

Регулярные выражения

Алексей Владыкин

СПбГУ ИТМО

11 февраля 2011

- Regular Expressions, RegExps
- Мощное средство для манипуляций с текстом
- С легкостью решают задачи поиска, извлечения и замены фрагментов текста

- Perl — Practical Extraction and Report Language

Корректными регулярными выражениями являются:

- пустая строка;
- a — отдельный символ;
- $[a-z]$, $\backslash d$, $\backslash D$ — диапазон символов;
- \wedge , $\$$, $\.$ — спецсимвол;
- $re1\ re2$ — $re1$, затем $re2$;
- $re1|re2$ — $re1$ или $re2$;
- (re) — то же, что просто re (используется для группировки);
- re^* — повторение re 0 или больше раз;
- re^+ — повторение re 1 или больше раз;
- $re?$ — повторение re 0 или 1 раз.

Операции с регулярными выражениями в Perl:

- $\$x = \sim m/re/$ — сопоставление переменной $\$x$ с регулярным выражением re ;
- $\$x = \sim s/re/subs/$ — замена вхождений re в переменную $\$x$ выражением $subs$.

Пример: заменить все вхождения «one» на «two».

```
my $line;  
while ($line = <STDIN>) {  
    $line =~ s/one/two/g;  
    print $line;  
}
```

Короче:

```
while (<>) {  
    s/one/two/g;  
    print;  
}
```




Пример: сжать все пробельные последовательности до одного пробела, убрать пробелы в начале и конце строк.

```
my $line;  
while ($line = <STDIN>) {  
    $line =~ s/^[ \t]+|[ \t]+$//g;  
    $line =~ s/[ \t]+/ /g;  
    print $line;  
}
```

Пример: найти и вывести все подстроки в одинарных или двойных кавычках.

```
my $line;  
while ($line = <STDIN>) {  
    print "$2\n" while ($line =~ m/(['"])(.*?)\1/g);  
}
```

Рекомендуемая литература

-  Фридл Д.
Регулярные выражения. 2-е изд.
СПб: Питер, 2003. — 464 с.: ил.
-  Уолл Л., Кристиансен Т., Орвант Д.
Программирование на Perl. — Пер. с англ.
СПб: Символ-Плюс, 2004. — 1152 с., ил.
-  Хопкрофт Д., Мотвани Р., Ульман Д.
Введение в теорию автоматов, языков и вычислений, 2-е изд.:
Пер. с англ.
М.: Вильямс, 2002. — 528 с.: ил. // Глава 3