

“Шпаргалка” по регулярным выражениям

Обращение к символам

[A - Я]

Одна заглавная буква кириллического алфавита. Для латинского алфавита [A - Z].

[a - яё]

Одна строчная буква кириллического алфавита. Для латинского алфавита [a - z].

[A - Яa - яё]

Одна буква кириллического алфавита заглавная или строчная.

Для латинского алфавита [A - Za - z].

[0 - 9]

Одна любая цифра.

[^0 - 9a - zA - Z]

Отрицание задается ^ **внутри** квадратных скобок. Например, [^0-9a-zA-Z]kitty найдет !kitty, &kitty, *kitty, но не 8kitty или Akitty

\b

Граница слова

\d

Одна любая цифра. Равнозначно [0 - 9].

\s

“пробельный символ” (сокращение от spacing character), к ним относятся пробел, перевод (конец) строки и табуляция

\t

табуляция

\n

перевод (конец) строки.

Операторы повторений

+

1 и более вхождений предыдущего символа, например, Sa+m найдет Sam, Saam, Saaam, и т.п.

*

0 и более вхождений предыдущего символа, например, Sa*m найдет Sm, Sam, Saam, и т.п.

?

0 или 1 от предыдущего знака, иными словами “может присутствовать, а может отсутствовать”. Например, Sa?m найдет Sm и Sam, но не Saam.

Можно задавать фиксированное число повторений символа, для это есть специальные операторы.

(Этого материала не было на занятии, но он пригодится для выполнения ДЗ)

`{n}`

Захватывает n повторений элемента, к которому он применяется, например, `юу{3}ху` найдет юуууху, но не найдет юхууууху.

`{n, }`

Захватывает n и более повторений элемента, к которому он применяется, например, `юу{3, }ху` найдет юуууху, юуууууху, но не найдет юууху.

`{m, n}`

Захватывает от m до n повторений элемента, к которому он применяется, например, `юу{1, 3}ху` найдет юуху, юууху, юуууху, но не найдет юуууууху.

Экранирование знаков

Знаки, например, `? . () + -`, которые выступают в качестве операторов языка регулярных выражений нужно экранировать, чтобы находить последовательности с этими знаками в тексте. Надо только указать, что мы используем их не как операторы.

Например, выражение `[0-9]{2} \. [0-9]{2} \. 2016` найдет последовательности 0340952016 или 03и09о2016 и т.п.

Если же мы напишем регулярное выражение так `[0-9]{2} \. [0-9]{2} \. 2016`, оно найдет даты 2016 года, например, 03.09.2016

Примеры

`[aeiou]+` – найдет последовательности вроде eau в слове beautiful или iou в слове continious.

`beg.n` – найдет формы слова begin, т.е., began и begun.

`при+` – найдет последовательности при, прии, приии, приииии и т.п.

`[а-яё]{3} \s [а-яё]+` – найдет последовательность “слово из трех букв – пробельный символ – слово сколь угодно длины”.

`[0-9] \s [а-яё]+` – найдет последовательность “цифра – слово сколь угодно длины”, например, 2 дня, 3 месяца, но не 2016 год.