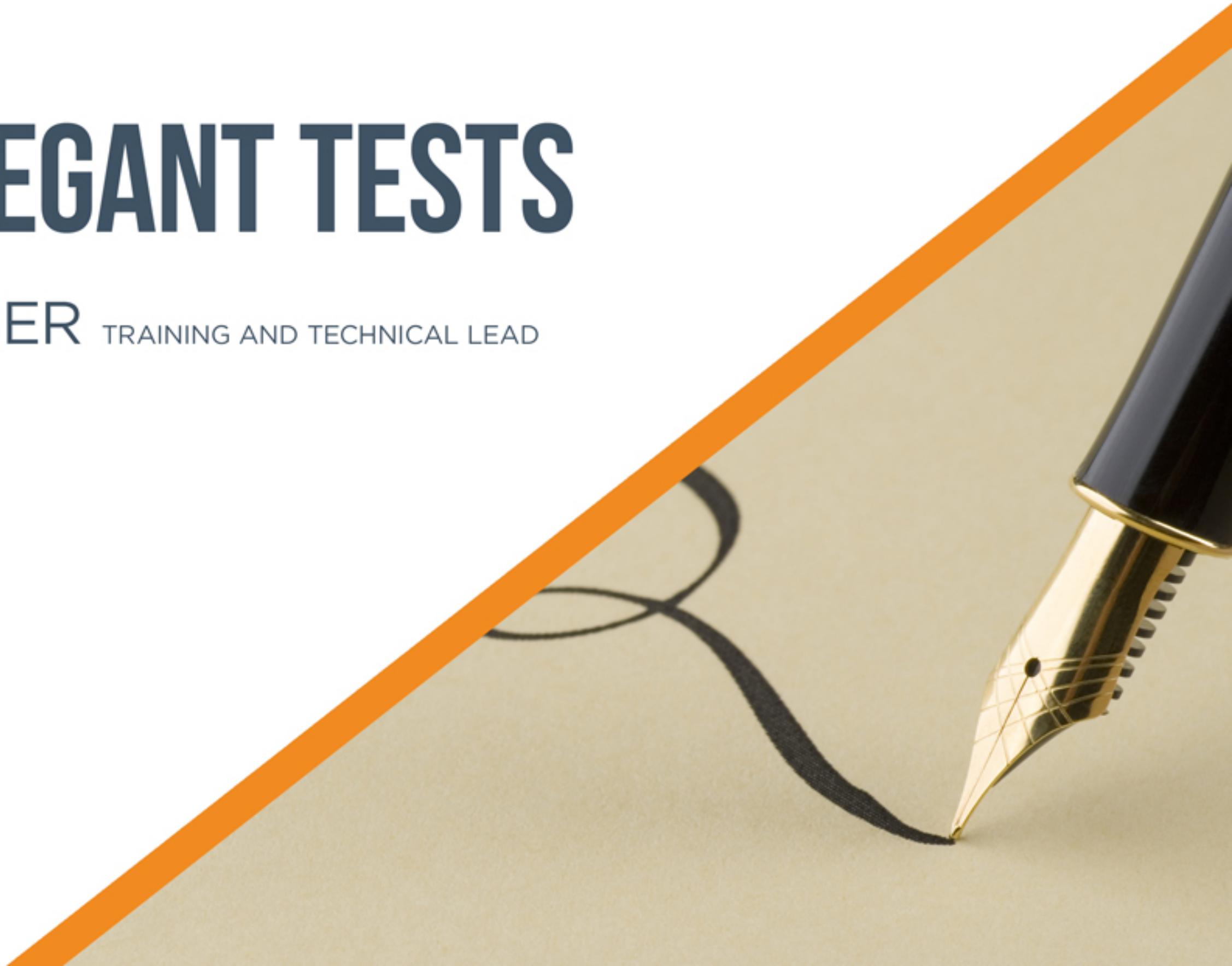


WEBINAR

WRITING ELEGANT TESTS

FRANKLIN WEBBER TRAINING AND TECHNICAL LEAD



Franklin Webber

Training and Technical Content Lead

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



Ceci n'est pas un entraîneur

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



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
1 require 'spec_helper'
2
3 describe 'ark::default' do
4   context 'when no attributes are specified, on an unspecified platform' do
5     let(:chef_run) do
6       runner = ChefSpec::SoloRunner.new
7       runner.converge(described_recipe)
8     end
9
10    it 'installs necessary packages' do
11      expect(chef_run).to install_package('libtool')
12      expect(chef_run).to install_package('autoconf')
13      expect(chef_run).to install_package('unzip')
14      expect(chef_run).to install_package('rsync')
15      expect(chef_run).to install_package('make')
16      expect(chef_run).to install_package('gcc')
17      expect(chef_run).to install_package('autogen')
18    end
19
20    it "does not install the gcc-c++ package" do
21      expect(chef_run).not_to install_package("gcc-c++")
22    end

```

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
1 require 'spec_helper'-
2 -
3 describe_recipe 'ark::default' do
4   context 'when no attributes are specified, on an unspecified platform' do
5     let(:installed_packages) do
6       %w[ libtool autoconf unzip rsync make gcc autogen ]
7     end
8
9     it_behaves_like 'installs packages'
10
11    let(:packages_not_installed) do
12      %w[ gcc-c++ seven_zip ]
13    end
14
15    it_behaves_like 'does not install packages'
16
17    it "apache mirror" do
18      expect(attribute('apache_mirror')).to eq "http://apache.mirrors.tds.net"
19    end
20
21    it "prefix root" do
22      expect(attribute('prefix_root')).to eq "/usr/local"
```

