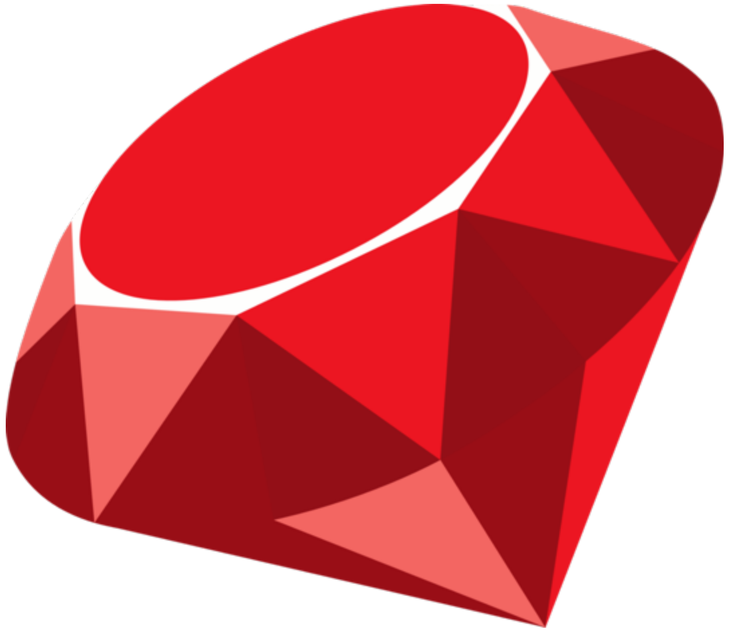
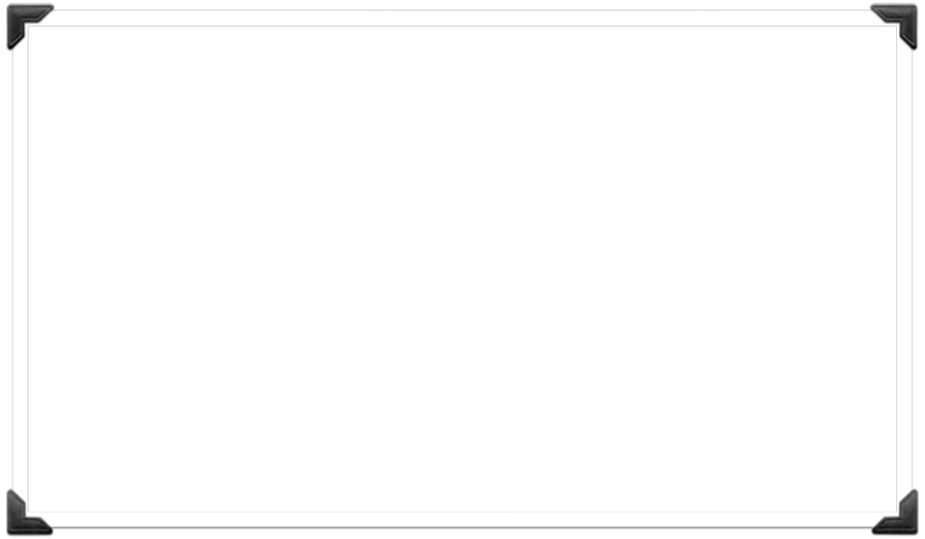
Ruby Gems

Designing and Maintaining Software (DAMS)

Louis Rose





How is Ruby code shared?

