Ruby Koans

learn Ruby with Neo

Ruby Koans

The Path to Ruby Enlightenment

Jim Weirich
Chief Scientist
Neo Ohio
@jimweirich

Joe O'Brien
Software Artisan
Neo Ohio
@objo

Who are Koans for?

- Know little (or nothing) about Ruby
- Have a background in some other language
  - (Java, C#, Python, etc)

Structure

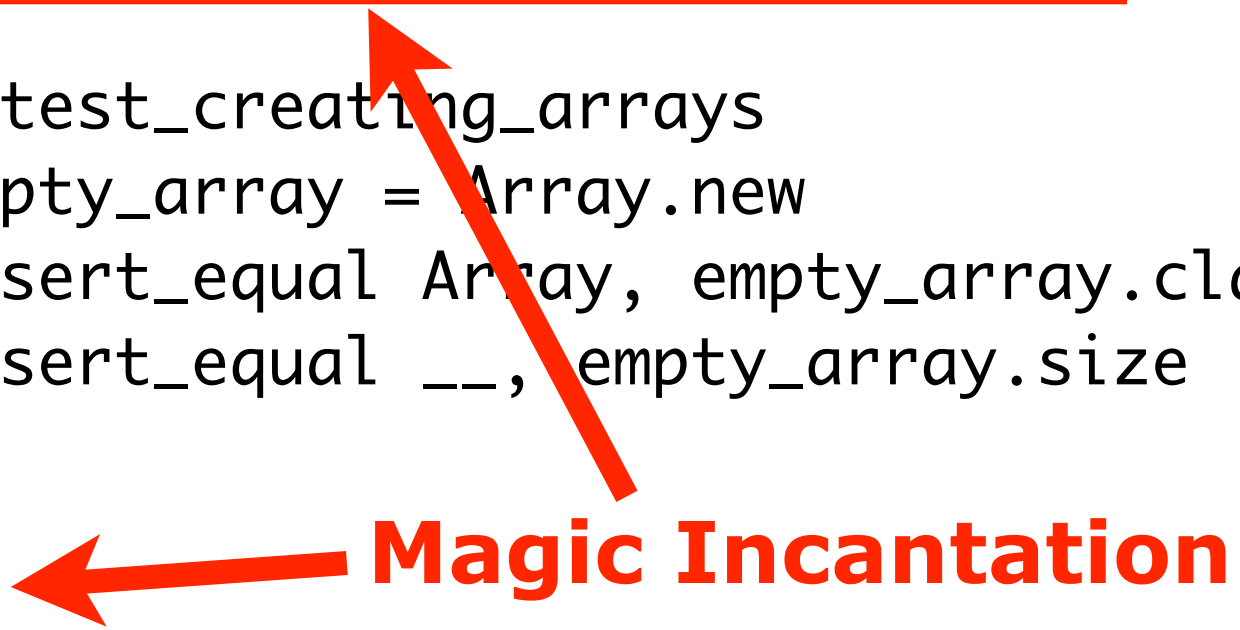
Each file focuses on one area of the Ruby language.

Typical Koan

```
require 'edgecase'

class AboutArrays < EdgeCase::Koan

  def test_creating_arrays
    empty_array = Array.new
    assert_equal Array, empty_array.class
    assert_equal __, empty_array.size
  end
  ...
end
```

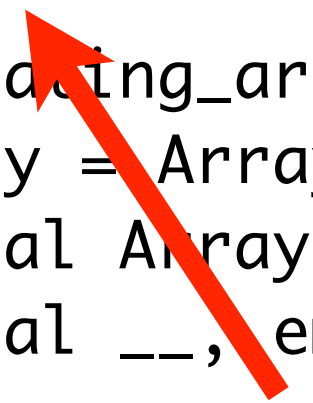
Typical Koan

```
require 'edgecase'

class AboutArrays < EdgeCase::Koan

  def test_creating_arrays
    empty_array = Array.new
    assert_equal Array, empty_array.class
    assert_equal __, empty_array.size
  end

  ...
end
```



