Ruby Standard Library

Sheetal Sonare 04305025

Computer Science and Engineering
IIT Bombay

November 29, 2004

abbrev abbrev(words,pattern=nil)

- Given a set of strings, it calculate the set of unambiguous abbreviations for those strings and return a hash where the keys are all the possible abbreviations and the values are full strings.
- Example
  require 'abbrev'
  require 'pp'
  pp Abbrev::abbrev(['car','cone']).sort
- Generates [["ca","car"], ["car","car"], ["co","cone"], ["con","cone"], ["cone","cone"]]

base64

- Encode into lines of a given length (b64encode).
- Decode the special format specified in RFC2047 for the representation of email headers(decode_b)
- Example
  require "base64"
  enc = Base64.encode64('Send reinforcements')
  "U2VuZCByZWluZm9yY2VtZW50cw==\n"
  plain = Base64.decode64(enc)
  "Send reinforcements"

benchmark

- The Benchmark module provides methods to measure and report the time used to execute Ruby code.
- Example
  require 'benchmark'
  put Benchmark.measure {"a"*1_000_000}
  It may give.
  1.166667 0.050000 1.216667 (0.571355)
  This report shows the user CPU time, system CPU time, the sum of the user and system CPU times, and the elapsed real time. The unit of time is seconds.

cgi

- This holds the CGI class. This class provides functionality for retrieving HTTP request parameters, managing cookies, and generating HTML output.
- Require files
  English nkf delegate tempfile stringio tempfile shellwords

complex

- complex.rb implements the Complex class for complex numbers. Additionally, some methods in other Numeric classes are redefined or added to allow greater interoperability with Complex numbers.
- Complex number can be created in following manner:
  - Complex(a,b)
  - Complex.polar(radius, theta)

date

- This file provides two classes for working with dates and times.
- The first class, Date, represents dates. It works with years, months, weeks, and days.
- The second, DateTime, extends Date to include hours, minutes, seconds, and fractions of a second. It provides basic support for time zones.

drb

- dRuby is a distributed object system for Ruby.
- It allows an object in one Ruby process to invoke methods on an object in another Ruby process on the same or a different machine.

English

- Include the English library file in a Ruby script, and you can reference the global variables such as VAR{$_} using less cryptic names, listed in the following table.
  % vref{tab:english}.
- It allows an object in one Ruby process to invoke methods on an object in another Ruby process on the same or a different machine.

fileutils

- FileUtils::Verbose
  This module has all methods of FileUtils module, but it outputs messages before acting. This equates to passing the :verbose flag to methods in FileUtils.
- FileUtils::NoWrite
  This module has all methods of FileUtils module, but never changes files/directories. This equates to passing the :noop flag to methods in FileUtils.
- FileUtils::DryRun
  This module has all methods of FileUtils module, but never changes files/directories. This equates to passing the :noop and :verbose flags to methods in FileUtils.

find

- This is the Find module for processing all files under a given directory.
- The Find module supports the top-down traversal of a set of file paths.
- Example find(*paths) {—path— ...}
  Calls the associated block with the name of every file and directory listed as arguments, then recursively on their subdirectories, and so on.

ftools

- ftools adds several (class, not instance) methods to the File class, for copying, moving, deleting, installing, and comparing files, as well as creating a directory path.

generator

- This library provides the Generator class, which converts an internal iterator (i.e. an Enumerable object) to an external iterator. In that form, you can roll many iterators independently.

gserver

- GServer implements a generic server, featuring thread pool management, simple logging, and multi-server management.
- Methods used
- Methods
  connections in_service? join new serve shutdown start stop stop stopped?

net/ftp

- This class implements the File Transfer Protocol.
- require 'net/ftp'
- example

```
ftp = Net::FTP.new('ftp.netlab.co.jp')
ftp.login
files = ftp.chdir('pub/lang/ruby/contrib')
files = ftp.list('n*')
ftp.getbinaryfile('nif.rb-0.91.gz', 'nif.gz', 1024)
ftp.close
```

observer

- This implements the Observer object-oriented design pattern.
- The observable object must:
  - assert that it has changed.
  - call notify_observers

ostruct

- ostruct (OpenStruct) allows the creation of data objects with arbitrary attributes.
- Example
- require 'ostruct'
  record = OpenStruct.new
  record.name = "John Smith"
  record.age = 70
  record.pension = 300
  puts record.name -> "John Smith"
  puts record.address -> nil

ping

- This module contains routines to test for the reachability of remote hosts.
- Currently the only routine implemented is pingecho().
- Required files
  timeout socket

set

- This library provides the Set class.
- Deals with a collection of unordered values with no duplicates.
- It is a hybrid of Arrays intuitive inter-operation facilities and Hashs fast lookup.
- Example

```
require 'set'
s1 = Set.new [1, 2] -><Set: 1, 2>
s2 = [1, 2].to_set -><Set: 1, 2>
s1 == s2 -> true
s1.add("foo") -><Set: 1, 2, "foo">
s1.merge([2, 6]) -><Set: 6, 1, 2, "foo">
s1.subset? s2 ->false
s2.subset? s1 -> true
```