План

1.Регулярные Выражения

2.Назначение

3.Источник для изучения

4.Односимвольные Шаблоны

5.Квантификаторы

6.Функции для работы с регулярками живут в модуле re.

7.Использование дополнительных флагов в питоне

8.Тонкости экранирования в питоне

9.Перечисления (операция «ИЛИ»)

10.Скобки плюс перечисления

11.Match-объекты

12.Шаблоны, соответствующие не конкретному тексту, а позиции

13.Сложные шаблоны, соответствующие позиции (*lookaround* и Co)

**1.Регулярные Выражения**

Регулярные выражения, или коротко, *регулярки* — это очень мощный инструмент для работи со строками.

**2.Назначение**

Для чего исполь зуеться:

1. Парсер текста

2. Валидаторы

3. Для изменения текста

ОПАСНО

Регулярные выражения не подходят для парсинга xml и тем более html. Используйте другие инструменты: BeautifulSoup4.

Использование регулярных выражений при парсинге html подобно залатыванию резиновой лодки шилом.

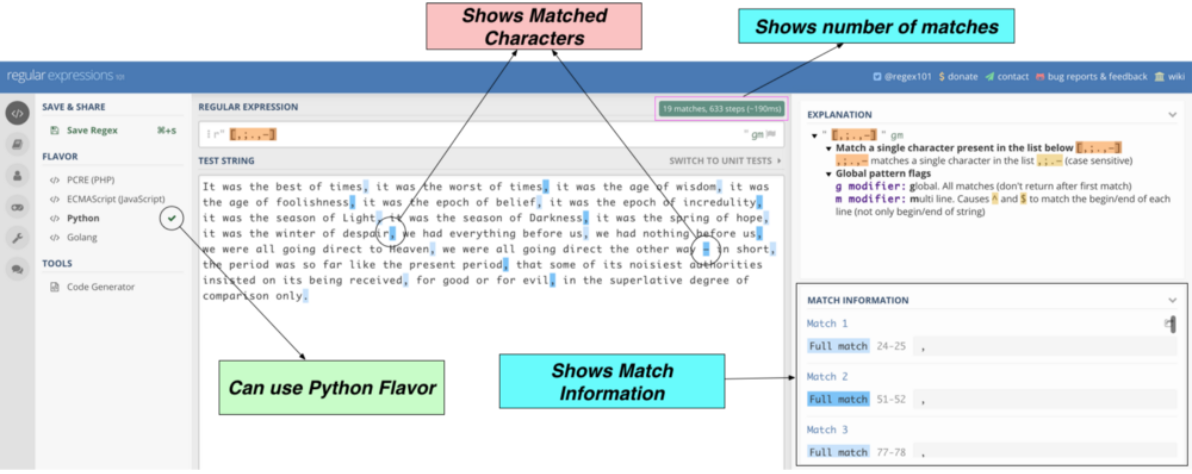
Закон Мёрфи для парсинга html и xml при помощи регулярок гласит: Парсинг html и xml регулярками иногда работает, но в точности до того момента, когда правильность результата будет *очень* важна.

**3.Источник для изучения**

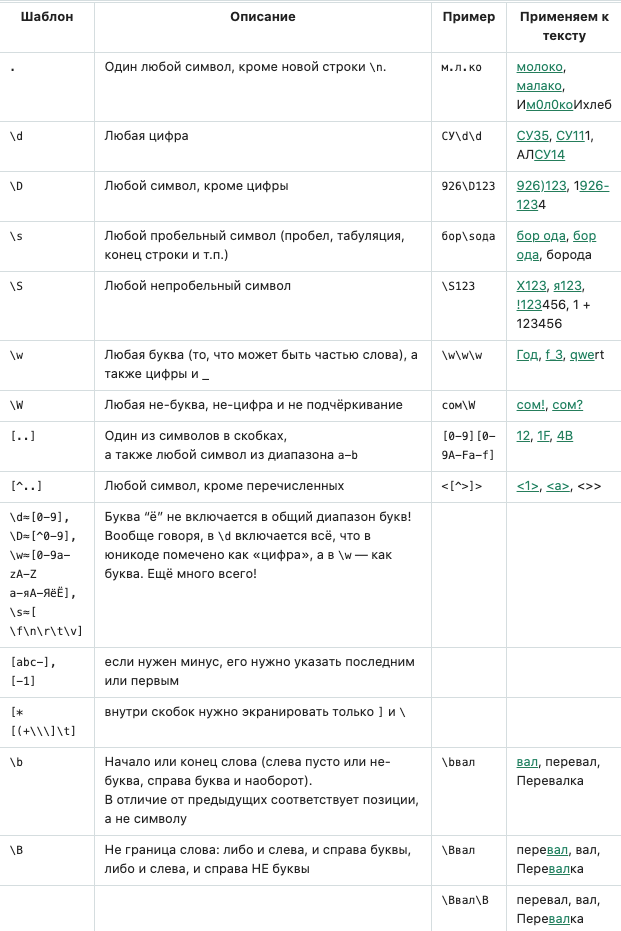
Источник: [https://habr.com/ru/post/349860/#Gruppiruyuschie\_skobki\_\_i\_matchobekty\_v\_pitone](https://habr.com/ru/post/349860/" \l "Gruppiruyuschie_skobki__i_matchobekty_v_pitone)

