

Approval Testing

Introduction

Emily Bache

Technical Agile Coach

Conference Speaker

Based in Gothenburg, Sweden

Author of

“The Coding Dojo Handbook”

@emilybache

emily.bache@proagile.se




ProAgile

@emilybache

Category: gilded-rose

ronjeffries.com/categories/gilded-rose/



About Search Site Categories

Category: gilded-rose

[Gilded Rose Baker's Dozen](#) (May 2, 2020)
Some final thoughts on the Gilded Rose exercise. Well, final for now ... [[Practices, gilded-rose, ruby, tdd](#)]

[Gilded Rose 12](#) (Apr 30, 2020)
To round out a dozen, we simplify things a lot. Too much? What do you think? [[Practices, gilded-rose, ruby, tdd](#)]

[Gilded Rose 11](#) (Apr 29, 2020)
Experiment with lambda. [[Practices, gilded-rose, ruby, tdd](#)]

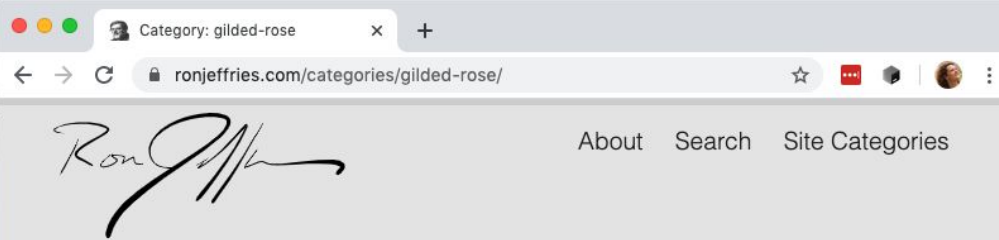
[Gilded Rose 10](#) (Apr 28, 2020)
Today I think I'll double-check that double-decrement issue. Then I want to see if we can combine those two parallel if nests. [[Practices, gilded-rose, ruby, tdd](#)]

[Gilded Rose 9](#) (Apr 27, 2020)
Some really nice progress today ... and one open question. [[Practices, gilded-rose, ruby, tdd](#)]

```
def update_quality()
  @items.each do |item|
    if item.name != "Aged Brie" and item.name != "Backstage passes"
      if item.quality > 0
        if item.name != "Sulfuras, Hand of Ragnaros"
          item.quality = item.quality - 1
        end
      end
    else
      if item.quality < 50
        item.quality = item.quality + 1
        if item.name == "Backstage passes to a TAFKAL80ETC concert"
          if item.sell_in < 11
            if item.quality < 50
              item.quality = item.quality + 1
            end
          end
          if item.sell_in < 6
            if item.quality < 50
              item.quality = item.quality + 1
            end
          end
        end
      end
    end
  end
end

if item.name != "Sulfuras, Hand of Ragnaros"
  item.sell_in = item.sell_in - 1
end

if item.sell_in < 0
  if item.name != "Aged Brie"
```



Category: gilded-rose

[Gilded Rose Baker's Dozen](#) (May 2, 2020)
Some final thoughts on the Gilded Rose exercise. Well, final for now ... [[Practices, gilded-rose, ruby, tdd](#)]

[Gilded Rose 12](#)
To round out a dozen, we s... What do you think?

[Gilded Rose 11](#)
Experiment with lambda.

[Gilded Rose 10](#)
Today I think I'll double-che... Then I want to see if we can nests.

