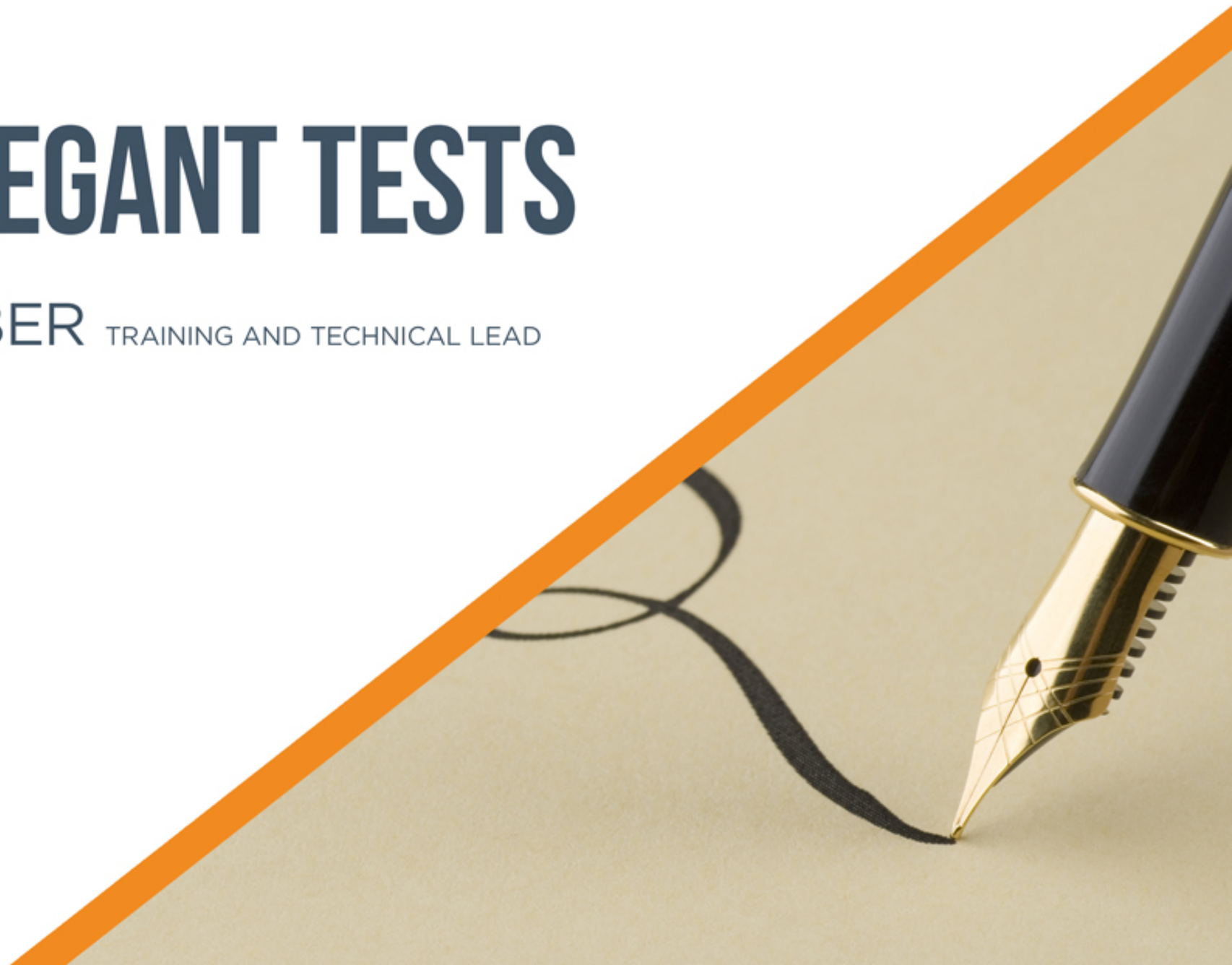
WEBINAR

WRITING ELEGANT TESTS

FRANKLIN WEBBER TRAINING AND TECHNICAL LEAD



CHEF™

Franklin Webber

Training and Technical Content Lead

- twitter.com/franklinwebber
- github.com/burtlo
- linkedin.com/in/fwebber

Ceci n'est pas un entraîneur

CHEF

Who I Think You Are

Chef Cookbook Author

- Author and maintain Chef Cookbooks with ChefSpec Tests
- Maintainer of a test suite that contains lots of duplication
- Interested in furthering your knowledge of Ruby's RSpec

chef.io/webinars/?commid=191263

chef.io/webinars/?commid=194483

OR

Training

training.chef.io

- Chef Essentials
- Test-Driven Cookbook Development

CHEF

What You Will Learn

The Focus

- **Reduce duplication in our expectations, examples, and scenario setup**
- Create re-useable test helpers
- Extract these testing resources into a Ruby gem

spec_helper.rb default_spec.rb

```
require 'spec_helper'

describe 'ark::default' do
  context 'when no attributes are specified, on an unspecified platform' do
    let(:chef_run) do
      runner = ChefSpec::SoloRunner.new
      runner.converge(described_recipe)
    end

    it 'installs necessary packages' do
      expect(chef_run).to install_package('libtool')
      expect(chef_run).to install_package('autoconf')
      expect(chef_run).to install_package('unzip')
      expect(chef_run).to install_package('rsync')
      expect(chef_run).to install_package('make')
      expect(chef_run).to install_package('gcc')
      expect(chef_run).to install_package('autogen')
    end

    it "does not install the gcc-c++ package" do
      expect(chef_run).not_to install_package("gcc-c++")
    end
```


CHEF

What You Will Learn

The Focus

- Reduce duplication in our expectations, examples, and scenario setup
- **Create re-useable test helpers**
- Extract these testing resources into a Ruby gem

CHEF

What You Will Learn

The Focus

- Reduce duplication in our expectations, examples, and scenario setup
- Create re-useable test helpers
- **Extract these testing resources into a Ruby gem**

CHEF

Schedule

- **Introduction (5 minutes)**
- **techniques (40 minutes)**

  Introduce 7 concepts with demonstration, and review
- **questions (12 minutes)**
- **wrap up (3 minutes)**

CHEF

Example Code Repository

```
> git clone https://github.com/chef-training/elegant_tests-repo.git
```

```
Cloning into 'elegant_tests-repo'...
remote: Counting objects: 67, done.
remote: Compressing objects: 100% (41/41), done.
remote: Total 67 (delta 12), reused 67 (delta 12), pack-reused 0
Unpacking objects: 100% (67/67), done.
```