Оригинальная документация: <https://docs.python.org/3/library/re.html>;

Он-лайн отладка регулярок [https://regex101.com](https://regex101.com/r/F8dY80/3) (не забудьте поставить галочку Python в разделе FLAVOR слева).



**4.Односимвольные Шаблоны (Code)**



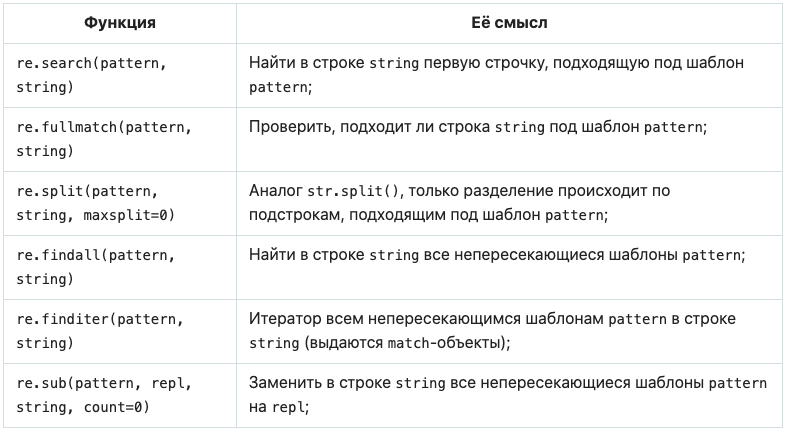
**5. Квантификаторы (указание количества повторений) (Text, Code)**

По умолчанию квантификаторы *жадные*. Этот подход решает очень важную проблему — проблему границы шаблона.



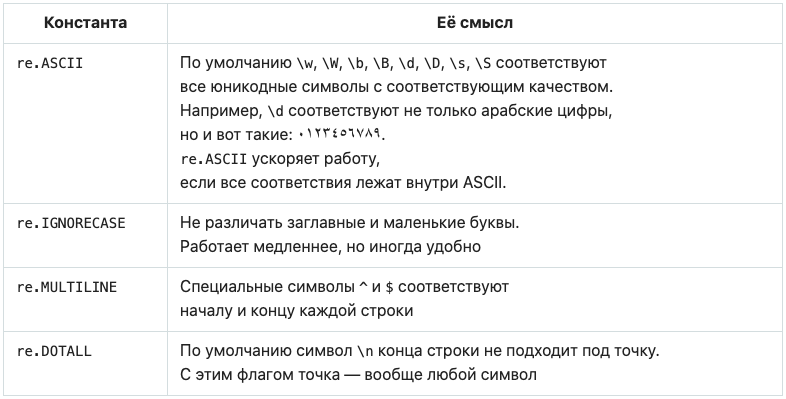
**6.Функции для работы с регулярками живут в модуле re. (Code)**

Основные функции:



**7.Использование дополнительных флагов в питоне (Code)**

Каждой из функций, перечисленных выше, можно дать дополнительный параметр flags, что несколько изменит режим работы регулярок. В качестве значения нужно передать сумму выбранных констант, вот они:



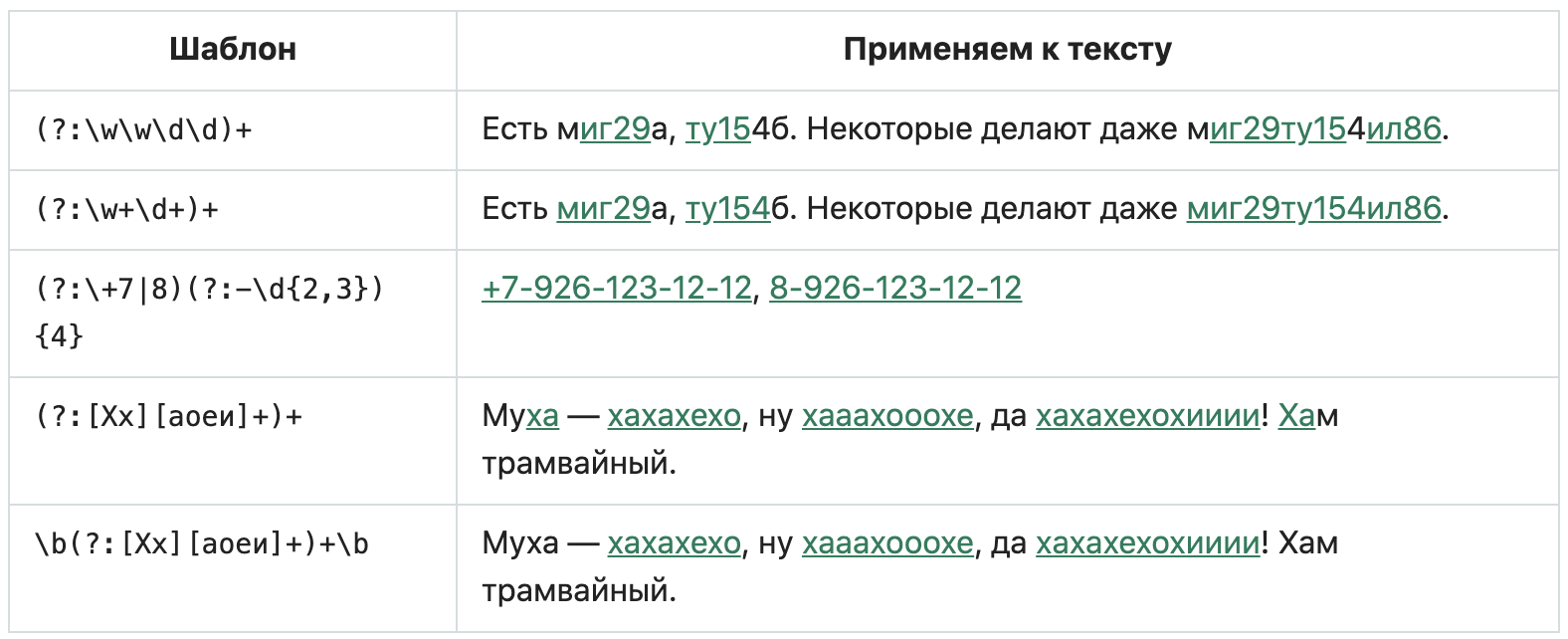
**8. Тонкости экранирования в питоне (Code)**

Символ ‘r’

