

Python 01

re

| Оператор | Описание |
|------------|---|
| . | Один любой символ, кроме новой строки \n. |
| ? | 0 или 1 вхождение шаблона слева |
| + | 1 и более вхождений шаблона слева |
| * | 0 и более вхождений шаблона слева |
| \w | Любая цифра или буква (\W — все, кроме буквы или цифры) |
| \d | Любая цифра [0-9] (\D — все, кроме цифры) |
| \s | Любой пробельный символ (\S — любой непробельный символ) |
| \b | Граница слова |
| [..] | Один из символов в скобках ([^..] — любой символ, кроме тех, что в скобках) |
| \ | Экранирование специальных символов (\. означает точку или \+ — знак «плюс») |
| ^ и \$ | Начало и конец строки |
| {n,m} | От n до m вхождений ({,m} — от 0 до m) |
| a b | a или b |
| () | Группирует выражение и возвращает найденный текст |
| \t, \n, \r | Символ табуляции, новой строки и возврата каретки соответственно |

r“.”

r“[.]”

‘hello python hello’

‘^hello’ ‘hello\$’

| Запись | Эквивалент |
|-----------------|-----------------------------|
| <code>\d</code> | <code>[0-9]</code> |
| <code>\D</code> | <code>[^0-9]</code> |
| <code>\s</code> | <code>[\t\n\r\f\v]</code> |
| <code>\S</code> | <code>[^ \t\n\r\f\v]</code> |
| <code>\w</code> | <code>[a-zA-Z0-9_]</code> |
| <code>\W</code> | <code>[^a-zA-Z0-9_]</code> |

match() & group()

- >>> import re
- >>> a='Hello Python Hello Django'
- >>> res1=re.match(r'Hello',a)
- >>> res1
- <_sre.SRE_Match object; span=(0, 5), match='Hello'>
- >>> print(res1)
- <_sre.SRE_Match object; span=(0, 5), match='Hello'>
- >>> res2=res1.group(0)
- >>> print (res2)
- Hello
- >>>
- >>> res3=res1.group()
- >>> res3
- 'Hello'

```
>>> import re
>>> a='Hello Python Hello Django'
>>> res1=re.match(r'Hello',a)
>>> res1
<_sre.SRE_Match object; span=(0, 5), match='Hello'>
>>> print(res1)
<_sre.SRE_Match object; span=(0, 5), match='Hello'>
>>> res2=res1.group(0)
>>> print (res2)
Hello
>>> res3=res1.group()
>>> res3
'Hello'
```

compile()

- >>> b=re.compile('Hello')
- >>> res11=b.findall(a)
- >>> res11_=re.findall(r'Hello',a)
- >>> res11
- ['Hello', 'Hello']

```
>>> b=re.compile('Hello')
>>> res11=b.findall(a)
>>> res11
['Hello', 'Hello', 'Hello']
```

Поиск по шаблону и поиск ПОЗИЦИИ

- `>>> p1=re.compile('[a-z]+')`
- `>>> p1`
- `re.compile('[a-z]+')`
- `>>> print(p1)`
- `re.compile('[a-z]+')`
- `>>> m1=p1.match('hello')`
- `>>> m1`
- `<_sre.SRE_Match object; span=(0, 5), match='hello'>`
- `>>> print(m1.group())`
- `hello`
- `>>> print(m1.span())`
- `(0, 5)`

```
>>> p1=re.compile('[a-z]+')
>>> p1
re.compile('[a-z]+')
>>> print(p1)
re.compile('[a-z]+')
>>> m1=p1.match('hello')
>>> m1
<_sre.SRE_Match object; span=(0, 5), match='hello'>
>>> print(m1.group())
hello
>>> print(m1.span())
(0, 5)
```

search() & findall()

- >>> res4=re.match(r'Python',a)
- >>> res4
- >>> print(res4)
- None
- >>> res5=re.search(r'Python',a)
- >>> print(res5.group())
- Python
- >>> a=a+' Hello'
- >>> a
- 'Hello Python Hello Django Hello'
- >>> res6=re.findall('Hello',a)
- >>> print(res6)
- ['Hello', 'Hello', 'Hello']
- >>>

```
>>> res5=re.search(r'Python',a)
>>> print(res5.group())
Python
>>> a=a+' Hello'
>>> a
'Hello Python Hello Django Hello'
>>> res6=re.findall('Hello',a)
>>> print(res6)
['Hello', 'Hello', 'Hello']
>>>
```


Поиск по шаблону

- >>> p2=re.compile('\d+')
• >>> p1=re.compile('[0-9]+')
- >>> a3='1 Hello 2 Python, 3 Hello 4 Django'
- >>> p1.findall(a3)
- ['1', '2', '3', '4']
- >>> p2.findall(a3)
- ['1', '2', '3', '4']
- >>>

```
>>> p2=re.compile('\d+')
>>> p1=re.compile('[0-9]+')
>>> a3='1 Hello 2 Python, 3 Hello 4 Django'
>>> p1.findall(a3)
['1', '2', '3', '4']
>>> p2.findall(a3)
['1', '2', '3', '4']
...

