Chapter 7

perlretut

Perl regular expressions tutorial

7.1 DESCRIPTION

This page provides a basic tutorial on understanding, creating and using regular expressions in Perl. It serves as a complement to the reference page on regular expressions *perlre*. Regular expressions are an integral part of the `m//`, `s///`, `qr//` and `split` operators and so this tutorial also overlaps with Regexp Quote-Like Operators in *perlop* and split in *perlfunc*.

Perl is widely renowned for excellence in text processing, and regular expressions are one of the big factors behind this fame. Perl regular expressions display an efficiency and flexibility unknown in most other computer languages. Mastering even the basics of regular expressions will allow you to manipulate text with surprising ease.

What is a regular expression? A regular expression is simply a string that describes a pattern. Patterns are in common use these days; examples are the patterns typed into a search engine to find web pages and the patterns used to list files in a directory, e.g., `ls *.txt` or `dir *.*`. In Perl, the patterns described by regular expressions are used to search strings, extract desired parts of strings, and to do search and replace operations.

Regular expressions have the undeserved reputation of being abstract and difficult to understand. Regular expressions are constructed using simple concepts like conditionals and loops and are no more difficult to understand than the corresponding `if` conditionals and `while` loops in the Perl language itself. In fact, the main challenge in learning regular expressions is just getting used to the terse notation used to express these concepts.

This tutorial flattens the learning curve by discussing regular expression concepts, along with their notation, one at a time and with many examples. The first part of the tutorial will progress from the simplest word searches to the basic regular expression concepts. If you master the first part, you will have all the tools needed to solve about 98% of your needs. The second part of the tutorial is for those comfortable with the basics and hungry for more power tools. It discusses the more advanced regular expression operators and introduces the latest cutting edge innovations in 5.6.0.

A note: to save time, ’regular expression’ is often abbreviated as regexp or regex. Regexp is a more natural abbreviation than regex, but is harder to pronounce. The Perl pod documentation is evenly split on regexp vs regex; in Perl, there is more than one way to abbreviate it. We’ll use regexp in this tutorial.

7.2 Part 1: The basics

7.2.1 Simple word matching

The simplest regexp is simply a word, or more generally, a string of characters. A regexp consisting of a word matches any string that contains that word:

```
    "Hello World" =~ /World/;  # matches
```