

# Шпаргалка по регулярным выражениям

Регулярные выражения – это шаблоны, используемые для сопоставления комбинаций символов в строках, для поиска в строке, описываемые с помощью специального синтаксиса. В Python есть модуль **RE**, обеспечивающий поддержку работы с регулярными выражениями.

## Метасимволы

Символы	Объяснение
a	Соответствует символу a
ab	Соответствует строке ab
a   b	Соответствует a ИЛИ b
\$	Конец строки
i	Игнорировать регистр
s	Соответствует всему, включая перевод строки
u	Соответствует классам символов Unicode
x	Разрешает пробелы и комментарии (подробные)
^	Начало строки
*	От нуля вхождений. То же самое, что {0,}
+	От одного вхождения. То же самое, что {1,}
?	От нуля до одного вхождения. То же самое, что {0, 1}
{a, b}	Сопоставьте a с b раз
{a,}	Совпадает хотя бы раз
{,b}	Совпадение до b раз.
{a}	Совпадение ровно одному
{a, b}?	Сопоставляет выражение с a количеством раз и игнорирует b

## Специальные последовательности

Базовая концепция регулярных выражений состоит в поиске по образцу. Специальные последовательности ниже определяют набор(ы) символов. По заданной последовательности может быть найден обширный набор символов.

Классы	Объяснение
\d	Любая цифра. То же самое, что [0-9]
\D	Любой символ, кроме цифры. То же самое, что [^0-9]
\w	Любая буква, цифра и нижнее подчёркивание
\W	Любой символ, кроме буквы, цифры и нижнего подчёркивания
\s	Любой пробельный символ (пробел, новая строка, табуляция и тому подобное)
\S	Любой символ, кроме пробельного
\n	Соответствует символу новой строки
\t	Соответствует символу табуляции
\b	Начало или конец слова
\Z	Конец строки. То же самое, что \$
\A	Начало строки. То же самое, что ^

## Наборы

Определённая группа символов, заключённых в квадратные скобки. Результат поиска по такому шаблону (образцу) будет соответствовать любому из нескольких символов в наборе.

Наборы	Объяснение
[a-z]	Возвращает совпадение для любого символа нижнего регистра в алфавитном порядке от a до z
[xyz]	Возвращает совпадение, в котором присутствует один из указанных символов (x, y или z)
[x \ -z]	Соответствует x, или -z.
[-x]	Совпадение или -x.
[a-d0-9]	Соответствует символам от a до d или от 0 до 9.
[^ xy4]	Соответствует символам, которые не являются x, y или 4.
[0-5][0-9]	Совпадения для любых двузначных чисел от 00 до 59.
[^ ab5]	Добавление ^ исключает любой символ в наборе. Здесь оно соответствует символам, которые не являются a, b или 5.

## Методы модуля регулярных выражений (RE)

Модуль RE предлагает следующий набор методов:

Методы	Объяснение
re.match	Ищет pattern в начале строки string и возвращает Match-объект, если таковой имеется
re.search	Ищет pattern по всей строке string. Возвращает Match-объект с первым совпадением, остальные не находит
re.fullmatch	Возвращает объект соответствия тогда и только тогда, когда вся строка соответствует шаблону. В противном случае она вернёт значение None
re.compile	Собирает регулярное выражение в объект для будущего использования в других re-функциях. Ничего не ищет, всегда возвращает Pattern-объект
re.sub	Заменяет в строке string все pattern на repl. Возвращает строку в изменённом виде
re.escape	Возвращает строку со всеми не алфавитно-цифровыми символами с обратным слешем
re.split	Разделяет строку string по подстрокам, соответствующим pattern
re.findall	Ищет pattern по всей строке string. Возвращает список со всеми найденными совпадениями
re.subn	Возвращает новую строку вместе с номером замены

## Примеры:

### 1. Корректность URL:

```
import re

pattern = ("((http|https://)(www.)?" +
           "[a-zA-Z0-9@:%._\\+~#?&//=]" +
           "{2,256}\\.[a-z]" +
           "{2,6}\\b([-a-zA-Z0-9@:%" +
           "._\\+~#?&//=]*)")
url = 'https://www.yandex.ru' # http://www.yandex.ru - не будет корректным

result = re.search(pattern, url, re.I)
print(f'{result[0]} - корректный URL') if result else print(f'URL: {url} - не корректен')
```

### Корректность E-mail:

```
import re

pattern = '([A-Za-z0-9]+[.-_])*[A-Za-z0-9]+@[A-Za-z0-9]+(\\.[A-Z|a-z]{2,})+'
email = 'abc@yandex.ru' # abc@yandex.r - не будет корректным

result = re.search(pattern, email, re.I)
print(f'{result[0]} - корректный E-mail') if result else print(f'E-mail: {email} - не корректен')
```