

小・中学生にもできる！ ゲームプログラ
ミング ～ **Ruby** を使ってゲームの作
り方を学ぼう～

高尾宏治 著

版 発行

目次

第 1 章	はじめに	1
1.1	プログラムってどんなもの？	1
1.2	僕や私にもできるの？	1
1.3	お金はかかるの？	1
第 2 章	はじめに	2
2.1	プログラムってどんなもの？	2
2.2	僕や私にもできるの？	2
2.3	お金はかかるの？	2
第 3 章	Smalruby のインストール	3
3.1	Microsoft Windows 8.1 の場合	4
3.1.1	STEP1: Ruby と DevKit	4
3.1.2	STEP2: Smalruby	14
3.2	Mac OS X Mountain Lion の場合	16
3.2.1	STEP1: Xcode	16
3.2.2	STEP2: Homebrew	18
3.2.3	Ruby	19
3.2.4	SDL	23
3.2.5	Smalruby	24
3.3	Mac OS X Mavericks の場合	26
3.4	GNU/Linux (Ubuntu) の場合	26
第 4 章	サンプルプログラムで遊ぶ	27
4.1	ゲームの動かし方	27
4.2	サンプルプログラム 1	27
4.3	サンプルプログラム 2	27
4.4	サンプルプログラム 3	27
4.5	サンプルプログラム 4	27
4.6	サンプルプログラム 5	27

第 1 章

はじめに

- 1.1 プログラムってどんなもの？
- 1.2 僕や私にもできるの？
- 1.3 お金はかかるの？

第 2 章

はじめに

2.1 プログラムってどんなもの？

2.2 僕や私にもできるの？

2.3 お金はかかるの？

第3章

Smalruby のインストール

まずは Smalruby をコンピュータに入れて使えるようにしよう。このことを「Smalruby をインストールする」ともいうんだ。少し時間がかかったり、難しい言葉が画面にでてくるのでおうちの人と一緒にやってみよう。

きみが持っているコンピュータにはオペレーティングシステム^{*1}というソフトウェアが入っていて、その種類によってやり方が違うんだ。ここでは次の4つについて説明するよ。

- Microsoft Windows 8.1
- Mac OS X Mountain Lion
- Mac OS X Mavericks
- GNU/Linux (Ubuntu)

それじゃあ、それぞれのオペレーティングシステムのやり方を説明するので、きみの使っているものを選んで Smalruby をインストールしよう。

コラム: どのオペレーティングシステムを使っているの？

きみが使っているのはどのオペレーティングシステムかな？ 次の画面をヒントに調べてみよう。もし分からなかったらおうちの人に聞いてみよう。

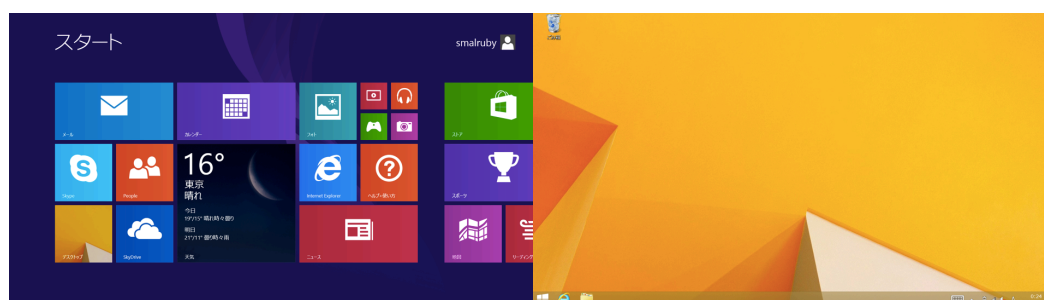


図 3.1 Microsoft Windows 8.1 の画面

^{*1} <http://ja.wikipedia.org/wiki/オペレーティングシステム>

```
--[[path = (not exist)]]--  
Mac OS X Mountain Lion のスクリーンショット  
  
Mac OS X Mountain Lion の画面  
  
--[[path = (not exist)]]--  
Mac OS X Mavericks のスクリーンショット  
  
Mac OS X Mavericks の画面  
  
--[[path = (not exist)]]--  
GNU/Linux (Ubuntu) のスクリーンショット  
  
GNU/Linux (Ubuntu) の画面
```

3.1 Microsoft Windows 8.1 の場合

Microsoft Windows 8.1 を使っている場合は Smalruby といっしょに次のソフトウェアもインストールする必要があるんだ。

- Ruby^{*2} と DevKit^{*3}
- Smalruby

それでは上から順番にインストールしてみよう。

3.1.1 STEP1: Ruby と DevKit

まずはインターネットから Ruby と、DevKit という 2 つのソフトウェアを取ってきます。インターネットなどからソフトウェアなどを取ってくることを**ダウンロード**というんだ。よく使う言葉だから覚えておこう。

インターネットを開いて「<http://rubyinstaller.org>」にアクセスします。たくさんの英語が表示されているけど落ち着いて一緒にやっていこう。

^{*2} <http://ja.wikipedia.org/wiki/Ruby>

^{*3} Ruby に関連したソフトウェアをインストールするときに使うソフトウェアです。英語ではあるけど <http://rubyinstaller.org/add-ons/devkit/> に説明があるよ。

