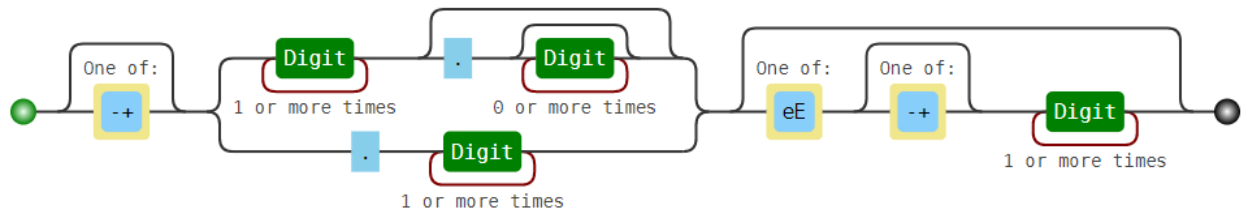


## Примеры регулярных выражений

| Регулярка   | Её смысл   |
|---|--|
| <code>simple text</code>                                    | В точности текст «simple text»   |
| <code>\d{5}</code>  | Последовательности из 5 цифр<br><code>\d</code> означает любую цифру<br><code>{5}</code> — ровно 5 раз   |
| <code>\d\d/\d\d/\d{4}</code>                                | Даты в формате ДД/ММ/ГГГГ<br>(и прочие куски, на них похожие,<br>например, 98/76/5432)   |
| <code>\b\w{3}\b</code>                                      | Слова в точности из трёх букв<br><code>\b</code> означает границу слова<br>(с одной стороны буква, а с другой —<br>нет)<br><code>\w</code> — любая буква,<br><code>{3}</code> — ровно три раза |
| <code>[-+]? \d+</code>                                      | Целое число, например, 7, +17, -42,<br>0013 (возможны ведущие нули)<br><code>[-+]?</code> — либо -, либо +, либо пусто<br><code>\d+</code> — последовательность из 1 или<br>более цифр         |
| <code>[-+]?(?:\d+(?:\.\d*)? \.\d+)(?:[eE][-+]? \d+)?</code> | Действительное число, возможно в<br>экспоненциальной записи<br>Например, 0.2, +5.45, -.4, 6e23, -<br>3.17E-14.<br>См. ниже картинку.   |

RegExp: `/[-+]?(?:\d+(?:\.\d*)?|\.\d+)(?:[eE][-+]?\d+)?/`



<https://www.debuggex.com/>

## Шаблоны, соответствующие одному символу

Во всех примерах ниже соответствия регулярному выражению выделяются бирюзовым цветом с подчёркиванием.

| Шаблон | Описание   | Пример   | Применяем к тексту   |
|--------|--|----------|--|
| .      | Один любой символ, кроме новой строки \n.                        | м.л.ко   | <u>молоко</u> , <u>малако</u> ,<br><u>ИмОлОко</u> Ихлеб    |
| \d     | Любая цифра  | СУ\d\d   | <u>СУ35</u> , <u>СУ111</u> ,<br><u>АЛСУ14</u>              |
| \D     | Любой символ, кроме цифры  | 926\D123 | <u>926)123</u> , <u>1926-1234</u>                          |
| \s     | Любой пробельный символ (пробел, табуляция, конец строки и т.п.) | бор\sода | <u>бор ода</u> , <u>бор ода</u> , борода                   |
| \S     | Любой непробельный символ  | \S123    | <u>X123</u> , <u>я123</u> , <u>!123</u> 456,<br>1 + 123456 |
| \w     | Любая буква (то, что может быть частью слова), а также           | \w\w\w   | <u>Год</u> , <u>f_3</u> , <u>qwert</u>                     |

| Шаблон  | Описание   | Пример           | Применяем к тексту   |
|---|--|------------------|--|
|   | цифры и _  |                  |  |
| \w  | Любая не-буква, не-цифра и не подчёркивание  | com\w            | <a href="#">com!</a> , <a href="#">com?</a>                  |
| [. . ]  | Один из символов в скобках, а также любой символ из диапазона a-b  | [0-9][0-9A-Fa-f] | <a href="#">12</a> , <a href="#">1F</a> , <a href="#">4B</a> |
| [^ . . ]  | Любой символ, кроме перечисленных  | <[^>]>           | <a href="#">&lt;1&gt;</a> , <a href="#">&lt;a&gt;</a> , <>   |
| \d=[0-9],<br>\D=[^0-9],<br>\w=[0-9a-zA-Z<br>a-яA-ЯёЁ],<br>\s=[<br>\f\n\r\t\v] | Буква “ё” не включается в общий диапазон букв!<br>Вообще говоря, в \d включается всё, что в юникоде помечено как «цифра», а в \w — как буква. Ещё много всего! |                  |  |
| [abc-], [-1]  | если нужен минус, его нужно указать последним или первым   |                  |  |
| [*[(+\\)\t]   | внутри скобок нужно экранировать только ] и \  |                  |  |
| \b  | Начало или конец слова (слева пусто или не-буква, справа буква и наоборот).  | \bвал            | <a href="#">вал</a> , перевал,                               |

| Шаблон | Описание  | Пример  | Применяем к тексту                        |
|--------|---|---------|---|
|        | В отличие от предыдущих соответствует позиции, а не символу                     |         | Перевалка                                 |
| \B     | Не граница слова: либо и слева, и справа буквы, либо и слева, и справа НЕ буквы | \Bвал   | пере <u>вал</u> , вал, Пере <u>вал</u> ка |
|        |   | \Bвал\B | перевал, вал, Пере <u>вал</u> ка          |