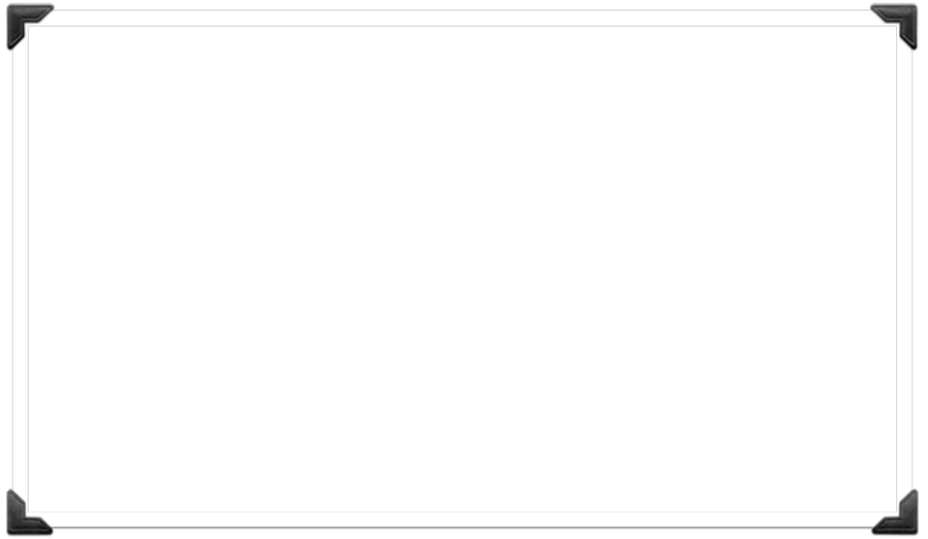
Gems are Ruby's software packages

A gem contains:

- Ruby code (including tests, etc)
- Documentation
- Metadata (such as a version number and list of authors

The gem command is used for searching for, installing, updating, and removing Ruby gems.

