RubyGems Ruby package manager

CONCEPTS

gem installs gems machine-wide. For a project, list them in a Gemfile and use bundle (see Bundler) so everyone gets the same versions.

Gemspec your gem's metadata
↓ `gem build`
.gem file packaged archive
↓ `gem push`
rubygems.org the default source
↓ `gem install`
GEM_HOME installed gems
↓ `require`
Your Ruby code

Without write access to the system Ruby, gem install fails with a permissions error. Use --user-install, or a Ruby version manager that keeps gems in your home.

INSTALL

`gem install <gem>`
Install the latest version

`gem install <gem> -v "~> 1.2"`
Install a version matching a constraint

`gem install <gem> --user-install`
Install into your home directory

`gem install <gem> --no-document`
Skip generating docs (much faster)

`gem install <gem> --prerelease`
Allow prerelease versions

LIST & SEARCH

`gem list`
List installed gems

`gem list <regexp>`
Filter installed gems by name

`gem search <regexp>`
Search remote gems by name

`gem info <gem>`
Version, license, homepage, install path

UPDATE & REMOVE

`gem outdated`
List gems with newer versions

UPDATE & REMOVE (cont.)

`gem update`
Update all installed gems

`gem update <gem>`
Update one gem

`gem update --system`
Update RubyGems itself

`gem uninstall <gem>`
Remove a gem

`gem uninstall <gem> -v <version>`
Remove one version

`gem cleanup`
Remove old versions of gems

SOURCES

`gem sources`
Show configured sources

`gem sources -a <url>`
Add a source

BUILD & PUBLISH

`gem build <name>.gemspec`
Package a gem into a .gem file

`gem signin`
Sign in and store an API key

`gem push <name>-<version>.gem`
Upload a gem to rubygems.org

ENVIRONMENT

`gem env`
Show RubyGems configuration

`gem env home`
Where gems are installed

`gem --version`
Print the RubyGems version

`gem help <command>`
Help for one command

Version constraints (-v)

Constraint	Meaning
`"1.2.3"`	exactly 1.2.3
`"≥ 1.2"`	1.2 or newer
`"~> 1.2"`	1.2 up to below 2.0
`"~> 1.2.3"`	1.2.3 up to below 1.3
`"< 2"`	anything below 2