

# JabirOS Developer Guide

Muhammadsreza Haghiri

December 7, 2015

## Contents

<b>1</b>	<b>Introduction</b>	<b>3</b>
1.1	A brief history of JabirOS . . . . .	3
1.2	Required knowledge for reading this book . . . . .	3
1.3	What will you learn? . . . . .	3
<b>2</b>	<b>A brief review on Ruby</b>	<b>4</b>
2.1	Structure of a Ruby script . . . . .	4
2.2	Blocks in Ruby . . . . .	4
2.2.1	A simple <i>do</i> block . . . . .	4
2.2.2	<i>if...else</i> block . . . . .	5
2.3	Classes . . . . .	5
2.4	Gemfile . . . . .	5
2.5	Ruby in command line . . . . .	5
2.5.1	-r . . . . .	5
2.5.2	-e . . . . .	5
2.6	What will you learn in future? . . . . .	6
<b>3</b>	<b>Setup Environment</b>	<b>7</b>
3.1	How to Install SDK . . . . .	7
3.2	First Program . . . . .	7

# 1 Introduction

JabirOS<sup>1</sup>, is a FreeBSD derivative, and a general purpose unix-like operating system. It can be used in servers, mainframe computers, personal computers, and embedded computers like Raspberry Pi or Cubieboard.

After reading this document, you can make JabirOS applications using software development kit<sup>2</sup>.

## 1.1 A brief history of JabirOS

JabirOS, was a GNU/Linux based operating system first. It was a remaster of Ubuntu<sup>3</sup> GNU/Linux. Actually, this operating system was made for fun by Reza Bagherzadeh and me. Then, we wanted to make a technical, and purer unix-like operating system. In this steps, Muhammad Esmaili had joined us, so we made JabirOS from FreeBSD.

## 1.2 Required knowledge for reading this book

- You need to know Ruby<sup>4</sup> language.
- You need to know how to write classes and objects in Ruby, and use them in other codes.
- You need to know how to work with JabirOS (or FreeBSD, etc. Every FreeBSD based operating systems are suitable for learning).
- You need to know how to run or package your final product.

## 1.3 What will you learn?

- Program in JabirOS SDK
- Run your program
- Debug your programs
- Publish your program

---

<sup>1</sup><http://jabirproject.org>

<sup>2</sup><https://github.com/JabirTech/SDK>

<sup>3</sup><http://ubuntu.com>

<sup>4</sup><http://ruby-lang.org>

## 2 A brief review on Ruby

Ruby is one of the easiest language on this world. It's a well-designed language, too. If you want to develop your ideas as fast as you can, you **must** know how to code in ruby. In this part, we're going to review Ruby.

### 2.1 Structure of a Ruby script

A ruby script is usually made up of two parts, like other script languages. If you download a ruby script, you will see something like this:

```
#!/usr/bin/ruby
```

REST OF THE CODE

or it may look like this:

```
#!/usr/bin/env ruby
```

REST OF THE CODE

It actually depends on who programmed it. A lot of people prefer to use the extension *\*.rb* in their ruby scripts.<sup>5</sup>

### 2.2 Blocks in Ruby

If you code in python, you need to use indents in blocks. In ruby, indents are optional, but every block usually starts with a *do* and ends with an *end*. A function block starts with *def* and ends with *end*. And other blocks have similar endings, but starts are different, and it depends on which kind of block you want to use. Here are some blocks :

#### 2.2.1 A simple *do* block

```
a = [1, 3, 4]
a.each do |n|
  puts n
end
```

---

<sup>5</sup>When you use this extension, you'll no more need to use shbang structure

### 2.2.2 *if...else* block

```
if n == 0
  puts n
else
  puts n-1
end
```

## 2.3 Classes

Ruby classes are as simple as other blocks in this language. But there are two tips you must know about classes (and also modules<sup>6</sup>).

1. If you want to use a method directly, don't forget to use self keyword before methods.
2. Classes and Modules are considered as constants. So, they shall be started by a capitalized letter.

## 2.4 Gemfile

This is not actually a part of ruby programming, but it's necessary. You write required gems in this file, and when you run *bundler*, it'll install all gems in an isolated form.

## 2.5 Ruby in command line

Ruby has some switches you must know. Of course, JabirOS programming standard suggests using this switches for running your programs.

### 2.5.1 -r

This switch, means *require*. When you don't want to use "require" keyword in your code, you use this switch. But, using this switch instead of "require" is **required** in JabirOS SDK.

### 2.5.2 -e

This option is used for running a certain ruby command in terminal, and see the result. If you don't want to use IRB<sup>7</sup>, this option is the best choice.

---

<sup>6</sup>Modules in ruby are almost similar.

<sup>7</sup>Interactive Ruby

## **2.6 What will you learn in future?**

In this chapter, we reviewed Ruby basics. In next chapters, we'll learn how to install SDK, and how to use it. If you don't have Ruby installed on your computer, you can install it by following the instruction of its website, [ruby-lang.org](http://ruby-lang.org).

## 3 Setup Environment

### 3.1 How to Install SDK

First, you need to install *git* on your JabirOS machine. It's available in *pkg* repositories.

Then, you need to clone our SDK repository, like this :

```
git clone https://github.com/JabirTech/SDK
```

Now, you need to copy SDK folder to a certain path. We suggest this :

```
/usr/local/share/SDK
```

and now, you can code in JabirOS SDK.

### 3.2 First Program

**IRB**, is ruby's REPL. We here write our first program using this REPL. For running REPL with SDK, type this command in command line :

```
irb -r /usr/local/share/SDK/sdk.rb
```

and then, in REPL, type :

```
JabirOS::Terminal.writeln("Hello, World!")
```

Congragulations, you wrote your first program using JabirOS SDK. But, SDK's Libraries are a lot! And, you'll learn all of them after reading this book.