



RUBY

programming language



tutorialspoint
SIMPLY EASY LEARNING

www.tutorialspoint.com



<https://www.facebook.com/tutorialspointindia>



<https://twitter.com/tutorialspoint>

About the Tutorial

Ruby is a scripting language designed by Yukihiro Matsumoto, also known as Matz. It runs on a variety of platforms, such as Windows, Mac OS, and the various versions of UNIX.

This tutorial gives a complete understanding on Ruby.

Audience

This tutorial has been prepared for beginners to help them understand the basic to advanced concepts related to Ruby Scripting languages.

Prerequisites

Before you start practicing with various types of examples given in this tutorial, we are making an assumption that you are already aware of computer programs and programming languages in general.

Copyright & Disclaimer

© Copyright 2015 by Tutorials Point (I) Pvt. Ltd.

All the content and graphics published in this e-book are the property of Tutorials Point (I) Pvt. Ltd. The user of this e-book is prohibited to reuse, retain, copy, distribute or republish any contents or a part of contents of this e-book in any manner without written consent of the publisher.

We strive to update the contents of our website and tutorials as timely and as precisely as possible, however, the contents may contain inaccuracies or errors. Tutorials Point (I) Pvt. Ltd. provides no guarantee regarding the accuracy, timeliness or completeness of our website or its contents including this tutorial. If you discover any errors on our website or in this tutorial, please notify us at contact@tutorialspoint.com

Table of Contents

About the Tutorial	
Audience.....	i
Prerequisites.....	i
Copyright & Disclaimer	i
Table of Contents.....	ii
 1. RUBY – OVERVIEW	 1
Features of Ruby.....	1
Tools You Will Need	2
What is Next?	2
 2. RUBY – ENVIRONMENT SETUP	 3
Try it Option Online	3
Local Environment Setup	3
Ruby Installation on Linux/Unix	3
Using yum to Install Ruby	4
Ruby Installation on Windows	4
Ruby Command Line Options.....	5
Ruby Environment Variables.....	7
Popular Ruby Editors	9
Interactive Ruby (IRb)	9
What is Next?	10
 3. RUBY – SYNTAX	 11
Whitespace in Ruby Program	11
Line Endings in Ruby Program	11
Ruby Identifiers	11

Reserved Words.....	12
Here Document in Ruby	12
Ruby BEGIN Statement	13
Ruby END Statement	14
Ruby Comments.....	15
4. RUBY – CLASSES AND OBJECTS.....	16
Defining a Class in Ruby	17
Variables in a Ruby Class.....	17
Creating Objects in Ruby Using new Method.....	18
Custom Method to Create Ruby Objects	18
Member Functions in Ruby Class	19
Simple Case Study.....	20
5. RUBY – VARIABLES, CONSTANTS AND LITERALS	24
Ruby Global Variables	24
Ruby Instance Variables.....	25
Ruby Class Variables	26
Ruby Local Variables	27
Ruby Constants	27
Ruby Pseudo-Variables	28
Ruby Basic Literals	28
Integer Numbers.....	28
Floating Numbers.....	29
String Literals	29
Backslash Notations.....	30
Ruby Arrays	31
Ruby Hashes	31

Ruby Ranges	32
6. RUBY – OPERATORS	33
Ruby Arithmetic Operators	33
Ruby Comparison Operators	33
Ruby Assignment Operators	35
Ruby Parallel Assignment.....	35
Ruby Bitwise Operators	36
Ruby Logical Operators	37
Ruby Ternary Operator	38
Ruby Range Operators	38
Ruby defined? Operators	38
Ruby Dot "." and Double Colon "::" Operators.....	40
Ruby Operators Precedence.....	41
7. RUBY – COMMENTS	43
Ruby Multiline Comments	43
8. RUBY – IF...ELSE, CASE, UNLESS.....	45
Ruby if...else Statement	45
Ruby if modifier.....	46
Ruby unless Statement.....	46
Ruby unless modifier	47
Ruby case Statement	47
9. RUBY – LOOPS.....	50
Ruby while Statement	50
Ruby while modifier	51
Ruby until Statement.....	51