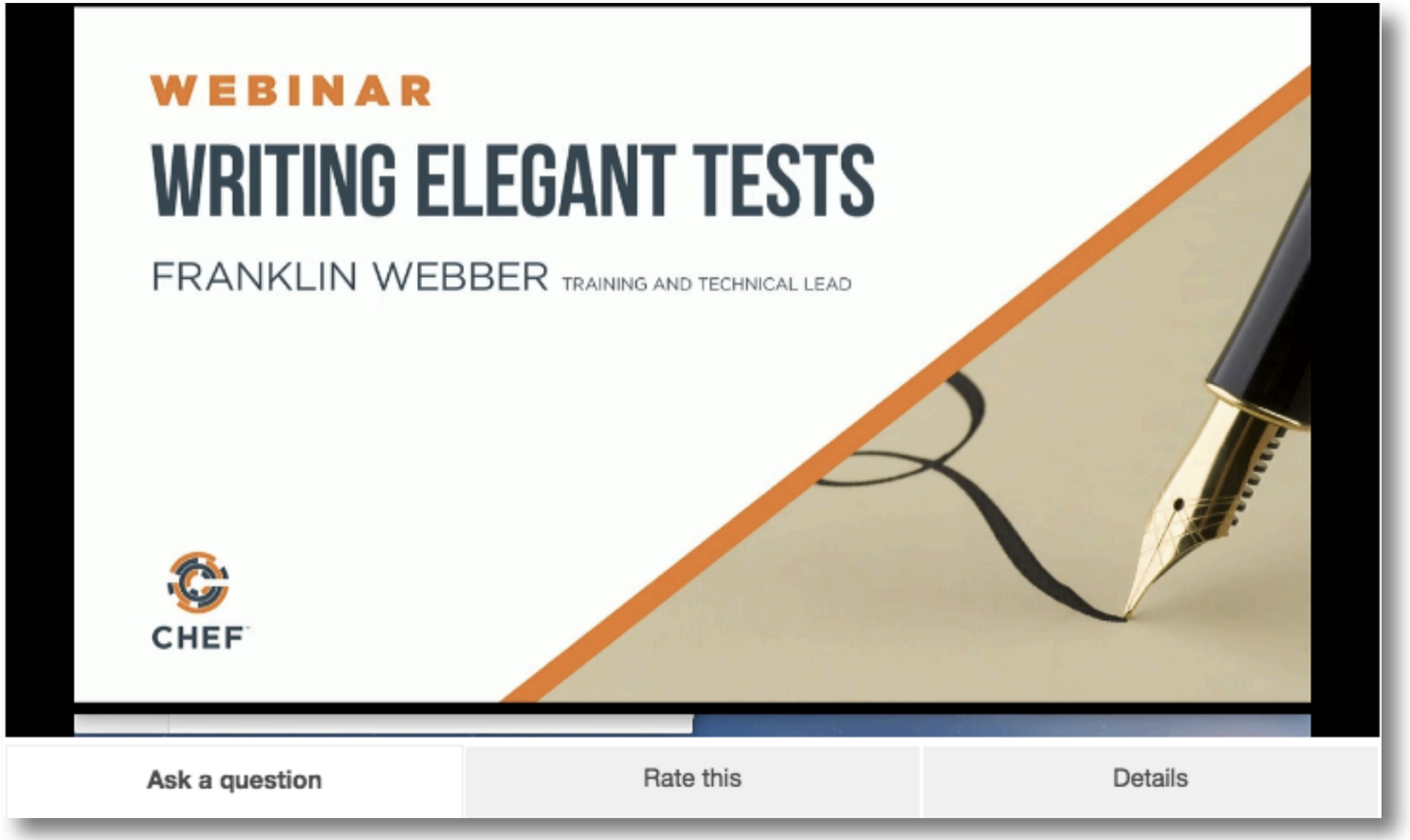

CHEF

Let Us Know What You Think

"Ask a Question" and I will answer it.

"Rate this" presentation to leave your feedback and help me do my work better.

To share, click on the appropriate professional/social network in **"Details"**

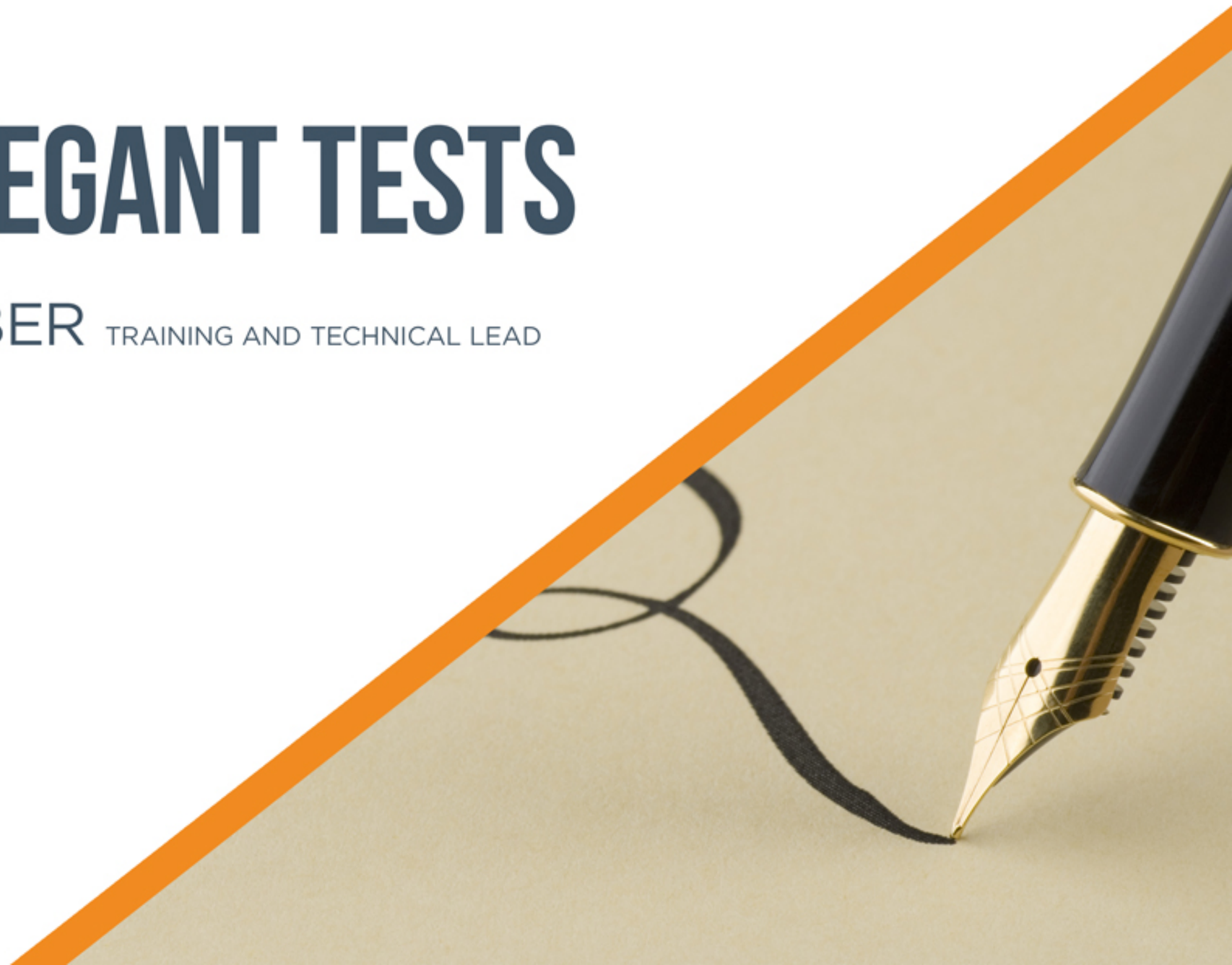


CHEF

WEBINAR

WRITING ELEGANT TESTS

FRANKLIN WEBBER TRAINING AND TECHNICAL LEAD



CHEF™

Agenda

TECHNIQUES

- ❑ let
- ❑ shared_examples
- ❑ def method
- ❑ shared_context
- ❑ require
- ❑ alias_example_group_to
- ❑ creating a Ruby gem
- ❑ questions
- ❑ wrap up

CHEF

let

"Since we cannot change reality, let us change the eyes which see reality."

~ Nikos Kazantzakis

- ❑ **Concepts**
- ❑ **Demonstration**
- ❑ **Review**

CHEF

CONCEPT

`let`

Use let to define a memoized helper method.

The value will be cached across multiple calls in the same example but not across examples. It is also lazy-evaluated: it is not evaluated until the first time the method it defines is invoked. See **`let!`** if you want to force the invocation before each example.

