1. 字符串匹配和搜索：可以使用**std::regex\_match()**函数和**std::regex\_search()**函数来判断一个字符串是否匹配指定的正则表达式，并且可以查找字符串中是否存在满足正则表达式的子字符串。
2. 字符串替换：可以使用**std::regex\_replace()**函数来对一个字符串中满足正则表达式的部分进行替换。
3. 正则表达式分割：可以使用**std::** regex\_search **()**搜索位置。
4. 正则表达式迭代器：可以使用**std::sregex\_iterator**和**std::sregex\_token\_iterator**类来实现正则表达式的迭代操作，例如遍历一个字符串中所有匹配正则表达式的子串。

正则语法：

一，元字符是构造[正则表达式](https://so.csdn.net/so/search?q=%E6%AD%A3%E5%88%99%E8%A1%A8%E8%BE%BE%E5%BC%8F&spm=1001.2101.3001.7020)的一种基本元素。

. ：匹配除换行符以外的任意字符

w：匹配字母或数字或下划线或汉字

s：匹配任意的空白符

d：匹配数字

b：匹配单词的开始或结束

^：匹配字符串的开始

$：匹配字符串的结束

二、重复限定符

正则没提供办法处理这些重复的元字符吗？答案肯定是有的。

\*：重复零次或更多次

+：重复一次或更多次

?：重复零次或一次

{n}：重复n次

{n,}：重复n次或更多次

{n,m}：重复n到m次