Ruby until modifier.....	52
Ruby for Statement	53
Ruby break Statement.....	54
Ruby next Statement.....	55
Ruby redo Statement	56
Ruby retry Statement	56
10. RUBY – METHODS	58
Return Values from Methods.....	59
Ruby return Statement.....	59
Variable Number of Parameters	60
Class Methods.....	61
Ruby alias Statement.....	62
Ruby undef Statement.....	62
11. RUBY – BLOCKS	64
The yield Statement	64
Blocks and Methods.....	66
BEGIN and END Blocks	66
12. RUBY – MODULES AND MIXINS	68
Ruby require Statement	69
Ruby include Statement	70
Mixins in Ruby	71
13. RUBY – STRINGS.....	73
Expression Substitution	73
General Delimited Strings	73
Escape Characters	74

Character Encoding	75
String Built-in Methods	75
String unpack Directives.....	85
14. RUBY – ARRAYS	89
Creating Arrays	89
Array Built-in Methods	91
Array pack Directives	99
15. RUBY – HASHES.....	103
Creating Hashes	103
Hash Built-in Methods	104
16. RUBY – DATE AND TIME	109
Getting Current Date and Time	109
Getting Components of a Date & Time.....	109
Time.utc, Time.gm and Time.local Functions	110
Timezones and Daylight Savings Time	112
Formatting Times and Dates	112
Time Formatting Directives	113
Time Arithmetic	114
17. RUBY – RANGES	116
Ranges as Sequences	116
Ranges as Conditions	118
Ranges as Intervals	118
18. RUBY – ITERATORS.....	120
Ruby each Iterator.....	120
Ruby collect Iterator.....	121

19. RUBY – FILE I/O	123
The puts Statement	123
The gets Statement	123
The puts Statement	124
The print Statement	124
Opening and Closing Files	125
The File.new Method.....	125
The File.open Method	125
Reading and Writing Files	126
The sysread Method.....	126
The syswrite Method.....	127
The each_byte Method.....	127
The IO.readlines Method.....	128
The IO.foreach Method	128
Renaming and Deleting Files	128
File Modes and Ownership	129
File Inquiries	130
Directories in Ruby.....	132
Navigating Through Directories.....	132
Creating a Directory	132
Deleting a Directory	133
Creating Files & Temporary Directories.....	133
Built-in Functions	133
File Class and Methods	134
Directory Class and Methods	139
20. RUBY – EXCEPTIONS.....	142

Using retry Statement	143
Using raise Statement	144
Using ensure Statement	146
Using else Statement.....	147
Catch and Throw	148
Class Exception	149
21. RUBY OBJECT ORIENTED	151
Ruby Class Definition	151
Define Ruby Objects	151
The initialize Method	151
The instance Variables	152
The accessor & setter Methods.....	152
The instance Methods.....	155
The class Methods and Variables	155
The to_s Method.....	156
Access Control	157
Class Inheritance.....	159
Methods Overriding.....	160
Operator Overloading.....	161
Freezing Objects.....	162
Class Constants	164
Create Object Using Allocate.....	165
Class Information	166
22. RUBY – REGULAR EXPRESSIONS	167
Regular-Expression Modifiers	167
Regular-Expression Patterns	168

Regular-Expression Examples.....	171
Anchors.....	173
Special Syntax with Parentheses	174
Search and Replace	174
23. RUBY – DBI	177
Architecture of a DBI Application	177
Prerequisites.....	178
Obtaining and Installing Ruby/DBI	178
Database Connection	179
INSERT Operation	180
Using do Statement	180
Using prepare and execute.....	181
READ Operation	183
Fetching the Result	184
Update Operation	190
DELETE Operation	191
Performing Transactions	192
COMMIT Operation	193
ROLLBACK Operation	193
Disconnecting Database	193
Handling Errors	194
Code Blocks with Methods.....	195
Driver-specific Functions and Attributes	196
24. RUBY – WEB APPLICATIONS	199
Writing CGI Scripts	199
Using cgi.rb	199