`https://goo.gl/BJp0IQ`

CHEF

Diagramming the let Helper Method

```
describe 'ark::default' do
  context when no attributes are ...
    let(:chef_run) do
      runner = ChefSpec::SoloRunner.new
      runner.converge(described_recipe)
      runner
    end

    it 'installs necessary packages' do
      expect(chef_run).to install_p...
      expect(chef_run).to install_p...
    end

    it "does not install the gcc-c+...
      expect(chef_run).not_to insta...
    end
```

1. let is a RSpec helper method
2. Ruby Symbol
3. Code Block
4. Invocation

CHEF

Invoking the Helper Method

```
describe 'ark::default' do
  context 'when no attributes are ...
    let(:chef_run) do
      runner = ChefSpec::SoloRunner.new
      runner.converge(described_recipe)
      runner
    end

    it 'installs necessary packages' do
      expect(chef_run).to install_p...
      expect(chef_run).to install_p...
    end

    it "does not install the gcc-c+...
      expect(chef_run).not_to insta...
    end
```

1. chef_run sends a message
2. RSpec invokes the contents of the block
3. RSpec stores the contents of the execution
4. chef_run sends a message
5. RSpec retrieves the stored execution

CHEF

Cached Within each Example

```
describe 'ark::default' do
  context 'when no attributes are ...
    let(:chef_run) do
      runner = ChefSpec::SoloRunner.new
      runner.converge(described_recipe)
      runner
    end

    it 'installs necessary packages' do
      expect(chef_run).to install_p...
      expect(chef_run).to install_p...
    end

    it "does not install the gcc-c+...
      expect(chef_run).not_to insta...
    end
```

1. chef_run is loaded and stored
2. chef_run uses the stored invocation
3. chef_run is loaded and stored

CHEF

A chef_run with Node Attributes

```
describe 'ark::default' do
  context 'when no attributes are specified, on CentOS' do
    let(:chef_run) do
      runner = ChefSpec::SoloRunner.new({ platform: 'centos',
                                          version: '6.7' })
      runner.converge(described_recipe)
    end

    # ... EXAMPLES WITHIN CONTEXT
  end
end
```


CHEF

`let`

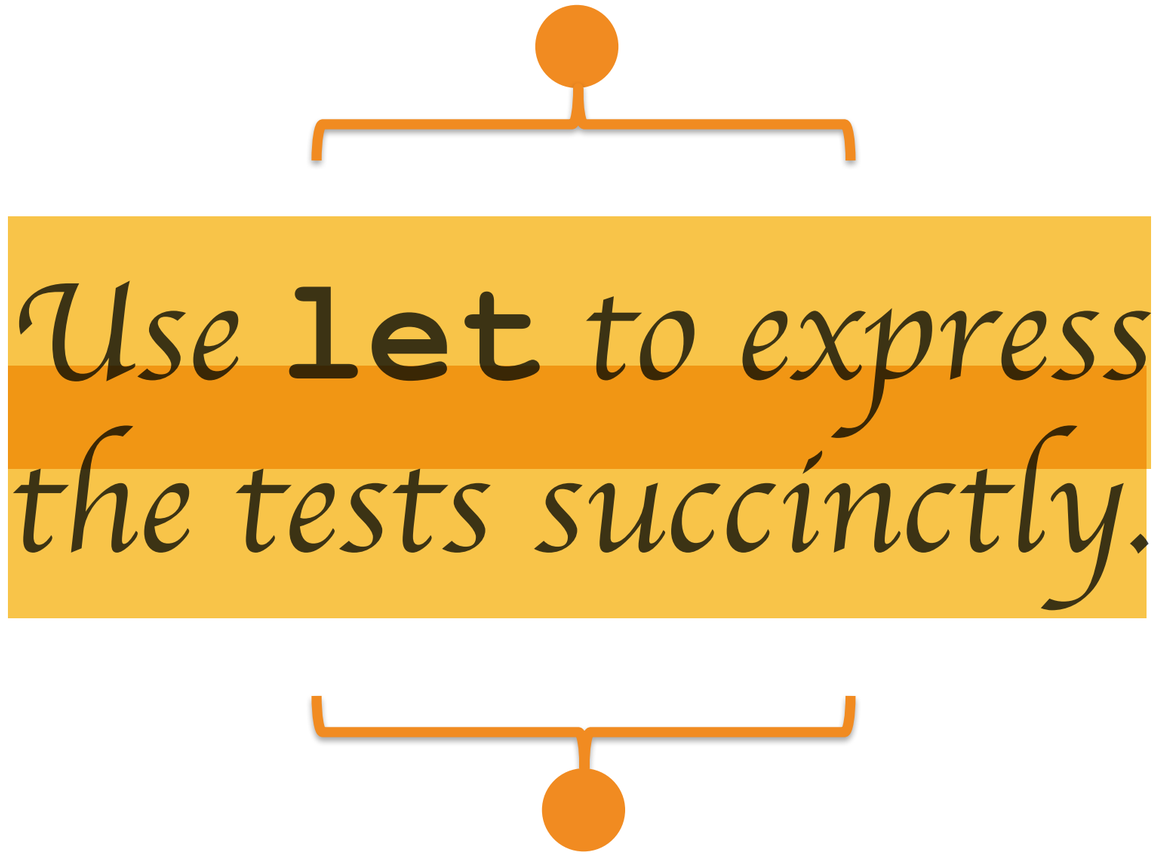
"If the path be beautiful, let us not ask where it leads."

~ Anatole France

- ✓ **Concepts**
- ❑ **Demonstration**
- ❑ **Review**

CHEF

Objective



CHEF

`let`

"Let us always meet each other with smile, for the smile is the beginning of love."
~ Mother Teresa

- ✓ **Concepts**
- ✓ **Demonstration**
- ❑ **Review**

CHEF

EXAMPLE

Example

https://goo.gl/ChkP47

CHEF™

CONCEPT

Using let for clarity

The use of the let to create the chef_run saves us from having the write the same code over-and-over again within each example.

We can define our own let helpers to increase the readability of our test code. Extracting important details and giving them a name.

https://goo.gl/BJp0IQ

CHEF

A chef_run with Node Attributes

```
describe 'ark::default' do
  context 'when no attributes are specified, on CentOS' do
    let(:chef_run) do
      runner = ChefSpec::SoloRunner.new({ platform: 'centos',
                                          version: '6.7' })
      runner.converge(described_recipe)
    end

    # ... EXAMPLES WITHIN CONTEXT
  end
end
```


CHEF

Using let to Create Clearer Examples

```
describe 'ark::default' do
  context 'when no attributes are specified, on CentOS' do
    let(:chef_run) do
      runner = ChefSpec::SoloRunner.new(node_attributes)
      runner.converge(described_recipe)
    end

    let(:node_attributes) do
      { platform: 'centos', version: '6.7' }
    end

    # ... EXAMPLES WITHIN CONTEXT
  end
```


CHEF

`let`

"Let us not pray to be sheltered from dangers but to be fearless when facing them."