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
1 require 'chefspec'      1
2 require 'chefspec/berkshelf'  2
3
4 at_exit { ChefSpec::Coverage.report! }  3
5
6 RSpec.configure do |config|  4
7   config.color = true  5
8   config.alias_example_group_to :describe_recipe, type: :recipe  6
9 end  7
10
11 shared_context 'converging recipe', type: :recipe do  8
12   let(:chef_run) do  9
13     runner = ChefSpec::SoloRunner.new(node_attributes)  10
14     runner.converge(described_recipe)  11
15   end  12
16
17   let(:node_attributes) do  13
18     {}  14
19   end  15
20
21   let(:node) do  16
22     chef_run.node  17
23   end  18
24
25   let(:runner) do  19
26     ChefSpec::SoloRunner.new(node_attributes)  20
27   end  21
28
29   let(:subject) do  22
30     described_recipe  23
31   end  24
32
33   let(:node_name) do  25
34     'test-kitchen'  26
35   end  27
36
37   let(:node_status) do  28
38     'ok'  29
39   end  30
40
41   let(:node_status_color) do  31
42     'green'  33
43   end  34
44
45   let(:node_status_message) do  35
46     'The run was successful'  37
47   end  38
48
49   let(:node_status_message_color) do  39
50     'green'  41
51   end  42
52
53   let(:node_status_message_type) do  43
54     'info'  45
55   end  46
56
57   let(:node_status_message_level) do  47
58     'INFO'  49
59   end  50
60
61   let(:node_status_message_level_color) do  51
62     'green'  53
63   end  54
64
65   let(:node_status_message_level_type) do  55
66     'INFO'  57
67   end  58
68
69   let(:node_status_message_level_type_color) do  59
70     'green'  61
71   end  62
72
73   let(:node_status_message_level_type_level) do  63
74     'INFO'  65
75   end  66
76
77   let(:node_status_message_level_type_level_color) do  67
78     'green'  69
79   end  70
80
81   let(:node_status_message_level_type_level_type) do  71
82     'INFO'  73
83   end  74
84
85   let(:node_status_message_level_type_level_type_color) do  75
86     'green'  77
87   end  78
88
89   let(:node_status_message_level_type_level_type_level) do  79
90     'INFO'  81
91   end  82
92
93   let(:node_status_message_level_type_level_type_level_color) do  83
94     'green'  85
95   end  86
96
97   let(:node_status_message_level_type_level_type_level_type) do  87
98     'INFO'  90
99   end  100
100
101   let(:node_status_message_level_type_level_type_level_type_color) do  101
102     'green'  103
103   end  104
104
105   let(:node_status_message_level_type_level_type_level_type_level) do  105
106     'INFO'  108
107   end  109
108
109   let(:node_status_message_level_type_level_type_level_type_level_color) do  111
110     'green'  112
111   end  113
112
113   let(:node_status_message_level_type_level_type_level_type_level_level) do  115
114     'INFO'  117
115   end  118
116
117   let(:node_status_message_level_type_level_type_level_type_level_level_color) do  119
118     'green'  120
119   end  121
120
121   let(:node_status_message_level_type_level_type_level_type_level_level_level) do  123
122     'INFO'  125
123   end  126
124
125   let(:node_status_message_level_type_level_type_level_type_level_level_level_color) do  127
126     'green'  128
127   end  129
128
129   let(:node_status_message_level_type_level_type_level_type_level_level_level_level) do  131
130     'INFO'  133
131   end  134
132
133   let(:node_status_message_level_type_level_type_level_type_level_level_level_level_color) do  135
134     'green'  136
135   end  137
136
137   let(:node_status_message_level_type_level_type_level_type_level_level_level_level_level_color) do  139
138     'green'  140
139   end  141
140
141   let(:node_status_message_level_type_level_type_level_type_level_level_level_level_level_color) do  143
142     'green'  144
143   end  145
144
145   let(:node_status_message_level_type_level_type_level_type_level_level_level_level_level_level_color) do  147
146     'green'  148
147   end  149
148
149   let(:node_status_message_level_type_level_type_level_type_level_level_level_level_level_level_color) do  151
150     'green'  152
151   end  153
152
153   let(:node_status_message_level_type_level_type_level_type_level_level_level_level_level_level_level_color) do  155
154     'green'  156
155   end  157
156
157   let(:node_status_message_level_type_level_type_level_type_level_level_level_level_level_level_level_color) do  159
158     'green'  160
159   end  161
160
161   let(:node_status_message_level_type_level_type_level_type_level_level_level_level_level_level_level_color) do  163
162     'green'  164
163   end  165
164
165   let(:node_status_message_level_type_level_type_level_type_level_level_level_level_level_level_level_color) do  167
166     'green'  168
167   end  169
168
169   let(:node_status_message_level_type_level_type_level_type_level_level_level_level_level_level_level_color) do  171
170     'green'  172
171   end  173
172
173   let(:node_status_message_level_type_level_type_level_type_level_level_level_level_level_level_level_color) do  175
174     'green'  176
175   end  177
176
177   let(:node_status_message_level_type_level_type_level_type_level_level_level_level_level_level_level_color) do  179
180     'green'  181
181   end  182
182
183   let(:node_status_message_level_type_level_type_level_type_level_level_level_level_level_level_level_color) do  185
186     'green'  187
187   end  188
188
189   let(:node_status_message_level_type_level_type_level_type_level_level_level_level_level_level_level_color) do  191
192     'green'  193
193   end  194
194
195   let(:node_status_message_level_type_level_type_level_type_level_level_level_level_level_level_level_color) do  197
198     'green'  199
199   end  200
200
201   let(:node_status_message_level_type_level_type_level_type_level_level_level_level_level_level_level_color) do  203
204     'green'  205
205   end  206
206
207   let(:node_status_message_level_type_level_type_level_type_level_level_level_level_level_level_level_color) do  209
210     'green'  211
211   end  212
212
213   let(:node_status_message_level_type_level_type_level_type_level_level_level_level_level_level_level_color) do  215
216     'green'  217
217   end  218
218
219   let(:node_status_message_level_type_level_type_level_type_level_level_level_level_level_level_level_color) do  221
222     'green'  223
223   end  224
224
