Шпаргалка по регулярным выражениям

devanych.ru/technologies/shpargalka-po-regulyarnym-vyrazheniyam

Регулярное выражение представляет собой «строку-шаблон», написанную на формальном языке поиска, по которой производится поиск в целевом строковом представлении (контенте). «Строка-шаблон» состоит из строковых, цифровых и специальных символов, а также заключается между ограничителями шаблона (`/RegExp/`). В роли ограничителей шаблона нельзя использовать буквы и цифры, чаще всего используются `/` , `#` и `~` .

> В шпаргалке описываются регулярные выражения, работающие с библиотекой PCRE.

Метасимволы

Зарезервированные специальные символы. Для использования метасимвола, как обычного литерала, необходимо экранировать его, для этого нужно поставить `\` непосредственно перед экронируемым метасимволом, например `.` — это совпадение с любым символом кроме пробельного, а `\.` — это совпадение только с точкой.

Позиционные:

- `^` — начало строки (`/^RegExp/`), внутри символьного класса трактуется как литерал или знак отрицания (зависит от расположения в наборе).
- `$` — конец строки (`/RegExp$/`), внутри символьного класса трактуется как литерал.
- `\A` — начало текста (`/\ARegExp/`), похож на `^` , но в мультистроковом режиме `\A` будет всегда обозначать начало всего текста, а `^` — начало каждой строки;
- `\z` — конец текста (`/RegExp\z/`), похож на `$` , но в мультистроковом режиме `\z` будет всегда обозначать конец всего текста, а `$` — конец каждой строки;
- `\Z` — похож на `\z` , но если последним символом текста является перевод строки, то `\Z` будет занимать позицию, находящуюся перед последним переводом строки, а `\z` всегда будет на позиции в самом конце текста;
- `\b` — обозначает границу слова, для обращения к первой букве слова `www` «W» используется так (`\bw`), а к последней букве так (`w\b`);
- `\B` — обратное от `\b` , для обращения к средней (второй) букве «w» слова `www` используется так (`\Bw\B`);
- `\G` — останавливается на позиции окончания повторяющихся подряд символов, например: `\Gw` остановится на четвертой позиции после `www` , при поиске в строке `www.example.com` .

Группирующие:

- `(` — открывает вложенное выражение;
- `)` — закрывает вложенное выражение;
- `|` — логическое «или», может использоваться внутри (`(abc|def|ghi)`) и вне (`abc|def|ghi`) группы.

Пробельные:

- `\f` — конец страницы;
- `\n` — новая строка;
- `\r` — возврат каретки;
- `\t` — табуляция;
- `\v` — вертикальная табуляция.

Квантификаторы (для поиска последовательностей):

- `{number}` — точное количество вхождений;
- `{min,max}` — диапазон вхождений от `min` до `max` ;
- `?` — ноль или одно вхождение (эквивалентно `{0,1}`);
- `+` — одно или более одно вхождения (эквивалентно `{1,}`);
- `*` — ноль, одно или более одно вхождения (эквивалентно `{0,}`).

Объединяющие (для символьных классов):

- `[` — открывает символьный класс;
- `]` — закрывает символьный класс;
- `-` — задает диапазон символов в символьном классе (`[0-9]`);
- `^` — если `^` располагается в самом начале, то это означает отрицание всех символов, входящих в состав данного символьного класса (`/[^0-9]/`), на другой позиции трактуется как литерал;
- `\d` — целое число (`[0-9]`);
- `\D` — любой символ кроме целочисленного (`[^0-9]`);
- `\s` — любой пробельный символ (`[\f\n\r\t\v ]`);
- `\S` — любой символ кроме пробельного (`[^\f\n\r\t\v ]`);
- `\w` — целое число, буква и подчеркивание (`[a-zA-Z0-9_]`);
- `\W` — любой символ кроме целого числа, буквы и подчеркивания (`[^a-zA-Z0-9_]`).

Квантификаторы

Располагаются следом за символьным классом, группой или одиночным символом, указывая количество их повторений, например `.*` обозначает любые символы в любом количестве.

По умолчанию квантификаторы являются «жадными», например если произвести поиск всех тегов в HTML-коде `<.*>` , то все теги будут трактоваться как один, так как после первого совпадения следующие теги будут соответствовать `.*` . Для решения задачи нужно или уточнить искомый результат `<[^>]*>` , или сделать квантификатор «ленивым», поставив `?` после квантификатора `<.*?>` .