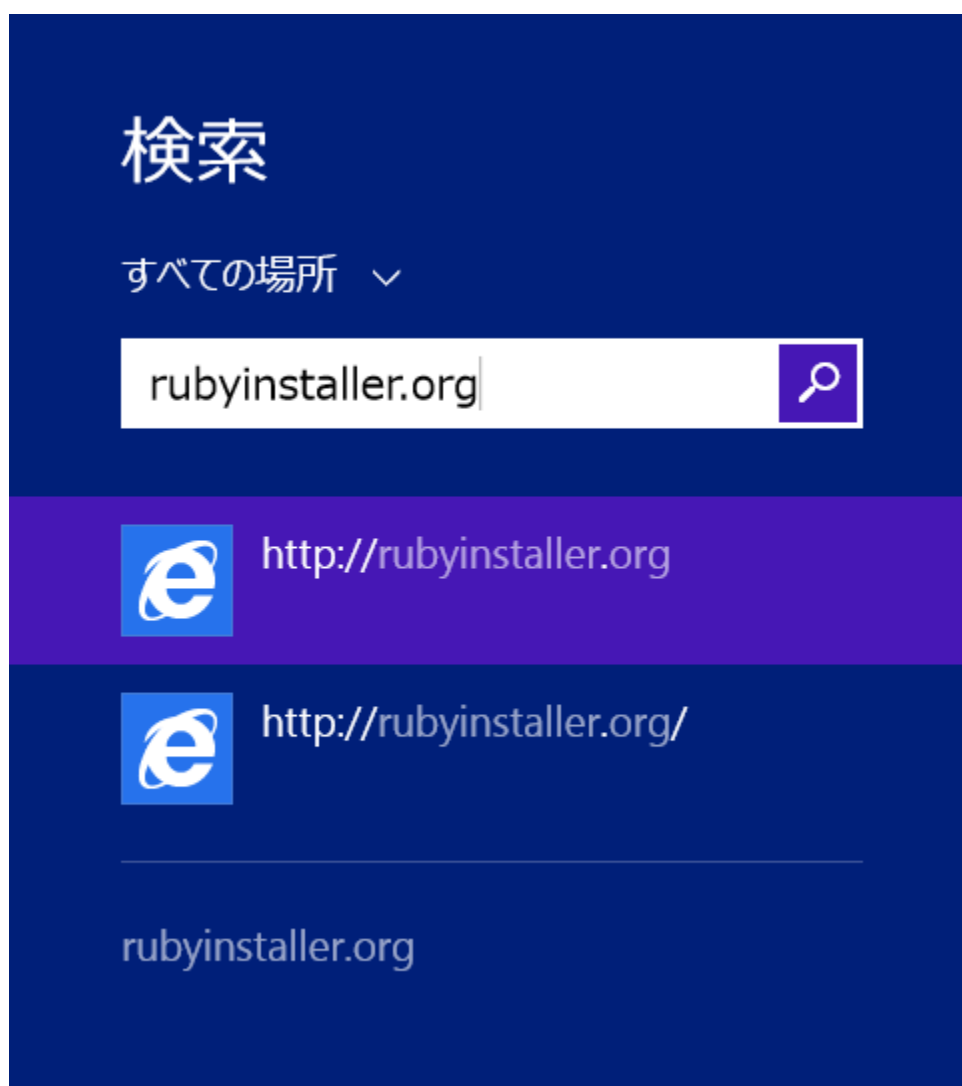


図 3.2 検索に「rubyinstaller.org」と入力することで「http://rubyinstaller.org」にアクセスすることもできるよ

表示された画面の「Download」ボタンを押します。



図 3.3 RubyInstaller for Windows のウェブサイト

表示された画面の左側に「Ruby 2.0.0-p247」と表示されているのが見つけれられるかな。見つかったらそれを押してね。そのすぐ下に「Ruby 2.0.0-p247 (x64)」というものもあるけど、これじゃないから間違えないでね。

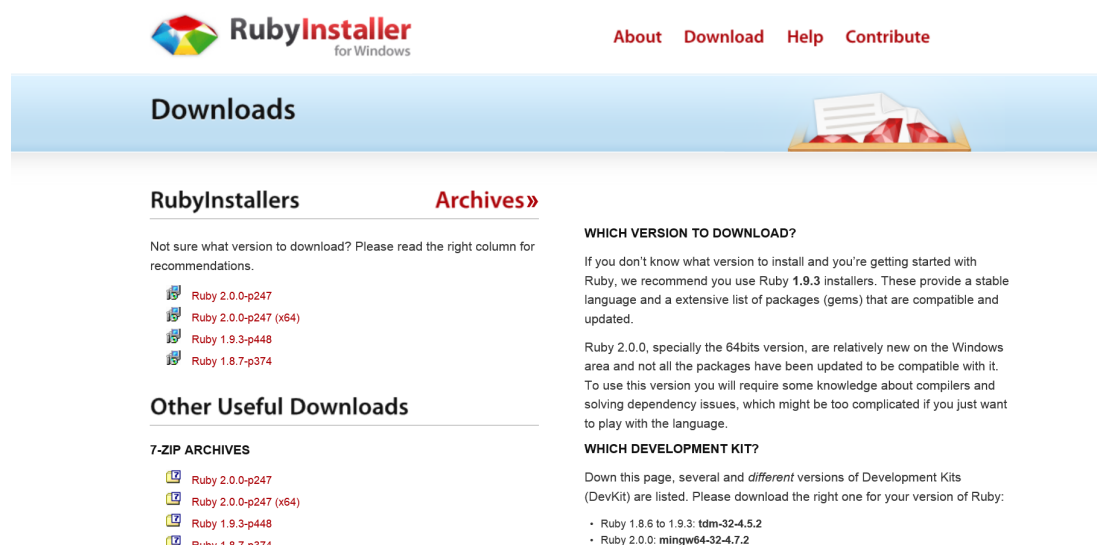


図 3.4 Ruby 2.0.0-p247 を押す

画面の下側に「dl.bintray.com から rubyinstaller-2.0.0-p247.exe(16.0MB) を実行または保存しますか?」と表示されているよね。その横の「保存 (S)」ボタンを押してね。

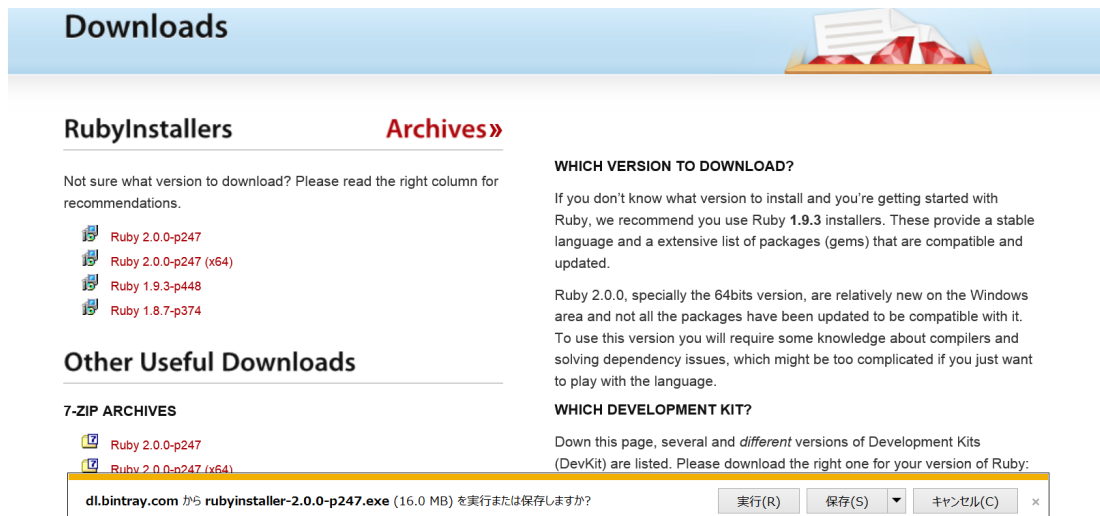


図 3.5 保存 (S) ボタンを押す

少し待っていると「rubyinstaller-2.0.0-p247.exe のダウンロードが完了しました。」と表示されるはず。これで Ruby をダウンロードできたよ。それじゃあその横の「実行 (R)」ボタンを押して、Ruby をインストールしよう。

