For the Test We Love

❤️ 💎

Why Do I Test 🤨

It's NOT about correctness

It's about stability

How Do I Test

- Manual Testing

- Automated Testing

- Elegant Automated Testing (RSpec 😘)

FizzBuzz

- When the number divided by 3
  > Fizz

- When the number divided by 5
  > Buzz

- when the number divided by 15
  > FizzBuzz

- else
  > number

FizzBuzz

- Given inputs [1,3,5,7,9,11,13,15]

> [1, Fizz, Buzz, 7, Fizz, 11, 14, FizzBuzz]

FizzBuzz

```ruby
module FizzBuzz
  def self.check(numbers)
    numbers.map do |number|
      if number % 15 == 0
        'FizzBuzz'
      elsif number % 3 == 0
        'Fizz'
      elsif number % 5 == 0
        'Buzz'
      else
        number.to_s
      end
    end
  end
end
```

Manual

```
→ irb -r ./fizzbuzz.rb
2.3.3 :001 > FizzBuzz.check
[1,3,5,7,9,11,13,15]

 => ["1", "Fizz", "Buzz", "7", "Fizz",
"11", "13", "FizzBuzz"]
2.3.3 :002 >
```

Automated Testing

```
→ ruby fizzbuzz.automated.rb
test with input [1, 3, 5, 7, 9, 11, 13, 15]
gets ["1", "Fizz", "Buzz", "7", "Fizz",
"11", "13", "FizzBuzz"]
>>> huray!

test with input [1, 2, 3, 4, 5] gets
["something stupid"]
>>> F*CK!

-----summary-----
From 2 cases
test passes 1, and failed 1
```

Elegant Automated Testing

```
→ rspec fizzbuzz.rspec.rb -fd

FizzBuzz
  returns ["1", "Fizz", "Buzz", "13", "FizzBuzz"] when given
[1,3,5,13,15]
  returns ["something stupid"] when given [1,2,3,4,5] (FAILED - 1)
```

Elegant Automated Testing

```
Failures:

  1) FizzBuzz returns ["something stupid"] when given [1,2,3,4,5]
     Failure/Error: expect(FizzBuzz.check([1,2,3,4,5])).to
eq(["something stupid"])

       expected: ["something stupid"]
            got: ["1", "2", "Fizz", "4", "Buzz"]

       (compared using ==)
     # ./fizzbuzz.rspec.rb:12:in `block (2 levels) in <top
(required)>'

Finished in 0.01576 seconds (files took 0.12338 seconds to load)
2 examples, 1 failure

Failed examples:

rspec ./fizzbuzz.rspec.rb:11 # FizzBuzz returns ["something
stupid"] when given [1,2,3,4,5]
```

Better Automated Testing

SAFE

Better Automated Testing

- **S** - single purpose

- **A** - align with its own kind

- **F** - fast!

- **E** - easy to spot errors

Demo

https://github.com/offfffz/rbtd-fizzbuzz