

Regex In Depth 🧐



Python Mauritius UserGroup (pymug)

More info: mscc.mu/python-mauritius-usergroup-pymug/

Why	Where
codes	github.com/pymug
share events	twitter.com/pymugdotcom
ping professionals	linkedin.com/company/pymug
all info	pymug.com
tell friends by like	facebook.com/pymug

Abdur-Rahmaan Janhangeer

Python

Help people get into OpenSource

Regex In Depth 🧐

History of regex

(from notes)

```
import re

pattern = re.compile(r'eee')
result = pattern.match('eeee')

print(result.group())
```

eee

```
import re

pattern = re.compile(r'ike')
result = pattern.search('like sike')

print(result.group())
```

ike


```
import re

pattern = re.compile(r'ike')
result = pattern.findall('like sike')

print(result)
```

['ike', 'ike']

```
import re

pattern = r'ike'
string = 'like sike'

result = re.findall(pattern, string)

print(result)
```

```
pattern = re.compile(  
    r"...",  
    re.MULTILINE)
```

Patterns

```
pattern = r"letter"  
string = "i wrote some letters from letters"
```

```
['letter', 'letter']
```

```
pattern = r"l.tt.r"  
string = '''i wrote some letters  
from letters from the latter'''
```

['letter', 'letter', 'latter']

\d digit

```
pattern = r"\d\d\d.\d\d\d\d\d\d\d"
string = '''Personal: +230 5764321,
Office: +230 6712345'''
```

['230 5764321', '230 6712345']

\s - space

\D - non digit

```
try: r"\+\d\d\d\s\d\d\d\d\d\d"
```

```
r".\d\d\d\s\d\d\d\d\d\d"
```

```
r".\d\d\d\s\d\d\d\d\d\d"
```

```
pattern = r"\d\d\d\D\d\d\d\D\d\d\d"
string = '''Personal: +230 576-4321,
Office: +230 671 2345'''
```

```
r'\d'
```



```
pattern = r"\d{3}\D\d{3}\D\d{4}"  
string = '''Personal: +230 576-4321,  
Office: +230 671 2345'''
```

\w, \W

```
pattern = r"\w"  
string = '''abc123_!^%£&£%$'''
```

['a', 'b', 'c', '1', '2', '3', '_']

^ start, \$ end

```
pattern = r"^\\d$"  
string = '''234234234234234234'''
```

[abcd] range of chars

```
pattern = r"l[aeiou]ce"  
string = '''lice lace leece lyce lvce'''
```

['lice', 'lace']

try: [a-z] [A-Z] [0-9] [a-zA-z]

[^exclude]

```
pattern = r"l[^aeiou]ce"  
string = '''lice lace leece lyce lvce'''
```

['lyce', 'lvce']

Repeat x{1, 3}

```
pattern = r"lo{2,5} and behold"  
string = '''loo and behold lo and behold  
looooo and behold looo and behold  
loooooo and behold'''
```

['loo and behold', 'looooo and behold', 'looo and behold']

0 or more

```
pattern = r"lo* and behold"  
string = '''  
loo and behold  
lo and behold  
looooo and behold  
looo and behold  
loooooo and behold  
l and behold'''
```

['loo and behold', 'lo and behold', 'looooo and behold', 'looo
and behold', 'loooooo and behold', 'l and behold']

at least once

```
pattern = r"lo+ and behold"  
string = '''  
loo and behold  
lo and behold  
looooo and behold  
looo and behold  
loooooo and behold  
l and behold'''
```

['loo and behold', 'lo and behold', 'looooo and behold', 'looo
and behold', 'loooooo and behold']

0 or 1

```
pattern = r"lo? and behold"  
string = '''  
loo and behold  
lo and behold  
looooo and behold  
loo and behold  
loooooo and behold  
l and behold'''
```

['lo and behold', 'l and behold']

Capturing

```
pattern = re.compile(
    r"i bought (cat)?\s?(dog)?\s?(mouse)?"
)
string = '''
i bought cat
i bought dog
i bought mouse
i bought cat dog
i bought cat mouse
'''
```

```
[('cat', '', ''), ('', 'dog', ''), ('', '', 'mouse'), ('cat', 'dog', ''), ('cat', '', 'mouse')]
```

(?:)

```
pattern = re.compile(
    r"i bought (?:cat)?\s?(?:dog)?\s?(?:mouse)?"
)
string = '''
i bought cat
i bought dog
i bought mouse
i bought cat dog
i bought cat mouse
'''
```

['i bought cat \n', 'i bought dog\n', 'i bought mouse', 'i bought cat dog\n', 'i bought cat mouse']

Or

```
pattern = re.compile(
    r"(?:Mr|Mrs){1}\.\s\w*")
string = '''
Mrs. sam Dr. sam Mr. Sam Miss. Sam
'''
```

Capturing and backref

```
import re

pattern = re.compile(
    r"(a)l\1")
string = '''
ala alo ali
'''

result = re.search(pattern, string)

print(result.group())
```

look ahead if there is
not return match

```
pattern = re.compile(
    r"c(?=[aeiou])+")
string = '''
coo ca ci cw
'''
```

['c', 'c', 'c']

?! not followed by

?<= look behind

?<! not behind

Misc

\b