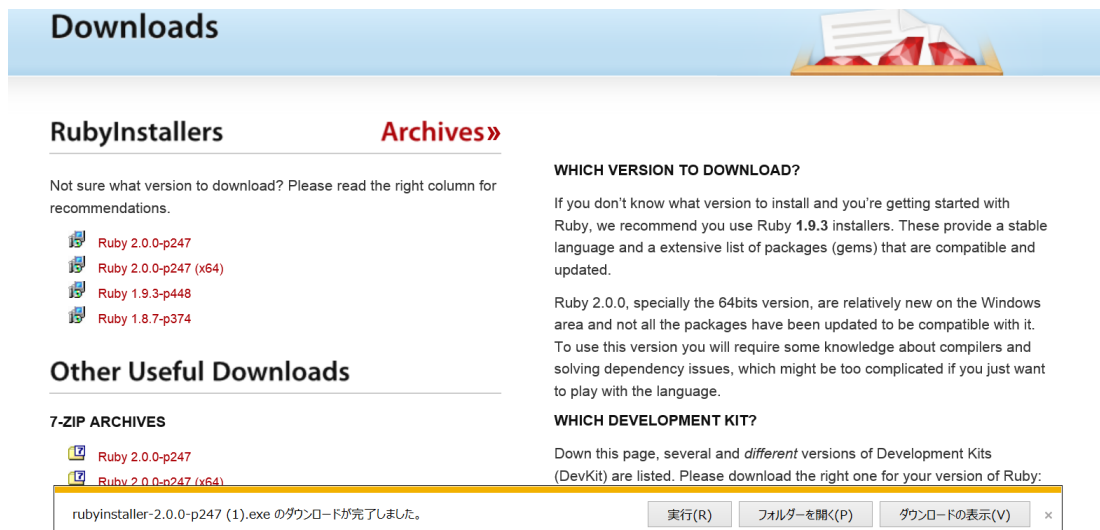


図 3.6 実行 (R) ボタンを押す

ここからは画面の表示に従って Ruby をインストールするよ。

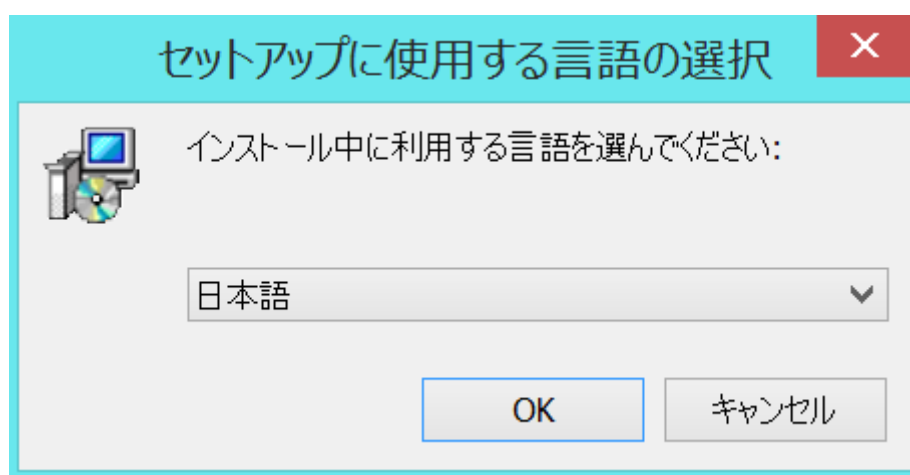


図 3.7 「OK」ボタンを押す

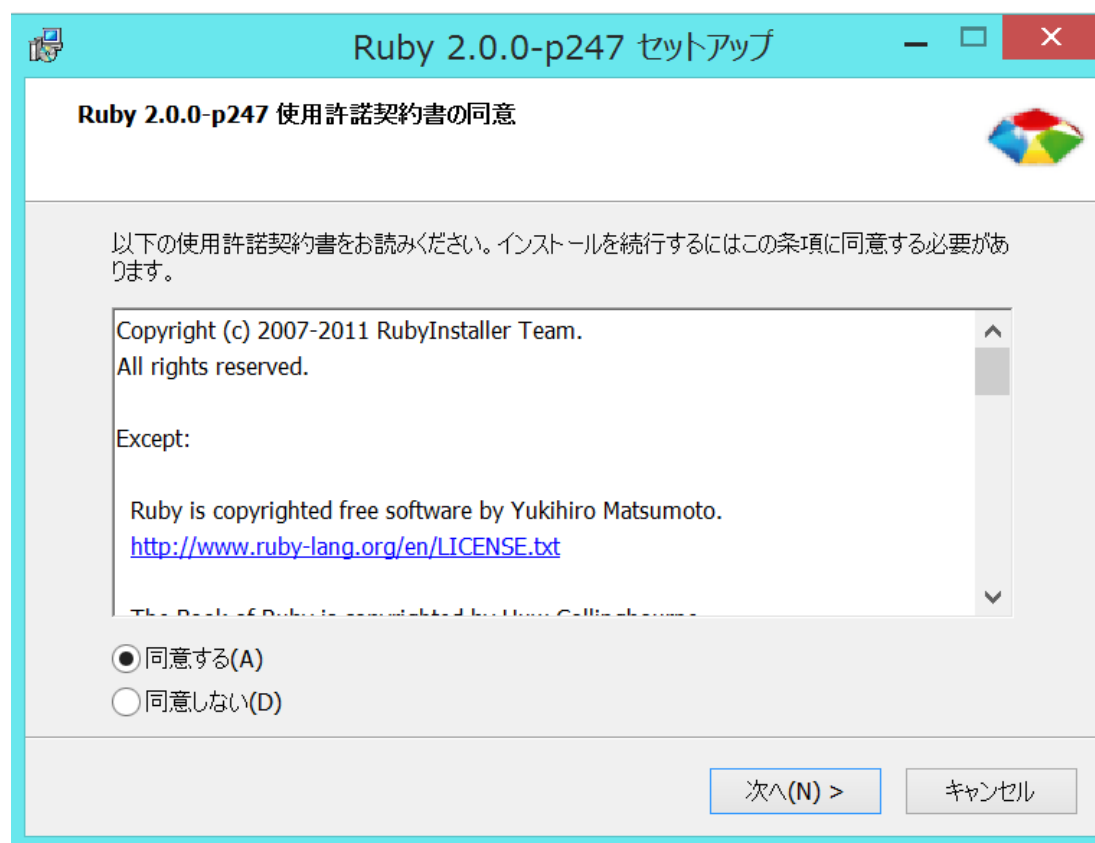


図 3.8 「同意する (A)」を押してから「次へ (N) >」ボタンを押す

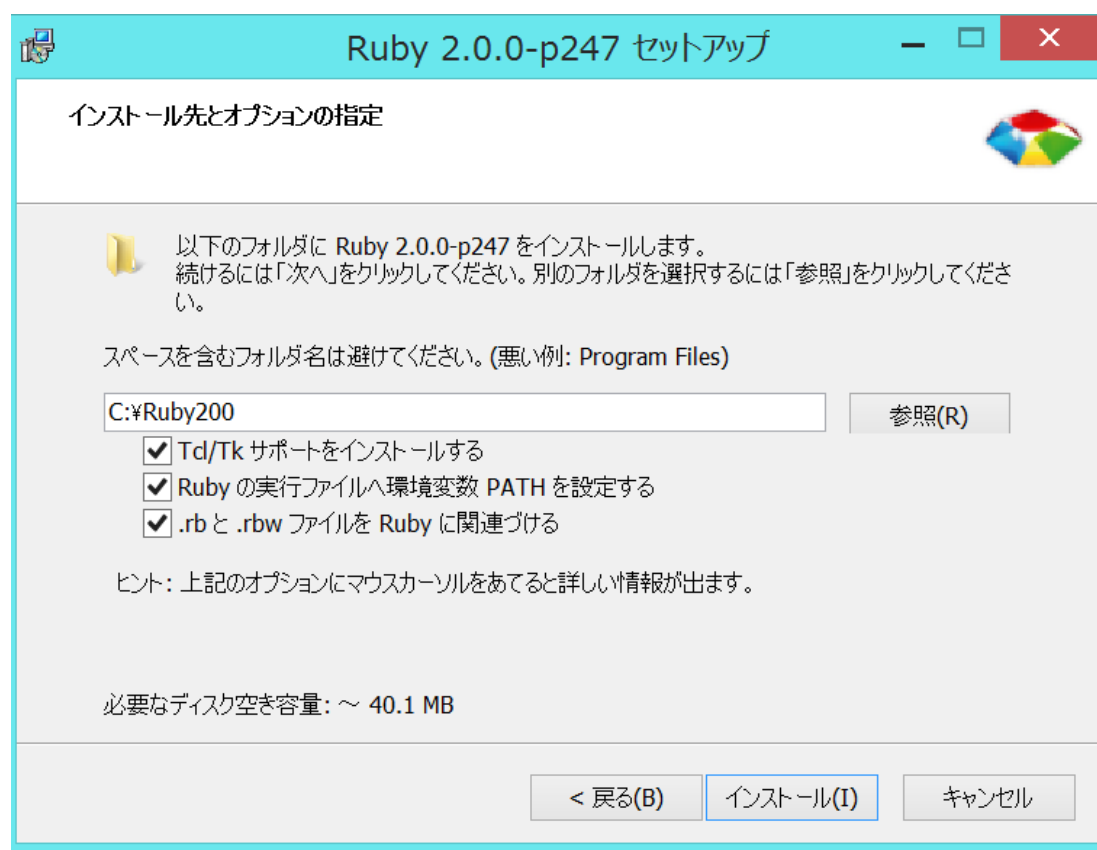


図 3.9 「Tcl/Tk サポートをインストールする」「Ruby の実行ファイルへ環境変数 PATH を設定する」「.rb と .rbw ファイルを Ruby に関連づける」を押してから「インストール (I)」ボタンを押す