Form Processing.....	200
Creating Forms and HTML.....	201
Quoting Strings	203
Useful Methods in CGI Class.....	203
Ruby CGI	204
Cookies and Sessions	210
Ruby CGI Cookies	210
Ruby CGI Sessions	212
Web Hosting Servers.....	214
25. RUBY – SENDING EMAIL	215
Sending an HTML e-mail using Ruby	216
Sending Attachments as an e-mail	217
26. RUBY – SOCKET PROGRAMMING	220
What are Sockets?	220
A Simple Client.....	221
A Simple Server.....	221
Multi-Client TCP Servers	222
A Tiny Web Browser	223
Further Readings.....	224
27. RUBY – XML, XSLT, XPATH	225
What is XML?	225
XML Parser Architectures and APIs	225
Parsing and Creating XML using Ruby	225
DOM-like Parsing	227
SAX-like Parsing	228
XPath and Ruby	230

x

XSLT and Ruby	231
Further Reading	232
28. RUBY – WEB SERVICES	233
What is SOAP?	233
Installing SOAP4R.....	233
Writing SOAP4R Servers.....	233
Writing SOAP4R Clients.....	237
29. RUBY – TK GUIDE	240
Introduction.....	240
Installation.....	240
Simple Tk Application	240
Ruby/Tk Widget Classes	241
TkFrame	242
TkButton.....	245
TkLabel	248
TkEntry	251
TkCheckButton.....	256
TkRadioButton.....	261
TkListbox.....	265
TkComboBox.....	272
TkMenu	274
TkMenubutton.....	280
Tk.messageBox	284
TkScrollbar	286
TkCanvas.....	291
TkScale.....	300

TkText	305
TkToplevel	310
TkSpinbox	312
TkProgressBar	318
Dialog Box.....	321
Tk::Tile::Notebook	323
Tk::Tile::Paned	326
Tk::Tile::Separator	328
Ruby/Tk Font, Colors, and Images.....	330
Standard Configuration Options	334
Ruby/Tk Geometry Management	339
grid	339
Pack	340
Place	342
Ruby/Tk Event Handling	343
The configure Method	345
The cget Method	346
30. RUBY – LDAP	347
Ruby/LDAP Installation	347
Establish LDAP Connection.....	347
Adding an LDAP Entry	348
Modifying an LDAP Entry	350
Deleting an LDAP Entry	351
Modifying the Distinguished Name	352
Performing a Search.....	353
Handling Errors	355

Further Reading	355
31. RUBY – MULTITHREADING	356
Creating Ruby Threads	356
Thread Lifecycle	357
Threads and Exceptions	358
Thread Variables	358
Thread Priorities	359
Thread Exclusion	359
Handling Deadlock	361
Thread States	362
Thread Class Methods	363
Thread Instance Methods	365
32. RUBY – BUILT-IN FUNCTIONS	368
Functions for Numbers	374
Functions for Float	377
Functions for Math	378
Conversion Field Specifier	379
Test Function Arguments	381
33. RUBY – PREDEFINED VARIABLES	384
34. RUBY – PREDEFINED CONSTANTS	388
35. RUBY – ASSOCIATED TOOLS	390
Standard Ruby Tools	390
RubyGems	390
Ruby Debugger	394
Interactive Ruby	398

Ruby Profiler	401
Additional Ruby Tools	402
eRuby: Embedded Ruby	402
ri: Ruby Interactive Reference.....	403

1. RUBY – OVERVIEW

Ruby is a pure object-oriented programming language. It was created in 1993 by Yukihiro Matsumoto of Japan.

You can find the name Yukihiro Matsumoto on the Ruby mailing list at www.ruby-lang.org. Matsumoto is also known as Matz in the Ruby community.

Ruby is "A Programmer's Best Friend".

