

## re

```
--re.match(rule, target)
```

从字符串开头匹配，成功，匹配对象。失败，None。

```
--re.search(rule, target)
```

查找第一次出现

FE:

```
>>> my_str = 'app123董沉鑫'
```

```
>>> re.match('123', my_str)
```

```
>>> re.match('123', my_str).group()
```

Traceback (most recent call last):

```
File "<pyshell#14>", line 1, in <module>
```

```
    re.match('123', my_str).group()
```

AttributeError: 'NoneType' object has no attribute 'group'

```
>>> re.search('123', my_str).group()
```

```
'123'
```

```
--re.findall(pattern, string[, flag=re.M])
```

返回列表, flag可以有

re.I 忽略 大小写

re.M 多行模式

re.L 表示特色字符，依赖当前环境

re.S '.' 且包括换行符在内所有字符

re.U 同re.L，但依赖于Unicode字符库

```
--re.split(pattern, string[, nums])
```

根据表达式分割成列表，可设定次数

```
--re.sub(pattern, repl, string[, count=0][, flags=0])
```

返回替换后的字符串

FE:

```
>>> import re
```

```
>>> replacestr = re.sub(r'\d+', '222', inputstr)
```

```
>>> replacestr
```

```
'hello 222 world 222'
```

```
--re.subn(pattern, repl, string[, count=0][, flags=0])
```

返回(匹配后的字符串, 匹配次数)

FE:

```
>>> re.subn(r'\d+', '222', inputstr)
```

```
('hello 222 world 222', 2)
```

```
--re.group([int])
```

返回子组

```
--re.compile(rule)
```

没有match方法，有findall/find/search方法

FE:

```
>>> rule = r'(?:[\w_]+)'
```

```
>>> rule = re.compile(r'(?:[\w_]+)')
```

```
>>> rule.findall('@fef fef@fef')
```

```
['@fef', '@fef']
```

```
>>> rule.match('@fefef 你好啊')
```

Traceback (most recent call last):

```
File "<pyshell#13>", line 1, in <module>
    rule.match('@fefef 你好啊')
AttributeError: 'str' object has no attribute 'match'
```