~ Rabindranath Tagore

- ✓ **Concepts**
- ✓ **Demonstration**
- ✓ **Review**

CHEF

Agenda

TECHNIQUES

- ✓ let
- ❑ shared_examples
- ❑ def method
- ❑ shared_context
- ❑ require
- ❑ alias_example_group_to
- ❑ creating a Ruby gem
- ❑ questions
- ❑ wrap up

CHEF

shared_examples

- ❑ Concepts
- ❑ Demonstration
- ❑ Review

CHEF

CONCEPT

shared_examples

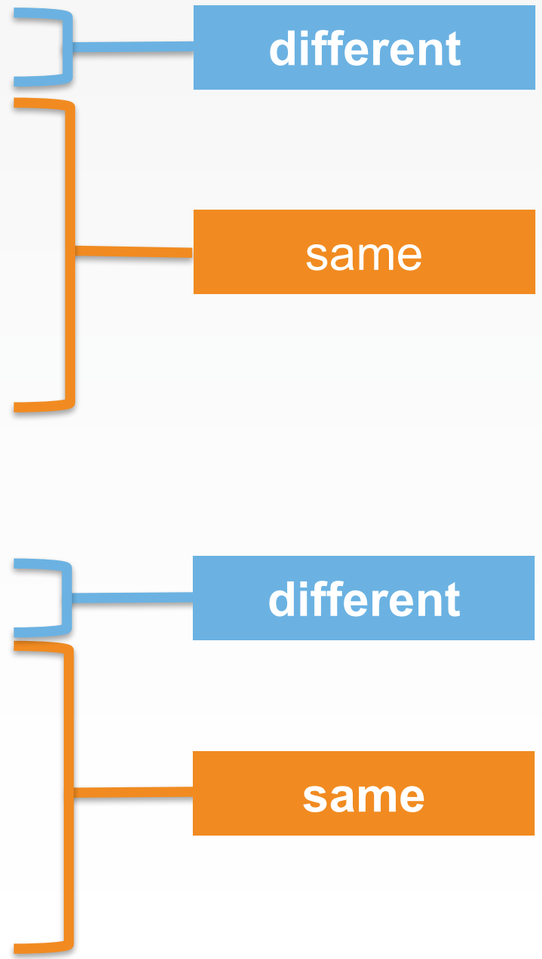
Shared examples let you describe behavior of classes or modules. When declared, a shared group's content is stored. It is only realized in the context of another example group, which provides any context the shared group needs to run.

https://goo.gl/yi12tM

CHEF

Finding Similar Expressed Expectations

```
context 'when no attributes are specified, on an unspecif...orm' do
  let(:installed_packages) ...
  it 'installs the necessary packages' do
    installed_packages.each do |name|
      expect(chef_run).to install_package(name)
    end
  end
end
context 'when no attributes are specified, on CentOS' do
  let(:installed_packages) ...
  it 'installs the necessary packages' do
    installed_packages.each do |name|
      expect(chef_run).to install_package(name)
    end
  end
end
```



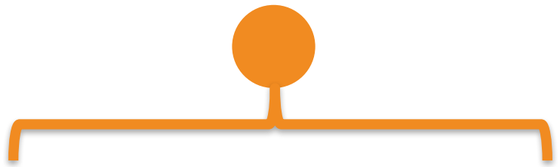
CHEF

shared_examples

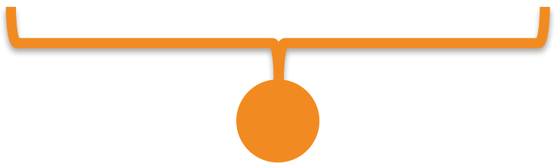
- ✓ Concepts
- ❑ Demonstration
- ❑ Review

CHEF

Objective



Use **`shared_examples`** *to express similarities.*



CHEF

shared_examples

- ✓ Concepts
- ✓ Demonstration
- ❑ Review

CHEF

EXAMPLE

Example

https://goo.gl/ChkP47

CHEF

CONCEPT

shared_examples

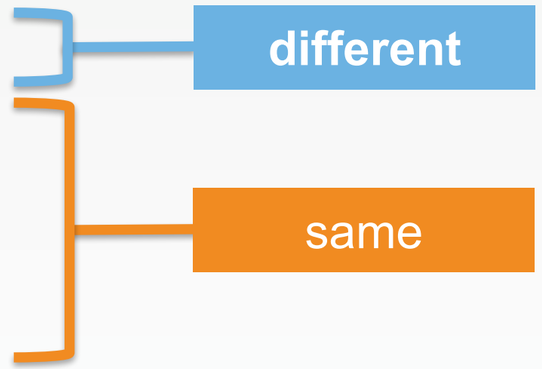
Shared examples let you describe behavior of classes or modules. When declared, a shared group's content is stored. It is only realized in the context of another example group, which provides any context the shared group needs to run.

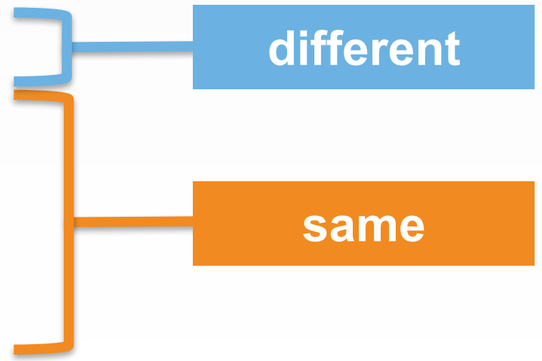
https://goo.gl/yi12tM

CHEF

Finding Similar Expressed Expectations

```
context 'when no attributes are specified, on an unspecif...orm' do
  let(:installed_packages) ...
  it 'installs the necessary packages' do
    installed_packages.each do |name|
      expect(chef_run).to install_package(name)
    end
  end
end
context 'when no attributes are specified, on CentOS' do
  let(:installed_packages) ...
  it 'installs the necessary packages' do
    installed_packages.each do |name|
      expect(chef_run).to install_package(name)
    end
  end
end
```





CHEF

A Place to Share Examples

```
shared_examples 'installs packages' do
  it 'installs the necessary packages' do
    installed_packages.each do |name|
      expect(chef_run).to install_package(name)
    end
  end
end
context 'when no attributes are specified, on an unspecif...orm' do
  let(:installed_packages) ...
  it_behaves_like 'installs packages'
end
context 'when no attributes are specified, on CentOS' do
  let(:installed_packages) ...
  it_behaves_like 'installs packages'
end
```

shared_examples

- ✓ Concepts
- ✓ Demonstration
- ✓ Review

CHEF

Agenda

TECHNIQUES

- ✓ let
- ✓ shared_examples
- ❑ def method
- ❑ shared_context
- ❑ require
- ❑ alias_example_group_to
- ❑ creating a Ruby gem
- ❑ questions
- ❑ wrap up

CHEF

def method

- ❑ Concepts
- ❑ Demonstration
- ❑ Review

CHEF

CONCEPT

Create helper method

You can define a Ruby method that takes care of some of the tedious work of retrieving node attributes.

CHEF™

CONCEPT

A Ruby Method with One Parameter

```
def file(name)
  # contents of method
  # last line automatically returns the value
end
```

The method named 'file' has a single parameter named 'name'.

CHEF

def method

- ✓ Concepts
- ❑ Demonstration
- ❑ Review

CHEF

Objective

CHEF

def method

- ✓ Concepts
- ✓ Demonstration
- ❑ Review

CHEF

EXAMPLE

Example

https://goo.gl/9mRNlD

(lowercase L)

CHEF

CONCEPT

Create helper method

You can define a Ruby method that takes care of some of the tedious work of retrieving node attributes.

CHEF

CONCEPT

A Ruby Method with One Parameter

```
def file(name)
  # contents of method
  # last line automatically returns the value
end
```

The method named 'file' has a single parameter named 'name'.

