

# cheat sheets.

\$ cheat rvm

RVM home page: <https://rvm.io>

Install RVM

-----

See <https://rvm.io/rvm/install/>

```
curl -L get.rvm.io | bash -s stable
```

On OS X you can use RailsInstaller which will bring all dependencies for you:  
installer: <https://github.com/railsinstaller/railsinstaller-nix/downloads>  
background: <http://www.engineyard.com/blog/2012/railsinstaller-for-os-x/>

Update RVM

-----

```
rvm get stable
```

In case of problems try first with development version  
(maybe it's already fixed):

```
rvm get head
```

Very old installations might not support those update methods, just run the installer and reopen your terminal.

Install Ruby dependencies

-----

Follow instructions from:

```
rvm requirements
```

List known rubies

-----

```
rvm list known
```

Install Ruby

-----

```
rvm install 1.9.3                # Latest known patch level
rvm install 1.9.3 -j 3           # Parallel compile, set to # of CPU cores
rvm install 1.9.3 --patch falcon # Use a patch (falcon for performance)
rvm install 1.9.2-p318           # Patchlevel 318
rvm install rbx --1.9            # Rubinius with 1.9 mode set as default
```

List all rubies and gemsets

-----

```
rvm list          # List rubies only
rvm list gemsets  # List rubies and gemsets
rvm gemset list   # List gemsets for current ruby
```

Selecting Ruby for work

-----

```
rvm system        # For system ruby, with fallback to default
rvm use jruby     # For current session only
rvm use --default 1.9.3 # For current and new sessions
rvm use --ruby-version rbx # For current session and this project
```

RVM will automatically use a ruby and gemset when you `cd` to a project directory.

Read more on project files:

- <https://rvm.io/workflow/projects/#ruby-versions>

Using ruby and gems

After selecting Ruby work as usual:

```
ruby -v
gem install haml
haml
```

Temporarily selecting another Ruby or gemset

```
rvm 1.8.7 do gem install rspec      # in the given ruby
rvm 1.8.7,1.9.2 do gem install haml # in this two rubies
rvm @global do gem install gist     # in @global gemset of current ruby
```

Gemsets

RVM by default allows creating multiple environments for one ruby - called \*gemsets\*.

Gemsets can be specified together with ruby name using gemsets separator(@):

- ruby-1.9.3-p125@my-project

During installation of Ruby, RVM creates two gemsets:

- default - automatically selected when no @gemset specified: rvm use 1.9.3
- global - super gemset, inherited by all other gemsets for the given ruby

Working with gemsets:

```
rvm use 1.8.7                # use the ruby to manage gemsets for
rvm gemset create project_name # create a gemset
rvm gemset use project_name   # use a gemset in this ruby
rvm gemset list               # list gemsets in this ruby
rvm gemset delete project_name # delete a gemset
rvm 1.9.1@other_project_name   # use another ruby and gemset
rvm 1.9.3@_project --create --rvmrc # use and create gemset & .rvmrc
```

Install RVM for all users

Discouraged; make sure to read <http://rvm.io/rvm/installation/>

More help ?

A lot of resources is available:

- in your terminal: `rvm help`
- <https://rvm.io/>
- IRC #rvm @ freenode - <http://webchat.freenode.net/?channels=#rvm>