225   let(:node_status_message_level_type_level_type_level_type_level_level_level_level_level_level_level_color) do  227
228     'green'  229
229   end  230
230
231   let(:node_status_message_level_type_level_type_level_type_level_level_level_level_level_level_level_color) do  233
234     'green'  235
235   end  236
236
237   let(:node_status_message_level_type_level_type_level_type_level_level_level_level_level_level_level_color) do  239
240     'green'  241
241   end  242
242
243   let(:node_status_message_level_type_level_type_level_type_level_level_level_level_level_level_level_color) do  245
246     'green'  247
247   end  248
248
249   let(:node_status_message_level_type_level_type_level_type_level_level_level_level_level_level_level_color) do  251
252     'green'  253
253   end  254
254
255   let(:node_status_message_level_type_level_type_level_type_level_level_level_level_level_level_level_color) do  257
258     'green'  259
259   end  260
260
261   let(:node_status_message_level_type_level_type_level_type_level_level_level_level_level_level_level_color) do  263
264     'green'  265
265   end  266
266
267   let(:node_status_message_level_type_level_type_level_type_level_level_level_level_level_level_level_color) do  269
270     'green'  271
271   end  272
272
273   let(:node_status_message_level_type_level_type_level_type_level_level_level_level_level_level_level_color) do  275
276     'green'  277
277   end  278
278
279   let(:node_status_message_level_type_level_type_level_type_level_level_level_level_level_level_level_color) do  281
282     'green'  283
283   end  284
284
285   let(:node_status_message_level_type_level_type_level_type_level_level_level_level_level_level_level_color) do  287
288     'green'  289
289   end  290
290
291   let(:node_status_message_level_type_level_type_level_type_level_level_level_level_level_level_level_color) do  293
294     'green'  295
295   end  296
296
297   let(:node_status_message_level_type_level_type_level_type_level_level_level_level_level_level_level_color) do  299
299   end  300
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
559
560
561
562
563
564
565
566
567
568
569
569
570
571
572
573
574
575
576
577
578
579
579
580
581
582
583
584
585
586
587
588
589
589
590
591
592
593
594
595
596
597
598
599
599
600
601
602
603
604
605
606
607
608
609
609
610
611
612
613
614
615
616
617
617
618
619
619
620
621
622
623
624
625
626
626
627
628
628
629
629
630
631
632
633
634
635
635
636
636
637
637
638
638
639
639
640
640
641
641
642
642
643
643
644
644
645
645
646
646
647
647
648
648
649
649
650
650
651
651
652
652
653
653
654
654
655
655
656
656
657
657
658
658
659
659
660
660
661
661
662
662
663
663
664
664
665
665
666
666
667
667
668
668
669
669
670
670
671
671
672
672
673
673
674
674
675
675
676
676
677
677
678
678
679
679
680
680
681
681
682
682
683
683
684
684
685
685
686
686
687
687
688
688
689
689
690
690
691
691
692
692
693
693
694
694
695
695
696
696
697
697
698
698
699
699
700
700
701
701
702
702
703
703
704
704
705
705
706
706
707
707
708
708
709
709
710
710
711
711
712
712
713
713
714
714
715
715
716
716
717
717
718
718
719
719
720
720
721
721
722
722
723
723
724
724
725
725
726
726
727
727
728
728
729
729
730
730
731
731
732
732
733
733
734
734
735
735
736
736
737
737
738
738
739
739
740
740
741
741
742
742
743
743
744
744
745
745
746
746
747
747
748
748
749
749
750
750
751
751
752
752
753
753
754
754
755
755
756
756
757
757
758
758
759
759
760
760
761
761
762
762
763
763
764
764
765
765
766
766
767
767
768
768
769
769
770
770
771
771
772
772
773
773
774
774
775
775
776
776
777
777
778
778
779
779
780
780
781
781
782
782
783
783
784
784
785
785
786
786
787
787
788
788
789
789
790
790
791
791
792
792
793
793
794
794
795
795
796
796
797
797
798
798
799
799
800
800
801
801
802
802
803
803
804
804
805
805
806
806
807
807
808
808
809
809
810
810
811
811
812
812
813
813
814
814
815
815
816
816
817
817
818
818
819
819
820
820
821
821
822
822
823
823
824
824
825
825
826
826
827
827
828
828
829
829
830
830
831
831
832
832
833
833
834
834
835
835
836
836
837
837
838
838
839
839
840
840
841
841
842
842
843
843
844
844
845
845
846
846
847
847
848
848
849
849
850
850
851
851
852
852
853
853
854
854
855
855
856
856
857
857
858
858
859
859
860
860
861
861
862
862
863
863
864
864
865
865
866
866
867
867
868
868
869
869
870
870
871
871
872
872
873
873
874
874
875
875
876
876
877
877
878
878
879
879
880
880
881
881
882
882
883
883
884
884
885
885
886
886
887
887
888
888
889
889
890
890
891
891
892
892
893
893
894
894
895
895
896
896
897
897
898
898
899
899
900
900
901
901
902
902
903
903
904
904
905
905
906
906
907
907
908
908
909
909
910
910
911
911
912
912
913
913
914
914
915
915
916
916
917
917
918
918
919
919
920
920
921
921
922
922
923
923
924
924
925
925
926
926
927
927
928
928
929
929
930
930
931
931
932
932
933
933
934
934
935
935
936
936
937
937
938
938
939
939
940
940
941
941
942
942
943
943
944
944
945
945
946
946
947
947
948
948
949
949
950
950
951
951
952
952
953
953
954
954
955
955
956
956
957
957
958
958
959
959
960
960
961
961
962
962
963
963
964
964
965
965
966
966
967
967
968
968
969
969
970
970
971
971
972
972
973
973
974
974
975
975
976
976
977
977
978
978
979
979
980
980
981
981
982
982
983
983
984
984
985
985
986
986
987
987
9
```

Schedule

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

Introduce 7 concepts with demonstration, and review

- ❖ **questions (12 minutes)**
- ❖ **wrap up (3 minutes)**

Example Code Repository



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

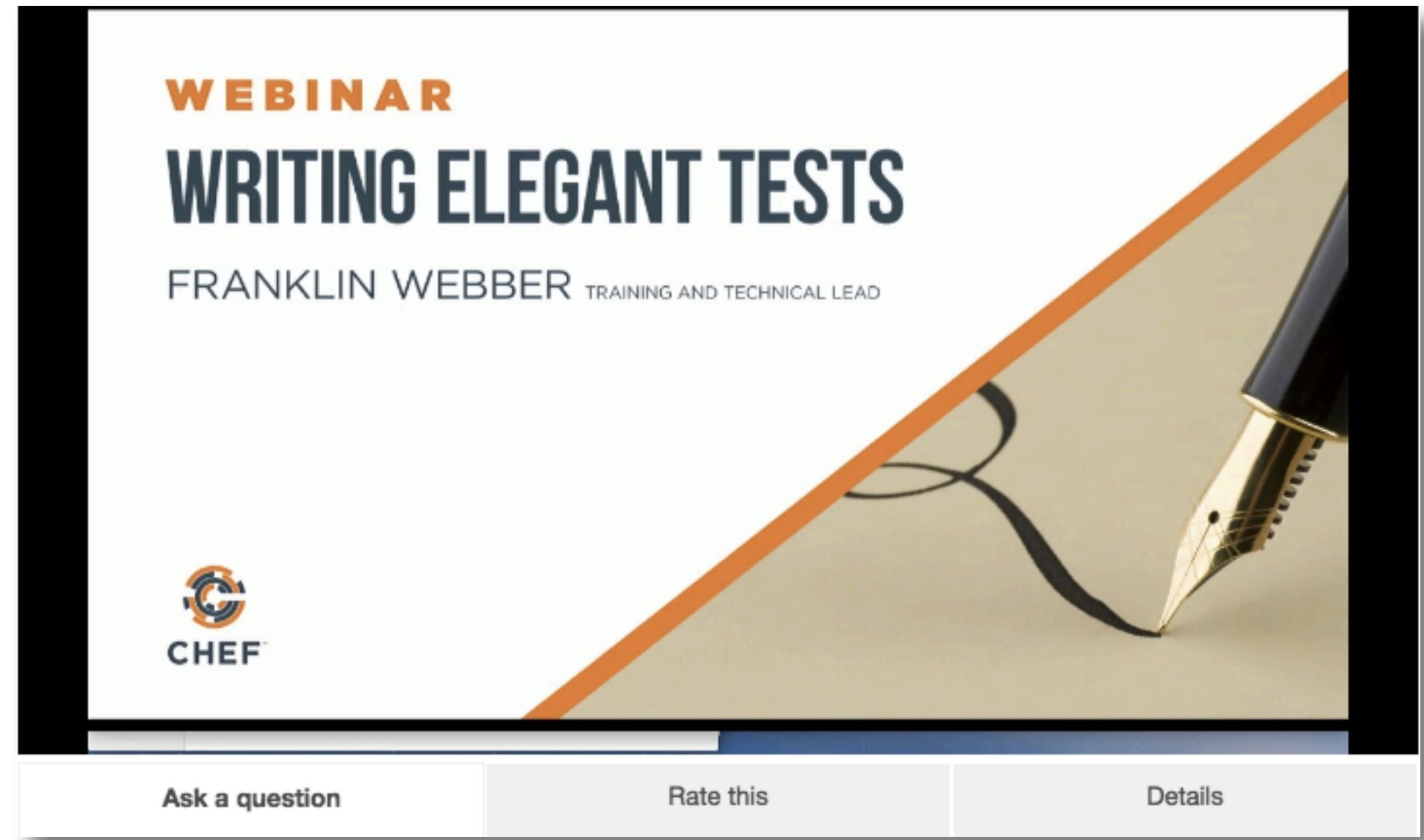
```
Cloning into 'elegant_tests-repo'...
remote: Counting objects: 67, done.
remote: Compressing objects: 100% (41/41), done.
remote: Total 67 (delta 12), reused 67 (delta 12), pack-reused 0
Unpacking objects: 100% (67/67), done.
```

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.

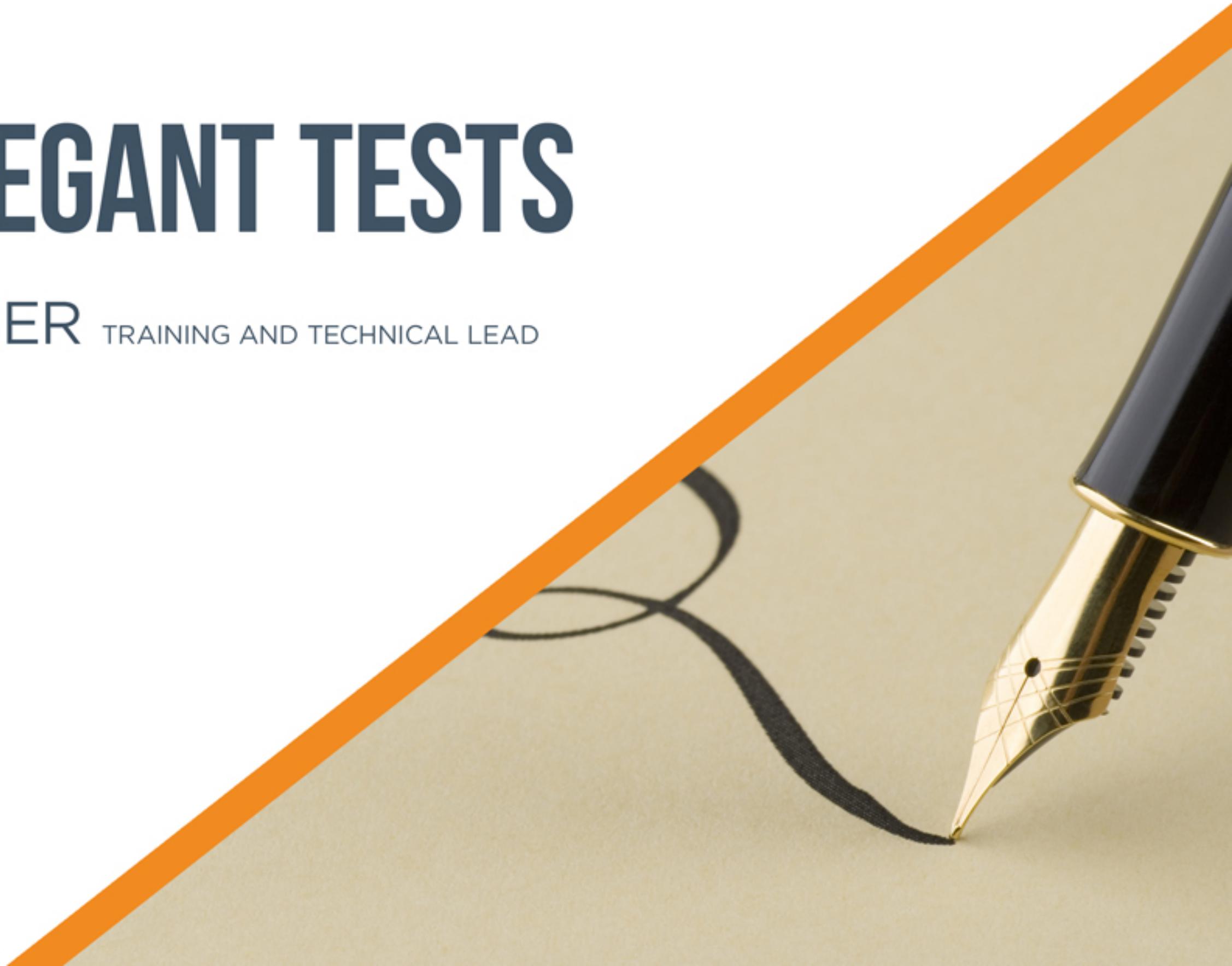
To share, click on the appropriate professional/social network in "Details"



WEBINAR

WRITING ELEGANT TESTS

FRANKLIN WEBBER TRAINING AND TECHNICAL LEAD



Agenda

TECHNIQUES

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

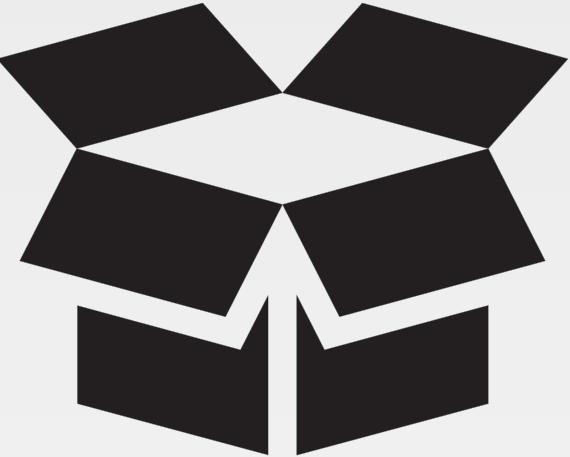
let

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

~ Nikos Kazantzakis

- Concepts
- Demonstration
- Review

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>

Diagramming the let Helper Method