[Gilded Rose 9](#) (Apr 27, 2020)
Some really nice progress today ... and one open question. [[Practices, gilded-rose, ruby, tdd](#)]

```
def update_quality()
  @items.each do |item|
    if item.name != "Aged Brie" and item.name != "Backstage passes"
      if item.quality > 0
        if item.name != "Sulfuras, Hand of Ragnaros"
          item.quality = item.quality - 1
        end
      end
    else
      if item.quality < 50
        item.quality = item.quality + 1
        if item.name == "Backstage passes to a TAFKAL80ETC concert"
          if item.sell_in < 11
            if item.quality < 50
```

I needed more tests

I hate writing tests for existing code, and I know I'm not the only one. And there were times in this exercise where I was really uncertain whether I had broken the old code or not. More tests would have helped me. A time or two, I wrote them. More often, I did not.

```
item.sell_in = item.sell_in - 1
end
if item.sell_in < 0
  if item.name != "Aged Brie"
```

Poll

Do you like writing tests?

On a scale of 1-5 how much do you enjoy the following coding activities?

- Coding a new function from scratch without writing any tests
- Coding a new function from scratch using Test-Driven Development
- Writing tests for an existing (poorly designed) function
- Refactoring an existing (poorly designed) function that lacks tests
- Refactoring an existing (poorly designed) function that has good tests

Supermarket Receipt with Approval Testing

Demo

<https://github.com/emilybache/SupermarketReceipt-Refactoring-Kata>

Approval Testing Characteristics

First impressions of Approval
Testing

- Think about what you just saw. If you had to explain the main idea to someone else, what would you say?
- Note down your explanation in a sentence or two.

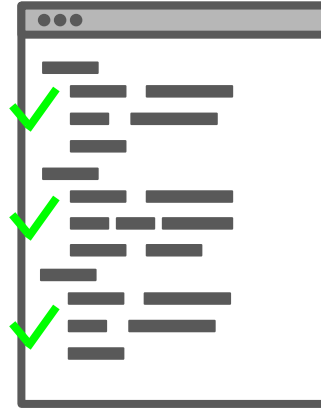
Approval Testing Characteristics

Unit Testing Fundamentals



A Unit is a Small Piece of code

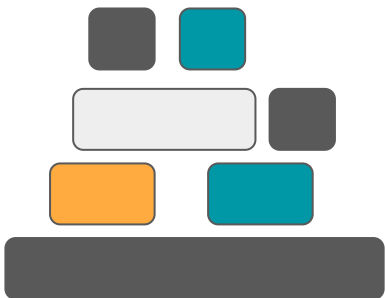
A method or function
A module or class
A group of related classes



An Automated Unit Test

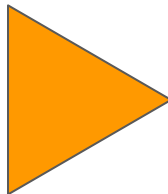
Is designed by a human
Runs without intervention
Reports either 'pass' or 'fail'