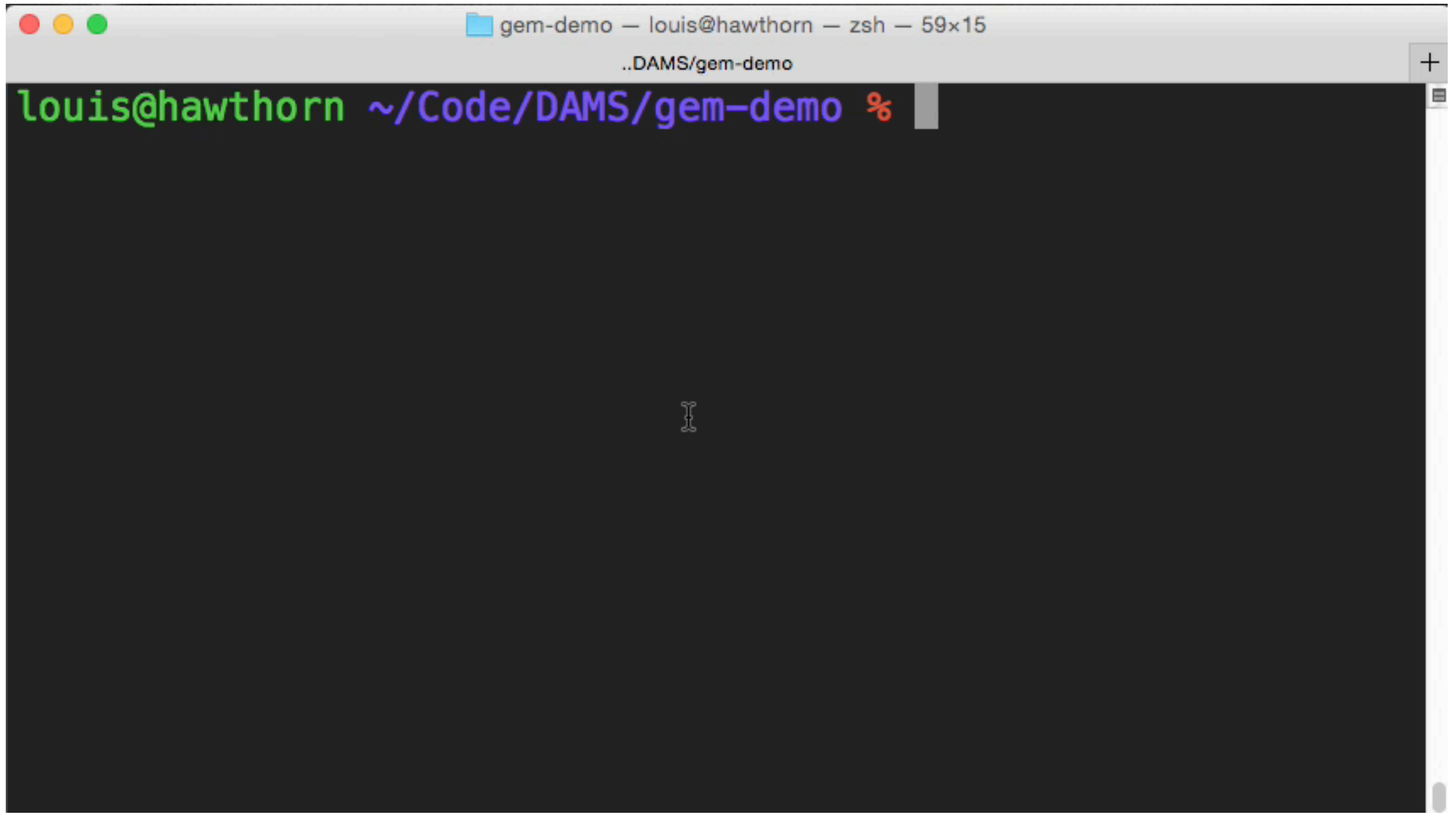
https://rubygems.org



Example: generating fake data

Suppose our customer asks for realistic but fake user data.

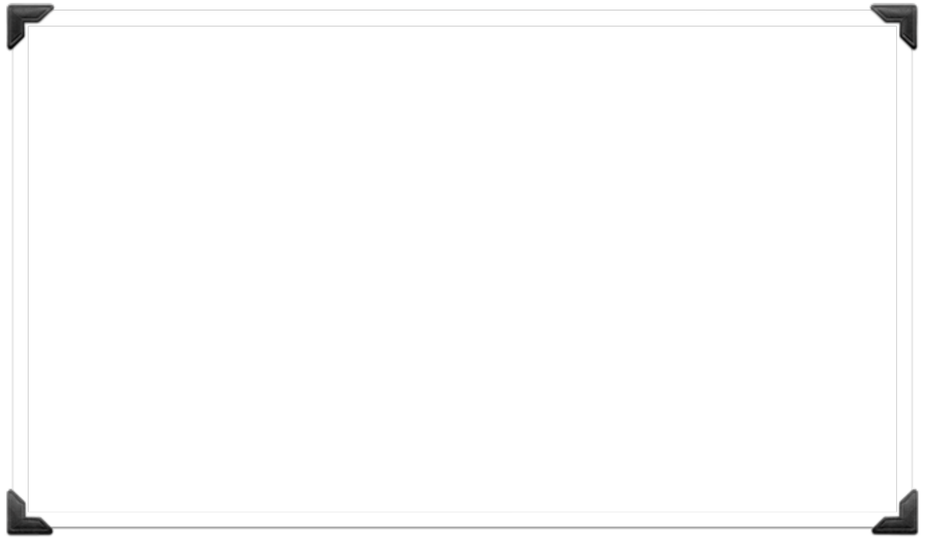
There's a Ruby gem for that: faker.



Problem: how to discover gems?