```
describe 'ark::default' do
  context 'when no attributes are ...' 2
    1 let(:chef_run) do
      runner = ChefSpec::SoloRunner.new
      runner.converge(described_recipe)
      runner
    end

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

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

3

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

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
  end

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

```

2

1

4

- 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

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

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
  end
end
```

... EXAMPLES WITHIN CONTEXT

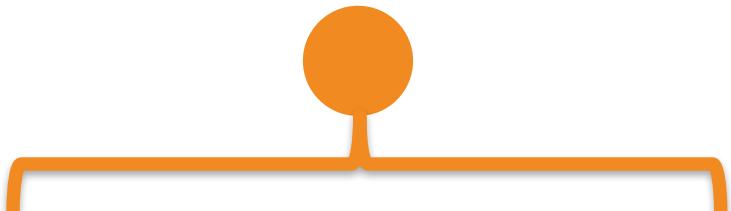
let

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

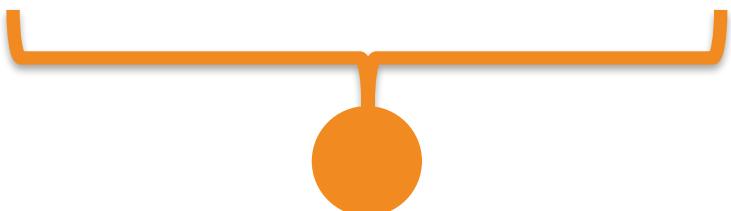
~ Anatole France

- Concepts
- Demonstration
- Review

Objective



*Use `let` to express
the tests succinctly.*



let

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

~ Mother Teresa

- Concepts
- Demonstration
- Review

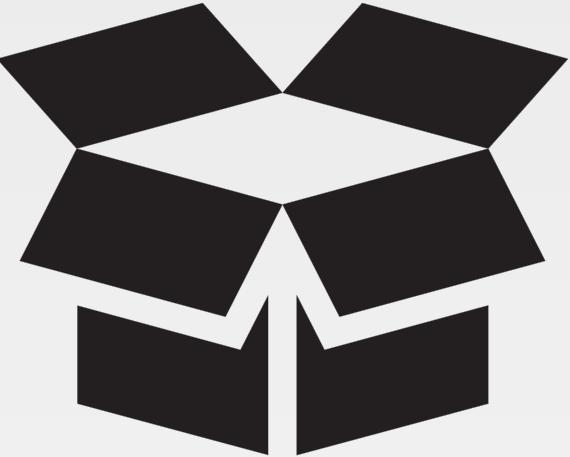
EXAMPLE



Example

<https://goo.gl/ChkP47>

CONCEPT



Using let for clarity

The use of the let to create the `chef_run` saves us from having to 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>

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
  end
end
```

... EXAMPLES WITHIN CONTEXT

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
end
```

let

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

~ Rabindranath Tagore

- ✓ Concepts
- ✓ Demonstration
- ✓ Review

Agenda

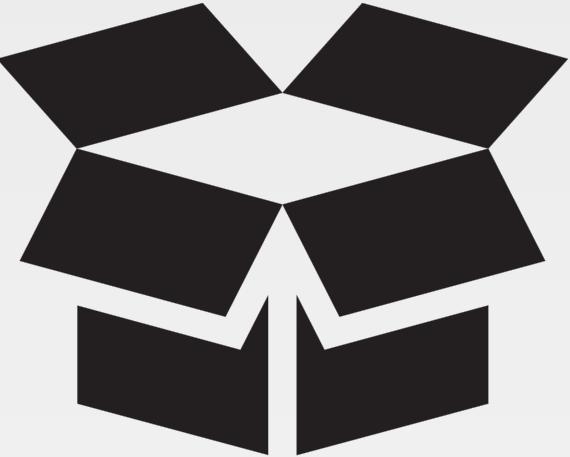
TECHNIQUES

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

shared_examples

- Concepts
- Demonstration
- Review

CONCEPT



shared_examples

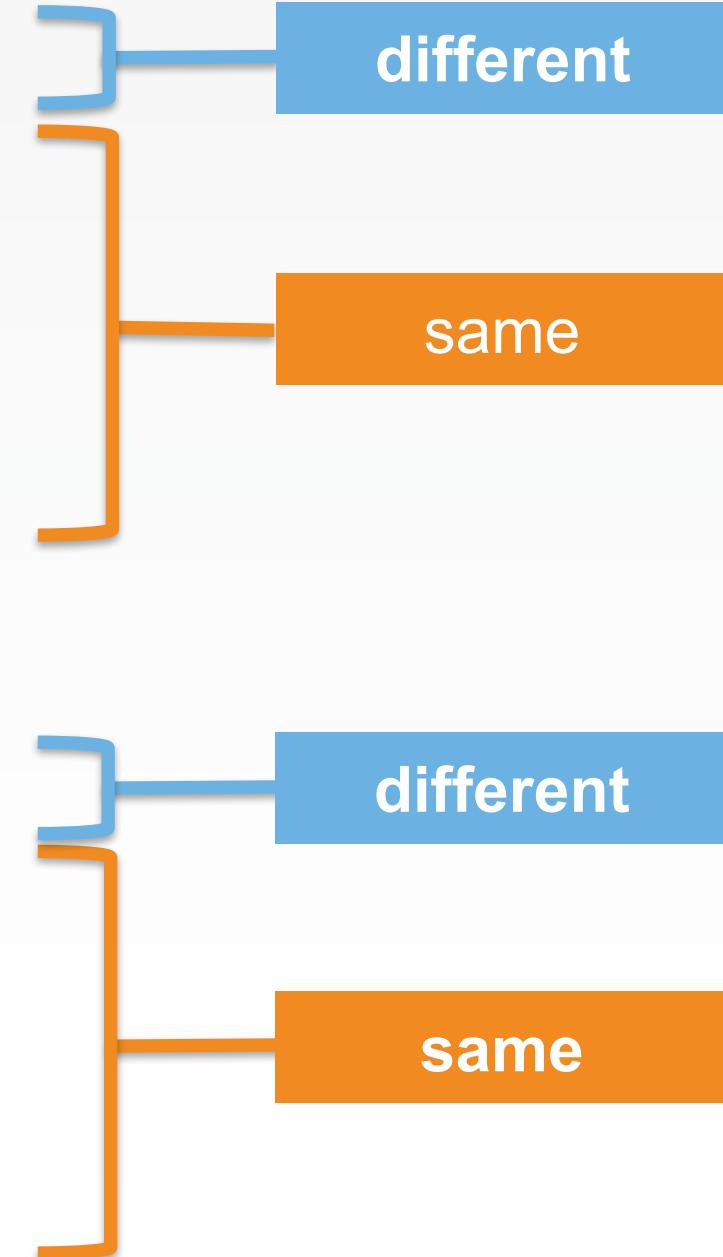
Shared examples let you describe behavior of classes or modules. When declared, a shared group's content is stored. It is only realized in the context of another example group, which provides any context the shared group needs to run.