CHEF

Viewing the Repetition of Retrieving Attributes

~/ark/spec/unit/recipes/default_spec.rb

```
it "apache mirror" do
  attribute = chef_run.node['ark']['apache_mirror']
  expect(attribute).to eq "http://apache.mirrors.tds.net"
end

it "prefix root" do
  attribute = chef_run.node['ark']['prefix_root']
  expect(attribute).to eq "/usr/local"
end
```

Refactoring with let to help ease the pain

~/ark/spec/unit/recipes/default_spec.rb

```
    let(:node) do
      chef_run.node
    end

    it "apache mirror" do
      attribute = node['ark']['apache_mirror']
      expect(attribute).to eq "http://apache.mirrors.tds.net"
    end

    it "prefix root" do
      attribute = node['ark']['prefix_root']
      expect(attribute).to eq "/usr/local"
    end
```

Implementing a Helper Method

~/ark/spec/unit/recipes/default_spec.rb

```
let(:node) do
  chef_run.node
end

def attribute(name)
  node[described_cookbook][name]
end

it "apache mirror" do
  expect(attribute('apache_mirror')).to eq "http://apache.mirr....net"
end

it "prefix root" do
  expect(attribute('prefix_root')).to eq "/usr/local"
end
```

def method

- ✓ Concepts
- ✓ Demonstration
- ✓ Review

CHEF

Agenda

TECHNIQUES

- ✓ let
- ✓ shared_examples
- ✓ def method
- ❑ shared_context
- ❑ require
- ❑ alias_example_group_to
- ❑ creating a Ruby gem
- ❑ questions
- ❑ wrap up

CHEF

shared_context

- ❑ Concepts
- ❑ Demonstration
- ❑ Review

CHEF

CONCEPT

shared_context

Use shared_context to define a block that will be evaluated in the context of example groups either explicitly, using include_context, or implicitly by matching metdata.

http://goo.gl/R0ujTA

CHEF

shared_context

- ✓ Concepts
- ❑ Demonstration
- ❑ Review

CHEF

Objective

Use **`shared_context`**
to save all that you wrote.

CHEF

shared_context

- ✓ Concepts
- ✓ Demonstration
- ❑ Review

CHEF

EXAMPLE

Example

https://goo.gl/9mRNlD

(lowercase L)

CHEF

CONCEPT

shared_context

Use shared_context to define a block that will be evaluated in the context of example groups either explicitly, using include_context, or implicitly by matching metdata.

http://goo.gl/R0ujTA

CHEF

Repeating More than Examples

~/ark/spec/unit/recipes/default_spec.rb

```
describe 'ark::default' do
  context 'when no attributes are specified, on an uns...platform' do
    let(:chef_run) do
      runner = ChefSpec::SoloRunner.new(node_attributes)
      runner.converge(described_recipe)
    end

    let(:node_attributes) do
      {}
    end

    def attribute(name) ...
```


CHEF

Repeating More than Examples

~/ark/spec/unit/recipes/default_spec.rb

```
shared_context 'converged recipe' do
 let(:chef_run) do
   runner = ChefSpec::SoloRunner.new(node_attributes)
   runner.converge(described_recipe)
 end

  let(:node_attributes) do
    {}
  end

  def attribute(name) ...
end

describe 'ark::default' do
  context 'when no attributes are specified, on an uns...platform' do
    include_context 'converged recipe'
```


CHEF

shared_context

- ✓ Concepts
- ✓ Demonstration
- ✓ Review

CHEF

Agenda

TECHNIQUES

- ✓ let
- ✓ shared_examples
- ✓ def method
- ✓ shared_context
- ❑ require
- ❑ alias_example_group_to
- ❑ creating a Ruby gem
- ❑ questions
- ❑ wrap up

CHEF

require

- ❑ Concepts
- ❑ Demonstration
- ❑ Review

CHEF

CONCEPT

require

Loads the given filename, returning true if successful and false if the feature is already loaded.

If the filename does not resolve to an absolute path, it will be searched for in the directories listed in $LOAD_PATH ($:).

http://goo.gl/cLKY37

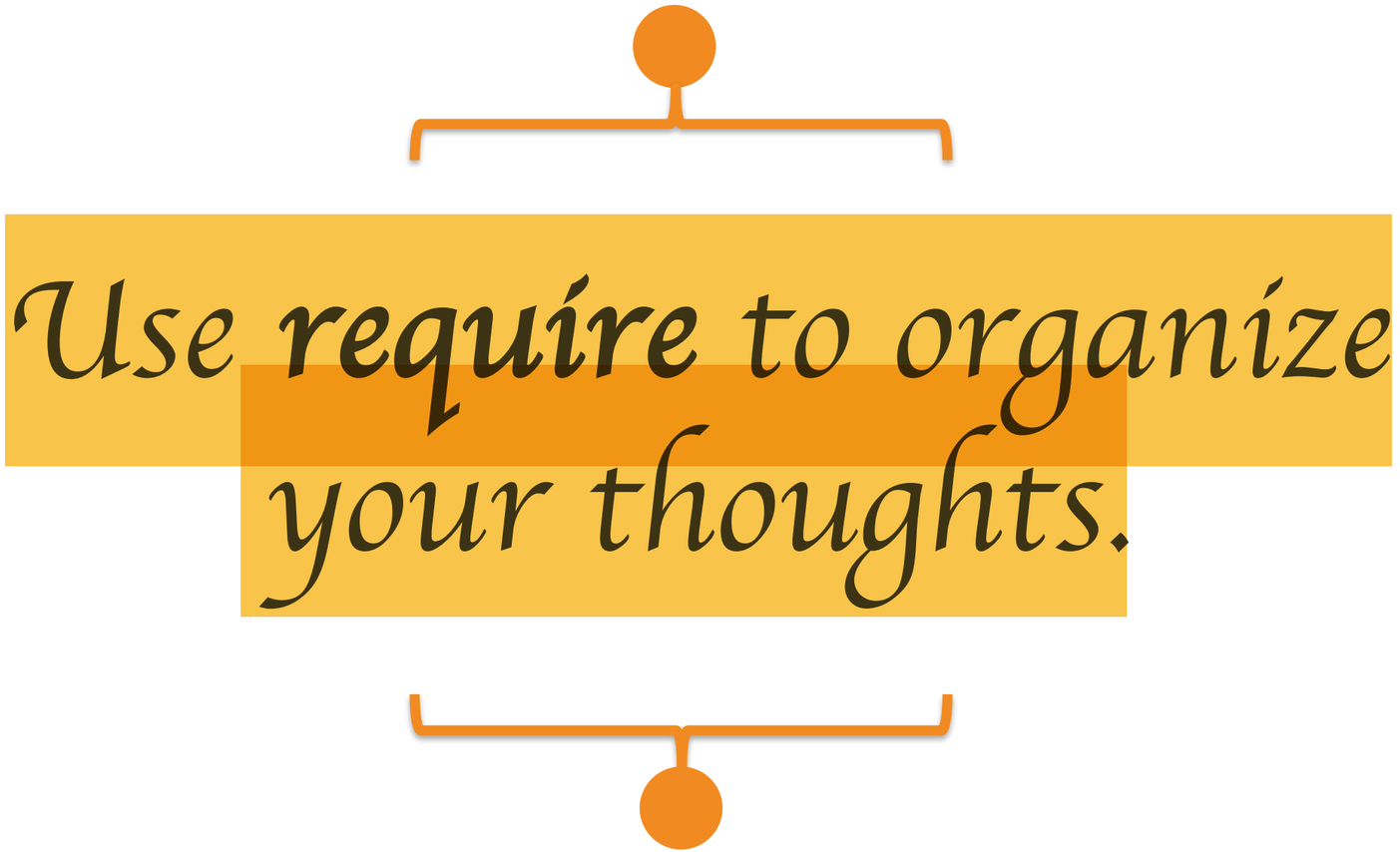
CHEF

require

- ✓ Concepts
- ❑ Demonstration
- ❑ Review

CHEF

Objective



CHEF

require

- ✓ Concepts
- ✓ Demonstration
- ❑ Review

CHEF

EXAMPLE

Example

https://goo.gl/9mRNlD

(lowercase L)

CHEF

CONCEPT

require

Loads the given filename, returning true if successful and false if the feature is already loaded.