The Three Parts of a (Unit) Test



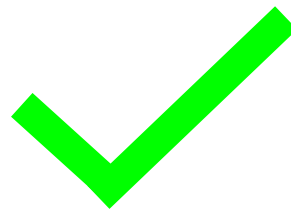
Arrange

Set up the object to be tested, and collaborators



Act

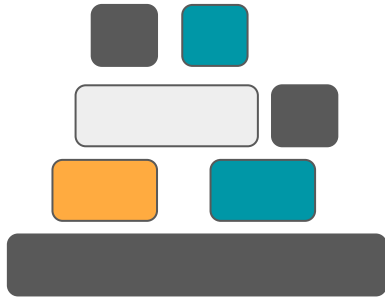
Exercise the unit under test



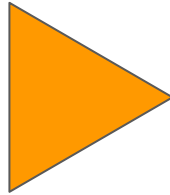
Assert

Make claims about what happened

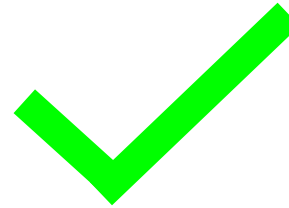
Three parts of an Approval Test



Arrange



Act



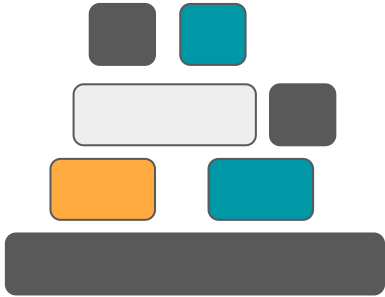
Compare with
Approved output

Compare with Approved Output

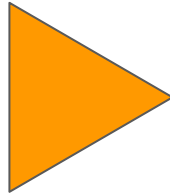
Approved	Actual
toothbrush 0.99	toothbrush 0.99 ✓
toothbrush 0.99	toothbrush 0.99
toothbrush 0.99	toothbrush 0.99
3 for 2(toothbrush) -0.99	β for 2(toothbrush) -0.00
Total: 1.98	Total: 2.97

You inspect and approve actual program output when you create a test case

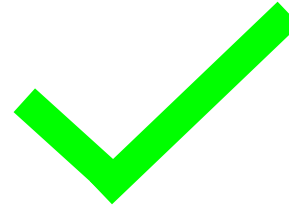
Approval Tests should be reliable



Arrange Test Data,
Fake time,
Stub network ...

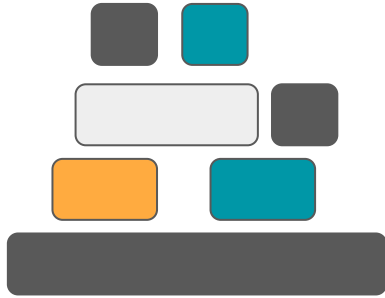


Act

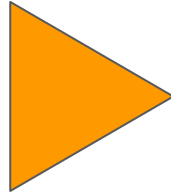


Compare with Approved
output

Three/Four parts of an Approval Test



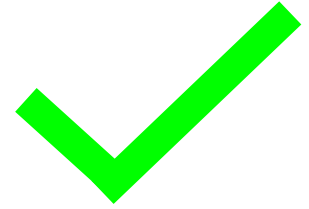
Arrange



Act

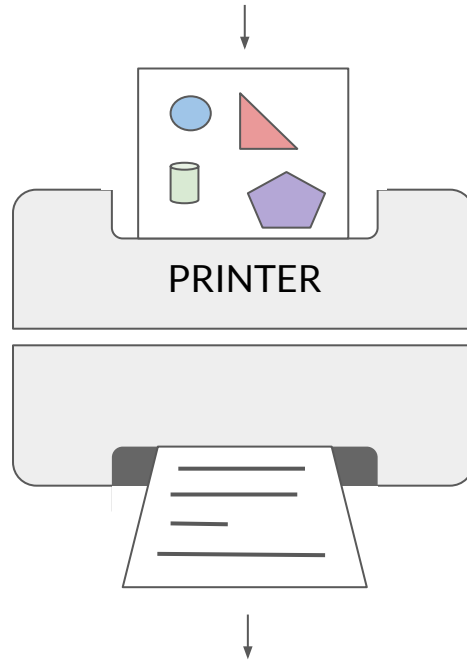


Print



Compare with
Approved output

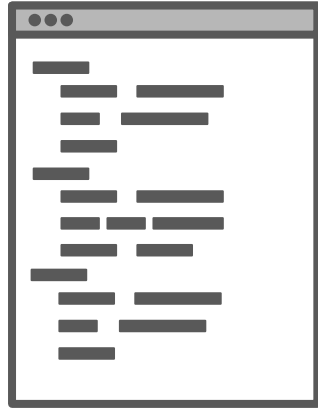
Design a reusable Printer for Complex Objects



Or use a standard library to:

- Serialize
- Pickle
- toString
- toJson

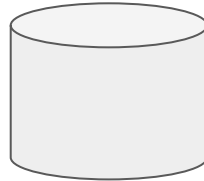
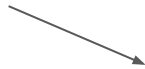
Store Approved output with test sourcecode