<https://goo.gl/yi12tM>

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

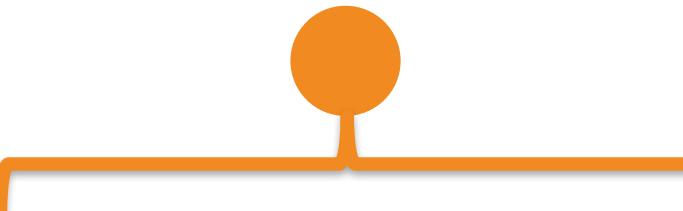
context 'when no attributes are specified, on CentOS' do
  let(:installed_packages) ...
  it 'installs the necessary packages' do
    installed_packages.each do |name|
      expect(chef_run).to install_package(name)
    end
  end
end
```



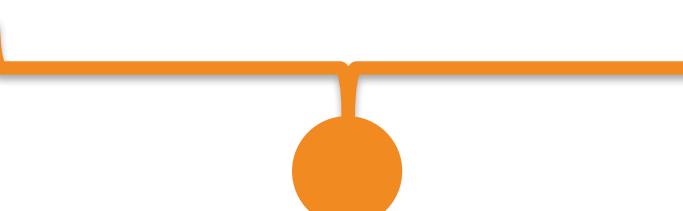
shared_examples

- Concepts
- Demonstration
- Review

Objective



*Use shared examples to
express similarities.*



shared_examples

- Concepts
- Demonstration
- Review

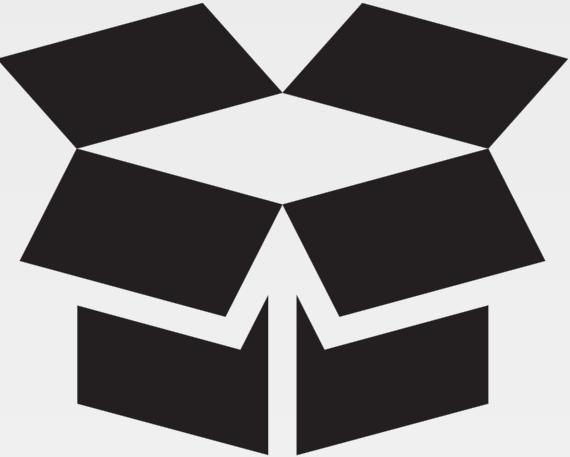
EXAMPLE



Example

<https://goo.gl/ChkP47>

CONCEPT



shared_examples

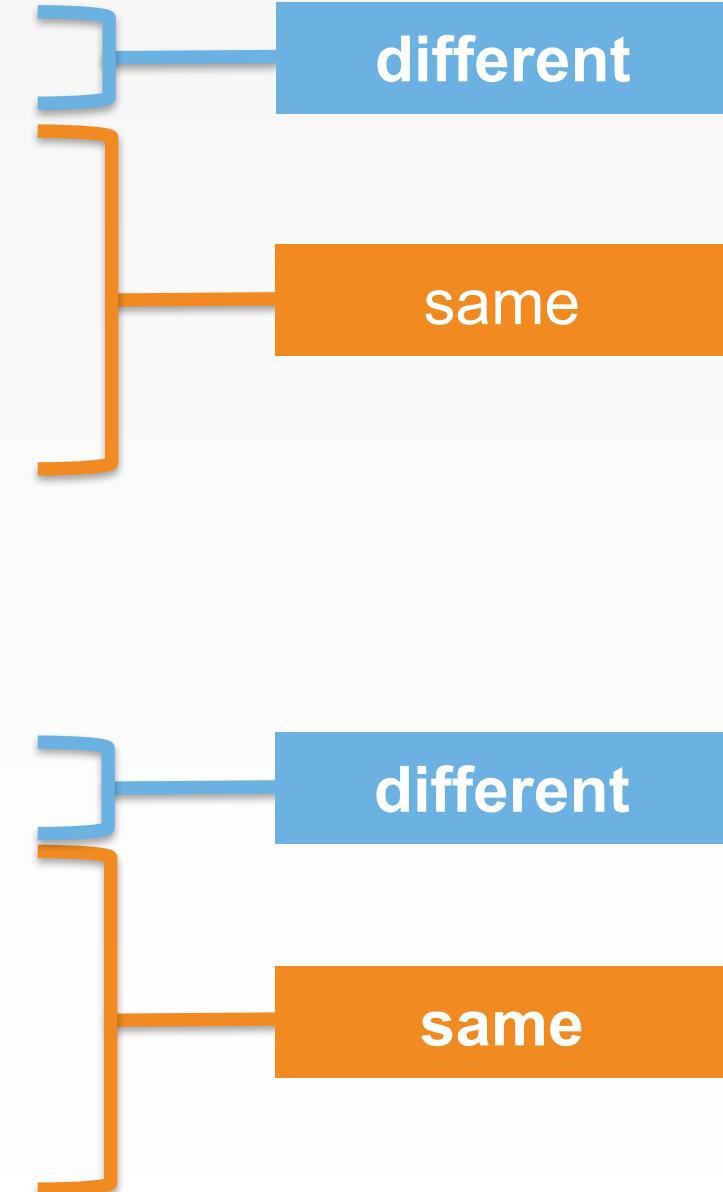
Shared examples let you describe behavior of classes or modules. When declared, a shared group's content is stored. It is only realized in the context of another example group, which provides any context the shared group needs to run.

<https://goo.gl/yi12tM>

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
```



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

Agenda

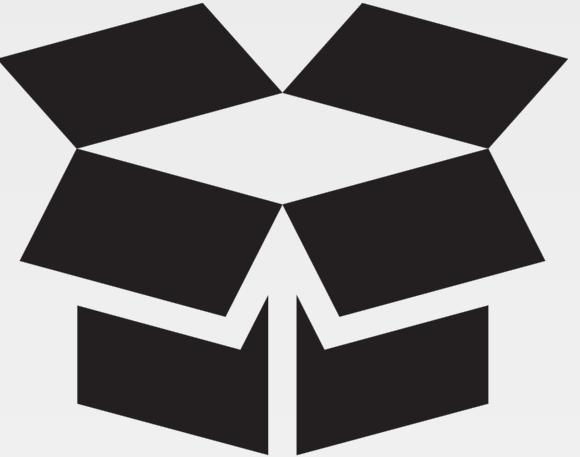
TECHNIQUES

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

def method

- Concepts
- Demonstration
- Review

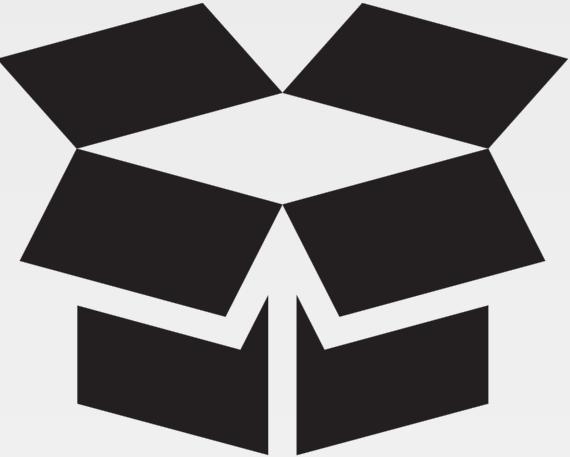
CONCEPT



Create helper method

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

CONCEPT



A Ruby Method with One Parameter

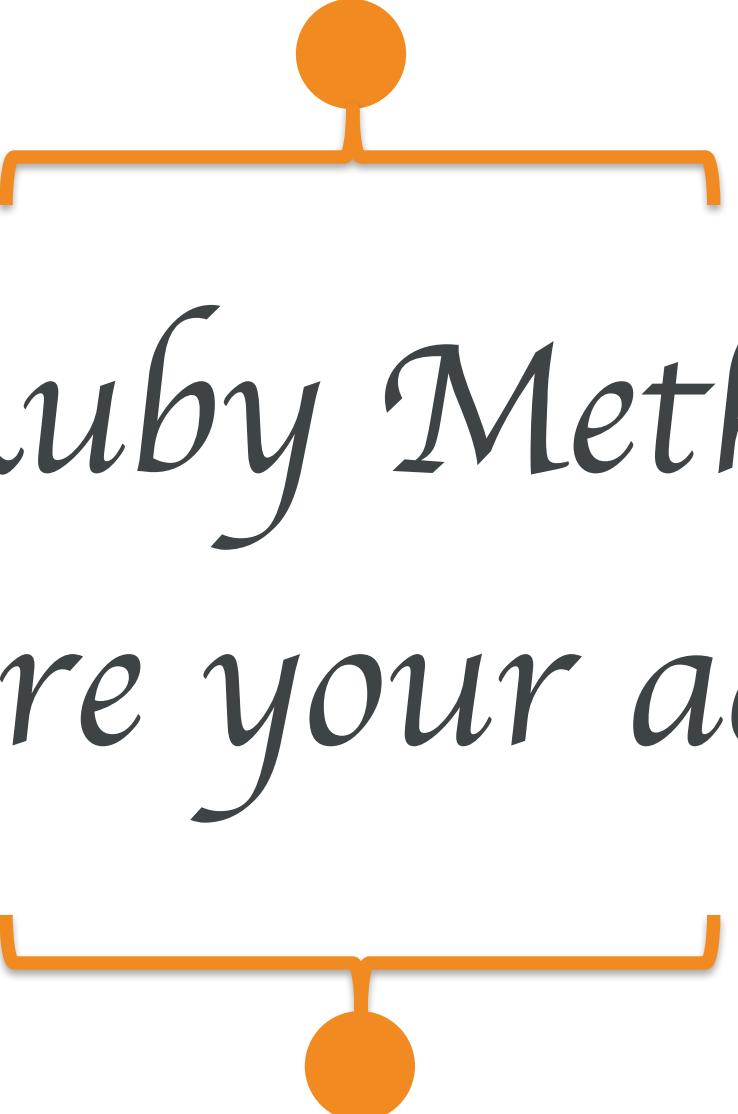
```
def file(name)
  # contents of method
  # last line automatically returns the value
end
```

