

贪婪和非贪婪模式

Python里数量词默认是贪婪的（在少数语言里也可能是默认非贪婪），总是尝试匹配尽可能多的字符；

非贪婪则相反，总是尝试匹配尽可能少的字符。

在 `*`, `?`, `+`, `{m,n}` 后面加上 `?` 使贪婪变成非贪婪。

```
>>> s="This is a number 234-235-22-423"
>>> r=re.match(".*(\d+-\d+-\d+-\d+)",s)
>>> r.group(1)
'4-235-22-423'
>>> r=re.match(".*?(\d+-\d+-\d+-\d+)",s)
>>> r.group(1)
'234-235-22-423'
>>>
```

正则表达式模式中使用到通配字，那它在从左到右的顺序求值时，会尽量“抓取”满足匹配最长字符串，在我们上面的例子里面，“.”会从字符串的起始处抓取满足模式的最长字符，其中包括我们想得到的第一个整型字段中的大部分，“\d+”只需一位字符就可以匹配，所以它匹配了数字“4”，而“.”则匹配了从字符串起始到这个第一位数字4之前的所有字符。

```
>>> re.match(r"aa(\d+)", "aa2343ddd").group(1)
'2343'
>>> re.match(r"aa(\d+?)", "aa2343ddd").group(1)
'2'
>>> re.match(r"aa(\d+)ddd", "aa2343ddd").group(1)
'2343'
>>> re.match(r"aa(\d+?)ddd", "aa2343ddd").group(1)
'2343'
>>>
```

练习：

字符串为：

```

```

请提取url地址

参考答案

```
re.search(r"https://.*?\.jpg", test_str)
```

千锋Python人工智能学院