Ruby has features that are similar to those of Smalltalk, Perl, and Python. Perl, Python, and Smalltalk are scripting languages. Smalltalk is a true object-oriented language. Ruby, like Smalltalk, is a perfect object-oriented language. Using Ruby syntax is much easier than using Smalltalk syntax.

Features of Ruby

- Ruby is an open-source and is freely available on the Web, but it is subject to a license.
- Ruby is a general-purpose, interpreted programming language.
- Ruby is a true object-oriented programming language.
- Ruby is a server-side scripting language similar to Python and PERL.
- Ruby can be used to write Common Gateway Interface (CGI) scripts.
- Ruby can be embedded into Hypertext Markup Language (HTML).
- Ruby has a clean and easy syntax that allows a new developer to learn very quickly and easily.
- Ruby has similar syntax to that of many programming languages such as C++ and Perl.
- Ruby is very much scalable and big programs written in Ruby are easily maintainable.
- Ruby can be used for developing Internet and intranet applications.
- Ruby can be installed in Windows and POSIX environments.
- Ruby support many GUI tools such as Tcl/Tk, GTK, and OpenGL.
- Ruby can easily be connected to DB2, MySQL, Oracle, and Sybase.
- Ruby has a rich set of built-in functions, which can be used directly into Ruby scripts.

Tools You Will Need

For performing the examples discussed in this tutorial, you will need a latest computer like Intel Core i3 or i5 with a minimum of 2GB of RAM (4GB of RAM recommended). You also will need the following software:

- Linux or Windows 95/98/2000/NT or Windows 7 operating system
- Apache 1.3.19-5 Web server
- Internet Explorer 5.0 or above Web browser
- Ruby 1.8.5

This tutorial will provide the necessary skills to create GUI, networking, and Web applications using Ruby. It also will talk about extending and embedding Ruby applications.

What is Next?

The next chapter guides you to where you can obtain Ruby and its documentation. Finally, it instructs you on how to install Ruby and prepare an environment to develop Ruby applications.

2. RUBY – ENVIRONMENT SETUP

Try it Option Online

We already have set up Ruby Programming environment online, so that you can execute almost all the tutorial examples online at the same time when you are doing your theory work. This gives you confidence in what you are reading and to check the result with different options. Feel free to modify any example and execute it online.

Try the following example using the **Try it** option available on our website at the top right corner of the sample code box given below:

```
#!/usr/bin/ruby -w

puts "Hello, Ruby!";
```

For most of the examples given in this tutorial, you will find a **Try it** option on our website code sections at the top right corner that will take you to the online compiler. So just make use of it and enjoy your learning.

Local Environment Setup

If you are still willing to set up your environment for Ruby programming language, then let's proceed. This tutorial will teach you all the important topics related to environment setup. We would recommend you to go through the following topics first and then proceed further:

- **Ruby Installation on Linux/Unix** : If you are planning to have your development environment on Linux/Unix Machine, then go through this chapter.
- **Ruby Installation on Windows** : If you are planning to have your development environment on Windows Machine, then go through this chapter.
- **Ruby Command Line Options** : This chapter list out all the command line options, which you can use along with Ruby interpreter.
- **Ruby Environment Variables** : This chapter has a list of all the important environment variables to be set to make Ruby Interpreter works.

Ruby Installation on Linux/Unix

Here are the steps to be followed to install Ruby on a Unix machine:

NOTE: Before proceeding, make sure you have root privilege.

- Download a zipped file having latest version of Ruby. Follow **Download Link**.

- After having downloaded the Ruby archive, unpack it and change into the newly created directory:

```
$ tar -xvzf ruby-1.6.7.tgz
$ cd ruby-1.6.7
```

- Now, configure and compile the source code as follows:

```
$ ./configure
$ make
```

- Finally, install Ruby interpreter as follows:

```
$ su -l root # become a root user
$ make install
$ exit      # become the original user again
```

- After installation, make sure everything is working fine by issuing the following command on the command-line:

```
$ ruby -v
ruby 1.6.7 (2002-06-04) [i386-netbsd]
```

- If everything is fine, this should output the version of the installed Ruby interpreter as shown above. You may have installed different version, so it will display a different version.