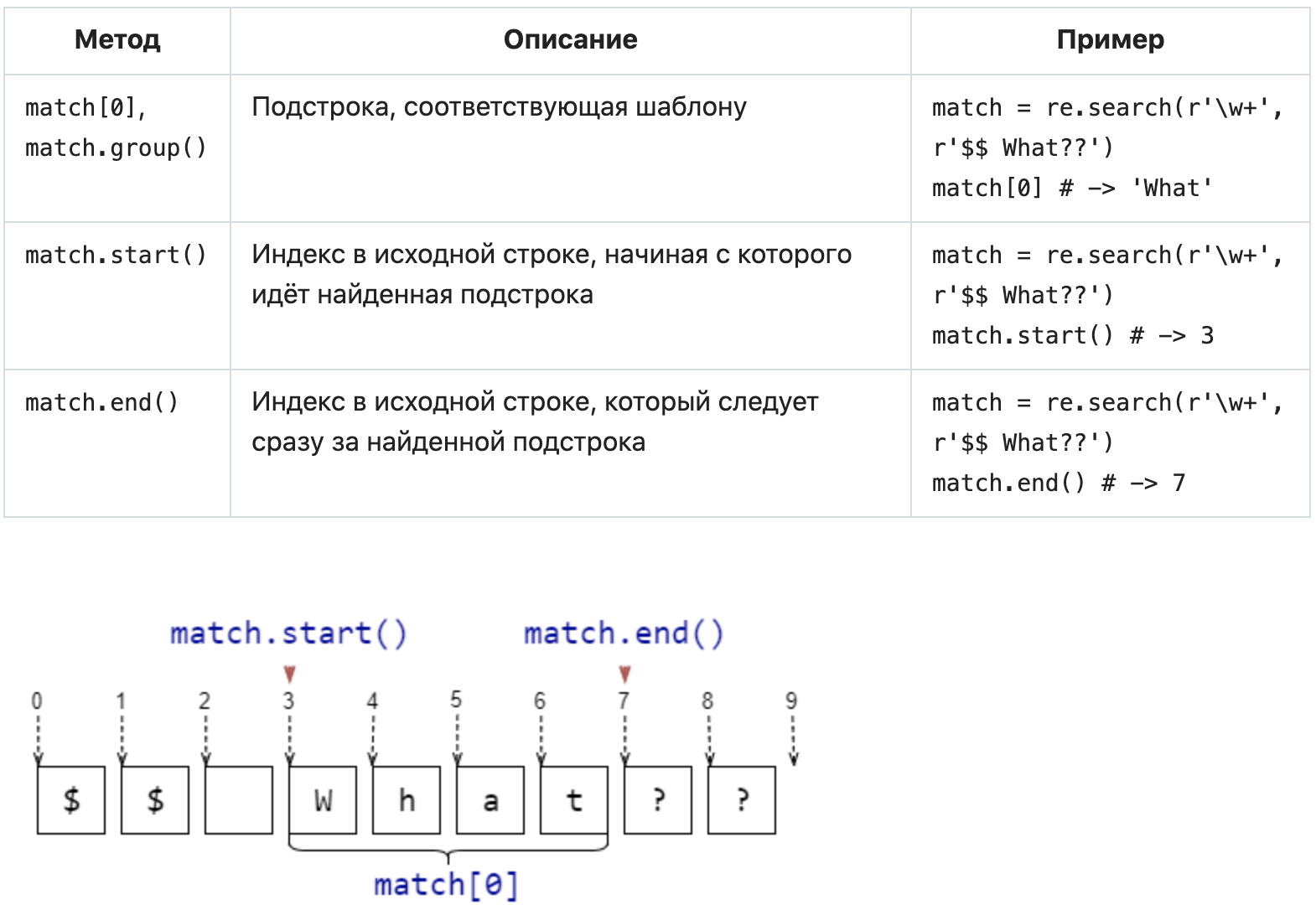
**9. Перечисления (операция «ИЛИ») (Code)**

Чтобы проверить, удовлетворяет ли строка хотя бы одному из шаблонов, можно воспользоваться аналогом оператора or, который записывается с помощью символа |.