```

Итератор finditer()

- >>> iterator=p1.finditer(a3)
- >>> for i in iterator:
- print (i.span(), ' ', i.group())
- #>>> iterator=p1.finditer(a3)
- #>>> for i in iterator:
- # print (i.group())

```
>>> iterator=p1.finditer(a3)
>>> for i in iterator:
        print (i.span(), ' ', i.group())
```

-
- (0, 1) 1
- (8, 9) 2
- (18, 19) 3
- (26, 27) 4

```
(0, 1) 1
(8, 9) 2
(18, 19) 3
(26, 27) 4
... |
```

split(), maxsplit – число разделений СТРОКИ

- >>> res7=re.split('Hello',a)
- >>> res7
- ['', ' Python ', ' Django ', '']
- >>> res9=re.split('Hello',a,maxsplit=1)
- >>> res9
- ['', ' Python Hello Django Hello']
- >>> res9=re.split('Hello',a,maxsplit=2)
- >>> res9
- ['', ' Python ', ' Django Hello']
- >>>

```
>>> a=a+' Hello'
>>> a
'Hello Python Hello Django Hello'
>>> res6=re.findall('Hello',a)
>>> print(res6)
['Hello', 'Hello', 'Hello']
>>> res7=re.split('Hello',a)
>>> res7
['', ' Python ', ' Django ', '']
>>> res9=re.split('Hello',a,maxsplit=1)
>>> res9
['', ' Python Hello Django Hello']
>>> res9=re.split('Hello',a,maxsplit=2)
>>> res9
['', ' Python ', ' Django Hello']
>>>
```

sub()

- `>>> res10=re.sub('WOW', 'Hello',a)`
- `>>> res10`
- `'Hello Python Hello Django Hello'`
- `>>> res10=re.sub('Hello','WOW',a)`
- `>>> res10`
- `'WOW Python WOW Django WOW'`

```
>>> res10=re.sub('Hello','WOW',a)
>>> res10
'WOW Python WOW Django WOW'
```

sub() & compile()

- `>>> p=re.compile(r'(exe|py|htm|html)')`
 - `>>> p.sub('files','i can use exe')`
 - `'i can use files'`
 - `>>> p.sub('files','i can use exe and py')`
 - `'i can use files and files'`
 - `>>> p.sub('files','i can use exe and py',count=1)`
 - `'i can use files and py'`
- ```
>>> p=re.compile(r'(exe|py|htm|html)')
>>> p.sub('files','i can use exe')
'i can use files'
>>> p.sub('files','i can use exe and py')
'i can use files and files'
>>> p.sub('files','i can use exe and py',count=1)
'i can use files and py'
>>> |
```
-

# Вставка разделителей

- `>>> p=re.compile('x*')`
- `>>> p.sub('-', 'abcdefg')`
- `'-a-b-c-d-e-f-g-'`

```
>>> p=re.compile('x*')
>>> p.sub('-', 'abcdefg')
'-a-b-c-d-e-f-g-'
>>> |
```

# Смена десятичной размерности на шестнадцатеричную в строке

- `>>> def change(m):`
- `val=int(m.group())`
- `return hex(val)`
  
- `>>> p=re.compile(r'\d+')`
- `>>> p.sub(change,'1000 pages in 10 sites')`
- `'0x3e8 pages in 0xa sites'`

# match() & search()

- `>>> print(re.match('super','superclass').span())`
- `(0, 5)`
- `>>> print(re.match('super','superclass').group())`
- `super`
- `>>> print(re.match('super','exsuperclass').span())`
- Traceback (most recent call last):
- File "`<pyshell#389>`", line 1, in `<module>`
- `print(re.match('super','exsuperclass').span())`
- `AttributeError: 'NoneType' object has no attribute 'span'`
- `>>> print(re.search('super','exsuperclass').span())`
- `(2, 7)`
- `>>> print(re.search('super','exsuperclass').group())`
- `super`



# subn()

- >>> p=re.compile(r'(exe|py|htm|html)')
- >>> p.subn('files','i can use exe files')
- ('i can use files files', 1)
- >>> p.subn('files','i can use py and exe')
- ('i can use files and files', 2)

```
>>> p=re.compile(r'(exe|py|htm|html)')
>>> p.subn('files','i can use exe files')
('i can use files files', 1)
```

```
>>> p.subn('files','i can use py and exe')
('i can use files and files', 2)
```

# Поиск всех символов, символов без пробелов, слов, слов в начале и конце строки

- `>>> res12=re.findall(r'.',a, re.DOTALL)`
- `>>> print(res12)`
- `['H', 'e', 'l', 'l', 'o', ' ', 'P', 'y', 't', 'h', 'o', 'n', ' ', 'H', 'e', 'l', 'l', 'o', ' ', 'D', 'j', 'a', 'n', 'g', 'o', ' ', 'H', 'e', 'l', 'l', 'o']`
- `>>> res13=re.findall(r'\w',a)`
- `>>> print(res13)`
- `['H', 'e', 'l', 'l', 'o', 'P', 'y', 't', 'h', 'o', 'n', 'H', 'e', 'l', 'l', 'o', 'D', 'j', 'a', 'n', 'g', 'o', 'H', 'e', 'l', 'l', 'o']`
- `>>> res14=re.findall(r'\w+',a)`
- `>>> print(res14)`
- `['Hello', 'Python', 'Hello', 'Django', 'Hello']`
- `>>> res15=re.findall('^w+',a)`
- `>>> res15`
- `['Hello']`
- `>>> res16=re.findall('\w+$',a)`
- `>>> res16`
- `['Hello']`

```
>>> res12=re.findall(r'.',a)
>>> print(res12)
['H', 'e', 'l', 'l', 'o', ' ', 'P', 'y', 't', 'h', 'o', 'n', ' ', 'H', 'e', 'l', 'l', 'o', ' ', 'D', 'j', 'a', 'n', 'g', 'o', ' ', 'H', 'e', 'l', 'l', 'o']
>>> res13=re.findall(r'\w',a)
>>> print(res13)
['H', 'e', 'l', 'l', 'o', 'P', 'y', 't', 'h', 'o', 'n', 'H', 'e', 'l', 'l', 'o', 'D', 'j', 'a', 'n', 'g', 'o', 'H', 'e', 'l', 'l', 'o']
>>> res14=re.findall(r'\w+',a)
>>> print(res14)
['Hello', 'Python', 'Hello', 'Django', 'Hello']
>>> res15=re.findall('^w+',a)
>>> res15
['Hello']
>>> res16=re.findall('\w+$',a)
>>> res16
['Hello']
```

