查找方法的使用

在Python中的查找匹配方法，常见的有下面四种，他们的用法大致相同，但是匹配出的结果却不同。

- match方法(只匹配字符串开头)
- search方法(扫描整个字符串，找到第一个匹配)
- findall方法(扫描整个字符串，找到所有的匹配)
- finditer方法(扫描整个字符串，找到所有的匹配，并返回一个可迭代对象)

match方法的使用

re.match尝试从字符串的起始位置匹配一个模式，如果不是起始位置匹配成功的话，match()就返回none。

函数语法:

```
re.match(pattern,string,flags=0)
```

参数	描述
pattern	匹配的正则表达式
string	要匹配的字符串。
flags	标志位，用于控制正则表达式的匹配方式，如：是否区分大小写，多行匹配等等。

我们可以使用group(num)函数来获取匹配表达式。

```
import re
result1 = re.match(r'H','Hello')
result2 = re.match(r'e','Hello')
print(result1.group(0)) # 'H' 匹配到的元素
print(result1.span()) # (0,1) 匹配到的元素所在位置
print(result2)  # None
```

search方法的使用

re.search 扫描整个字符串并返回第一个成功的匹配。

函数语法:

```
re.search(pattern, string, flags=0)
```

示例:

```
import re
result1 = re.search(r'He','Hello')
result2 = re.search(r'lo','Hello')

print(result1.group(0))  # He
print(result1.span()) # (0,2)
print(result2.group(0)) # lo
print(result2.span()) # (3,5)
```

re.match与re.search的区别

re.match只匹配字符串的开始，如果字符串开始不符合正则表达式，则匹配失败，函数返回None；而re.search匹配整个字符串，直到找到一个匹配。

示例:

```
result1 = re.search(r'天气','今天天气不错哟')
result2 = re.match(r'天气','今天天气不错哟')
print(result1)  # <re.Match object; span=(2, 4), match='天气'>
print(result2) # None
```

findall 方法的使用

在字符串中找到正则表达式所匹配的所有子串，并返回一个列表，如果没有找到匹配的，则返回空列表。

注意： match 和 search 是匹配一次 findall 匹配所有。

语法格式:

```
re.findall(pattern,string,flags=0)
```

示例代码:

```
ret = re.findall(r'\d+','he23ll34')
print(ret)  # ['23', '34']
ret = re.match(r'\d+','he23ll34')
print(ret) # None match只匹配开头，所以匹配到
ret = re.search(r'\d+','he23ll34')
print(ret) # <re.Match object; span=(2, 4), match='23'> search 只能匹配到一个数字
```

- 注意事项:

  findall方法匹配时，如果匹配规则里有分组，则只匹配分组数据。

```
ret = re.findall(r'\w+@(qq|126|163)\.com','123@qq.com;aa@163.com;bb@126.com')
print(ret)  # ['qq', '163', '126']  只匹配到了分组里的内容
```

如果正则表达式里存在多个分组，则会把多个分组匹配成元组。

```
ret = re.findall(r'\w+@(qq|126|163)(\.com)','123@qq.com;aa@163.com;bb@126.com'
)
print(ret) #[('qq', '.com'), ('163', '.com'), ('126', '.com')]
```

如果想要让findall匹配所有的内容，而不仅仅只是匹配正则表达式里的分组，可以使用 `?:` 来将分组标记为非捕获分组。

```
ret = re.findall(r'\w+@(?:qq|126|163)\.com','123@qq.com;aa@163.com;bb@126.com'
)
print(ret) # ['123@qq.com', 'aa@163.com', 'bb@126.com']
```

finditer方法的使用

和 findall 类似，在字符串中找到正则表达式所匹配的所有子串，并把它们作为一个迭代器返回。

```
ret = re.finditer(r'\d+','he23l134')  # 得到的结果是一个可迭代对象
for x in ret: # 遍历 ret 取出里面的每一项匹配
    print(x.group(), x.span()) # 匹配对象里的group保存了匹配的结果
```