**10. Скобки плюс перечисления (Code)**

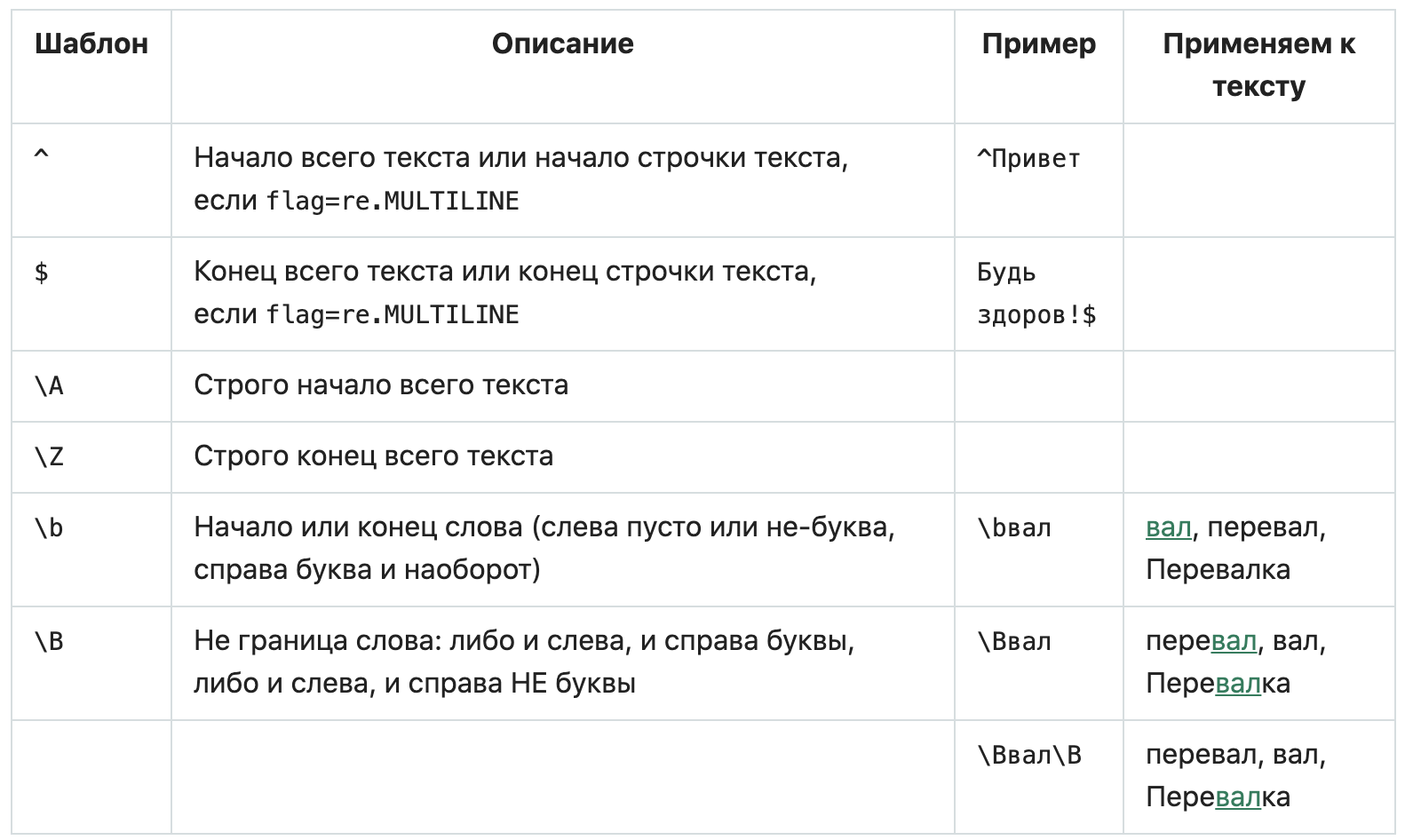


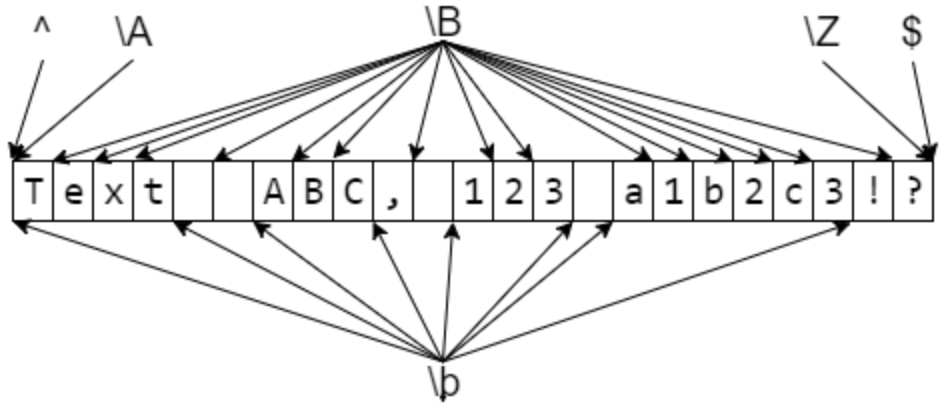
**11. Match-объекты (Code)**



**12. Шаблоны, соответствующие не конкретному тексту,  
а позиции**

Отдельные части регулярного выражения могут соответствовать не части текста, а позиции в этом тексте. То есть такому шаблону соответствует не подстрока, а некоторая позиция в тексте, как бы «между» буквами.





**13.Сложные шаблоны, соответствующие позиции  
(Code)**

Следующие шаблоны применяются в основном в тех случаях, когда нужно уточнить, что должно идти непосредственно перед или после шаблона, но при этом не включать найденное в match-объект.

