第十二章：正则表达式

Table of Contents

[1. 序言 2](#_Toc51594807)

[1.1 章节简介 2](#_Toc51594808)

[2. 正则表达式规则 2](#_Toc51594809)

[2.1 常用元字符 3](#_Toc51594810)

[2.2 常用限定符 3](#_Toc51594811)

[2.3 分支条件 3](#_Toc51594812)

[2.4 反义 3](#_Toc51594813)

[3. re模块 4](#_Toc51594814)

[3.1 切分字符串 6](#_Toc51594815)

[3.2 分组 6](#_Toc51594816)

[3.3 编译并使用正则表达式 7](#_Toc51594817)

# 序言

## 章节简介

正则表达式（Regular Expression）是用于描述一组字符串特征的模式，用来匹配特定的字符串。通过特殊字符+普通字符来进行模式描述，从而达到文本匹配目的工具。

正则表达式目前被集成到了各种文本编辑器/文本处理工具当中。比如用户登录系统时，进行用户名密码的验证。从大量信息中快速提取指定内容，在一批url中，查找指定url。将指定格式的文本进行正则匹配查找，找到之后进行特定替换等。本章节我们介绍正则表达式的规则，以及Python中关于正则表达式的re模块。本章节主要有以下一些内容：

* 正则表达式规则
* re模块

# 正则表达式规则

我们先来了解正则表达式如何规定的字符规则。要想真正的用好正则表达式，正确的理解元字符是最重要的事情。下表列出了主要的元字符和对它们的一个简短的描述。

## 常用元字符



## 常用限定符



## 分支条件

满足其中任意一种规则就当成匹配成功，需要使用分枝条件：使用管道|把不同的规则分开，这时候会从左到右地测试每个条件，如果满足了其中一个分枝，后面的规则就被忽略掉。

## 反义

下面的规则表示匹配不是某些字符的规则：



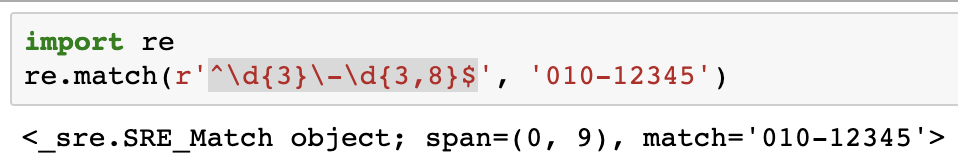
正则表达式的规则还有很多，我们经常使用以后就能熟悉它的语法，完全的定义规则，我们可以参考baidu百科的定义<https://baike.baidu.com/item/%E6%AD%A3%E5%88%99%E8%A1%A8%E8%BE%BE%E5%BC%8F/1700215?fr=aladdin#7> 。下一个小节里，我们会用一些示例为大家演示。

# re模块

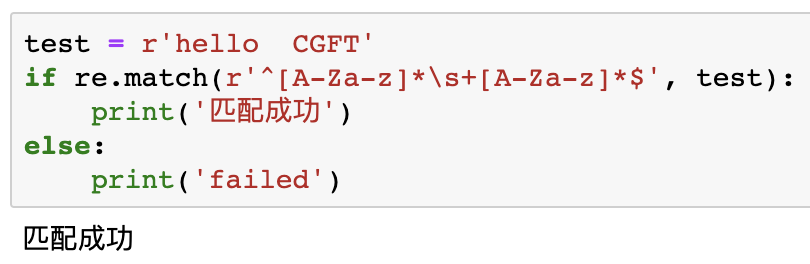
有了准备知识，我们就可以在Python中使用正则表达式了。Python提供re模块，包含所有正则表达式的功能。因为字符\在包括Python在内的很多编程语言里都有转义的作用，所以在定义字符串时，我们强烈建议大家在字符串前加上字符r，这样转义字符会被当做普通字符对待而不会发挥转义的作用。



我们先看看如何使用re模块。在下面的参数里，我们首先在match()方法的第一个参数里定义了一个正则表达式，根据前面的定义\d{3}表示可以匹配一个三位数的数字，\-代表字符-，而\d{3,8}表示匹配一个3到8位全部是数字组成的字符，^表示从起始位置开始匹配，$表示在字符的末尾匹配。将它们组合起来，就是匹配诸如“021-12345”这样的字符串。match()方法判断是否匹配，如果匹配成功，返回一个Match对象，否则返回None。下面的示例因为010-12345”匹配了前面定义的表达式，所以返回一个Match对象。

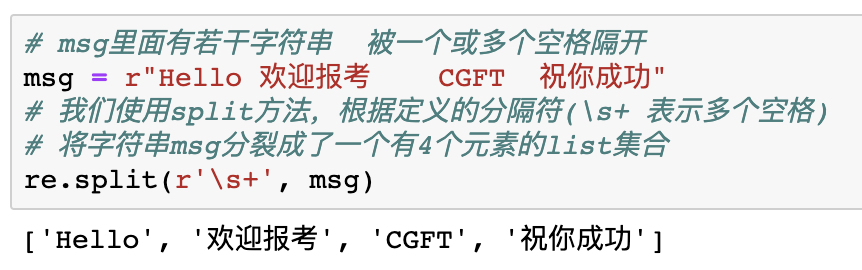


下面的示例判断字符串是否匹配，根据匹配情况输出信息。我们再来分析下这个正则表达式的意义，*^[A-Za-z]\**表示匹配任意多个字母，*\s+*表示匹配任意多个空格符，[A-Za-z]\*$表示任意多个字母结尾，将三者组合起来，就是所有以字母组成的2个字符串，中间如果用一个或多个空格隔开，那么这些字符串都会匹配^[A-Za-z]\*\s+[A-Za-z]\*$：



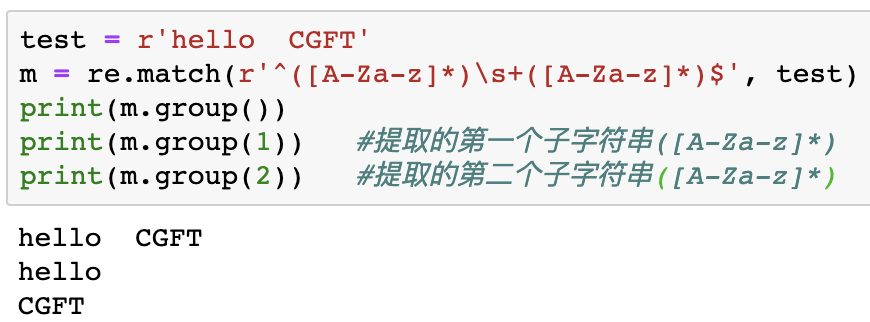
## 切分字符串

我们来看看如何用re.split()函数切分多个字符串。



## 分组

re.group()用来匹配正则表达式并提取对应的子字符串。下面这个示例中，我们在正则表达式里用()表示的就是要提取的分组。



## 编译并使用正则表达式

如果一个正则表达式需要经常使用尝试匹配很多字符串，我们一般先把这个正则编译(compile)成一个正则表达式对象，然后在后续程序中直接只用并匹配字符串。编译后生成Regular Expression对象，由于该对象自己包含了正则表达式，所以调用对应的方法时不用给出正则字符串。