If the filename does not resolve to an absolute path, it will be searched for in the directories listed in $LOAD_PATH ($:).

http://goo.gl/cLKY37

CHEF

Requiring the spec_helper file

~/ark/spec/unit/recipes/default_spec.rb

```
require 'spec_helper'

describe 'ark::default' do
  context 'when no attributes are specified, on an ...platform' do
    let(:chef_run) do
      runner = ChefSpec::SoloRunner.new
      runner.converge(described_recipe)
    end
```


CHEF

Viewing the spec_helper file

~/ark/spec/spec_helper.rb

```
require 'chefspec'
require 'chefspec/berkshelf'

at_exit { ChefSpec::Coverage.report! }

RSpec.configure do |config|
  config.color = true
end

# puts $LOAD_PATH
# ... define test helpers and content in this file
```


CHEF

Viewing the $LOAD_PATH

```
> chef exec rspec
```

```
/opt/chefdk/embedded/lib/ruby/gems/2.1.0/gems/uuidtools-2.1.5/lib
/Users/franklinwebber/ark/spec
/Users/franklinwebber/.chefdk/gem/ruby/2.1.0/gems/rspec-support-3.4.1/lib
/Users/franklinwebber/.chefdk/gem/ruby/2.1.0/gems/rspec-core-3.4.4/lib
/opt/chefdk/embedded/lib/ruby/gems/2.1.0/gems/net-ssh-3.1.1/lib
/opt/chefdk/embedded/lib/ruby/gems/2.1.0/gems/fauxhai-3.5.0/lib
/opt/chefdk/embedded/lib/ruby/gems/2.1.0/gems/diff-lcs-1.2.5/lib
/Users/franklinwebber/.chefdk/gem/ruby/2.1.0/gems/rspec-expectations-3.4.0/lib
/Users/franklinwebber/.chefdk/gem/ruby/2.1.0/gems/rspec-mocks-3.4.1/lib
/Users/franklinwebber/.chefdk/gem/ruby/2.1.0/gems/rspec-3.4.0/lib
```


CHEF

require

- ✓ Concepts
- ✓ Demonstration
- ✓ Review

CHEF

Agenda

TECHNIQUES

- ✓ let
- ✓ shared_examples
- ✓ def method
- ✓ shared_context
- ✓ require
- ❑ alias_example_group_to
- ❑ creating a Ruby gem
- ❑ questions
- ❑ wrap up

CHEF

alias_example_group_to

- ❑ Concepts
- ❑ Demonstration
- ❑ Review

CHEF

CONCEPT

alias_example_group_to

describe and **context** are the default aliases for **example_group**. You can define your own aliases for **example_group** and give those custom aliases default metadata.

http://goo.gl/DfUChL

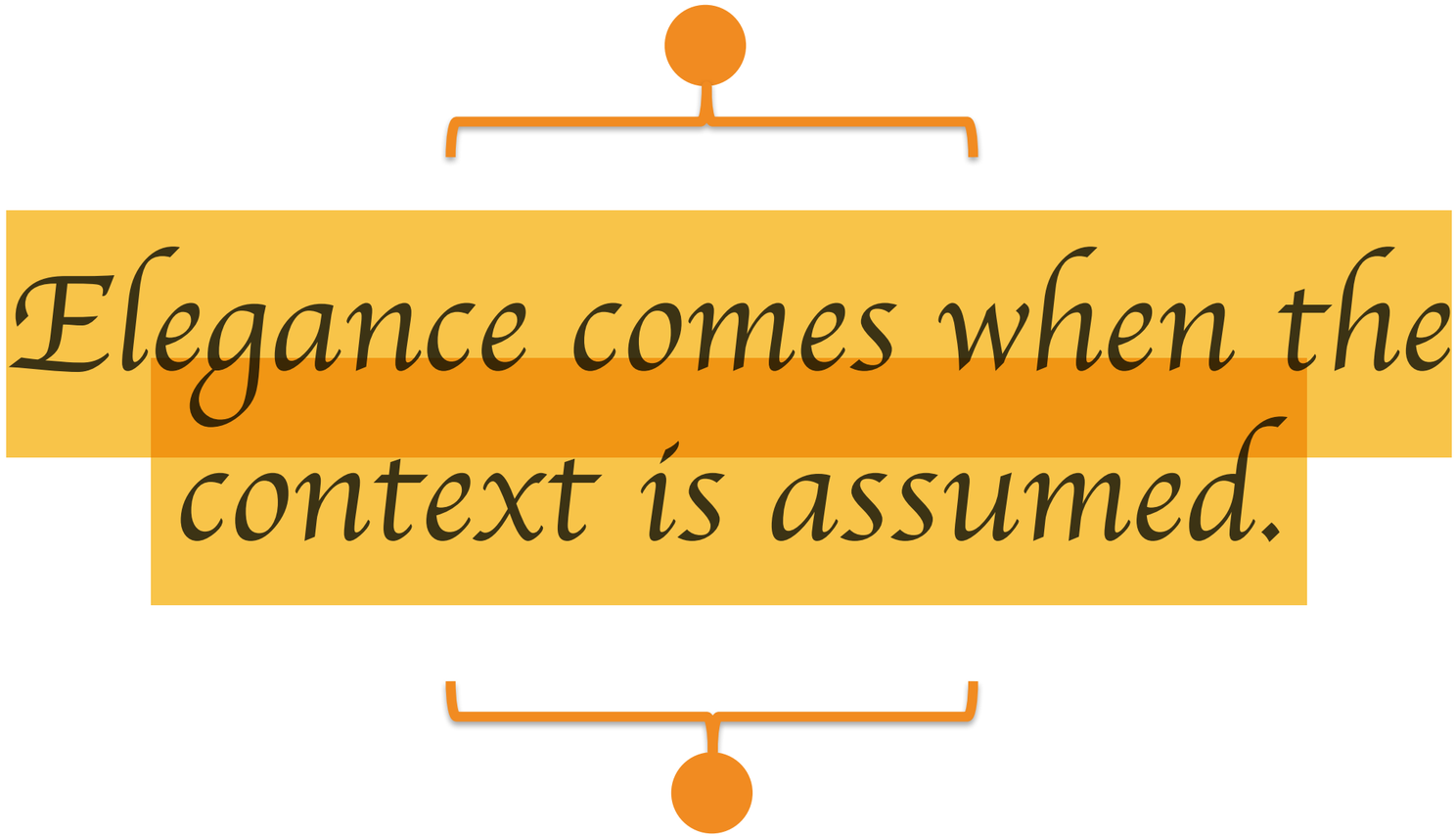
CHEF

alias_example_group_to

- ✓ Concepts
- ❑ Demonstration
- ❑ Review

CHEF

Objective



CHEF

EXAMPLE

Live Demonstration

https://goo.gl/9mRNlD

(lowercase L)

CHEF

alias_example_group_to

- ✓ Concepts
- ✓ Demonstration
- ❑ Review

CHEF

CONCEPT

alias_example_group_to

describe and **context** are the default aliases for **example_group**. You can define your own aliases for **example_group** and give those custom aliases default metadata.

http://goo.gl/DfUChL

CHEF

Viewing a Spec with include_context

~/ark/spec/unit/recipes/default_spec.rb

```
describe 'ark::default' do
  context 'when no attributes are specified, on an ...platform' do
    include_context 'converged recipe'

    # ... examples ...

  end

  # ... other contexts ...
end
```


CHEF

Refactoring the Spec to auto include the context

~/ark/spec/unit/recipes/default_spec.rb

```
describe_recipe 'ark::default' do
  context 'when no attributes are specified, on an uns...platform' do
    include_context 'converged recipe'

    # ... examples ...

  end

  # ... other contexts
end
```