#### Квантификаторы (указание количества повторений)

| Шаблон | Описание                          | Пример  | Применяем к тексту                              |
|--------|-----------------------------------|---------|---|
| {n}    | Ровно n повторений                | \d{4}   | 1, 12, 123, <u>1234</u> , 12345                 |
| {m,n}  | От m до n повторений включительно | \d{2,4} | 1, <u>12</u> , <u>123</u> , <u>1234</u> , 12345 |
| {m,}   | Не менее m повторений             | \d{3,}  | 1, 12, <u>123</u> , <u>1234</u> , <u>12345</u>  |
| {,n}   | Не более n повторений             | \d{,2}  | <u>1</u> , <u>12</u> , <u>123</u>               |

| Шаблон                                      | Описание   | Пример             | Применяем к тексту   |
|---|--|--------------------|--|
| ?   | Ноль или одно вхождение, синоним {0,1}   | валы?              | <a href="#">вал</a> , <a href="#">валы</a> , <a href="#">валов</a>                         |
| *   | Ноль или более, синоним {0, }  | СУ\d*              | <a href="#">СУ</a> , <a href="#">СУ1</a> , <a href="#">СУ12</a> , ...                      |
| +   | Одно или более, синоним {1, }  | a\)+               | <a href="#">a</a> ), <a href="#">a))</a> , <a href="#">a)))</a> , <a href="#">ba)])</a>    |
| *?<br>+?<br>??<br>{m,n}?<br>{,n}?<br>{m,}?? | По умолчанию<br>квантификаторы <i>жадные</i> —<br>захватывают максимально возможное<br>число символов.<br>Добавление ? делает их <i>ленивыми</i> ,<br>они захватывают минимально возможное<br>число символов | \(.*\)<br>\(..*?\) | <a href="#">(a + b) * (c + d) * (e + f)</a><br><a href="#">(a + b)</a> * (c + d) * (e + f) |

## Регулярные выражения в питоне

Функции для работы с регулярками находятся в модуле re. Основные функции:

| Функция                       | Её смысл   |
|-------------------------------|--|
| re.search(pattern, string)    | Найти в строке string первую строчку, подходящую под шаблон pattern; |
| re.fullmatch(pattern, string) | Проверить, подходит ли строка string под шаблон pattern;             |

| Функция   | Её смысл   |
|---|--|
| <code>re.split(pattern, string, maxsplit=0)</code>  | Аналог <code>str.split()</code> , только разделение происходит по подстрокам, подходящим под шаблон <code>pattern</code> ;           |
| <code>re.findall(pattern, string)</code>            | Найти в строке <code>string</code> все непересекающиеся шаблоны <code>pattern</code> ;   |
| <code>re.finditer(pattern, string)</code>           | Итератор по всем непересекающимся шаблонам <code>pattern</code> в строке <code>string</code> (выдаются <code>match</code> -объекты); |
| <code>re.sub(pattern, repl, string, count=0)</code> | Заменить в строке <code>string</code> все непересекающиеся шаблоны <code>pattern</code> на <code>repl</code> ;                       |

### Пример использования всех основных функций

```
import re

match = re.search(r'\d\d\D\d\d', r'Телефон 123-12-12')
print(match[0] if match else 'Not found')
# -> 23-12

match = re.search(r'\d\d\D\d\d', r'Телефон 1231212')
print(match[0] if match else 'Not found')
# -> Not found

match = re.fullmatch(r'\d\d\D\d\d', r'12-12')
print('YES' if match else 'NO')
# -> YES

match = re.fullmatch(r'\d\d\D\d\d', r'T. 12-12')
print('YES' if match else 'NO')
# -> NO

print(re.split(r'\W+', 'Где, скажите мне, мои очки??!'))
# -> ['Где', 'скажите', 'мне', 'мои', 'очки', '']

print(re.findall(r'\d\d\.\d\d\.\d{4}',
                r'Эта строка написана 19.01.2018, а могла бы и 01.09.2017'))
```

```
# -> ['19.01.2018', '01.09.2017']

for m in re.finditer(r'\d\d\.\d\d\.\d{4}', r'Эта строка написана 19.01.2018, а могла бы и 01.09.2017'):
    print('Дата', m[0], 'начинается с позиции', m.start())
# -> Дата 19.01.2018 начинается с позиции 20
# -> Дата 01.09.2017 начинается с позиции 45

print(re.sub(r'\d\d\.\d\d\.\d{4}',
             r'DD.MM.YYYY',
             r'Эта строка написана 19.01.2018, а могла бы и 01.09.2017'))
# -> Эта строка написана DD.MM.YYYY, а могла бы и DD.MM.YYYY
```