図 3.10 「完了 (F)」ボタンを押す

これで Ruby のインストールは完了だよ。

続けて DevKit もダウンロード・インストールしよう。

さっき Ruby をダウンロードしたインターネットの画面を少し下にずらすと「DEVELOPMENT KIT」と表示されているのがわかるかな。そこからもう少し下のほうに「DevKit-mingw64-32-4.7.2-20130224-1151-sfx.exe」という赤い文字が見つけれられるかな。目印としてその文字の上に「For use with Ruby 2.0 (32bits version only):」と書かれているよ。見つけれたらそれを押してね。

さっきと同じように画面の下に「保存」ボタンが表示されるから、そのボタンを押してね。

DEVELOPMENT KIT

For use with Ruby 1.8.7 and 1.9.3:

 [DevKit-tdm-32-4.5.2-20111229-1559-sfx.exe](#)




For use with Ruby 2.0 (32bits version only):

 [DevKit-mingw64-32-4.7.2-20130224-1151-sfx.exe](#)

For use with Ruby 2.0 (x64 – 64bits only)

 [DevKit-mingw64-64-4.7.2-20130224-1432-sfx.exe](#)

MD5 Checksums

-  [RubyInstallers & 7-zip Archives](#)
-  [Documentation Archives](#)
-  [Development Kit](#)

Sorry the inconvenience.

SPEED AND COMPATIBILITY

RubyInstaller is compiled with MinGW which offers improved speed and better RubyGem compatibility, including support for many more native C-based extensions such as [Ruby FFI](#), [Nokogiri](#), [FXRuby](#) and [many others](#).

CONVENIENCE

No additional software is needed if you want to use the executable versions of the RubyInstaller. If you would like to use the 7-Zip archived versions or the Ruby documentation, you will need to download 7-Zip from the [7-Zip website](#).

DOCUMENTATION

As an added convenience for Windows users, we've made available the Ruby Core and Standard Library documentation in Compiled HTML Help (CHM) format.

BUILD YOUR OWN NATIVE EXTENSIONS

The [RubyInstaller Development Kit \(DevKit\)](#) is a MSYS/MinGW based toolkit than enables you to build many of the native C/C++ extensions available for Ruby.

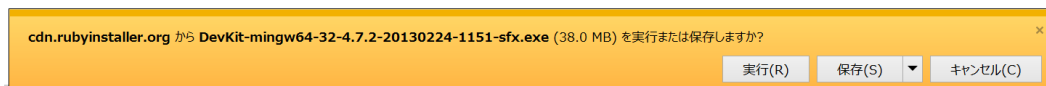
SUPPORT

図 3.11 DevKit-mingw64-32-4.7.2-20130224-1151-sfx.exe を押して、保存 (S) ボタンを押す

しばらくすると、これもさっきと同じように「実行 (R)」ボタンが表示されるから、そのボタンを押して DevKit をインストールしよう。

DEVELOPMENT KIT

For use with Ruby 1.8.7 and 1.9.3:

 [DevKit-tdm-32-4.5.2-20111229-1559-sfx.exe](#)


For use with Ruby 2.0 (32bits version only):

 [DevKit-mingw64-32-4.7.2-20130224-1151-sfx.exe](#)

For use with Ruby 2.0 (x64 – 64bits only)

 [DevKit-mingw64-64-4.7.2-20130224-1432-sfx.exe](#)

MD5 Checksums

-  [RubyInstallers & 7-zip Archives](#)
-  [Documentation Archives](#)
-  [Development Kit](#)

Sorry the inconvenience.

SPEED AND COMPATIBILITY

RubyInstaller is compiled with MinGW which offers improved speed and better RubyGem compatibility, including support for many more native C-based extensions such as [Ruby FFI](#), [Nokogiri](#), [FXRuby](#) and [many others](#).

CONVENIENCE

No additional software is needed if you want to use the executable versions of the RubyInstaller. If you would like to use the 7-Zip archived versions or the Ruby documentation, you will need to download 7-Zip from the [7-Zip website](#).

DOCUMENTATION

As an added convenience for Windows users, we've made available the Ruby Core and Standard Library documentation in Compiled HTML Help (CHM) format.

BUILD YOUR OWN NATIVE EXTENSIONS

The [RubyInstaller Development Kit \(DevKit\)](#) is a MSYS/MinGW based toolkit than enables you to build many of the native C/C++ extensions available for Ruby.

SUPPORT

Enjoy, happy Ruby coding, and let us know what you think or if you have any issues at our helpful and friendly [RubyInstaller Google Group](#).

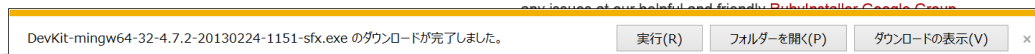


図 3.12 実行 (R) ボタンを押す

以下の画面のようにキーボードを操作して「C:\devkit」と入力して「extract」ボタンを押します。しばらくして画面が消えたら OK だよ。でも DevKit のインストールはこれだけでは終わりじゃなくて、もう少しやる必要があるんだ。

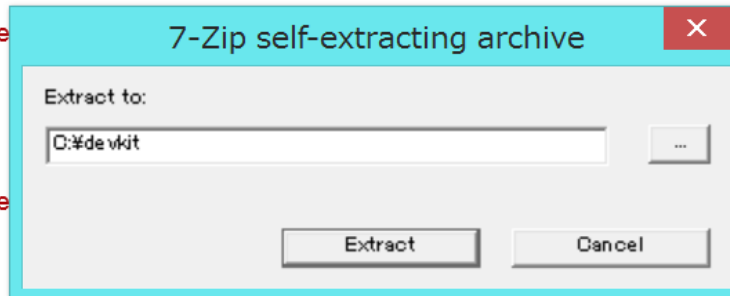
224-sfx.exe

better RubyGem compatibility, including support for C-based extensions such as **Ruby FFI**, **Noko**

only):

CONVENIENCE

224-1151-sfx.exe



ded if you want
ould like to use t
will need to do

only)

224-1432-sfx.exe

Windows users
Ruby Core and Standard Library documentat
(CHM) format.