Typical Koan

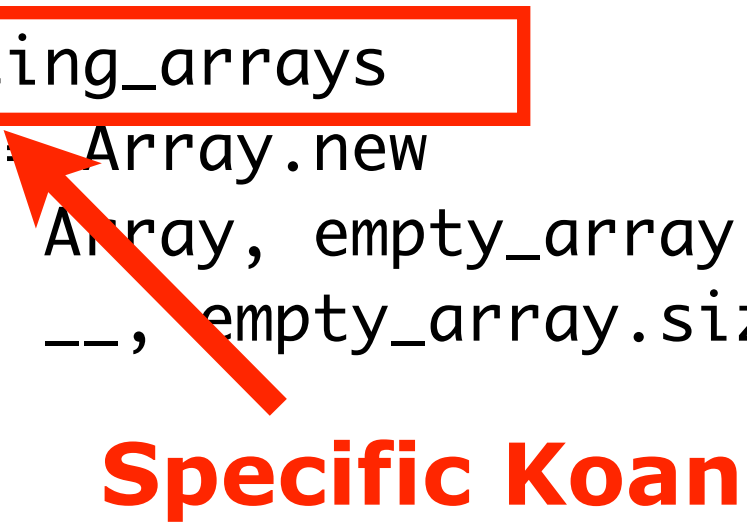
```
require 'edgecase'

class AboutArrays < EdgeCase::Koan

  def test_creating_arrays
    empty_array = Array.new
    assert_equal Array, empty_array.class
    assert_equal __, empty_array.size
  end
  ...
end
```



Koan Category

Typical Koan

```
require 'edgecase'

class AboutArrays < EdgeCase::Koan

  def test_creating_arrays
    empty_array = Array.new
    assert_equal Array, empty_array.class
    assert_equal __, empty_array.size
  end
  ...
end
```



Specific Koan

Typical Koan

```
require 'edgecase'

class AboutArrays < EdgeCase::Koan
  def test_creating_arrays
    empty_array = Array.new
    assert_equal Array, empty_array.class
    assert_equal __, empty_array.size
  end
  ...
end
```

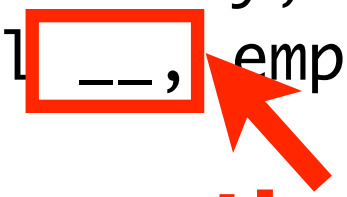
Typical Koan

```
require 'edgecase'

class AboutArrays < EdgeCase::Koan
  def test_creating_arrays
    empty_array = Array.new
    assert_equal Array, empty_array.class
    assert_equal __, empty_array.size
  end
  ...
end
```

Some Assertions Just pass

Typical Koan

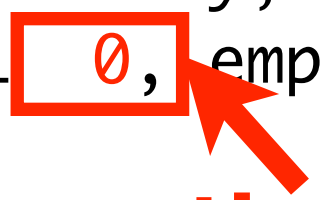
```
require 'edgecase'

class AboutArrays < EdgeCase::Koan
  def test_creating_arrays
    empty_array = Array.new
    assert_equal Array, empty_array.class
    assert_equal __, empty_array.size
  end
  ...
end
```



Most Assertions need User Input

Typical Koan

```
require 'edgecase'

class AboutArrays < EdgeCase::Koan
  def test_creating_arrays
    empty_array = Array.new
    assert_equal Array, empty_array.class
    assert_equal  0, empty_array.size
  end
  ...
end
```



Most Assertions need User Input

Typical Koan

```
def test_something
  ...
end

def test_something_else
  ...
end
```

Normally koans will be listed without breaks

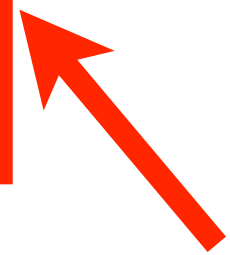
Typical Koan

```
# ------------------------------------
class Dog
end

def test_something_about_dog
end

def test_something_else_about_dog
end
# ------------------------------------
```

If the several koans are related to something defined outside a koan, they will be visually grouped.