- `>>> res17=re.findall('\w*',a)`
- `>>> res17`
- `['Hello', '', 'Python', '', 'Hello', '', 'Django', '', 'Hello', '']`

```
>>> res17=re.findall('\w*',a)
>>> res17
['Hello', '', 'Python', '', 'Hello', '', 'Django', '', 'Hello', '']
>>>
```

# Первые три символа каждого слова

- `>>> res18=re.findall('\w'*3,a)`
- `>>> res18`
- `['Hel', 'Pyt', 'hon', 'Hel', 'Dja', 'ngo', 'Hel']`
- `>>> res18=re.findall('\w\w\w',a)`
- `>>> res18`
- `['Hel', 'Pyt', 'hon', 'Hel', 'Dja', 'ngo', 'Hel']`

```
>>> res18=re.findall('\w'*3,a)
>>> res18
['Hel', 'Pyt', 'hon', 'Hel', 'Dja', 'ngo', 'Hel']
>>> res18=re.findall('\w\w\w',a)
>>> res18
['Hel', 'Pyt', 'hon', 'Hel', 'Dja', 'ngo', 'Hel']
>>> |
```

# Первые символы, используя границу слова \b

- >>> res19=re.findall(r'\b\w.',a)
- >>> res19
- ['He', 'Py', 'He', 'Dj', 'He']
- >>> res19=re.findall('\b\w.',a)
- >>> res19
- ['He', 'Py', 'He', 'Dj', 'He']
- >>> res19=re.findall('\b\w.',a)
- >>> res19
- []

```
>>> res19=re.findall(r'\b\w.',a)
>>> res19
['He', 'Py', 'He', 'Dj', 'He']
>>> res19=re.findall('\b\w.',a)
>>> res19
['He', 'Py', 'He', 'Dj', 'He']
>>> res19=re.findall('\b\w.',a)
>>> res19
[]
```

# Извлечение имен доменов

- >>> c1='vasya@mail.ru petya@yandex.ru seryazha@gmail.com'
- >>> c2=r'http://www.ifmo.ru http://openedu.ru http://python.org'
- >>> res20=re.findall(r'@\w+',c1)
- >>> res20
- ['@mail', '@yandex', '@gmail']
- >>> res20=re.findall(r'@\w+.\w+',c1)
- >>> res20=re.findall(r'@\w+[.]w+',c1)
- >>> res20
- ['@mail.ru', '@yandex.ru', '@gmail.com']
- >>> res21=re.findall(r'\w+.\w+',c1)
- >>> res21
- ['mail', 'petya', 'ru', 'gmail', 'm']
- >>> res21=re.findall(r'@\w+.\w+',c1)
- >>> res21
- ['ru', 'ru', 'com']

```
>>> c1='vasya@mail.ru petya@yandex.ru seryazha@gmail.com'
>>> c2=r'http://www.ifmo.ru http://openedu.ru http://python.org'
>>> res20=re.findall(r'@\w+',c1)
>>> res20
['@mail', '@yandex', '@gmail']
>>> res20=re.findall(r'@\w+.\w+',c1)
>>> res20=re.findall(r'@\w+[.]w+',c1)
>>> res20
['@mail.ru', '@yandex.ru', '@gmail.com']
>>> res21=re.findall(r'\w+.\w+',c1)
>>> res21
['mail', 'petya', 'ru', 'gmail', 'm']
>>> res21=re.findall(r'@\w+.\w+',c1)
>>> res21
['ru', 'ru', 'com']
```

- `>>> res22=re.findall(r'http://(\w+).(\w+)',c2)`
- `>>> res22`
- `[('www', 'ifmo'), ('openedu', 'ru'), ('python', 'org')]`

# Извлечение телефонных номеров и последних знаков номеров

- `>>> d1='Vasya 222-22-22 Petya 333-33-33 Seryezha 444-44-44'`
- `>>> res23=re.findall(r'\d{3}-\d{2}-\d{2}',d1)`
- `>>> res23`
- `['222-22-22', '333-33-33', '444-44-44']`
- `>>> res23=re.findall(r'\d{3}-\d{2}-(\d{2})',d1)`
- `>>> res23`
- `['22', '33', '44']`
- `>>> res23=re.findall(r'\d{3}-(\d{2})-(\d{2})',d1)`
- `>>> res23`
- `[('22', '22'), ('33', '33'), ('44', '44')]`
- `>>> res23=re.findall(r'\d{3}-(\d{2}-\d{2})',d1)`
- `>>> res23`
- `['22-22', '33-33', '44-44']`



```
>>> d1='Vasya 222-22-22 Petya 333-33-33 Seryezha 444-44-44'
```

```
>>> res23=re.findall(r'\d{3}-\d{2}-\d{2}',d1)
```

```
>>> res23
```

```
['222-22-22', '333-33-33', '444-44-44']
```

```
>>> res23=re.findall(r'\d{3}-\d{2}-\d{2}',d1)
```

```
>>> res23
```

```
['22', '33', '44']
```

```
>>> res23=re.findall(r'\d{3}-\d{2}-\d{2}',d1)
```

```
>>> res23
```

```
[('22', '22'), ('33', '33'), ('44', '44')]
```

```
>>> res23=re.findall(r'\d{3}-\d{2}-\d{2}',d1)
```

```
>>> res23
```

```
['22-22', '33-33', '44-44']
```

```
>>>
```