BUILD YOUR OWN NATIVE EXTENSIONS

図 3.13 「C:\devkit」と入力して「extract」ボタンを押す

スタート画面から「コマンド プロンプト」というアプリを探して起動します。「コマンド プロンプト」は「Windows システムツール」の近くにあるよ。

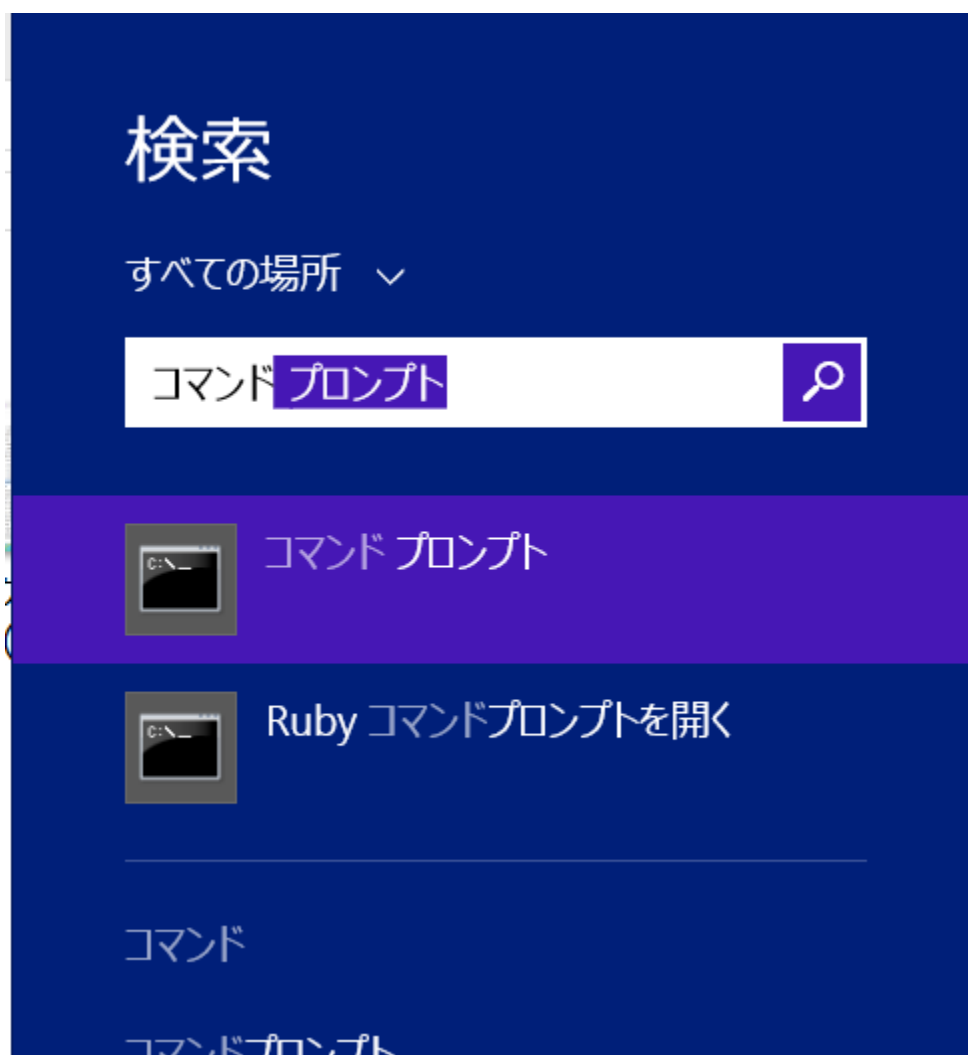


図 3.14 検索に「コマンド」と入力することで「コマンド プロンプト」を起動することもできるよ

キーボードを操作して以下を順番に入力するんだけど注意することがいくつかあるんだ。

```
cd \devkit
ruby dk.rb init
ruby dk.rb install
```

1 つは、「cd」と「\devkit」のように少し間が空いているところはスペースキーを打って必ず間を空けること。

もう 1 つは、「cd \devkit」まで入力したら Enter キーを打つこと。

それじゃあ、順番に入力していこう。

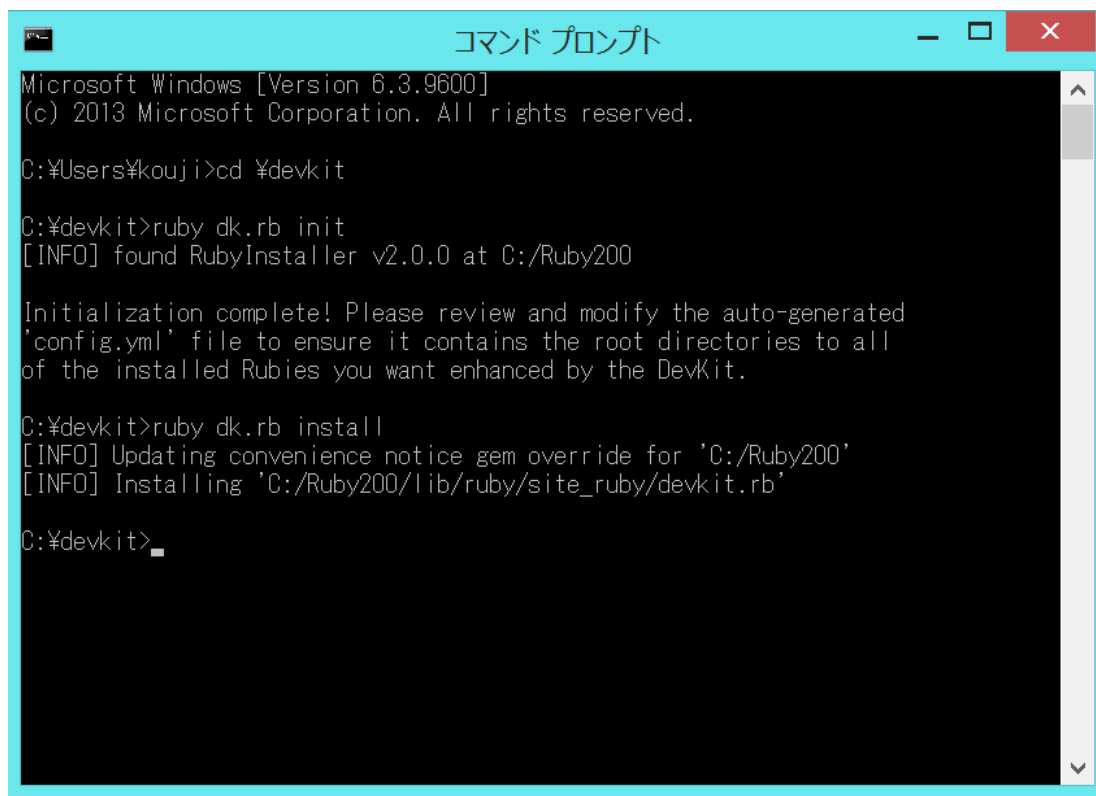


図 3.15 DevKit のインストール

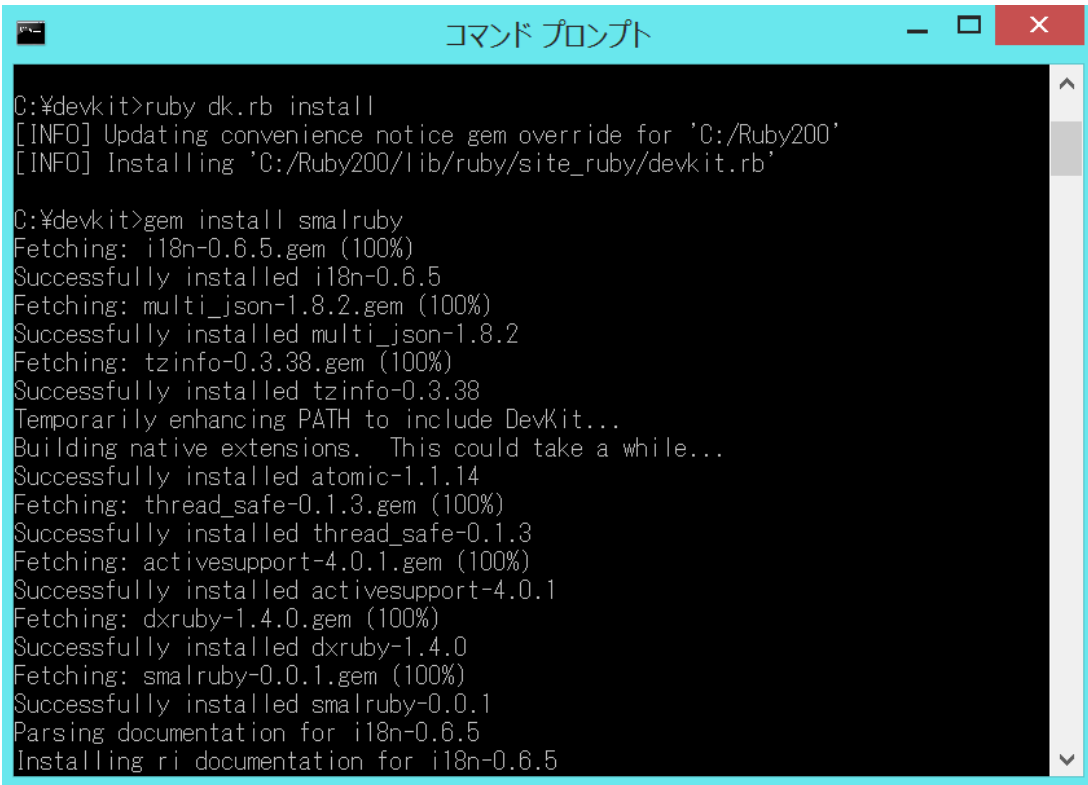
これで DevKit もインストールできた。コマンド プロンプトは立ち上げたままで次のステップに進んでね。