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

def method

- Concepts
- Demonstration
- Review

Objective



*Use Ruby Methods to
capture your actions.*

def method

- Concepts
- Demonstration
- Review

EXAMPLE



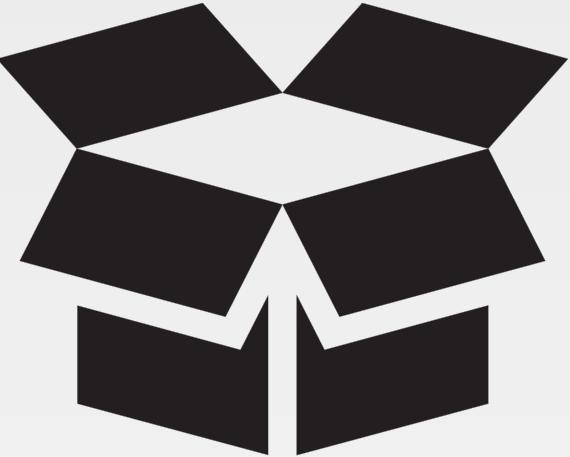
Example

<https://goo.gl/9mRN1D>



(lowercase L)

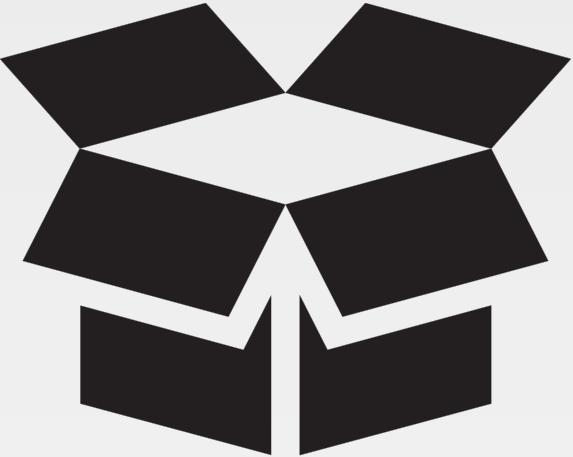
CONCEPT



Create helper method

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

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'.

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

Agenda

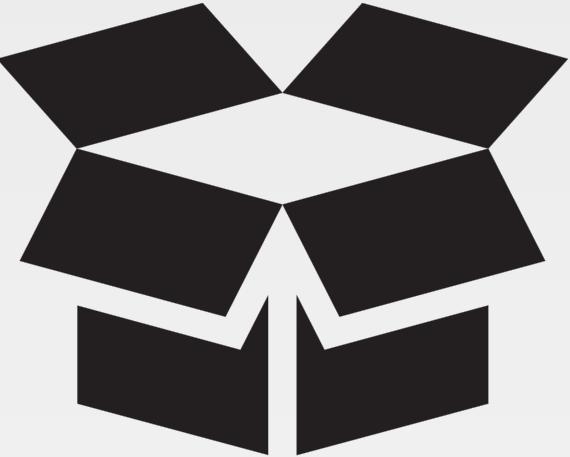
TECHNIQUES

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

shared_context

- Concepts
- Demonstration
- Review

CONCEPT



shared_context

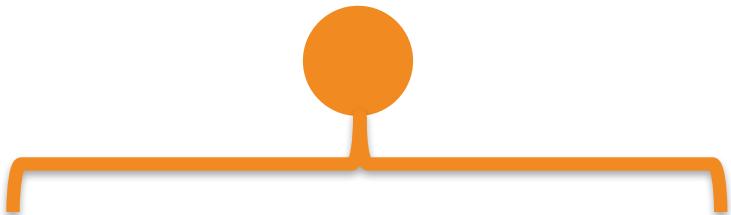
Use `shared_context` to define a block that will be evaluated in the context of example groups either explicitly, using `include_context`, or implicitly by matching metadata.

<http://goo.gl/R0ujTA>

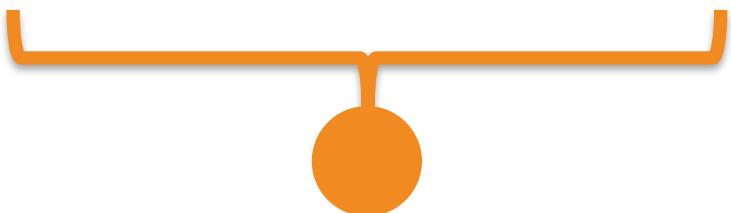
shared_context

- Concepts
- Demonstration
- Review

Objective



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



shared_context

- Concepts
- Demonstration
- Review

EXAMPLE



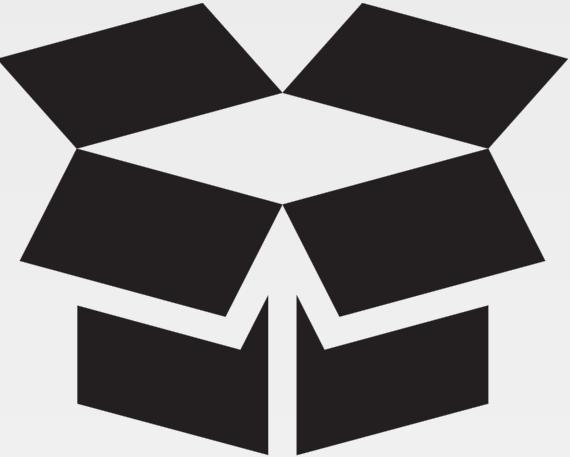
Example

<https://goo.gl/9mRN1D>



(lowercase L)

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 metadata.

<http://goo.gl/R0ujTA>

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) ...
```

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'
```

shared_context

- ✓ Concepts
- ✓ Demonstration
- ✓ Review

Agenda

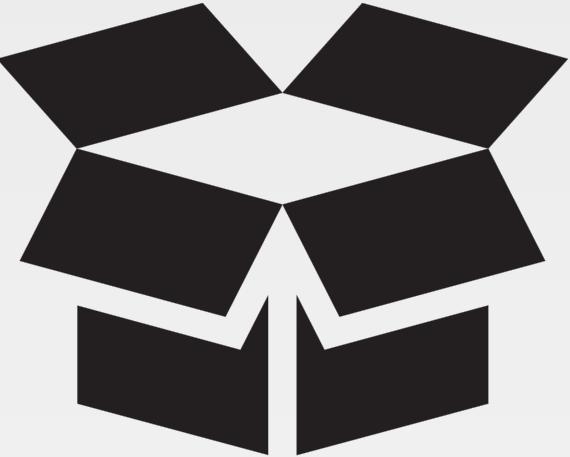
TECHNIQUES

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

require

- Concepts
- Demonstration
- Review

CONCEPT



require

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

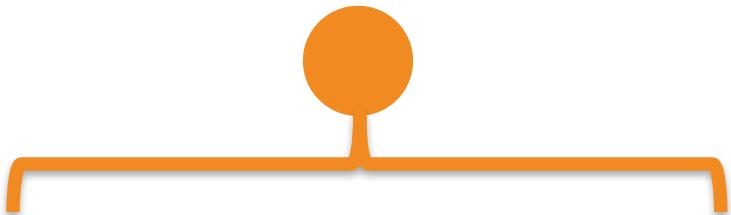
If the filename does not resolve to an absolute path, it will be searched for in the directories listed in `$LOAD_PATH ($:)`.

<http://goo.gl/cLKY37>

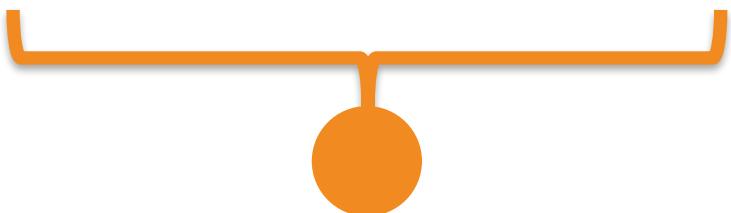
require

- Concepts
- Demonstration
- Review

Objective



*Use require to organize
your thoughts.*



require

- Concepts
- Demonstration
- Review

EXAMPLE



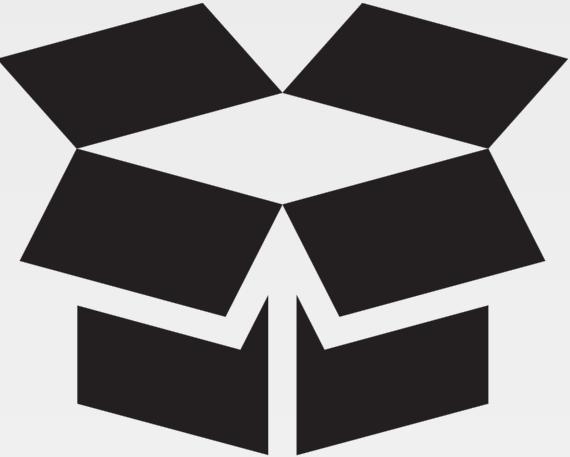
Example

<https://goo.gl/9mRN1D>



(lowercase L)

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>

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
  end
end
```

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
```