Существует еще один «сверхжадный» режим, его еще называют «ревнивым», он является самым быстродействующим и служит для поиска самого длинного варианта. Данный режим полезен для проверки существования подстроки в строке, а также для исключения из результатов поиска нежелаемых совпадений. Для включения «ревнивого» режима нужно поставить `+` после квантификатора.

Жадный	Ленивый	Ревнивый
`?`	`??`	`?+`
`+`	`+?`	`++`
`*`	`*?`	`*+`

Символьные классы

Наборы различных символов, помещенные в квадратные скобки. В некоторых случаях поведение метасимволов в символьных классах может изменяться по сравнению с их аналогами, находящимися на других позициях «строки-шаблона», например `.` внутри набора трактуется как литерал.

- `[abc]` — любой один символ из трех указанных: «a» или «b» или «c».
- `[^abc]` — любые символы кроме трех указанных: «a» или «b» или «c».
- `[a-d]` — символы в диапазоне от «a» до «d» (a, b, c, d).
- `[^a-d]` — любые символ кроме диапазона от «a» до «d» (a, b, c, d).
- `[0-9]` — целые числа от «0» до «9» (0, 1, 2, 3, 4, 5, 6, 7, 8, 9).
- `[^a-d1-4]` — любые символы кроме диапазона букв от «a» до «d» (a, b, c, d) и цифр от «1» до «4» (1, 2, 3, 4).

Группы (подмаски)

Группировка добавляет функционал «обратных ссылок», которые дают возможность запоминать найденные группы символов под порядковыми номерами и обращаться к ним по этим номерам как по ссылкам. Для обращения к обратным ссылкам в «строке-шаблоне» используется обратный слэш и присвоенный номер группе (`\1`), а для обращения в «строке-замене» — знак доллара (`$1`). Для примера подставим закрывающий HTML-тег заголовка:

```
/<h([1-6])>.*?<\/h\1>/
```

В некоторых случаях дополнительно к цифровым удобно использовать именованные группы (`(?P<group-name>...)` или (`(?<group-name>...)`), например для обработки динамических HTTP-роутов:

```
/^(?<category>[a-z0-9-]+)\/(?<post>[a-z0-9-]+)(?<extension>\.html)$/i
```

Встречаются ситуации, когда необходимо сгруппировать символы, но саму группу не запоминать. Для этого нужно после открывающейся скобки поставить знак вопроса и двоеточие (`(?:...)`).

Существует еще «атомарная группировка« (`(?>...)`). Она похожа на «ревнивую« квантификацию, точно также при первом найденном совпадении останавливает поиск в группе и является самой быстрой из группировок.

Подмаска дает возможность применять условия типа `if` и `if-else` :

- `(?(шаблон-условие)шаблон-если-успех)` ;
- `(?(шаблон-условие)шаблон-если-успех|шаблон-если-провал)` .

Также подмаску можно использовать для поиска конкретного фрагмента в целевой строке, по указанной подмаске будет производиться поиск, но при этом сама подмаска не будет включена в результат поиска.

Формат	Название	Пример	Результат
`(?=...)`	Позитивный просмотр вперёд	`.*(?=\.com)`	`example`.com example.org
`(?!...)`	Негативный просмотр вперёд	`.*(?!\.com)`	example.com `example`.org
`(?<=...)`	Позитивный просмотр назад	`(?<=example\.).*`	example.`com` example.`org`
`(?<!...)`	Негативный просмотр назад	`(?<!example\.).*`	example.com example.org

Модификаторы (флаги)

Модифицируют поведение регулярного выражения, стоящий перед модификатором `-` инвертирует его поведение (не распространяется на `U`). Флаги указываются после «строки-шаблона» в произвольном порядке (`/RegExp/ugi`).

- `g` — ищет все совпадения со «строкой-шаблоном» (по умолчанию поиск останавливается после первого совпадения).
- `i` — регистронезависимость («a» и «A» считаются эквивалентными).
- `m` — мультистроковость (по умолчанию целевая строка, в котором производится поиск, считается одной строкой).
- `s` — однострочность (контент считается одной строкой в отличие от режима по умолчанию, метасимвол `.` включает в себя пробельные символы).
- `u` — поддержка юникода («строка-шаблон» и целевая строка будут обрабатываться в кодировке UTF-8).

- `U` — инверсия жадности квантификаторов (по умолчанию квантификаторы становятся «ленивыми», вернуть им «жадность» можно, поставив после квантификатора `?`).
- `x` — все неэкранированные пробельные символы, которые находятся вне символьного класса, будут проигнорированы.

Коментарии (0)

Добавить комментарий

Латинские или кириллические буквы, не меньше 3 и не больше 30 символов.

E-mail никто не увидит.

|||