3.1.2 STEP2: Smalruby

続けて Smalruby をインストールするよ。

コマンド プロンプトで以下のコマンドを実行します。

```
>gem install smalruby
```

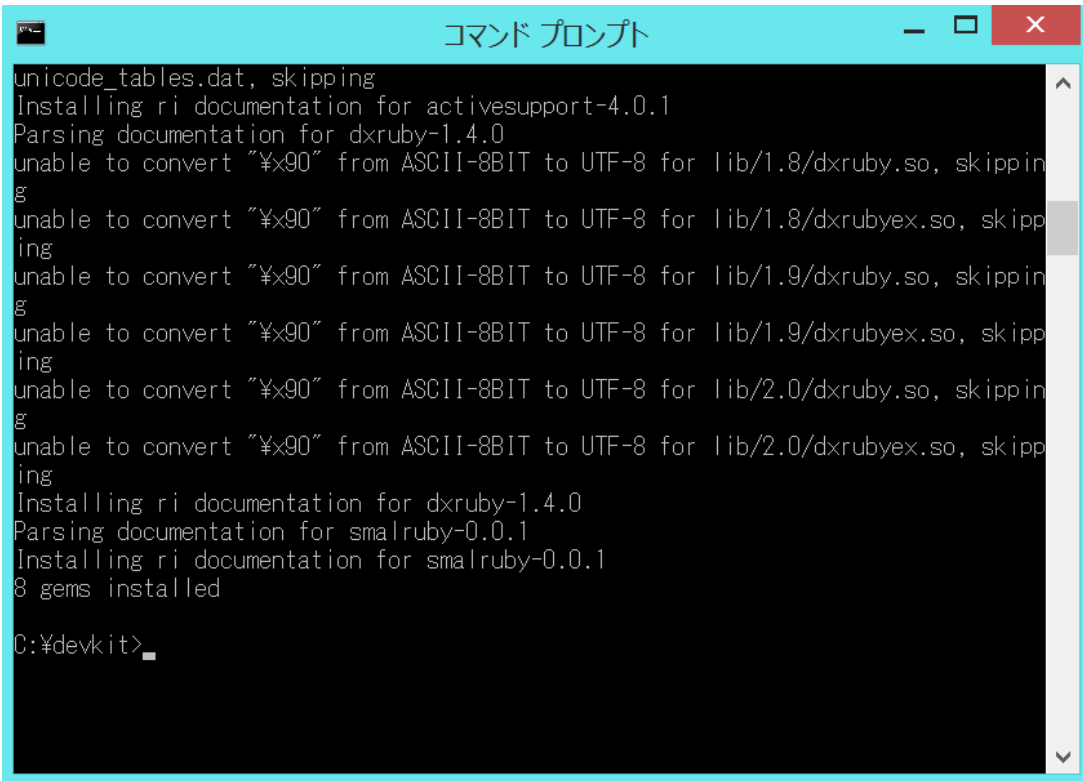



```
C:\¥devkit>ruby dk.rb install
[INFO] Updating convenience notice gem override for 'C:/Ruby200'
[INFO] Installing 'C:/Ruby200/lib/ruby/site_ruby/devkit.rb'

C:\¥devkit>gem install smalruby
Fetching: i18n-0.6.5.gem (100%)
Successfully installed i18n-0.6.5
Fetching: multi_json-1.8.2.gem (100%)
Successfully installed multi_json-1.8.2
Fetching: tzinfo-0.3.38.gem (100%)
Successfully installed tzinfo-0.3.38
Temporarily enhancing PATH to include DevKit...
Building native extensions. This could take a while...
Successfully installed atomic-1.1.14
Fetching: thread_safe-0.1.3.gem (100%)
Successfully installed thread_safe-0.1.3
Fetching: activesupport-4.0.1.gem (100%)
Successfully installed activesupport-4.0.1
Fetching: dxruby-1.4.0.gem (100%)
Successfully installed dxruby-1.4.0
Fetching: smalruby-0.0.1.gem (100%)
Successfully installed smalruby-0.0.1
Parsing documentation for i18n-0.6.5
Installing ri documentation for i18n-0.6.5
```

図 3.16 Smalruby のインストール

しばらくすると、「C:\devkit>」と表示されます。またその上には「8 gems installed」と表示されているはず。これで Smalruby をインストールできた。Ruby や DevKit に比べると簡単だったね。



```
unicode_tables.dat, skipping
Installing ri documentation for activesupport-4.0.1
Parsing documentation for dxruby-1.4.0
unable to convert "¥x90" from ASCII-8BIT to UTF-8 for lib/1.8/dxruby.so, skipping
unable to convert "¥x90" from ASCII-8BIT to UTF-8 for lib/1.8/dxrubyex.so, skipping
unable to convert "¥x90" from ASCII-8BIT to UTF-8 for lib/1.9/dxruby.so, skipping
unable to convert "¥x90" from ASCII-8BIT to UTF-8 for lib/1.9/dxrubyex.so, skipping
unable to convert "¥x90" from ASCII-8BIT to UTF-8 for lib/2.0/dxruby.so, skipping
unable to convert "¥x90" from ASCII-8BIT to UTF-8 for lib/2.0/dxrubyex.so, skipping
Installing ri documentation for dxruby-1.4.0
Parsing documentation for smalruby-0.0.1
Installing ri documentation for smalruby-0.0.1
8 gems installed

C:\devkit>
```

図 3.17 Smalruby のインストール完了！

これでプログラムを作る準備ができたよ。
それじゃあ、次の章に進もう！！

3.2 Mac OS X Mountain Lion の場合

Mac OS X Mountain Lion を使っている場合は Smalruby といっしょに次のソフトウェアもインストールする必要があるんだ。

- STEP1: Xcode
- STEP2: Homebrew
- STEP3: Ruby
- STEP4: SDL
- STEP5: Smalruby

それでは順番にインストールしてみよう。

3.2.1 STEP1: Xcode

はじめにインストールするのは Xcode だ。Xcode はきみが使っているコンピュータを使ってプログラムを作るためには必ず必要になるソフトウェアなんだ。まずはこれをインストールしよう。

Xcode をインストールするには Apple ID というものが必要になるんだ。きみが使っているコンピュータは Apple という会社を作っているんだけど、Apple の商品を買ったりするときに使うパスポートのようなもので、さまざまな場面で使うことになるんだ。もし、Apple ID が分からなければ、おうちの人に聞いてみよう。

Apple ID の用意ができたなら App Store を起動しよう。画面の左上の「りんごマーク」をクリックすると図 2.21 のようなものが表示されます。これをメニューといいます。