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
```

require

- ✓ Concepts
- ✓ Demonstration
- ✓ Review

Agenda

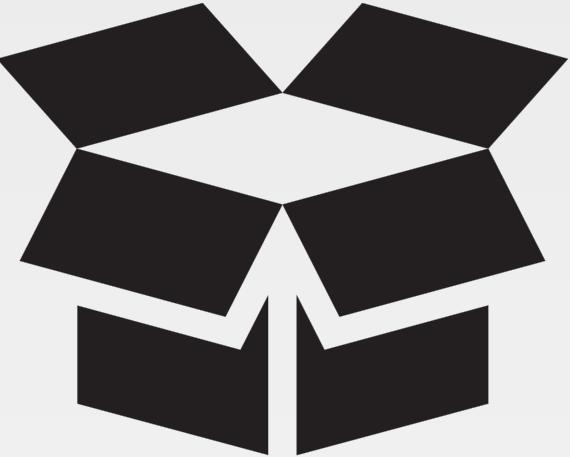
TECHNIQUES

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

alias_example_group_to

- Concepts
- Demonstration
- Review

CONCEPT



alias_example_group_to

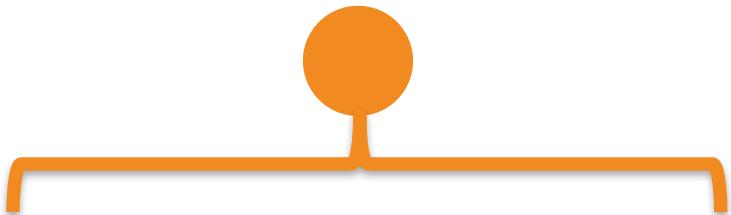
describe and **context** are the default aliases for **example_group**. You can define your own aliases for **example_group** and give those custom aliases default metadata.

<http://goo.gl/DfUCHL>

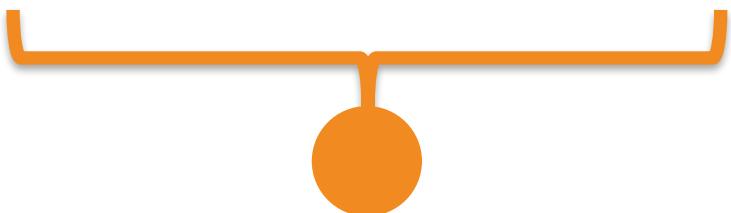
alias_example_group_to

- Concepts
- Demonstration
- Review

Objective



Elegance comes when the context is assumed.



EXAMPLE



Live Demonstration

<https://goo.gl/9mRN1D>

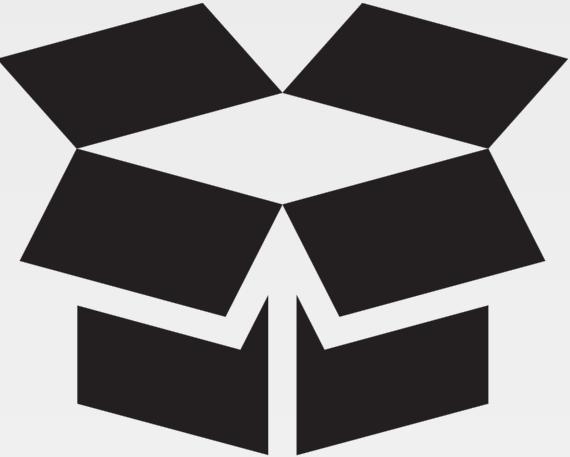


(lowercase L)

alias_example_group_to

- Concepts
- Demonstration
- Review

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>

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'

    # ...
  end

  # ...
end
```

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
```

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
```

alias_example_group_to

- ✓ Concepts
- ✓ Demonstration
- ✓ Review

Agenda

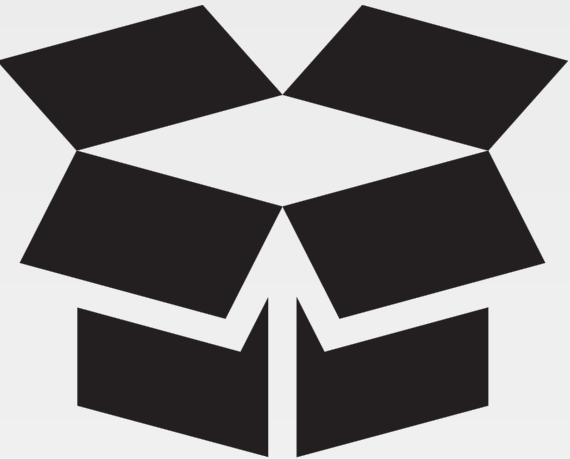
TECHNIQUES

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

creating a Ruby gem

- Concepts
- Demonstration

CONCEPT

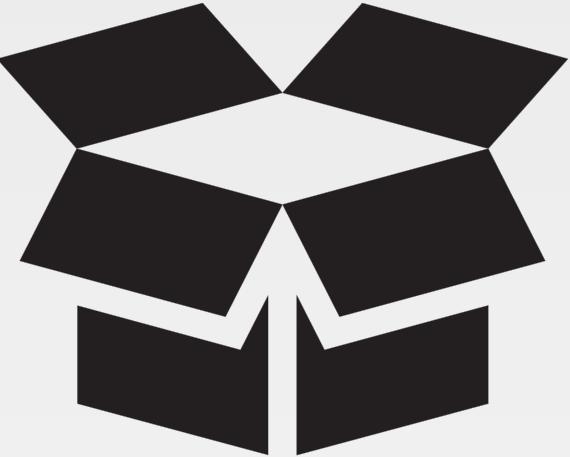


RubyGems

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>

CONCEPT



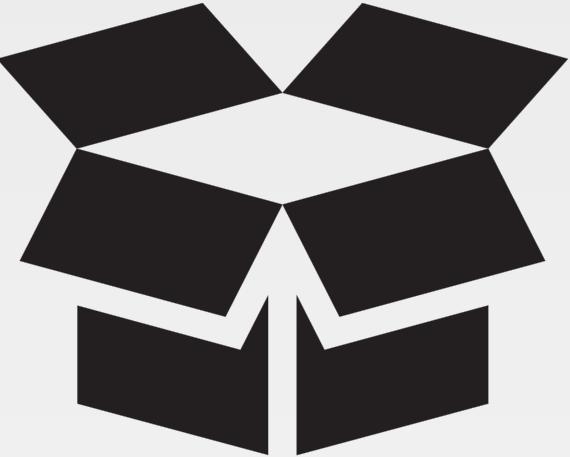
gem

```
> 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>

CONCEPT



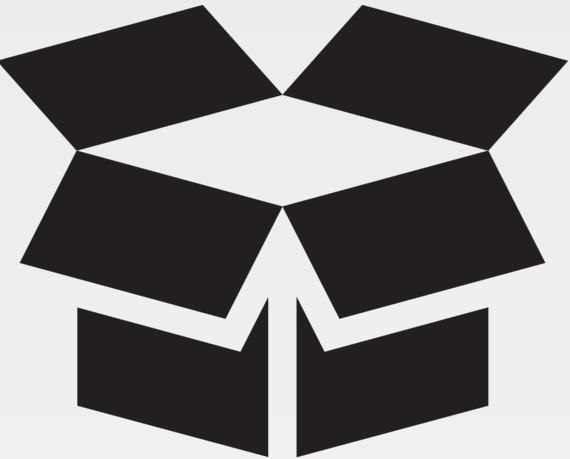
Bundler

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>

CONCEPT



bundle

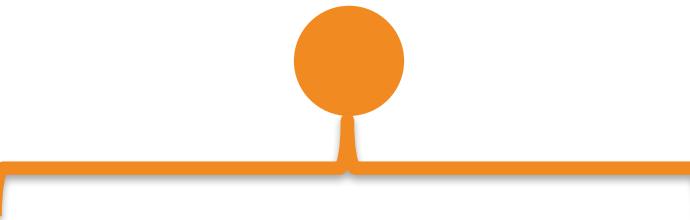
```
> chef exec bundle help
```

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

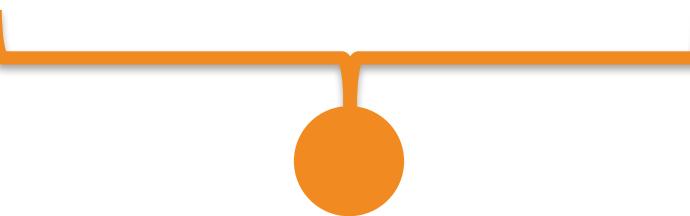
creating a Ruby gem

- Concepts
- Demonstration
- Review

Objective



Create Treasure!



creating a Ruby gem

- Concepts
- Demonstration
- Review

EXAMPLE



Example

github.com/burtlo/chefspec-ohai

Generating the Gem



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