# Поиск по набору символов

- `>>> a=a.replace('Hello', 'Yellow')`
- `>>> a`
- `'Yellow Python Yellow Django Yellow'`
- `>>> res24=re.findall(r'\w+',a)`
- `>>> res24`
- `['Yellow', 'Python', 'Yellow', 'Django', 'Yellow']`
- `>>> res24=re.findall(r'[yYdD]\w+',a)`
- `>>> res24`
- `['Yellow', 'ython', 'Yellow', 'Django', 'Yellow']`
- `>>> res24=re.findall(r'\b[yYdD]\w+',a)`
- `>>> res24`
- `['Yellow', 'Yellow', 'Django', 'Yellow']`

```
>>> a=a.replace('Hello', 'Yellow')
>>> a
'Yellow Python Yellow Django Yellow'
>>> res24=re.findall(r'\w+',a)
>>> res24
['Yellow', 'Python', 'Yellow', 'Django', 'Yellow']
>>> res24=re.findall(r'[yYdD]\w+',a)
>>> res24
['Yellow', 'ython', 'Yellow', 'Django', 'Yellow']
>>> res24=re.findall(r'\b[yYdD]\w+',a)
>>> res24
['Yellow', 'Yellow', 'Django', 'Yellow']
>>>
```

# Инвертирование группы

- `>>> res24=re.findall(r'\b[^yYdD]\w+',a)`
  - `>>> res24`
  - `[' Python', ' Yellow', ' Django', ' Yellow']`
  - `>>> res24=re.findall(r'\b[^yYdD ]\w+',a)`
  - `>>> res24`
  - `['Python']`
- ```
>>> res24=re.findall(r'\b[^yYdD]\w+',a)
>>> res24
[' Python', ' Yellow', ' Django', ' Yellow']
>>> res24=re.findall(r'\b[^yYdD ]\w+',a)
>>> res24
['Python']
```

Проверка на правильно введенный телефонный номер

- >>> t1='9513511 93279-19 95004x45'
- >>> t2=t1.split(' ')
- >>> t2
- ['9513511', '93279-19', '95004x45']
- >>> for i in t2:
- if re.match(r'[1-9]{1}[0-9]{6}',i)and len(i)==7:
- print(i,' yes')
- else:
- print(i, ' no')

-
- 9513511 yes
- 93279-19 no
- 95004x45 no

```
>>> for i in t2:
    if re.match(r'[1-9]{1}[0-9]{6}',i)and len(i)==7:
        print(i, ' yes')
    else:
        print(i, ' no')
```

```
9513511 yes
93279-19 no
95004x45 no
```

Использование нескольких разделителей и замена их пробелами

- `>>> a`
- `'Yellow Python Yellow Django Yellow'`
- `>>> res01=re.sub(r'[eo]',' ',a)`
- `>>> res01`
- `'Y ll w Pyth n Y ll w Djang Y ll w'`

Проверка на наличие

<h1></h1>

- `t1=r'<html><head><title>Hello</title></head><body><h1>MyPage</h1><td>1IvanIvanov2PetrPetrov3SidorSidorov</td></body></html>'`
- `>>> res03=re.findall(r'<h\d+>|</h\d+>',t1)`
- `>>> res03`
- `['<h1>', '</h1>']`

```
>>> t1=r'<html><head><title>Hello</title></head><body><h1>MyPage</h1><td>1IvanIvanov2PetrPetrov3SidorSidorov</td></body></html>'
```

```
>>> res03=re.findall(r'<h\d+>|</h\d+>',t1)
>>> res03
['<h1>', '</h1>']
...
```

Поиск имен и Фамилий в тексте

- `>>> res04=re.findall(r'\d([A-Z][A-Za-z]+)',t1)`
- `>>> res04`
- `['IvanIvanov', 'PetrPetrov', 'SidorSidorov']`
- `>>> res04=re.findall(r'\d([A-Z][A-Za-z]+)([A-Z][A-Za-z]+)',t1)`
- `>>> res04`
- `[('Ivan', 'Ivanov'), ('Petr', 'Petrov'), ('Sidor', 'Sidorov')]`
- `>>> res05=dict(res04)`
- `>>> res05`
- `{'Ivan': 'Ivanov', 'Petr': 'Petrov', 'Sidor': 'Sidorov'}`

```
>>> res04=re.findall(r'\d([A-Z][A-Za-z]+)',t1)
>>> res04
['IvanIvanov', 'PetrPetrov', 'SidorSidorov']
>>> res04=re.findall(r'\d([A-Z][A-Za-z]+)([A-Z][A-Za-z]+)',t1)
>>> res04
[('Ivan', 'Ivanov'), ('Petr', 'Petrov'), ('Sidor', 'Sidorov')]
>>> res05=dict(res04)
>>> res05
{'Ivan': 'Ivanov', 'Petr': 'Petrov', 'Sidor': 'Sidorov'}
```

Дополнительные флаги

| Флаг | Значение |
|---------------|--|
| IGNORECASE, I | Без учета регистра |
| LOCALE, L | Учет локализации |
| MULTILINE, M | Учет записи в несколько строк |
| UNICODE, U | Делает \w, \W, \b, \B, \d, \D, \s, \S соответствующими таблице Unicode |
| VERBOSE, X | Возможность многостроковых выражений |

Учет записи в несколько строк

- `>>> a4=""1 Hello`
- `2 Python`
- `3 Hello`
- `4 Django""`
- `>>> p3=re.compile('^d+')`
- `>>> p3.findall(a4)`
- `['1']`
- `>>> p3.findall(a4,re.MULTILINE) # не спасает`
- `[]`
- `>>> p3=re.compile('^d+',re.MULTILINE)`
- `>>> p3.findall(a4)`
- `['1', '2', '3', '4']`

```
>>> a4=""1 Hello
2 Python
3 Hello
4 Django""
>>> p3=re.compile('^d+')
>>> p3.findall(a4)
['1']
>>> p3.findall(a4,re.MULTILINE)
[]
>>> p3=re.compile('^d+',re.MULTILINE)
>>> p3.findall(a4)
['1', '2', '3', '4']
>>>
```