Using yum to Install Ruby

If your computer is connected to the Internet, then the easiest way to install Ruby or any other other RPM is using the **yum** utility. Give the following command at the command prompt and you will find Ruby gets installed on your computer.

```
$ yum install ruby
```

Ruby Installation on Windows

Here are the steps to install Ruby on a Windows machine.

NOTE: You may have different versions available at the time of installation.

- Download a zipped file having latest version of Ruby. Follow **Download Link**.
- After having downloaded the Ruby archive, unpack it and change into the newly created directory:
- Double-click the Ruby1.6.7.exe file. The Ruby installation wizard starts.

- Click Next to move to the Important Information page of the wizard and keep moving till Ruby installer completes installing Ruby.

You may need to set some environment variables if your installation has not setup them appropriately.

- If you use Windows 9x, add the following lines to your c:\autoexec.bat: set PATH="D:\(ruby install directory)\bin;%PATH%"
- Windows NT/2000 users need to modify their registries.
 - Click Control Panel | System Properties | Environment Variables.
 - Under System Variables, select Path and click EDIT.
 - Add your Ruby directory to the end of the Variable Value list and click OK.
 - Under System Variables, select PATHEXT and click EDIT.
 - Add .RB and .RBW to the Variable Value list and click OK.
- After installation, make sure everything is working fine by issuing the following command on the command-line:

```
$ruby -v
ruby 1.6.7
```

- If everything is fine, this should output the version of the installed Ruby interpreter as shown above. You may have installed different version, so it will display a different version.

Ruby Command Line Options

Ruby is generally run from the command line in the following way:

```
$ ruby [ options ] [.] [ programfile ] [ arguments ... ]
```

The interpreter can be invoked with any of the following options to control the environment and behavior of the interpreter.

Option	Description
-a	Used with -n or -p to split each line. Check -n and -p options.
-c	Checks syntax only, without executing program.
-C dir	Changes directory before executing (equivalent to -X).
-d	Enables debug mode (equivalent to -debug).

-F pat	Specifies pat as the default separator pattern (\$) used by split.
-e prog	Specifies prog as the program from the command line. Specify multiple -e options for multiline programs.
-h	Displays an overview of command-line options.
-i [ext]	Overwrites the file contents with program output. The original file is saved with the extension ext. If ext isn't specified, the original file is deleted.
-I dir	Adds dir as the directory for loading libraries.
-K [kcode]	Specifies the multibyte character set code (e or E for EUC (extended Unix code); s or S for SJIS (Shift-JIS); u or U for UTF-8; and a, A, n, or N for ASCII).
-l	Enables automatic line-end processing. Chops a newline from input lines and appends a newline to output lines.
-n	Places code within an input loop (as in while gets; ... end).
-O[octal]	Sets default record separator (\$/) as an octal. Defaults to \0 if octal not specified.
-p	Places code within an input loop. Writes \$_ for each iteration.
-r lib	Uses <i>require</i> to load <i>lib</i> as a library before executing.
-s	Interprets any arguments between the program name and filename arguments fitting the pattern -xxx as a switch and defines the corresponding variable.
-T [level]	Sets the level for tainting checks (1 if level not specified).
-v	Displays version and enables verbose mode
-w	Enables verbose mode. If program file not specified, reads from STDIN.
-x [dir]	Strips text before <i>#!/ruby</i> line. Changes directory to <i>dir</i> before executing if <i>dir</i> is specified.

-X dir	Changes directory before executing (equivalent to -C).
-y	Enables parser debug mode.
--copyright	Displays copyright notice.
--debug	Enables debug mode (equivalent to -d).
--help	Displays an overview of command-line options (equivalent to -h).
--version	Displays version.
--verbose	Enables verbose mode (equivalent to -v). Sets \$VERBOSE to true.
--yydebug	Enables parser debug mode (equivalent to -y).

