号内所有字符。
5. 逻辑或：
   1. (^\d{15}$) | (^\d{18}$) | (^\d{17})(\d | X | x)$ 匹配身份证
6. 转义：
   1. \ .
7. 分组：
   1. (……) {3} 对括号内重复操作三次
   2. 有括号的字符串，一般将整个字符串用括号括起
8. 使用：
   1. 正则表达式左右加上引号，并将其中的“ \ ”进行转义
      1. 或使用原始字符串r’……’
9. re模块：
   1. match = re.**match**(pattern, string, [flags]) 使用模式字符串匹配字符串，成功返回match对象，失败返回None
      1. flag：
         1. re.I 不区分大小写
         2. ……
   2. Match.**start**() Match.**end**() Match.**span**() 返回匹配位置的元组
   3. Match.**string** 匹配用的字符串 Match.**group**() 匹配到的数据
   4. match = re.**search**(pattern, string, [flags]) 使用模式字符串在整个字符串中搜索第一个匹配的值，成功。。。
   5. match = re.**findall**(……) 搜索整个字符串中所有符合表达式的字符串，返回**字符串列表**，失败返回空列表。
   6. result = re.sub(pattern, repl, string, count, flags) 替换字符串，返回字符串
      1. repl : 替换的字符串
      2. count ：替换的最大次数，默认为0，表示替换所有。
   7. result = re.split(pattern, string, [maxsplit], [flags]) 分割字符串，返回列表。
      1. maxsplit ：最大分割次数