Аналогичный поиск без созданного ранее шаблона

- `>>> res=re.findall('^d+',a4,re.MULTILINE)`
- `>>> res`
- `['1', '2', '3', '4']`
- `>>> res=re.findall('^d+',a4)`
- `>>> res`
- `['1']`

```
>>> res=re.findall('^d+',a4,re.MULTILINE)
>>> res
['1', '2', '3', '4']
>>> res=re.findall('^d+',a4)
>>> res
['1']
>>> |
```

re.IGNORECASE

- `>>> a5='Hello python Hello django'`
- `>>> res=re.findall('[A-Z]',a5)`
- `>>> res`
- `['H', 'H']`
- `>>> res=re.findall('[A-Z]',a5,re.IGNORECASE)`
- `>>> res`
- `['H', 'e', 'l', 'l', 'o', 'p', 'y', 't', 'h', 'o', 'n', 'H', 'e', 'l', 'l', 'o', 'd', 'j', 'a', 'n', 'g', 'o']`
- `>>> res=re.findall('([A-Z]+)',a5,re.IGNORECASE)`
- `>>> res`
- `['Hello', 'python', 'Hello', 'django']`

Группа без захвата содержимого

- `>>> m1=re.match('([dh])+','hellow python')`
- `>>> m1`
- `<_sre.SRE_Match object; span=(0, 1), match='h'>`
- `>>> m1.groups()`
- `('h',)`
- `>>> m2=re.match('(?:[dh])+','hellow python')`
- `>>> m2.groups()`
- `()`

Именованные группы

- `>>> p=re.compile(r'(?P<gname>\b\w+\b)')`
- `>>> m=p.search('Hello python')`
- `>>> m.group()`
- `'Hello'`
- `>>> m.group('gname')`
- `'Hello'`
- `>>> m.group(1)`
- `'Hello'`
- `>>> m.group(0)`
- `'Hello'`
- `>>>`

Простые опережающие проверки

- (?=...) Положительная проверка
- (?!...) Отрицательная проверка

Опережающие проверки шаблона

| Шаблон | Описание |
|----------|--|
| (?=...) | <i>lookahead assertion</i> , соответствует каждой позиции, сразу после которой начинается соответствие шаблону |
| (?!...) | <i>negative lookahead assertion</i> , соответствует каждой позиции, сразу после которой НЕ может начинаться шаблон |
| (?<=...) | <i>positive lookbehind assertion</i> , соответствует каждой позиции, которой может заканчиваться шаблон Длина шаблона должна быть фиксированной, (abc и a b) |
| (?<!...) | <i>negative lookbehind assertion</i> , соответствует каждой позиции, которой НЕ может заканчиваться шаблон |

- `>>> a='Python 3.3 Python 2.7 Python 3.5 Python 3.6'`
- `>>> p=re.compile('Python (?=3)')`
- `>>> p.findall(a)`
- `['Python ', 'Python ', 'Python ']`
- `# нашлись Python за которым идет 3`
- `>>> p=re.compile('Python (?!3)')`
- `>>> p.findall(a)`
- `['Python ']`
- `#нашлись Python за которым не идет 3`

- >>> a='Python3.3 Python2.7 Flask1.0 Python3.6 Django1.7 Python1.1'
- >>> p=re.compile('(?!Python)1')
- >>> p.findall(a)
- ['1']
- #1 НО ТОЛЬКО ЕСЛИ Python
- >>> p=re.compile('(?!Python)1')
- >>> p.findall(a)
- ['1', '1', '1']
- #1 НО ТОЛЬКО ЕСЛИ не Python

```

>>> a='Python3.3 Python2.7 Flask1.0 Python3.6 Django1.7 Python1.1'
>>> p=re.compile('(?!Python)1')
>>> p.findall(a)
['1', '1', '1']
>>> p=re.compile('(?!Python)1')
>>> p.findall(a)
['1']
>>>

```

Проверка на имя файла

- `>>> p=re.compile(r'.*[.]*$')`
- `>>> m=p.search('hello.exe')`
- `>>> m.group()`
- `'hello.exe'`
- `>>> m=p.search('hello')`
- `>>> m.group()`
- Traceback (most recent call last):
- File "`<pyshell#348>`", line 1, in `<module>`
- `m.group()`
- `AttributeError: 'NoneType' object has no attribute 'group'`

Проверка на расширение файла

НЕ exe

- `>>> p=re.compile(r'.*[.](?!exe$).*$', re.IGNORECASE)`
- `>>> m=p.search('hello.py')`
- `>>> m.group()`
- `'hello.py'`
- `>>> m=p.search('hello.exe')`
- `>>> m.group()`
- Traceback (most recent call last):
- File "<pysHELL#354>", line 1, in <module>
- m.group()
- AttributeError: 'NoneType' object has no attribute 'group'
- `p=re.compile(r'.*[.](?!exe$|py$).*$', re.IGNORECASE) # поиск не exe и не py`