A few choices:

- Your favourite search engine
- StackOverflow answers
- Github search (Rubyists love Github)
- My favourite: Ruby Toolbox (ruby-toolbox.com)

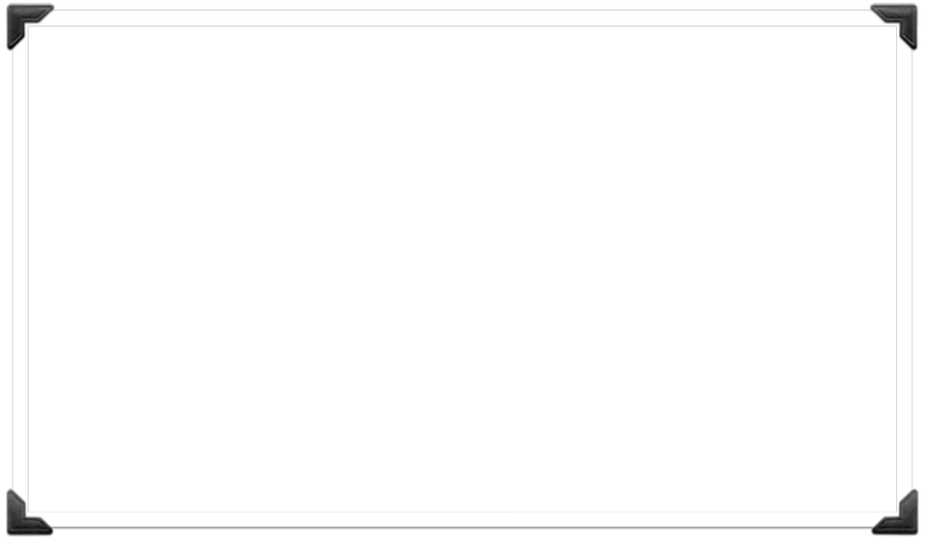


Problem: versioning

Which version of Faker did our app depend on?

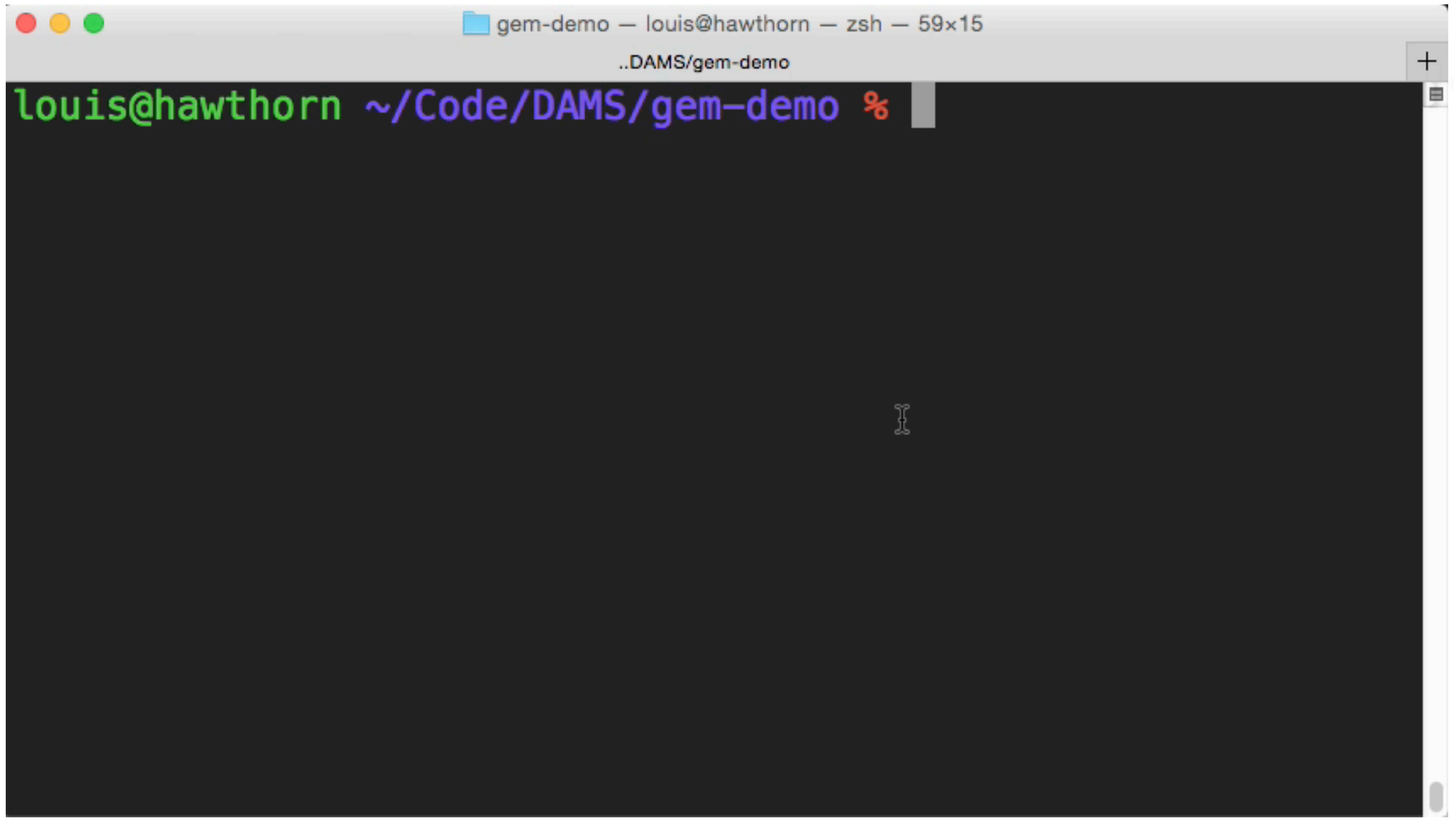
If we run our app in a year and on a different machine, will it work?