SomeTestCase.java



SomeTestCase.approved.txt

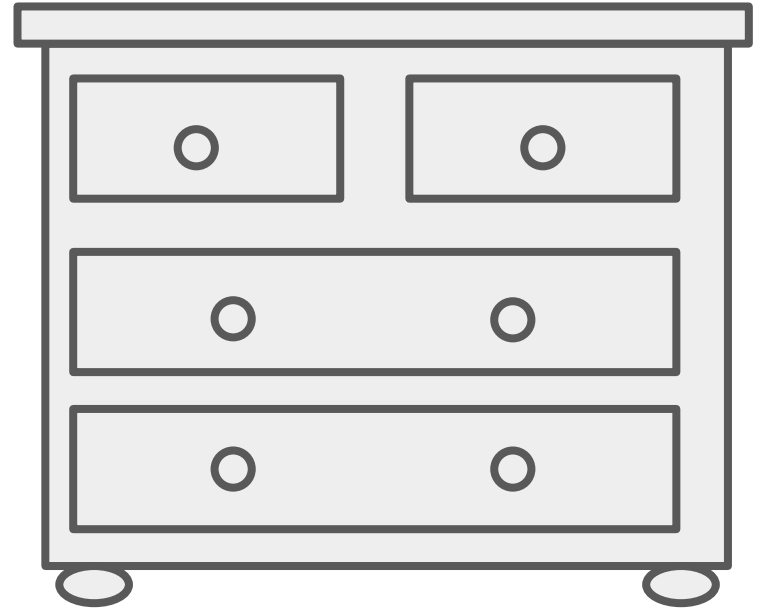


Put them in the same VCS repository

Refactoring Katas

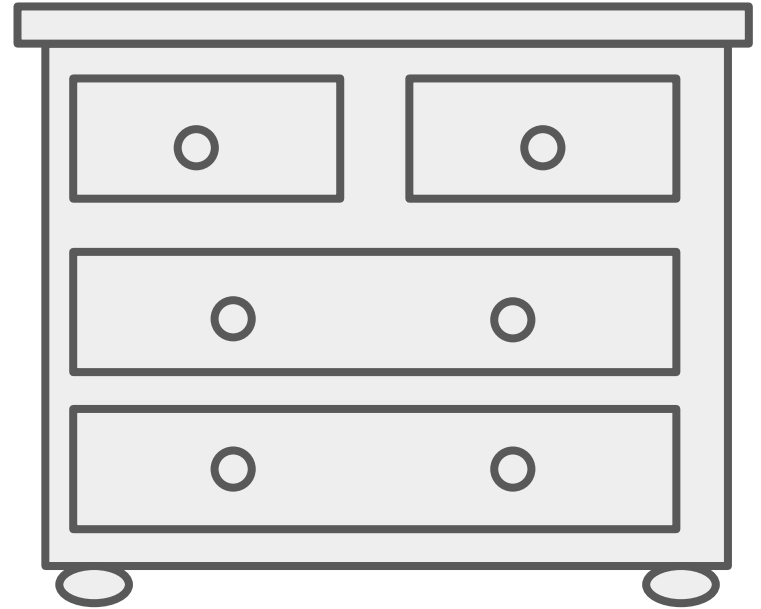
Legacy Code

- Valuable code that works
- Developers find it difficult
- We need to change it safely