Работа match() с html

- >>> h1=r'<html> <head> <title> Hello </title> </head> <body> <h1> Hi!</h1></body></html>'
- >>> len(h1)
- 79
- >>> print(re.match(r'<.*>',h1).span())
- (0, 79)
- >>> print(re.match(r'<.*>',h1).group())
- <html> <head> <title> Hello </title> </head> <body> <h1> Hi!</h1></body></html>
- >>> print(re.match(r'<.*?>',h1).span())
- (0, 6)
- >>> print(re.match(r'<.*?>',h1).group())
- <html>

```
>>> h1=r'<html> <head> <title> Hello </title> </head> <body> <h1> Hi!</h1></body>
</html>'
>>> print(re.match(r'<.*>',h1).span())
(0, 79)
>>> print(re.match(r'<.*>',h1).group())
<html> <head> <title> Hello </title> </head> <body> <h1> Hi!</h1></body></html>
>>> print(re.match(r'<.*?>',h1).span())
(0, 6)
>>> print(re.match(r'<.*?>',h1).group())
<html>
>>> |
```

ect

- >>> re.match(r'\w+@\w+\.\w+', 'vasya@ru')
- >>> re.match(r'\w+@\w+\.\w+', 'vasya@mail.ru')
- <_sre.SRE_Match object; span=(0, 13), match='vasya@mail.ru'>
- >>> p=re.compile(r'\w+@\w+\.\w+')
- >>> p.match('vasya@mail.ru')
- <_sre.SRE_Match object; span=(0, 13), match='vasya@mail.ru'>
- >>> p.match('vasya@ru')

```
>>> re.match(r'\w+@\w+\.\w+', 'vasya@ru')
>>> re.match(r'\w+@\w+\.\w+', 'vasya@mail.ru')
<_sre.SRE_Match object; span=(0, 13), match='vasya@mail.ru'>
>>> p=re.compile(r'\w+@\w+\.\w+')
>>> p.match('vasya@mail.ru')
<_sre.SRE_Match object; span=(0, 13), match='vasya@mail.ru'>
>>> p.match('vasya@ru')
>>> |
```

- >>> p2=re.compile(r'\w+@\w+[.]\w+')
- >>> p2.match('vasya@mail.ru')
- <_sre.SRE_Match object; span=(0, 13), match='vasya@mail.ru'>

```
>>> p2=re.compile(r'\w+@\w+[.]\w+')
>>> p2.match('vasya@mail.ru')
<_sre.SRE_Match object; span=(0, 13), match='vasya@mail.ru'>
```

Использование точки

- `>>> p3=re.compile(r'h.o')`
- `>>> p3.search('hello python')`
- `>>> p3=re.compile(r'h...o')`
- `>>> p3.search('hello python')`
- `<_sre.SRE_Match object; span=(0, 5), match='hello'>`

```
>>> p3=re.compile(r'h.o')
>>> p3.search('hello python')
>>> p3=re.compile(r'h...o')
>>> p3.search('hello python')
<_sre.SRE_Match object; span=(0, 5), match='hello'>
>>>
```

Использование квадратных скобок

- `>>> p4=re.compile(r'h[a-z]o')`
- `>>> p4.search('hello python').group()`
- Traceback (most recent call last):
- File "<pyshell#442>", line 1, in <module>
- `p4.search('hello python').group()`
- `AttributeError: 'NoneType' object has no attribute 'group'`
- `>>> p4=re.compile(r'h[a-z][a-z][a-z]o')`
- `>>> p4.search('hello python').group()`
- `'hello'`

```
>>> p4=re.compile(r'h[a-z]o')
>>> p4.search('hello python').group()
Traceback (most recent call last):
  File "<pyshell#442>", line 1, in <module>
    p4.search('hello python').group()
AttributeError: 'NoneType' object has no attribute 'group'
>>> p4=re.compile(r'h[a-z][a-z][a-z]o')
>>> p4.search('hello python').group()
'hello'|
```


w+ w{3} w{1,3}

- >>> p4=re.compile(r'h\w+o')
- >>> p4.search('hello python').group()
- 'hello'
- >>> p4=re.compile(r'h\wo')
- >>> p4.search('hello python')
- >>> p4=re.compile(r'h\w{1,3}o')
- >>> p4.search('hello python').group()
- 'hello'
- >>> p4=re.compile(r'h\w{3}o')
- >>> p4.search('hello python').group()
- 'hello'
- >>> p4=re.compile(r'h\w\w\wo')
- >>> p4.search('hello python').group()
- 'hello'
- >>>

```
>>> p4=re.compile(r'h\w+o')
>>> p4.search('hello python').group()
'hello'
>>> p4=re.compile(r'h\wo')
>>> p4.search('hello python')
>>> p4=re.compile(r'h\w{1,3}o')
>>> p4.search('hello python').group()
'hello'
>>> p4=re.compile(r'h\w{3}o')
>>> p4.search('hello python').group()
'hello'
>>> p4=re.compile(r'h\w\w\wo')
>>> p4.search('hello python').group()
'hello'
```

+ N *

- >>> p5=re.compile(r'h[a-z]+o')
- >>> p5.search('hello python').group()
- 'hello'
- >>> p6=re.compile(r'h[a-z]*o')
- >>> p6.search('hello python').group()
- 'hello'

```
>>> p5=re.compile(r'h[a-z]+o')
>>> p5.search('hello python').group()
'hello'
>>> p6=re.compile(r'h[a-z]*o')
>>> p6.search('hello python').group()
'hello'
....
```

Пробел решает многое

- `>>> p7=re.compile(r'^0-9]+o')`
- `>>> p7.search('hello python').group()`
- `'hello pytho'`
- `>>> p7=re.compile(r'^0-9]*o')`
- `>>> p7.search('hello python').group()`
- `'hello pytho'`
- `>>> p7=re.compile(r'^0-9]*o')`
- `>>> p7.search('hello python').group()`
- `'hello'`

```
>>> p7=re.compile(r'^0-9]+o')
>>> p7.search('hello python').group()
'hello pytho'
>>> p7=re.compile(r'^0-9]*o')
>>> p7.search('hello python').group()
'hello pytho'
>>> p7=re.compile(r'^0-9 ]*o')
>>> p7.search('hello python').group()
'hello'
>>>
```