Single character command-line options can be combined. The following two lines express the same meaning:

```
$ruby -ne 'print if /Ruby/' /usr/share/bin

$ruby -n -e 'print if /Ruby/' /usr/share/bin
```

Ruby Environment Variables

Ruby interpreter uses the following environment variables to control its behavior. The ENV object contains a list of all the current environment variables set.

Variable	Description
DLN_LIBRARY_PATH	Search path for dynamically loaded modules.
HOME	Directory moved to when no argument is passed to Dir::chdir. Also used by File::expand_path to expand "~".
LOGDIR	Directory moved to when no arguments are passed to

	Dir::chdir and environment variable HOME isn't set.
PATH	Search path for executing subprocesses and searching for Ruby programs with the -S option. Separate each path with a colon (semicolon in DOS and Windows).
RUBYLIB	Search path for libraries. Separate each path with a colon (semicolon in DOS and Windows).
RUBYLIB_PREFIX	Used to modify the RUBYLIB search path by replacing prefix of library path1 with path2 using the format path1;path2 or path1path2.
RUBYOPT	Command-line options passed to Ruby interpreter. Ignored in taint mode (Where \$SAFE is greater than 0).
RUBYPATH	With -S option, search path for Ruby programs. Takes precedence over PATH. Ignored in taint mode (where \$SAFE is greater than 0).
RUBYSHELL	Specifies shell for spawned processes. If not set, SHELL or COMSPEC are checked.

For Unix, use **env** command to see a list of all the environment variables.

```

HOSTNAME=ip-72-167-112-17.ip.secureserver.net
RUBYPATH=/usr/bin
SHELL=/bin/bash
TERM=xterm
HISTSIZE=1000
SSH_CLIENT=122.169.131.179 1742 22
SSH_TTY=/dev/pts/1
USER=amrood
JRE_HOME=/usr/java/jdk/jre
J2RE_HOME=/usr/java/jdk/jre
PATH=/usr/local/bin:/bin:/usr/bin:/home/guest/bin
MAIL=/var/spool/mail/guest
PWD=/home/amrood
INPUTRC=/etc/inputrc

```

```

JAVA_HOME=/usr/java/jdk
LANG=C
HOME=/root
SHLVL=2
JDK_HOME=/usr/java/jdk
LOGDIR=/usr/log/ruby
LOGNAME=amrood
SSH_CONNECTION=122.169.131.179 1742 72.167.112.17 22
LESSOPEN=|/usr/bin/lesspipe.sh %s
RUBYLIB=/usr/lib/ruby
G_BROKEN_FILENAMES=1
_=/bin/env

```

Popular Ruby Editors

To write your Ruby programs, you will need an editor:

- If you are working on Windows machine, then you can use any simple text editor like Notepad or Edit plus.
- **VIM** (Vi IMproved) is a very simple text editor. This is available on almost all Unix machines and now Windows as well. Otherwise, you can use your favorite vi editor to write Ruby programs.
- **RubyWin** is a Ruby Integrated Development Environment (IDE) for Windows.
- Ruby Development Environment (**RDE**) is also a very good IDE for windows users.

Interactive Ruby (IRb)

Interactive Ruby (IRb) provides a shell for experimentation. Within the IRb shell, you can immediately view expression results, line by line.

This tool comes along with Ruby installation so you have nothing to do extra to have IRb working.

Just type **irb** at your command prompt and an Interactive Ruby Session will start as given below:

```

$irb
irb 0.6.1(99/09/16)
irb(main):001:0> def hello
irb(main):002:1> out = "Hello World"
irb(main):003:1> puts out

```

```
irb(main):004:1> end
nil
irb(main):005:0> hello
Hello World
nil
irb(main):006:0>
```

Do not worry about what we did here. You will learn all these steps in subsequent chapters.

What is Next?

We assume now you have a working Ruby Environment and you are ready to write the first Ruby Program. The next chapter will teach you how to write Ruby programs.

End of ebook preview

If you liked what you saw...

Buy it from our store @ **<https://store.tutorialspoint.com>**