

## 参考

<https://qiita.com/oshou/items/6283c2315dc7dd244aef>

<https://qiita.com/HayneRyo/items/d493a2b3cec2322f167c>

## Ruby 関連ツール

gem

Ruby 用ライブラリの呼び方

bundler

gem の一つ。gem 同士のバージョン、互換性を保ちながら gem の導入、管理を行うツール。

rbenv

Ruby の複数のバージョンを管理、切替を行うツール

## Bundler

### インストール

```
gem install bundler
```

### 使い方

初期化 (Gemfile という雛形が作成される)

```
bundle init
```

Gemfile を編集

Gemfile にリスト化した gem を一括インストール

```
bundle install --path vendor/bundle
```

--path でインストール先を指定可能。

bundle でインストールされた gem パッケージを使用してコマンドを実行する

```
bundle exec コマンド
```

bundle exec を使わない場合は、システム側の gem を指定したことになる。

### rbenv

### インストール

ユーザごとにインストールする場合

必要なパッケージ導入

```
yum install -y openssl-devel readline-devel zlib-devel gcc make git
```

## git から取得

```
git clone https://github.com/sstephenson/rbenv.git ~/.rbenv
git clone https://github.com/sstephenson/ruby-build.git ~/.rbenv/plugins/ruby-build
```

## 環境変数設定

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

## システムワイドにインストールする場合

### 必要なパッケージ導入

```
yum install -y openssl-devel readline-devel zlib-devel gcc make git
```

## git から取得

```
git clone https://github.com/sstephenson/rbenv.git /opt/rbenv
git clone https://github.com/sstephenson/ruby-build.git /opt/rbenv/plugins/ruby-build
```

## グループの設定

```
groupadd rbenv
gpasswd -a root rbenv
gpasswd -a ユーザ rbenv
```

## 権限設定

```
chgrp -R rbenv /opt/rbenv
chmod g+w -R /opt/rbenv
```

## 環境変数設定

```
echo 'export RBENV_ROOT=/opt/rbenv' >> ~/.bash_profile
echo 'export PATH="$RBENV_ROOT/bin:$PATH"' >> ~/.bash_profile
echo 'eval "$(rbenv init -)"' >> ~/.bash_profile
```

## 使い方

### 利用可能な ruby のバージョン確認

```
rbenv install -l
```

### 指定のバージョンをインストール

```
rbenv install 2.4.0
```

### インストールした ruby のバージョンの一覧

```
rbenv versions
```

利用するバージョンを指定する

```
rbenv global 2.4.0
```

特定のディレクトリにバージョンを指定する

```
rbenv local 2.2.2
```

## 対話型

irb

```
irb
```

を実行すると対話式に ruby を実行できる。  
毎回戻り値や代入結果が表示が表示される。もし、表示が邪魔な場合は

```
conf.echo = false
```

で止められる。表示を再開するには

```
conf.echo = true
```

pry

irb を便利にしたもの。  
エコーが邪魔な場合は

```
_pry_.config.print = proc {}
```

で止められる。  
一時的にオフ、オンする場合は

```
_pry_.config._print = _pry_.config.print  
_pry_.config.print = proc{}
```

オンに戻す

```
_pry_.config.print = _pry_.config._print
```