

Регулярные выражения



RegExp

Регулярные выражения—формальный язык поиска и осуществления манипуляций с подстроками в тексте, основанный на использовании метасимволов.

Пример	Соответствие	Создаваемые группы
<code>a(bc b x)cc</code>	<div>abccaxcc</div> <div>abccaxcc</div>	<div>abccaxcc</div> <div>abccaxcc</div>
<code>a(?:bc b x)cc</code>	abccaxcc, abccaxcc	нет
<code>a(>?bc b x)cc</code>	<div>abccaxcc</div> <div>но не abccaxcc : вариант x найден, остальные проигнорированы</div>	нет
<code>a(>?x*)xa</code>	не найдётся axxxa : все x заняты, и нет возврата внутрь группы	

Метасимволы

. ^ \$ * + ? { [] \ | () – имеют особое значение в шаблонах

. - соответствует всем символам, кроме символа новой строки

^, \$ - начало и конец строки

*, +, ?, { } – квантификаторы

[] – символьный класс

\ - экранирующий слэш

| - «или»

() – образование группы

Квантификаторы

1. $*$ - от нуля и более вхождений
2. $+$ - больше одного вхождения
3. $?$ – ноль или одно вхождение
4. $\{m, n\}$ – от m по n вхождений включительно
5. $\{m\}$ – ровно m вхождений

"ba*" -> b , ba, baaaaa

"ba?" -> b, ba

"ba+" -> ba, baaa

"ba{3, 4}" – baaa, baaaa

Ленивая квантификация

"ba+?"

"ba*?"

Символьные класс

Записывается, как набор символов в квадратных скобках

[abc] – соответствует одному символу a или b или c

[0-9]+ - одна или более цифр

[a-zA-Z] – любой букве из диапазона

[^xyz] – любой символ, кроме x, y, z

Встроенные символьные классы

- `\d`

Соответствует любой цифре; эквивалент класса `[0-9]`.

- `\D`

Соответствует любому нечисловому символу; эквивалент класса `[^0-9]`.

- `\s`

Соответствует любому символу `whitespace`; эквивалент `[\t\n\r\f\v]`.

- `\S`

Соответствует любому не-`whitespace` символу; эквивалент `[^\t\n\r\f\v]`.

- `\w`

Соответствует любой букве или цифре; эквивалент `[a-zA-Z0-9_]`.

- `\W`

Наоборот; эквивалент `[^a-zA-Z0-9_]`.


Группировка

1. Обычная группа “(abc)”, далее в регулярке можно обратиться к этой группе через \1
2. Группа без обратной связи “(?:abc)”
3. Позитивный просмотр вперед “abc(=XXX)”
4. Негативный просмотр вперед “abc(!XXX)”
5. Позитивный просмотр назад “(?<=XXX)abc”
6. Негативный просмотр назад “(?<!XXX)abc”

Также группировка позволяет сохранить конкретную группу для последующего использования.

Пример: “(\d{4})-(\d{2})-(\d{2})”

**Спасибо за
внимание**

 **artezio_software**

 **info@artezio.com**

www.artezio.com