If we are developing several apps and they each require different versions of Faker, will our apps work?



Gem Management with Bundler

Bundler manages app-specific dependencies.



https://bundler.io

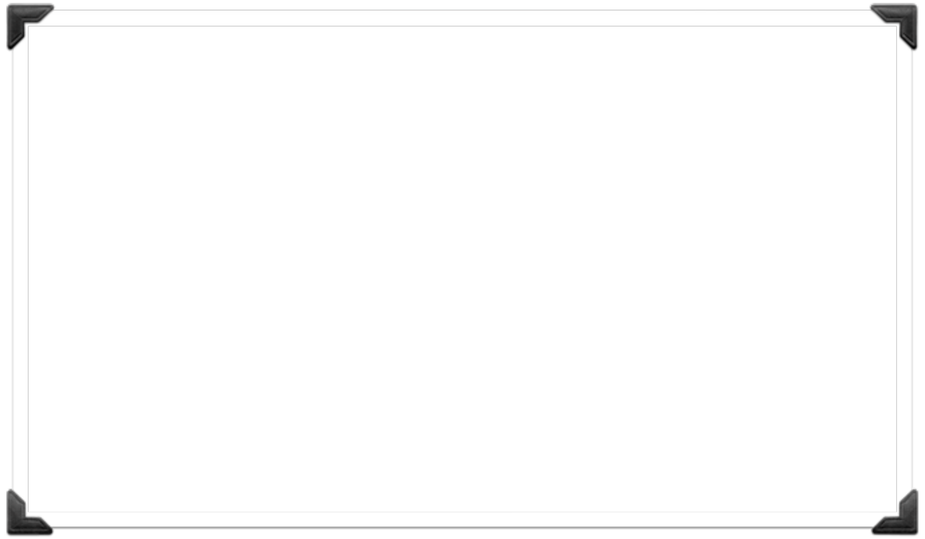
Warning!

When using terminal commands that are defined in a Gem file, like rspec, you might need to use:

```
% bundle exec rspec spec/test_my_app.rb
```

Rather than:

```
% rspec spec/test_my_app.rb
```



Summary & Advice

Ruby software packages are distributed via “gems”

Use bundler to manage your gems

Bundler will use the “gem” command under-the-hood

Be sure to store Gemfile and Gemfile.lock in, for example, your application’s Git repository