CHEF

Defining the alias and tying it to the shared_context

~/ark/spec/spec_helper.rb

```
require 'chefspec'
require 'chefspec/berkshelf'

at_exit { ChefSpec::Coverage.report! }

RSpec.configure do |config|
  config.color = true
  config.alias_example_group_to :describe_recipe, :type => :recipe
end

shared_context 'converged recipe', :type => :recipe do
```


CHEF

alias_example_group_to

- ✓ Concepts
- ✓ Demonstration
- ✓ Review

CHEF

Agenda

TECHNIQUES

- ✓ let
- ✓ shared_examples
- ✓ def method
- ✓ shared_context
- ✓ require
- ✓ alias_example_group_to
- ❑ creating a Ruby gem
- ❑ questions
- ❑ wrap up

CHEF

creating a Ruby gem

- ❑ Concepts
- ❑ Demonstration

CHEF™

CONCEPT

RubyGems

RubyGems is a package manager for the Ruby programming language that provides a standard format for distributing Ruby programs and libraries (in a self-contained format called a "gem"), a tool designed to easily manage the installation of gems, and a server for distributing them.

http://guides.rubygems.org/rubygems-basics

CHEF

CONCEPT

gem

```
> chef gem --help
```

The command allows you build gems for distribution, install gems locally, and push gems to Gem server.

http://guides.rubygems.org/command-reference

CHEF

CONCEPT

Bundler

Bundler provides a consistent environment for Ruby projects by tracking and installing the exact gems and versions that are needed.

Bundler is an exit from dependency hell, and ensures that the gems you need are present in development, staging, and production. Starting work on a project is as simple as bundle install.

http://bundler.io

CHEF

CONCEPT

bundle

```
> chef exec bundle help
```

The command allows you to install and update a project's dependencies. It will also allow you to generate a cookbook.

CHEF

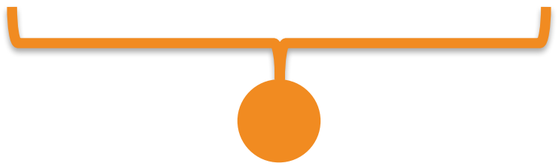
creating a Ruby gem

- ✓ Concepts
- ❑ Demonstration
- ❑ Review

CHEF

Objective

Create Treasure!



CHEF™

creating a Ruby gem

- ✓ Concepts
- ✓ Demonstration
- ❑ Review

CHEF

EXAMPLE

Example

`github.com/burtlo/chefspec-ohai`

CHEF

Generating the Gem

```
> chef exec bundle gem chefspec-myhelpers
```

```
Creating gem 'chefspec-myhelpers'...
      create  chefspec-myhelpers/Gemfile
      create  chefspec-myhelpers/.gitignore
      create  chefspec-myhelpers/lib/chefspec/myhelpers.rb
      create  chefspec-myhelpers/lib/chefspec/myhelpers/version.rb
      create  chefspec-myhelpers/chefspec-myhelpers.gemspec
      create  chefspec-myhelpers/Rakefile
      create  chefspec-myhelpers/README.md
      create  chefspec-myhelpers/bin/console
      create  chefspec-myhelpers/bin/setup
```

Updating the Gem Specification

~/chefspec-myhelpers/chefspec-myhelpers.gemspec

```
Gem::Specification.new do |spec|
  spec.name          = "chefspec-myhelpers"
  spec.version       = Chefspec::Myhelpers::VERSION
  spec.authors       = ["Franklin Webber"]
  spec.email         = ["franklin.webber@gmail.com"]

  spec.summary       = %q{Provides common ChefSpec Helpers.}
  spec.description   = %q{ChefSpec Helpers that define an alia...}
  spec.homepage      = "https://github.com/burtlo/chefspec...com"
  spec.license       = "MIT"
```


CHEF

Moving the Shared Context to the Gem

~/chefspec-myhelpers/lib/chefspec/myhelpers.rb

```
RSpec.configure do |config|
  config.alias_example_group_to :describe_recipe, :type => :recipe
end

shared_context 'converged recipe' do
  let(:chef_run) do
    runner = ChefSpec::SoloRunner.new(node_attributes)
    runner.converge(described_recipe)
  end

  let(:node_attributes) do
    {}
  end
```


CHEF

Changing into the Gem's Directory

```
> cd chefspec-myhelpers
```


CHEF

Creating the Gem

```
> chef gem build chefspec-myhelpers.gemspec
```

```
Successfully built RubyGem
 Name: chefspec-myhelpers
 Version: 0.1.0
 File: chefspec-myhelpers-0.1.0.gem
```


CHEF

Installing the Gem

```
> chef gem install chefspec-myhelpers-0.1.0.gem
```

```
 install chefspec-myhelpers-0.1.0.gem
Successfully installed chefspec-myhelpers-0.1.0
1 gem installed
```


CHEF

Listing the installed Gem

```
> chef gem list chefspec
```

```
*** LOCAL GEMS ***


chefspec (5.3.0)
chefspec-ohai (0.1.0)
chefspec-myhelpers (0.1.0)
```

Updating the Cookbook's spec_helper

~/ark/spec/spec_helper.rb

```
require 'chefspec'
require 'chefspec/berkshelf'
require 'chefspec/myhelpers'                                    +

at_exit { ChefSpec::Coverage.report! }

RSpec.configure do |config|
  config.color = true
  config.alias_example_group_to :describe_recipe, :type => :recipe-
end

shared_context 'converged recipe', :type => :recipe do          -
```


CHEF

creating a Ruby gem

- ✓ Concepts
- ✓ Demonstration
- ✓ Review

CHEF

Agenda

TECHNIQUES

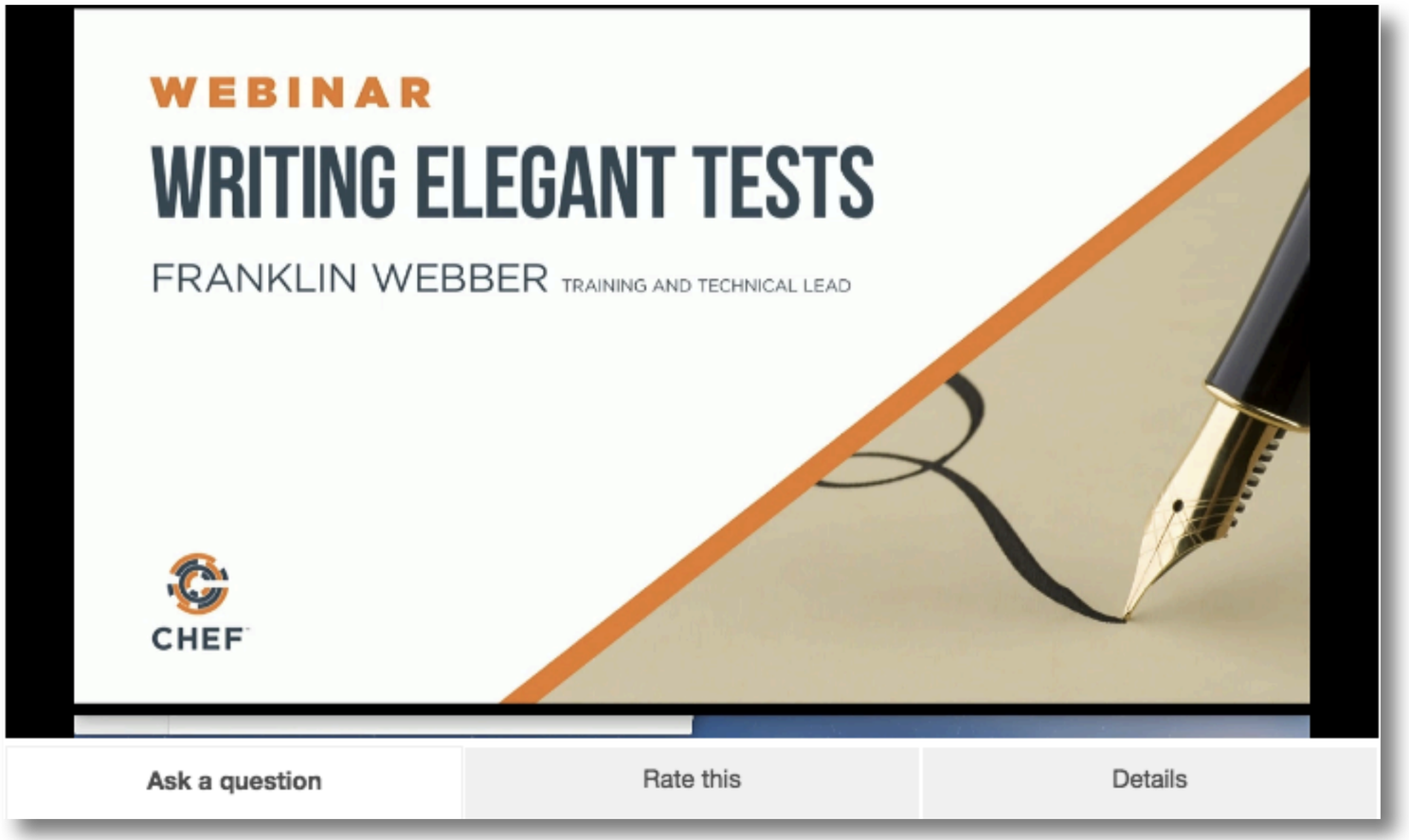
- ✓ let
- ✓ shared_examples
- ✓ def method
- ✓ shared_context
- ✓ require
- ✓ alias_example_group_to
- ✓ creating a Ruby gem
- ❑ questions
- ❑ wrap up