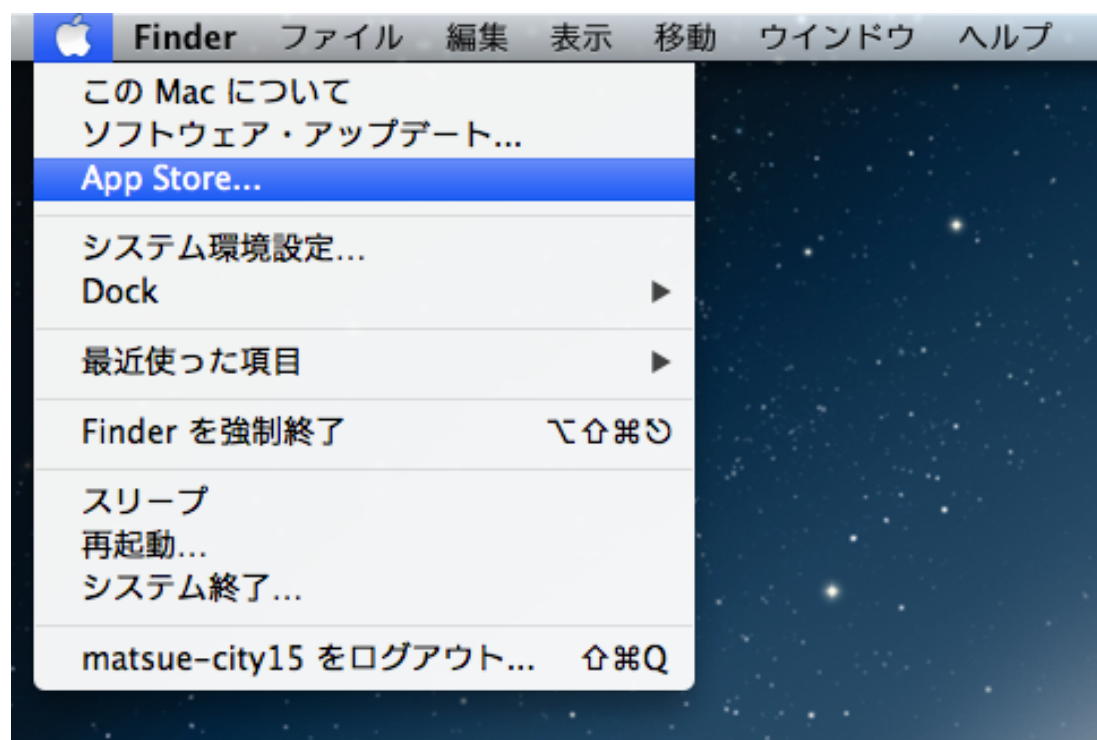


図 3.18 「App Store...」メニュー

メニューから「App Store...」をクリックします。すると図 2.22 のような画面が表示されます。これが App Store です。



図 3.19 App Store

App Store の画面の右上に虫眼鏡のマークがある。その横の部分をクリックしてキーボードで「Xcode」と入力します。すると、画面に Xcode のアイコンと「インストール」というボタンが表示される。「インストール」ボタンを押すと、さきほど用意した Apple ID とパスワードを入力するための画面が表示される。それぞれを入力して「サインイン」ボタンを押すとインストールが開始されます。

このときインターネットから大きなファイル（約 1.5GB）のファイルをダウンロードするため時間がかかります。ときどき画面を見ながら気長に待ちましょう。

インストールできたら、Xcode を起動します。メニューの [Xcode]-[Preference...] を選択します。Downloads タブの Components タブを選択します。Command Line Tools の Install ボタンをクリックします。インストールされるまでしばらく待ちます。

3.2.2 STEP2: Homebrew

ターミナルを起動します。

以下のコマンドを実行します。

```
$ /usr/bin/ruby -e "$(curl -fsSL https://raw.githubusercontent.com/mxcl/homebrew/go)"
```

以下の表示がでます。return キーを押します。

```
==> This script will install:
/usr/local/bin/brew
/usr/local/Library/...
/usr/local/share/man/man1/brew.1
==> The following directories will be made group writable:
/usr/local/.

Press ENTER to continue or any other key to abort
```

以下の表示がでます。コンピュータを使うときに入力するパスワードを入れて enter キーを押します。

```
==> /usr/bin/sudo /bin/chmod g+rxw /usr/local/.
Password:
```

以下の表示が出たら完了です。

```
==> Downloading and installing Homebrew...
remote: Finding bitmap roots...
remote: Counting objects: 136502, done.
remote: Compressing objects: 100% (45289/45289), done.
remote: Total 136502 (delta 94462), reused 132027 (delta 90171)
Receiving objects: 100% (136502/136502), 25.04 MiB | 906 KiB/s, done.
Resolving deltas: 100% (94462/94462), done.
From https://github.com/mxcl/homebrew
 * [new branch]      master      -> origin/master
HEAD is now at eb32df9 mess 0.151
==> Installation successful!
You should run 'brew doctor' *before* you install anything.
Now type: brew help
```

3.2.3 Ruby

Ruby をインストールします。

ターミナルを起動して以下のコマンドを実行する。

```
$ brew install rbenv ruby-build
```

以下のように表示されてインストールが完了する。

```

==> Downloading https://github.com/sstephenson/rbenv/archive/v0.4.0.tar.gz
##### 100.0%
==> Caveats
To use Homebrew's directories rather than ~/.rbenv add to your profile:
  export RBENV_ROOT=/usr/local/var/rbenv

To enable shims and autocompletion add to your profile:
  if which rbenv > /dev/null; then eval "$(rbenv init -)"; fi
(省略)
==> make install
^^f0^^9f^^8d^^ba /usr/local/Cellar/libyaml/0.1.4: 7 files, 340K, built in 13 seconds
==> Installing ruby-build
==> Downloading https://github.com/sstephenson/ruby-build/archive/v20131030.tar.
##### 100.0%
==> ./install.sh
^^f0^^9f^^8d^^ba /usr/local/Cellar/ruby-build/20131030: 89 files, 396K, built in 4 seconds

```

以下のコマンドを順番に実行します。

```

$ echo 'eval "$(rbenv init -)'" >> ~/.bash_profile
$ echo 'export PATH="$HOME/.rbenv/shims:$PATH"' >> ~/.bash_profile
$ source ~/.bash_profile

```

以下のコマンドを実行する。

```
$ rbenv install 1.9.3-p448
```

以下のように表示されます。

```

Downloading ruby-1.9.3-p448.tar.gz...
-> http://dqw8nmjcqpjn7.cloudfront.net/a893cff26bcf351b8975ebf2a63b1023
Installing ruby-1.9.3-p448...

```

しばらく待っていると以下のように表示されてインストールが完了します。

```

Installed ruby-1.9.3-p448 to /Users/matsue-city15/.rbenv/versions/1.9.3-p448
$

```

インストールされていることを確認するには以下のコマンドを実行します。

```
$ rbenv versions
```

以下のように表示されたら正しくインストールされています。

```
* system (set by /Users/matsue-city15/.rbenv/version)
  1.9.3-p448
```

以下のコマンドを実行する。

```
$ rbenv global 1.9.3-p448
```

以下のコマンドを実行して、表示された結果が一致していれば正しく設定できている。

```
$ ruby --version
ruby 1.9.3p448 (2013-06-27 revision 41675) [x86_64-darwin12.4.0]
```

コラム: RVM をアンインストールする

ターミナルを起動する。

以下のコマンドを実行する。

```
$ rvm seppuku
```

以下が表示されます。

```
Are you SURE you wish for rvm to implode?
This will recursively remove /usr/local/rvm and other rvm traces?
(anything other than 'yes' will cancel) >
```

以下のようにキーボードで yes を入力して enter キーを押す。

```
(anything other than 'yes' will cancel) > yes
```

以下のように表示されたら ok です。

```
Removing rvm-shipped binaries (rvm-prompt, rvm, rvm-sudo rvm-shell and rvm-auto-ruby)
Removing rvm wrappers in /usr/local/rvm/bin
Hai! Removing /usr/local/rvm
rm: /etc/profile.d/rvm.sh: Permission denied
/usr/local/rvm has been removed.

Note you may need to manually remove /etc/rvmrc and ~/.rvmrc if they exist still.
Please check all .bashrc .bash_profile .profile and .zshrc for RVM source lines and de
Also make sure to remove 'rvm' group if this was a system installation.
Finally it might help to relogin / restart if you want to have fresh environment (like
```