Произвольное количество доп СИМВОЛОВ s^*

- `>>> import re`
- `>>> z=re.compile(r'\d\s*\d\s*\d')`
- `>>> zz=z.search('aa1 3 5zz')`
- `>>> zz`
- `<_sre.SRE_Match object; span=(2, 10), match='1 3 5'>`
- `>>> zz=z.search('aa135zz')`
- `>>> zz`
- `<_sre.SRE_Match object; span=(2, 5), match='135'>`

```
[>>> import re
[>>> z=re.compile(r'\d\s*\d\s*\d')
[>>> zz=z.search('aa1 3 5zz')
[>>> zz
<_sre.SRE_Match object; span=(2, 10), match='1 3 5'>
>>> █
```

Подбор выражения по шагам

- >>> a=r'Почта vasya-pupkin@openedu.ru или ivanpobeditel@mail.ru'
- >>> p8=re.compile(r'[\w+-]@[\w+-]')
- >>> p8.search(a)
- <_sre.SRE_Match object; span=(17, 20), match='n@o'>
- >>> p8=re.compile(r'[\w-]+@[\w-]+')
- >>> p8.findall(a)
- ['vasya-pupkin@openedu', 'ivanpobeditel@mail']
- >>> p8=re.compile(r'[\w-]+@[\w\.-]+')
- >>> p8.findall(a)
- ['vasya-pupkin@openedu.ru', 'ivanpobeditel@mail.ru']

```
>>> p8=re.compile(r'[\w+-]@[\w+-]')
>>> p8.search(a)
<_sre.SRE_Match object; span=(17, 20), match='n@o'>
>>> p8=re.compile(r'[\w-]+@[\w-]+')
>>> p8.findall(a)
['vasya-pupkin@openedu', 'ivanpobeditel@mail']
>>> p8=re.compile(r'[\w-]+@[\w\.-]+')
>>> p8.findall(a)
['vasya-pupkin@openedu.ru', 'ivanpobeditel@mail.ru']
>>>
```

- >>> a='mail send to dgopenedu@openedu.ru'
- >>> res=re.search(r'\w+@\w+',a)
- >>> res.group()
- 'dgopenedu@openedu'
- >>> res=re.search(r'\w+@\w+\.\w+',a)
- >>> res.group()
- 'dgopenedu@openedu.ru'
- >>> a='mail send to: d.g-openedu@openedu-ru.ru'
- >>> res=re.search(r'[\w\.-]+\@([\w\.-]+)',a)
- >>> res.group()
- 'd.g-openedu@openedu-ru.ru'

```
[>>> a='mail send to dgopenedu@openedu.ru'
>>> res=re.search(r'\w+@\w+',a)
>>> res.group()
'dgopenedu@openedu'
>>> res=re.search(r'\w+@\w+\.\w+',a)
>>> res.group()
'dgopenedu@openedu.ru'
>>> a='mail send to: d.g-openedu@openedu-ru.ru'
>>> res=re.search(r'[\w\.-]+\@([\w\.-]+)',a)
>>> res.group()
'd.g-openedu@openedu-ru.ru'
```

Группы СИМВОЛОВ

- `>>> p9=re.compile(r'([\w-]+)([\w\.-]+)')`
- `>>> p9.search(a)`
- `<_sre.SRE_Match object; span=(6, 29), match='vasya-pupkin@openedu.ru'>`
- `>>> p9.search(a).groups()`
- `('vasya-pupkin', 'openedu.ru')`
- `>>> p9.search(a).group()`
- `'vasya-pupkin@openedu.ru'`
- `>>> p9.search(a).group(1)`
- `'vasya-pupkin'`
- `>>> p9.search(a).group(2)`
- `'openedu.ru'`

```
>>> p9=re.compile(r'([\w-]+)([\w\.-]+)')
>>> p9.search(a)
<_sre.SRE_Match object; span=(6, 29), match='vasya-pupkin@openedu.ru'>
>>> p9.search(a).groups()
('vasya-pupkin', 'openedu.ru')
>>> p9.search(a).group()
'vasya-pupkin@openedu.ru'

>>> p9.search(a).group(1)
'vasya-pupkin'
>>> p9.search(a).group(2)
'openedu.ru'
```

- >>> p9=re.compile(r'([\w-]+)([\w\.-]+)')
 - >>> a='ivanov ivanov@mail.ru petrov petrov@gmail.com'
 - >>> p9.search(a)
 - <_sre.SRE_Match object; span=(7, 21), match='ivanov@mail.ru'>
 - >>> p99=p9.search(a)
 - >>> p99.groups()
 - ('ivanov', 'mail.ru')
 - >>> p99.group(1)
 - 'ivanov'
 - >>> p99.group(2)
 - 'mail.ru'
 - >>>
- ```
[>>> p9=re.compile(r'([\w-]+)([\w\.-]+)')
>>> a='ivanov ivanov@mail.ru petrov petrov@gmail.com'
>>> p9.search(a)
<_sre.SRE_Match object; span=(7, 21), match='ivanov@mail.ru'>
>>> p99=p9.search(a)
>>> p99.groups()
('ivanov', 'mail.ru')
>>> p99.group(1)
'ivanov'
>>> p99.group(2)
'mail.ru'
>>> █
```



# Поиск всех почтовых ящиков и разбиение их на группы

- `>>> a='1st email vasya@mail.ru 2nd email petya@gmail.com'`
- `>>> res=re.findall(r'[\w\.-]+\@[\w\.-]+',a)`
- `>>> res`
- `['vasya@mail.ru', 'petya@gmail.com']`
- `>>> res=re.findall(r'([\w\.-]+\@([\w\.-]+))',a)`
- `>>> res`
- `[('vasya', 'mail.ru'), ('petya', 'gmail.com')]`

```
[>>> a='1st email vasya@mail.ru 2nd email petya@gmail.com'
>>> res=re.findall(r'[\w\.-]+\@[\w\.-]+',a)
>>> res
['vasya@mail.ru', 'petya@gmail.com']
>>> res=re.findall(r'([\w\.-]+\@([\w\.-]+))',a)
>>> res
[('vasya', 'mail.ru'), ('petya', 'gmail.com')]
... ■
```