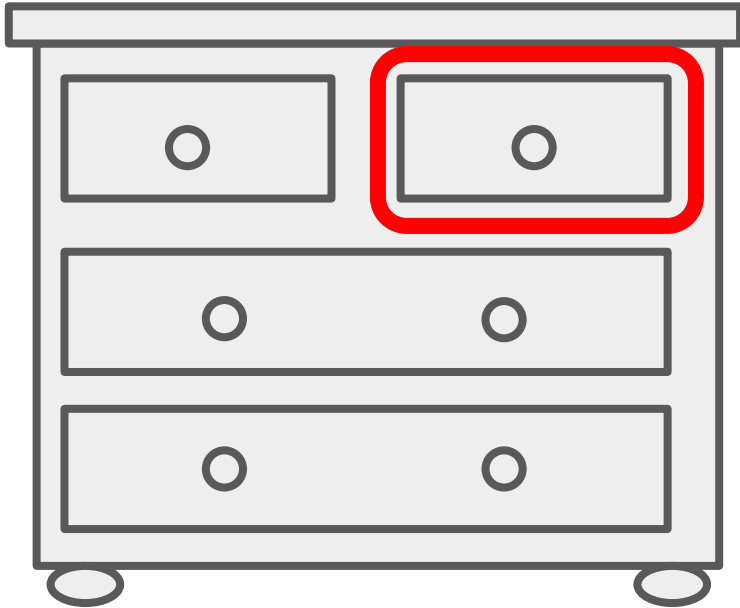


Refactoring Kata

- Practice handling Legacy code

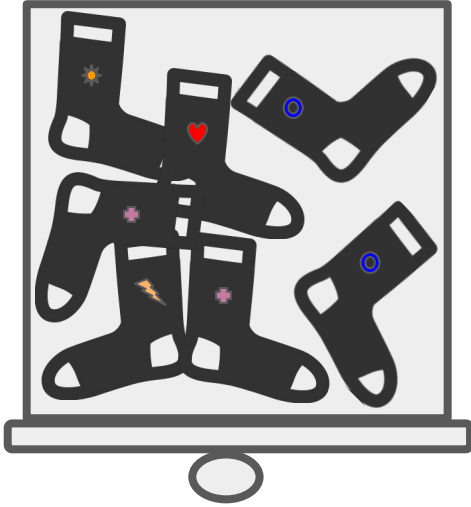


Extract a unit to test



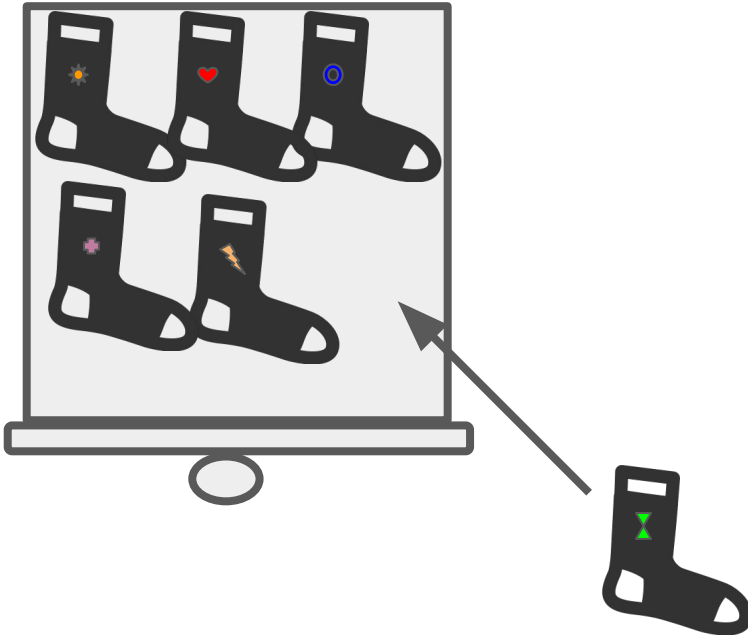
- Smaller pieces will have fewer behaviours
- May be bigger pieces than ordinary unit tests

Cover it in test cases



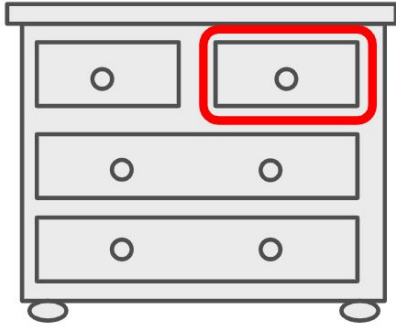
- Go through everything that's in there
- Find suitable test data