RailsInstaller でインストールされたファイルを削除する。

ターミナルを起動する。

以下のコマンドを実行する。

```
$ sudo rm /etc/profile.d/rvm.sh /etc/rvmrc
```

以下のように表示されたら、ログインするときに入力するパスワードを入力して enter キーを押す。

```
Password:
```

以下のように表示されたら削除できています。

```
$
```

設定ファイルを修正します。以下のコマンドを実行してエディタを起動します。そして最下部にある「source /etc/profile.d/rvm.sh」の行を削除します。カーソルキーで上記の行に移動して、キーボードの d を 2 回押す。ここで画面下部に赤い文字で「W10: Warning: Changing a readonly file」と表示されるが問題ない。行が消えたら、キーボードの「:」「w」「q」「!」を順番に押して ennter キーを押す。


```
$ sudo vi /etc/profile
```

これで rvm のアンインストールができました。

アンインストールした結果を反映させるために、いったん、ターミナルを終了させます。

3.2.4 SDL

SDL をインストールします。

```
$ brew install sdl
==> Downloading http://www.libsdl.org/release/SDL-1.2.15.tar.gz
##### 100.0%
==> ./configure --prefix=/usr/local/Cellar/sdl/1.2.15 --without-x
(ここで少し待つ)
==> make install
^^f0^^9f^^8d^^ba /usr/local/Cellar/sdl/1.2.15: 223 files, 2.1M, built in 45 seconds
```

以下のコマンドを実行する。

```
$ brew install sdl_image
==> Installing dependencies for sdl_image: jpeg, libpng, libtiff, webp
==> Installing sdl_image dependency: jpeg
==> Downloading https://downloads.sf.net/project/machomebrew/Bottles/jpeg-8d.mou
##### 100.0%
==> Pouring jpeg-8d.mountain_lion.bottle.tar.gz
(省略)
==> Installing sdl_image
==> Downloading http://www.libsdl.org/projects/SDL_image/release/SDL_image-1.2.1
##### 100.0%
==> ./configure --prefix=/usr/local/Cellar/sdl_image/1.2.12 --disable-sdltest
==> make install
^^f0^^9f^^8d^^ba /usr/local/Cellar/sdl_image/1.2.12: 8 files, 140K, built in 14 seconds
```

以下のコマンドを実行する。

```
$ brew install sdl_mixer
==> Downloading http://www.libsdl.org/projects/SDL_mixer/release/SDL_mixer-1.2.1
##### 100.0%
==> ./configure --prefix=/usr/local/Cellar/sdl_mixer/1.2.12
```

```
==> make install
^^f0^^9f^^8d^^ba /usr/local/Cellar/sdl_mixer/1.2.12: 8 files, 424K, built in 16 seconds
```

以下のコマンドを実行する。

```
$ brew install sdl_ttf
==> Installing sdl_ttf dependency: freetype
==> Downloading https://downloads.sf.net/project/machomebrew/Bottles/freetype-2.
##### 100.0%
==> Pouring freetype-2.5.0.1.mountain_lion.bottle.1.tar.gz
^^f0^^9f^^8d^^ba /usr/local/Cellar/freetype/2.5.0.1: 59 files, 2.6M
==> Installing sdl_ttf
==> Downloading http://www.libsdl.org/projects/SDL_ttf/release/SDL_ttf-2.0.11.ta
##### 100.0%
==> ./configure --prefix=/usr/local/Cellar/sdl_ttf/2.0.11 --disable-sdltest
==> make install
^^f0^^9f^^8d^^ba /usr/local/Cellar/sdl_ttf/2.0.11: 8 files, 100K, built in 12 seconds
```

以下のコマンドを実行する。

```
$ brew install https://gist.github.com/mitmul/5410467/raw/c4fa716635e951b61f489726976b10f0

==> Cloning git://github.com/flibitijibibo/libSGE.git
Cloning into '/Library/Caches/Homebrew/sge--git'...
remote: Counting objects: 78, done.
remote: Compressing objects: 100% (73/73), done.
remote: Total 78 (delta 5), reused 73 (delta 5)
Receiving objects: 100% (78/78), 267.15 KiB | 156 KiB/s, done.
Resolving deltas: 100% (5/5), done.
==> export HAVE_SDL='y'
==> make
==> make install
^^f0^^9f^^8d^^ba /usr/local/Cellar/sge/r030809: 4 files, 44K, built in 4 seconds
```

3.2.5 Smalruby

以下のコマンドを実行する。

```
$ cd ~
$ git clone https://github.com/smalruby/smalruby smalruby-lib
Cloning into 'smalruby-lib'...
remote: Counting objects: 249, done.
remote: Compressing objects: 100% (184/184), done.
remote: Total 249 (delta 115), reused 187 (delta 53)
```

```
Receiving objects: 100% (249/249), 82.76 KiB, done.  
Resolving deltas: 100% (115/115), done.
```

以下のコマンドを実行する。

```
$ cd smalruby-lib/
```

以下のコマンドを実行する。

```
$ echo 'gem: --no-ri --no-rdoc' > ~/.gemrc
```

以下のコマンドを実行する。

```
$ gem install bundler  
Successfully installed bundler-1.3.5  
1 gem installed
```

以下のコマンドを実行する。

```
$ rbenv rehash
```

以下のコマンドを実行する。

```
$ bundle  
Fetching gem metadata from https://rubygems.org/.....  
Fetching gem metadata from https://rubygems.org/..  
Resolving dependencies...  
Installing rake (10.1.0)  
Installing i18n (0.6.5)  
Installing minitest (4.7.5)  
Installing multi_json (1.8.2)  
Installing atomic (1.1.14)  
Installing thread_safe (0.1.3)  
Installing tzinfo (0.3.38)  
Installing activesupport (4.0.1)  
Installing ast (1.1.0)  
Using bundler (1.3.5)
```

```
(省略)
Installing rubocop (0.15.0)
Installing guard-rubocop (1.0.0)
Installing hashr (0.0.22)
Installing redcarpet (3.0.0)
Using smalruby (0.0.1) from source at /Users/matsue-city15/smalruby-lib
Installing travis-lint (1.7.0)
Installing yard (0.8.7.3)
Your bundle is complete!
Use 'bundle show [gemname]' to see where a bundled gem is installed.
```

以下のコマンドを実行する。

```
$ rbenv rehash
```

サンプルプログラムを起動する。画面をクリックすると `rsdl` が表示され、`esc` キーを押すと終了する。

```
$ bundle exec rsdl samples/finding_cars.rb
```

以下のコマンドを実行する。

```
$ rake build
smalruby 0.0.1 built to pkg/smalruby-0.0.1.gem.
```

以下のコマンドを実行する。

```
$ gem install pkg/smalruby-0.0.1.gem
Successfully installed smalruby-0.0.1
1 gem installed
```

3.3 Mac OS X Mavericks の場合

3.4 GNU/Linux (Ubuntu) の場合

第 4 章

サンプルプログラムで遊ぶ

- 4.1 ゲームの動かし方
- 4.2 サンプルプログラム 1
- 4.3 サンプルプログラム 2
- 4.4 サンプルプログラム 3
- 4.5 サンプルプログラム 4
- 4.6 サンプルプログラム 5

小・中学生にもできる！ ゲームプログラミング ～**Ruby**を使ってゲームの作り方を学ぼう～

2014 年 4 月 1 日 v1.0.0 版発行

著 者 高尾宏治

発行所

(C) 2013 Kouji Takao