# Определение группы без вывода результата

- `>>> res=re.findall(r'(?:[\w\.-]+)@([\w\.-]+)',a)`
- `>>> res`
- `['mail.ru', 'gmail.com']`

```
[>>> res=re.findall(r'(?:[\w\.-]+)@([\w\.-]+)',a)
[>>> res
['mail.ru', 'gmail.com']
```

- `>>> res=re.sub(r'([\w\.-]+)([\w\.-]+)',r'\1@openedu.ru',a)`
- `>>> res`
- `'1st email vasya@openedu.ru 2nd email petya@openedu.ru'`
- `>>> res=re.sub(r'([\w\.-]+)([\w\.-]+)',r'www@\2',a)`
- `>>> res`
- `'1st email www@mail.ru 2nd email www@gmail.com'`

```
[>>> res=re.sub(r'([\w\.-]+)([\w\.-]+)',r'\1@openedu.ru',a)
[>>> res
'1st email vasya@openedu.ru 2nd email petya@openedu.ru'
```

```
[>>> res=re.sub(r'([\w\.-]+)([\w\.-]+)',r'www@\2',a)
[>>> res
'1st_email www@mail.ru 2nd_email www@gmail.com'
```

# Поиск заголовка в html

- `import re`
- `f=open('01re.html','r')`
- `str=f.read()`
- `#mytable=re.findall(r'<title>\w+</title>',str)`
- `mytable=re.findall(r'<title>(.*?)</title>',str)`
- `print(mytable)`

```
MacBook-Pro-Dmitrij:django2018 dmitrijstennikov$ python3 01re.py
['<title>Table</title>']
MacBook-Pro-Dmitrij:django2018 dmitrijstennikov$ python3 02re.py
['Table']
```



# Получен список по строкам

- `import re`
- `f=open('01re.html','r')`
- `str=f.read()`
- `mytable=re.findall(r'<tr>(.*?)</tr>',str,re.DOTALL)`
- `print(mytable)`

```
[MacBook-Pro-Dmitriy:django2018 dmitrijstennikov$ python3 04re.py
['\n\t\t\t<td>row 1 col 1</td>\n\t\t\t<td>row 1 col 2</td>\n\t\t\t<td>row 1 col 3</td>\n\t\t\t', '\n\t\t\t<td>row 2 col 1</td>\n\t\t\t<td>row 2 col 2</td>\n\t\t\t<td>row 2 col 3</td>\n\t\t\t', '\n\t\t\t<td>row 3 col 1</td>\n\t\t\t<td>row 3 col 2</td>\n\t\t\t<td>row 3 col 3</td>\n\t\t\t']
```

```
[MacBook-Pro-Dmitriy:django2018 dmitrijstennikov$ python3 04re.py
['\n <td>row 1 col 1</td>\n <td>row 1 col 2</td>\n <td>row 1 col 3</td>
\n ', '\n <td>row 2 col 1</td>\n <td>row 2 col 2</td>\n <td>row 2 col 3</td>\n ', '\n <td>row 3 col 1</td>\n <td>row 3 col 2</td>\n <td>row 3 col 3</td>\n ']
```

- import re
- f=open('01re.html','r')
- mystr=f.read()
- mytable=re.findall(r'''
  - <tr>\s\*?
  - <td>(.\*?)</td>\s\*?
  - <td>(.\*?)</td>\s\*?
  - <td>(.\*?)</td>\s\*?
  - </tr>''',
  - mystr,
  - re.DOTALL | re.VERBOSE)
- print(mytable)

```
[MacBook-Pro-Dmitrij:django2018 dmitrijstennikov$ python3 05re.py
[('row 1 col 1</td>\n <td>row 1 col 2</td>\n <td>row 1 col 3</td>\n </tr>\n <tr>\n <td>row 2 col 1</td>\n <td>row 2 col 2</td>\n <td>row 2 col 3</td>\n </tr>\n <tr>\n <td>row 3 col 1', 'row 3 col 2', 'row 3 col 3')]
```

- import re
- f=open('01re.html','r')
- mystr=f.read()
- mytable=re.findall(r'''
  - <tr>\s\*?
  - <td>(.\*?)</td>\s\*?
  - <td>(.\*?)</td>\s\*?
  - <td>(.\*?)</td>\s\*?
  - </tr>''',
  - mystr,
  - re.DOTALL | re.VERBOSE)
- print(mytable)

```
MacBook-Pro-Dmitrij:django2018 dmitrijstennikov$ python3 06re.py]
[('row 1 col 1', 'row 1 col 2', 'row 1 col 3'), ('row 2 col 1', 'row 2 col 2', '
row 2 col 3'), ('row 3 col 1', 'row 3 col 2', 'row 3 col 3')]
```



- import re
- f=open('01re.html','r')
- mystr=f.read()
- mytable=re.findall(r"""
  - <tr>\s\*?
  - <td>(.\*?)</td>\s\*?
  - <td>(.\*?)</td>\s\*?
  - <td>(.\*?)</td>\s\*?
  - </tr>""",
  - mystr,
  - re.DOTALL | re.VERBOSE)
- for i in mytable:
- for j in i:
- print(j,end= ' | ')
- print()

```

[MacBook-Pro-Dmitrij:django2018 dmitrijstennikov$ python3 07re.py
row 1 col 1 | row 1 col 2 | row 1 col 3 |
row 2 col 1 | row 2 col 2 | row 2 col 3 |
row 3 col 1 | row 3 col 2 | row 3 col 3 |

```

- `>>> z=p9.findall(a)`
- `>>> z`
- `[('vasya-pupkin', 'openedu.ru'), ('ivanpobeditel', 'mail.ru')]`