Typical Koan

```
# THINK ABOUT IT:
#
# Is it better to use
#    obj.nil?
# or
#    obj == nil
# Why?
```



Sometimes you are just asked to think about an aspect of what you just learned.

path_to_enlightenment.rb

```
# The path to Ruby Enlightenment starts with the
# following:

require 'about_basics'
require 'about_nil'
require 'about_arrays'
require 'about_array_assignment'
require 'about_hashes'
require 'about_methods'
require 'about_control_statements'
require 'about_true_and_false'
...
```

Let’s Start

You Will Need

- Working version of Ruby
- The Ruby Koans folder
- Some kind of a text editor

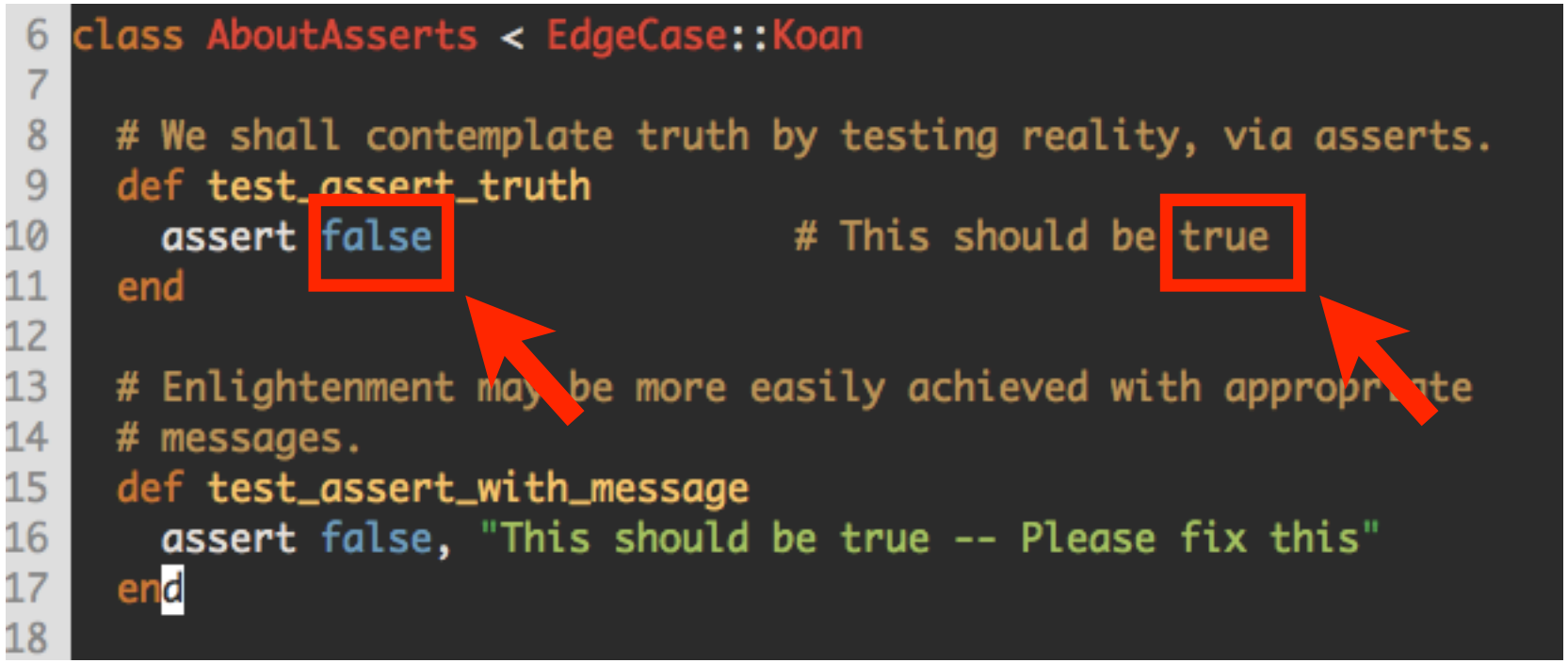
Path to Enlightenment

Path to Enlightenment

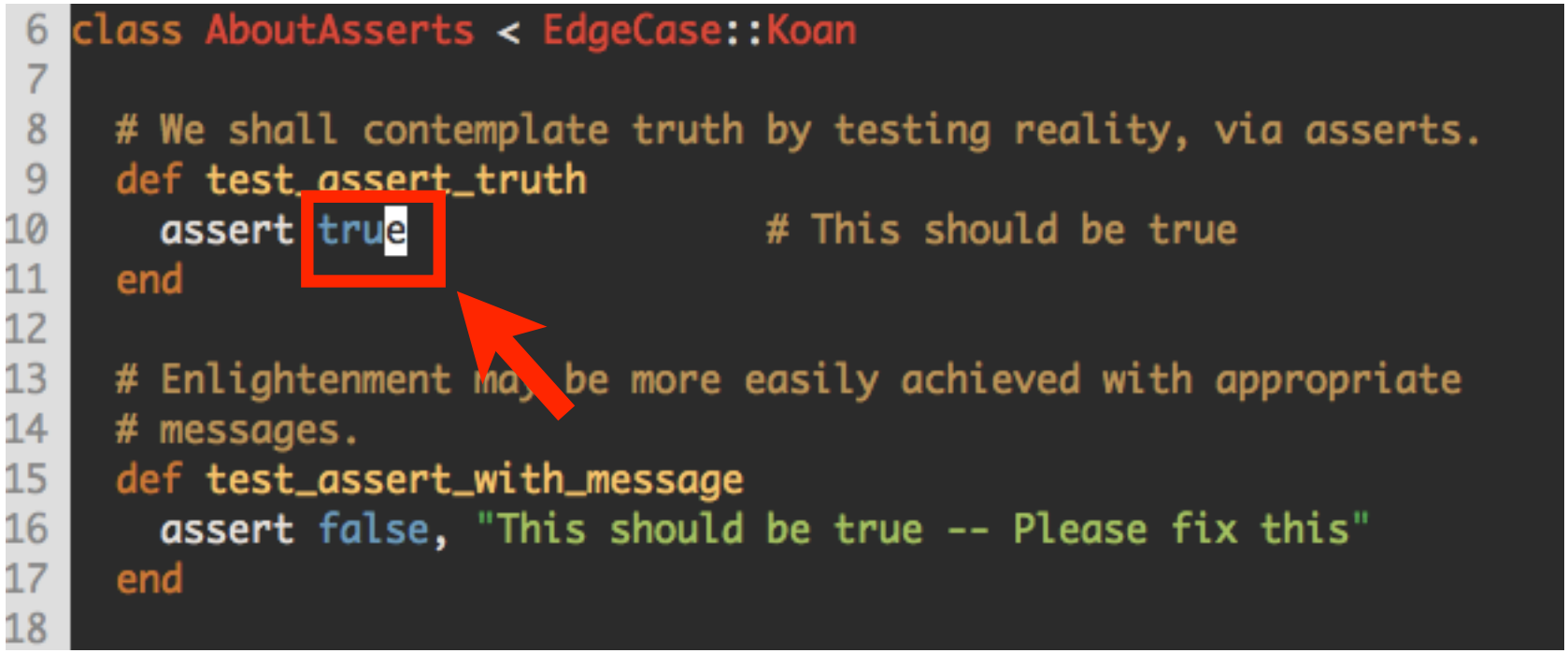
Path to Enlightenment

```
6  class AboutAsserts < EdgeCase::Koan
7
8    # We shall contemplate truth by testing reality, via asserts.
9    def test_assert_truth
10     assert false                  # This should be true
11   end
12
13   # Enlightenment may be more easily achieved with appropriate
14   # messages.
15   def test_assert_with_message
16     assert false, "This should be true -- Please fix this"
17   end
18
```

