不同语言的正则表达式api汇总

|  |  |  |
| --- | --- | --- |
|  | Js | java |
| 测试是否匹配 | RegExp对象 test() | Matcher matcher = pattern.matcher(str);      // 字符串是否与正则表达式相匹配      boolean rs = matcher.matches(); |
| 提取匹配信息 | RegExp。exec() | Matcher m = Pattern.matcher(str);  **while** (m.find()) { >>((m.group()));  } |
|  |  |  |
|  |  |  |
|  |  |  |

Python

## re.match与re.search的区别

Php

### **使用正则进行匹配**

在php中，提供了preg\_math()和preg\_match\_all函数进行正则匹配。关于这两个函数原型如下：

int preg\_match|preg\_match\_all ( string $pattern , string $subject [, array &$matches [, int $flags = 0 [, int $offset = 0 ]]] )

搜索subject与pattern给定的正则表达式的一个匹配.   
pattern:要搜索的模式，字符串类型。   
subject :输入字符串。   
matches:如果提供了参数matches，它将被填充为搜索结果。 matches[0]将包含完整模式匹配到的文本，matches[1]将包含第一个捕获子组匹配到的文本，以此类推。   
flags:flags可以被设置为以下标记值：PREG\_OFFSET\_CAPTURE 如果传递了这个标记，对于每一个出现的匹配返回时会附加字符串偏移量(相对于目标字符串的)。 注意：这会改变填充到matches参数的数组，使其每个元素成为一个由 第0个元素是匹配到的字符串，第1个元素是该匹配字符串 在目标字符串subject中的偏移量。   
offset:通常，搜索从目标字符串的开始位置开始。可选参数 offset 用于 指定从目标字符串的某个未知开始搜索(单位是字节)。   
返回值：preg\_match()返回 pattern 的匹配次数。 它的值将是0次（不匹配）或1次，因为 preg\_match()在第一次匹配后 将会停止搜索。 preg\_match\_all()不同于此，它会一直搜索subject直到到达结尾。 如果发生错误 preg\_match()返回 FALSE。