Refactor

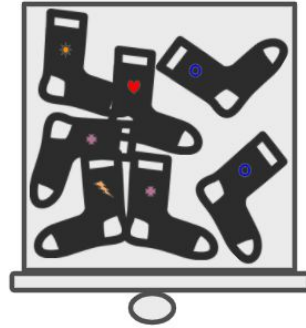


- Improve the design
- Make it easy to add the new feature

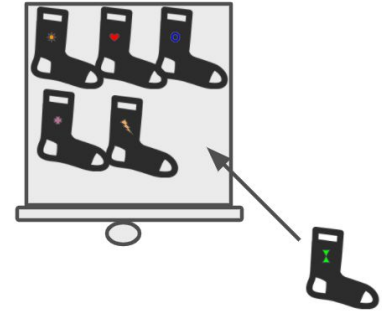
Form of a Refactoring Kata



Find seams and
create a unit



Cover with
tests



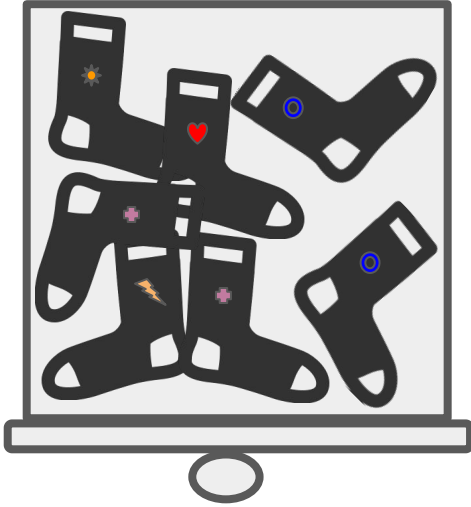
Refactor and
add new
functionality

Product Export with Approval Testing

Demo

<https://github.com/emilybache/Product-Export-Refactoring-Kata>

Use Approval Testing to Cover it in test cases

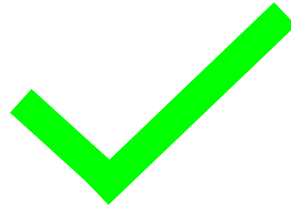


- Go through everything that's in there

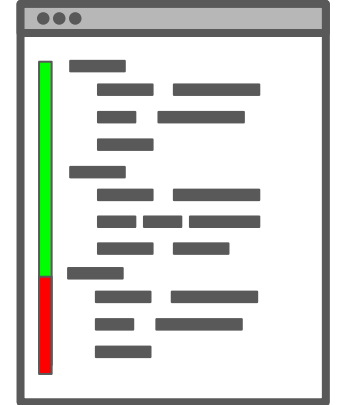
Approval Testing Legacy Code



We assume the code works



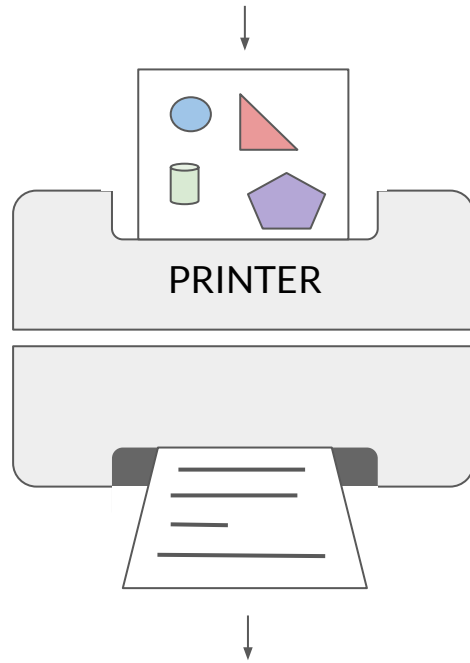
approve whatever the code does



Add more cases to increase coverage

Pre-Comparison Processing

Printers



- `verifyXML()`
- `verifyJSON()`

Text-Based Approval Tests

Poll results


Do you like writing tests?