```
Creating gem 'chefspect-myhelpers'...
  create  chefspec-myhelpers/Gemfile
  create  chefspec-myhelpers/.gitignore
  create  chefspec-myhelpers/lib/chefspect/myhelpers.rb
  create  chefspec-myhelpers/lib/chefspect/myhelpers/version.rb
  create  chefspec-myhelpers/chefspect-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 alias...}  
  spec.homepage       = "https://github.com/burtlo/chefspec...com"  
  spec.license        = "MIT"
```

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
end
```

Changing into the Gem's Directory



```
> cd chefspec-myhelpers
```

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`

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
```

Listing the installed Gem



```
> chef gem list chefspec
```

```
*** LOCAL GEMS ***
```

```
chefspect (5.3.0)
```

```
chefspect-ohai (0.1.0)
```

```
chefspect-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
```

-

creating a Ruby gem

- ✓ Concepts
- ✓ Demonstration
- ✓ Review

Agenda

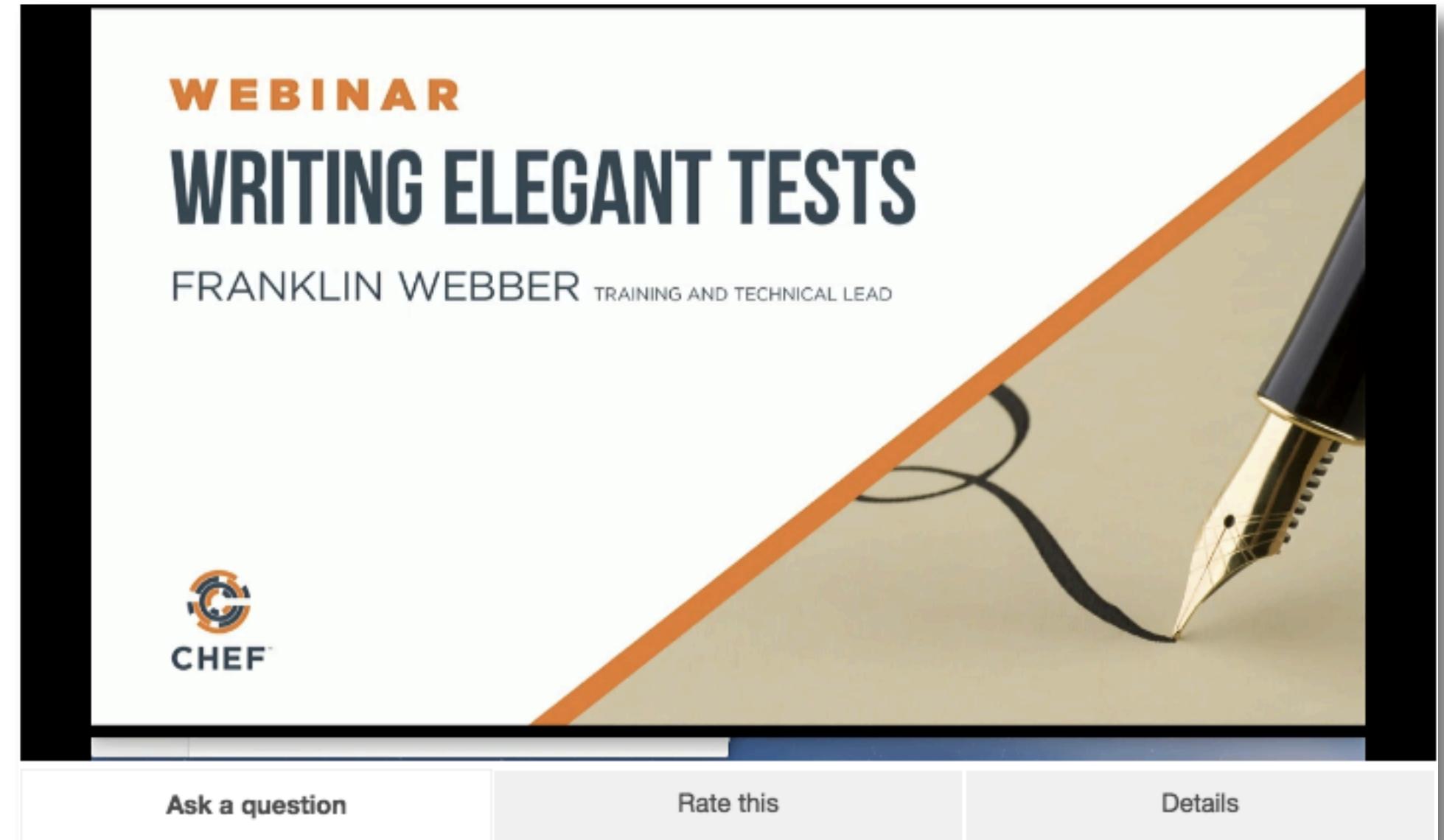
TECHNIQUES

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

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.



DISCUSSION



Q&A

What questions can I answer for you?

Agenda

TECHNIQUES

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

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

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>

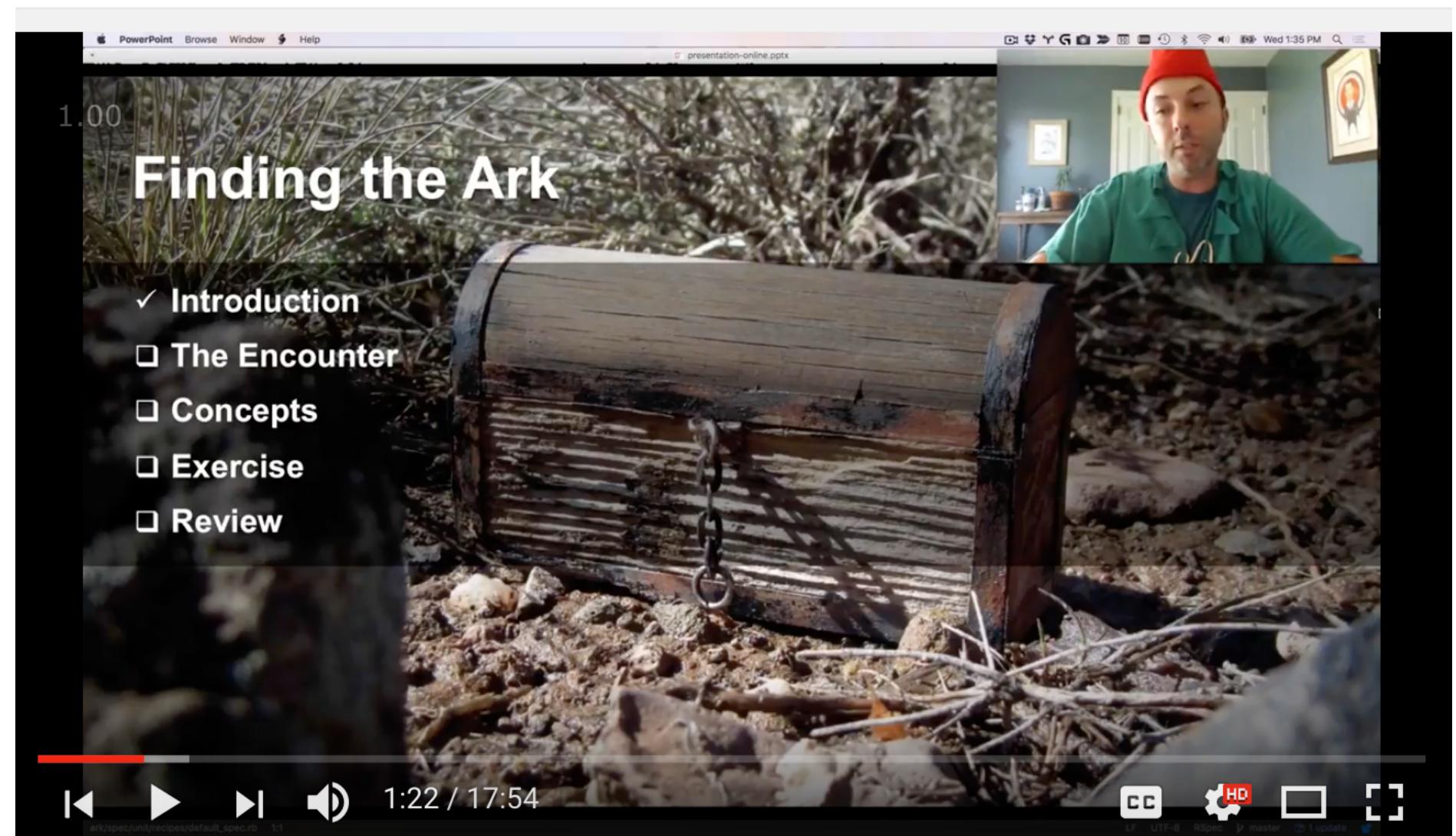
Ruby Gems

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

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>



Agenda

TECHNIQUES

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



CHEF™