###################### 通配符

‘ . ‘可以匹配除\n以外的任何字符，匹配字符’ . ‘时候必须用’ \ ’进行转义（使用re.S可以使.匹配\n）

注意区别>>> re.match(r’\w+.py’, ‘wkndpy’)

>>> re.match(r’\w\.py’, ‘elkn.py’)

###################### 字符集

‘ [a-zA-Z0-9] ‘可以匹配任意数字和大小写字母

‘ [^a-z] ’ 可以匹配小写字母外任何字符

###################### 选择符

r‘ python|perl ’ 只能匹配’python’或者’perl’

###################### 子模式

用’ ( ) ‘

###################### 可选匹配

子模式符( ) 后加上‘ ? ’ 表示( )中的内容可有可无

r’(http://)?(www\.)?python\.org’

可以匹配以下字符：

‘http://www.python.org’

‘http://python.org’

‘www.python.org’

‘python.org’

###################### 重复匹配

(abcde)\* 重复匹配0次或N次

(abcde)+ 重复匹配1次或N次

(abcde){m,n} 重复匹配m~n次

\w\* \s\* \d\*

###################### 开头或结尾

r’^abc\.py$’ 必须以a开头，以y结尾的字符串

###################### 用re进行字符串查找和替换（查找一个字符串中多个满足正则的字段）

strinfo = re.compile(正则表达式)

查找：查找的内容 = strinfo.findall(字符串)

替换：新字符串 = strinfo.sub(替换后的内容, 包含被替换内容的字符串)

例如查找cookie = “name: ‘Fujie’, gender: ‘male’, age: 23”中的所有属性名称

strinfo = re.compile(r’\w+:’)

attr = list(map(lambda item: item[:-1], strinfo.findall(cookie))) # [‘name’, ‘gender’, ‘age’]

或者

新字符串 = re.sub(正则表达式, 替换的内容, 包含被替换内容的字符串)

###################### 分组后提取字符串

result = re.match(r”^(\w+)\.(py)$”, “ab123.py”)

result.group()或result.group(0) # 得到匹配到的整个字符串

result.group(1) # 得到匹配到的子串”ab123”

result.group(2) # 得到匹配到的子串”py”

###################### 贪婪匹配

‘?’紧跟在{n,m}或者+,\*等限制符之后，匹配尽可能少的字符串

贪婪匹配>>> re.match(r’\w+?’, ‘wkndpdday’) 只会匹配到最少的字符，即w

>>> re.match(r’\w+’, ‘wkndpdday’) 默认采用贪婪匹配，结果为wkndpdday

###################### 忽略大小写

re.match(r”^(\w+)\.(py)$”, “ab123.py”, re.IGNORECASE)

re.I 大小写不敏感

re.L 做本地化识别（locale-aware）匹配

re.M 多行匹配，影响^和$

re.S 使.匹配包括换行符在内的所有字符

re.U 根据unicode字符集匹配字符，这个标志影响\w \W \b \B

re.X 该标志通过给予你更灵活的格式以便使你将正则表达式写的更容易