On a scale of 1-5 how much do you enjoy the following coding activities?

- Coding a new function from scratch without writing any tests
- Coding a new function from scratch using Test-Driven Development
- Writing tests for an existing (poorly designed) function
- Refactoring an existing (poorly designed) function that lacks tests
- Refactoring an existing (poorly designed) function that has good tests

Category: gilded-rose

ronjeffries.com/categories/gilded-rose/



About Search Site Categories

Category: gilded-rose

[Gilded Rose Baker's Dozen](#) (May 2, 2020)
Some final thoughts on the Gilded Rose exercise. Well, final for now ... [[Practices, gilded-rose, ruby, tdd](#)]

[Gilded Rose 12](#) (Apr 30, 2020)
To round out a dozen, we simplify things a lot. Too much? What do you think? [[Practices, gilded-rose, ruby, tdd](#)]

[Gilded Rose 11](#) (Apr 29, 2020)
Experiment with lambda. [[Practices, gilded-rose, ruby, tdd](#)]

[Gilded Rose 10](#) (Apr 28, 2020)
Today I think I'll double-check that double-decrement issue. Then I want to see if we can combine those two parallel if nests. [[Practices, gilded-rose, ruby, tdd](#)]

[Gilded Rose 9](#) (Apr 27, 2020)
Some really nice progress today ... and one open question. [[Practices, gilded-rose, ruby, tdd](#)]

```
def update_quality()
  @items.each do |item|
    if item.name != "Aged Brie" and item.name != "Backstage passes"
      if item.quality > 0
        if item.name != "Sulfuras, Hand of Ragnaros"
          item.quality = item.quality - 1
        end
      end
    else
      if item.quality < 50
        item.quality = item.quality + 1
        if item.name == "Backstage passes to a TAFKAL80ETC concert"
          if item.sell_in < 11
            if item.quality < 50
              item.quality = item.quality + 1
            end
          end
          if item.sell_in < 6
            if item.quality < 50
              item.quality = item.quality + 1
            end
          end
        end
      end
    end
  end
end

if item.name != "Sulfuras, Hand of Ragnaros"
  item.sell_in = item.sell_in - 1
end

if item.sell_in < 0
  if item.name != "Aged Brie"
```

Gilded Rose
with Approval Testing

Demo



@emilybache



Legacy Code



Arrange



Act

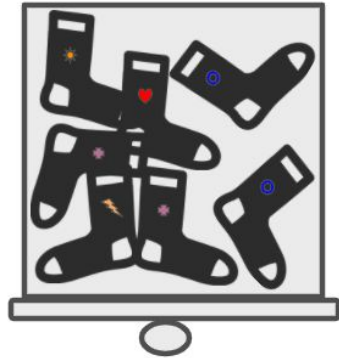


Print



Compare with
Approved output

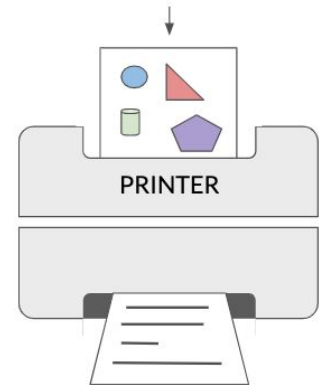
Approval Testing



Add Approval Tests to
existing code

Approved		Actual	
toothbrush	0.99	1 1	toothbrush 0.99
toothbrush	0.99	2 2	toothbrush 0.99
toothbrush	0.99	3 3	toothbrush 0.99
3 for 2(toothbrush)	-0.99	> 4 4 <<	B for 2(toothbrush) -0.00
		5 5	
Total:	1.98	> 6 6 <<	Total: 2.97

Diff-merge to view & update test results



Design the
output you
approve

Spare Slides