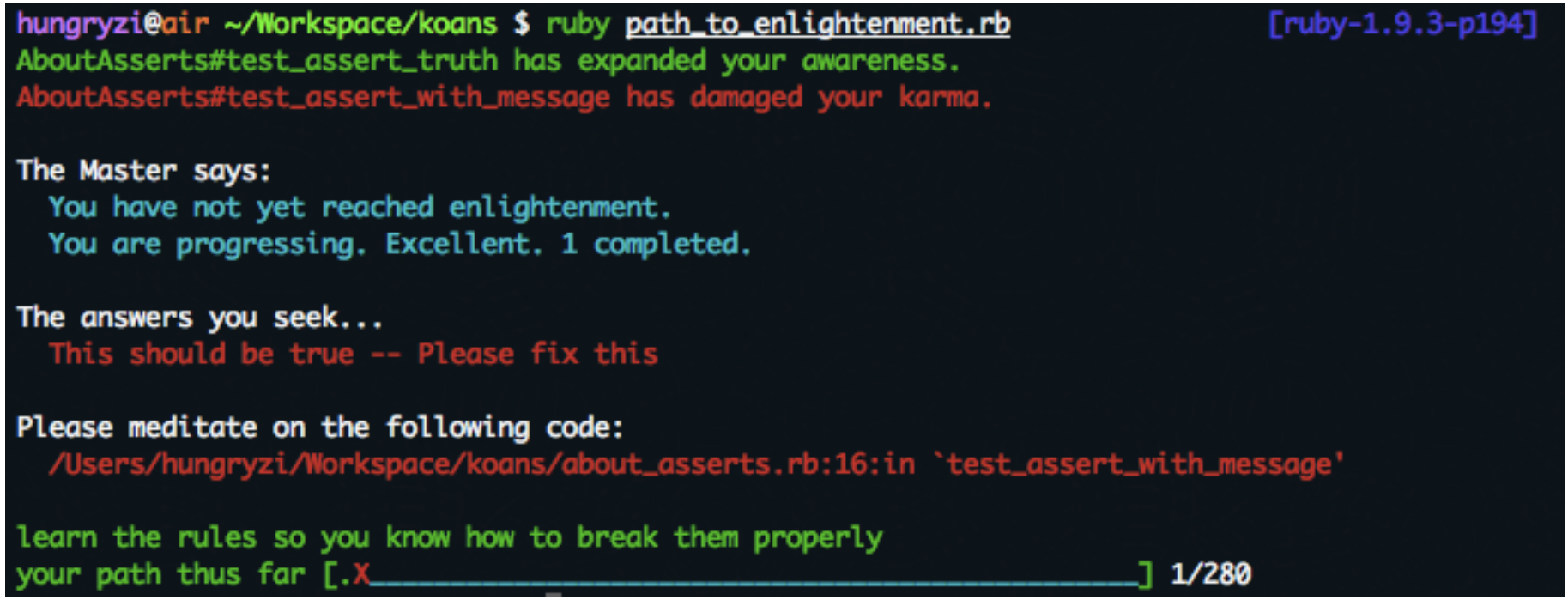
Path to Enlightenment



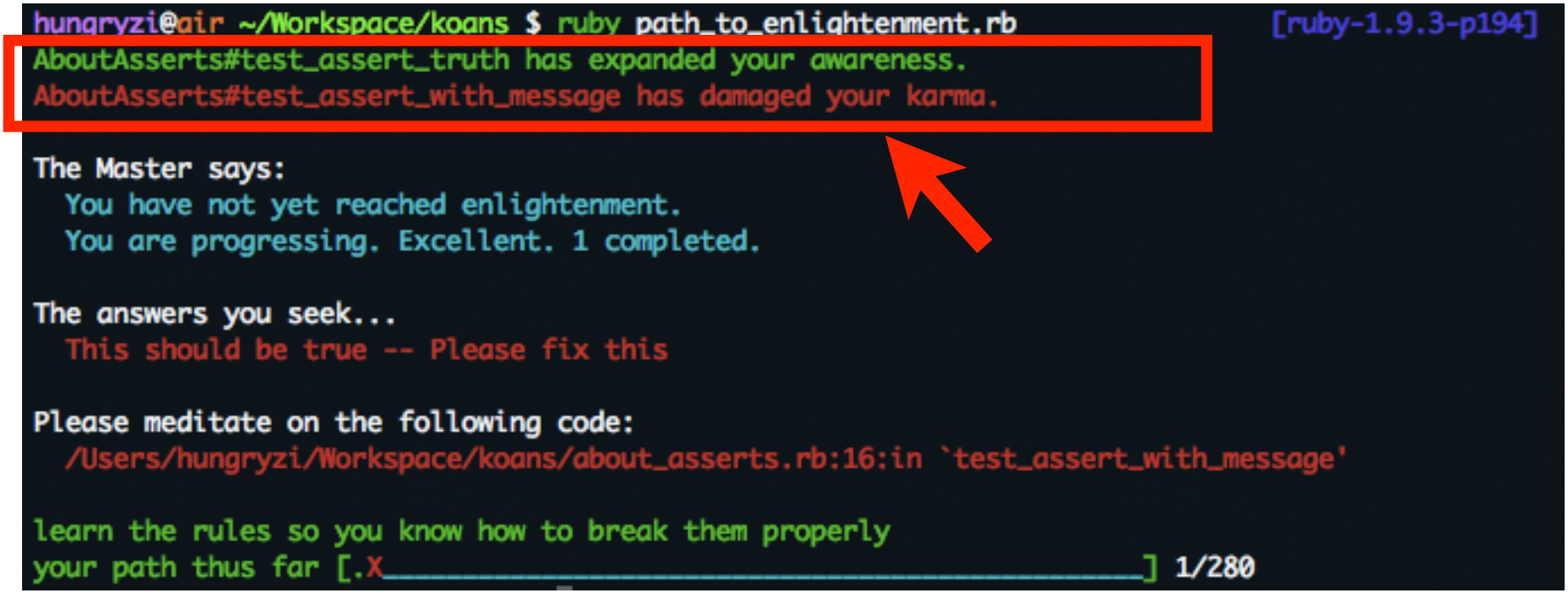
Path to Enlightenment