CHEF

Let Us Know What You Think

Email me at **franklin@chef.io**

"Rate This" presentation to leave your feedback and help me do my work better.



CHEF

DISCUSSION

Q&A

What questions can I answer for you?

CHEF

Agenda

TECHNIQUES

- ✓ let
- ✓ shared_examples
- ✓ def method
- ✓ shared_context
- ✓ require
- ✓ alias_example_group_to
- ✓ creating a Ruby gem
- ✓ questions
- ❑ wrap up

CHEF

Check The Attachments

All of the resources I am going to provide you can be found under **"Attachments"**.

Presentation Resources

Slides

https://github.com/chef-training/webinars/raw/master/writing_elegant_tests.pdf

Example Cookbook

https://github.com/chef-training/elegant_tests-repo

CHEF

RSpec Resources

let

https://relishapp.com/rspec/rspec-core/v/3-4/docs/helper-methods/let-and-let

shared_examples

https://relishapp.com/rspec/rspec-core/v/3-4/docs/example-groups/shared-examples

shared_context

https://relishapp.com/rspec/rspec-core/v/2-6/docs/example-groups/shared-context

alias_example_group_to

http://www.rubydoc.info/github/rspec/rspec-core/RSpec%2FCore%2FConfiguration%3Aalias_example_group_to

Ruby Resources

Ruby's require

http://www.rubydoc.info/stdlib/core/Kernel%3Arequire

Ruby Gems

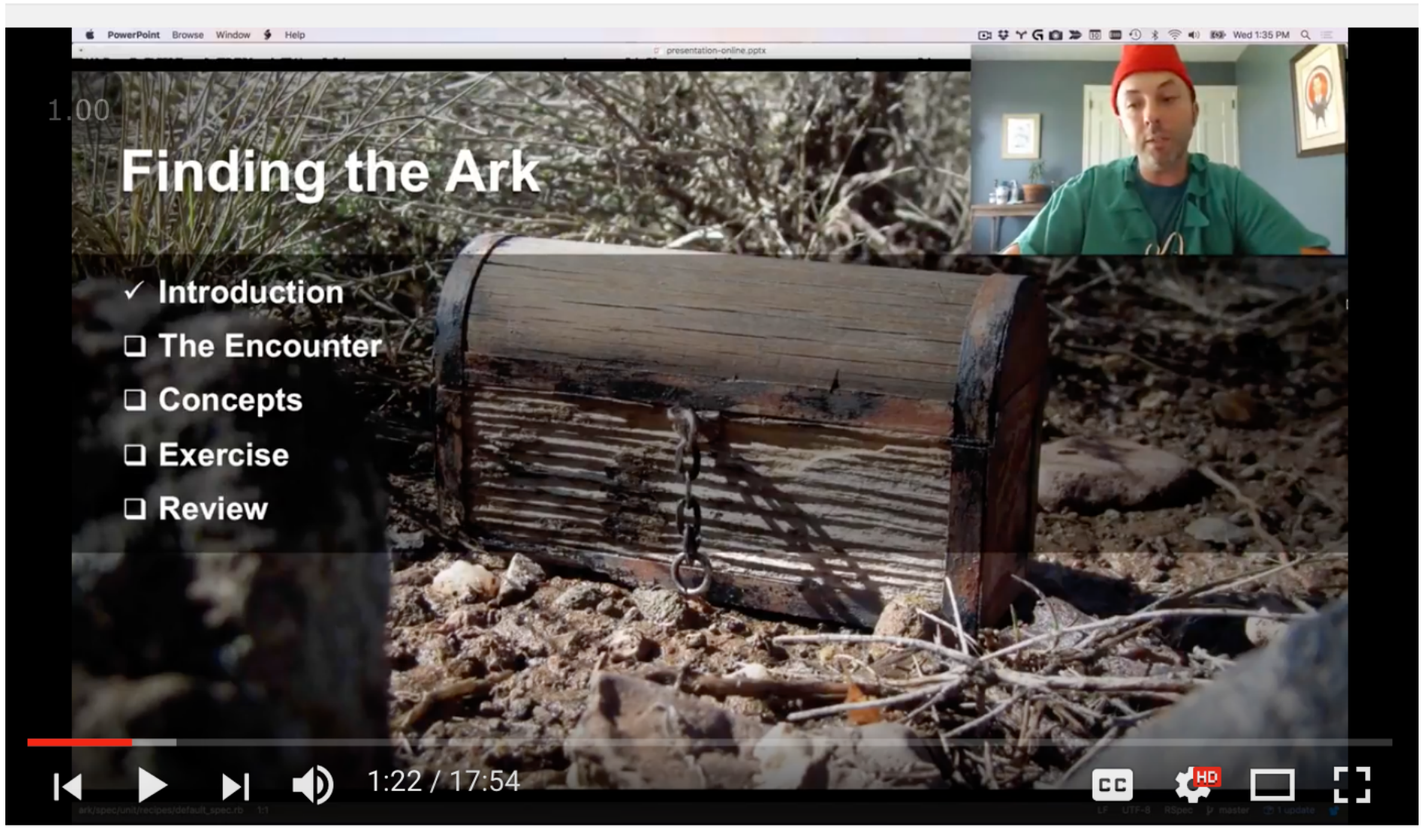
http://guides.rubygems.org/rubygems-basics/

CHEF

Advanced Dungeons and Testing Dragons

This adventure will test your mettle as we delve through these dark corners of cookbook development.

https://goo.gl/Qo2MfV



CHEF

Agenda

TECHNIQUES

- ✓ let
- ✓ shared_examples
- ✓ def method
- ✓ shared_context
- ✓ require
- ✓ alias_example_group_to
- ✓ creating a Ruby gem
- ✓ questions
- ✓ wrap up

CHEF

CHEF™