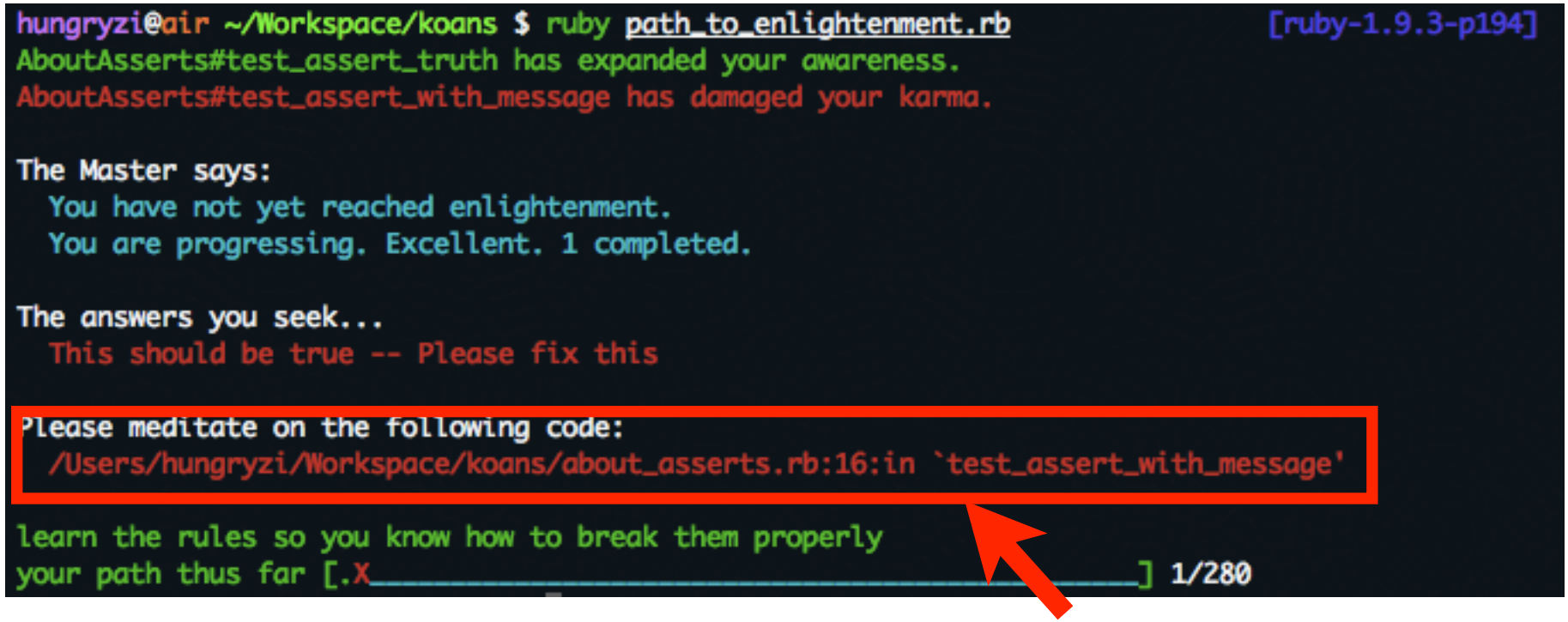
Path to Enlightenment



Path to Enlightenment



Path to Enlightenment



We are here to **help**!