```
puts 'iamhere'

# the subtle art of debugging
```

Andy Tiefenthaler

@pxlpnk

Setting expectations

- Whats a REPL? && REPL Driven Development
- Basics of debugging with PRY.
- Advanced wizardry with PRY.
- Everyone gets a REPL.
- Some cats, fictional charts and lisp.

What is a REPL

WENGER

Read

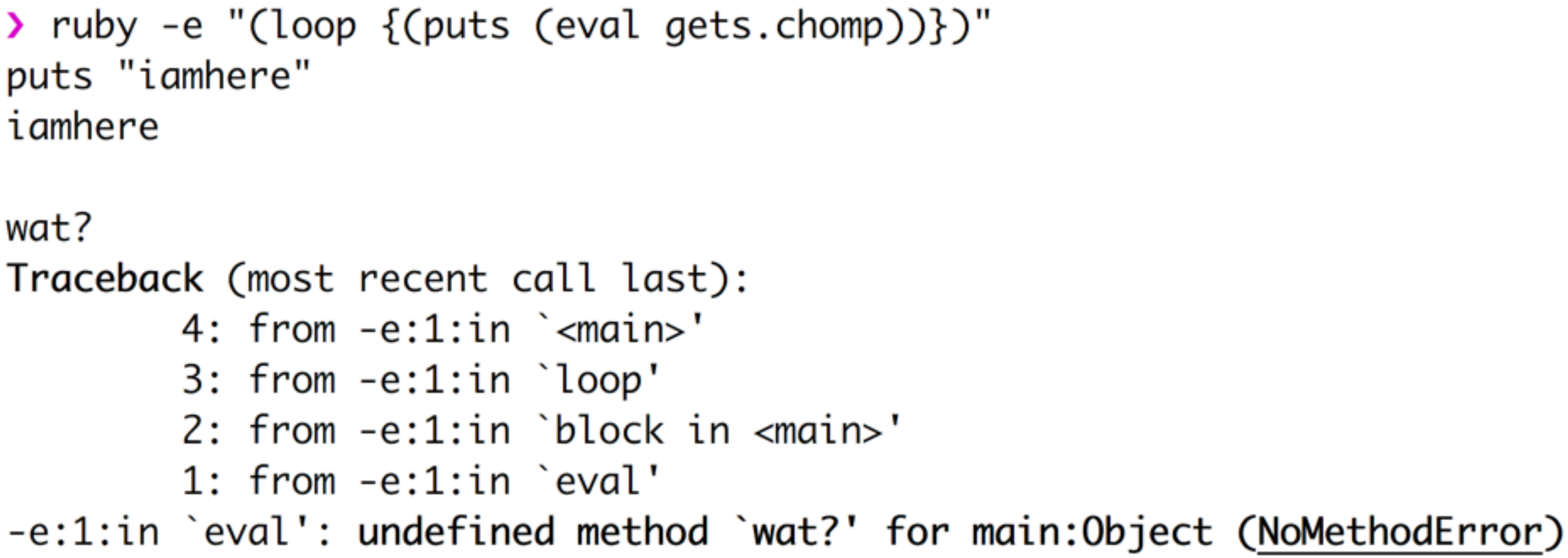
Eval

Print

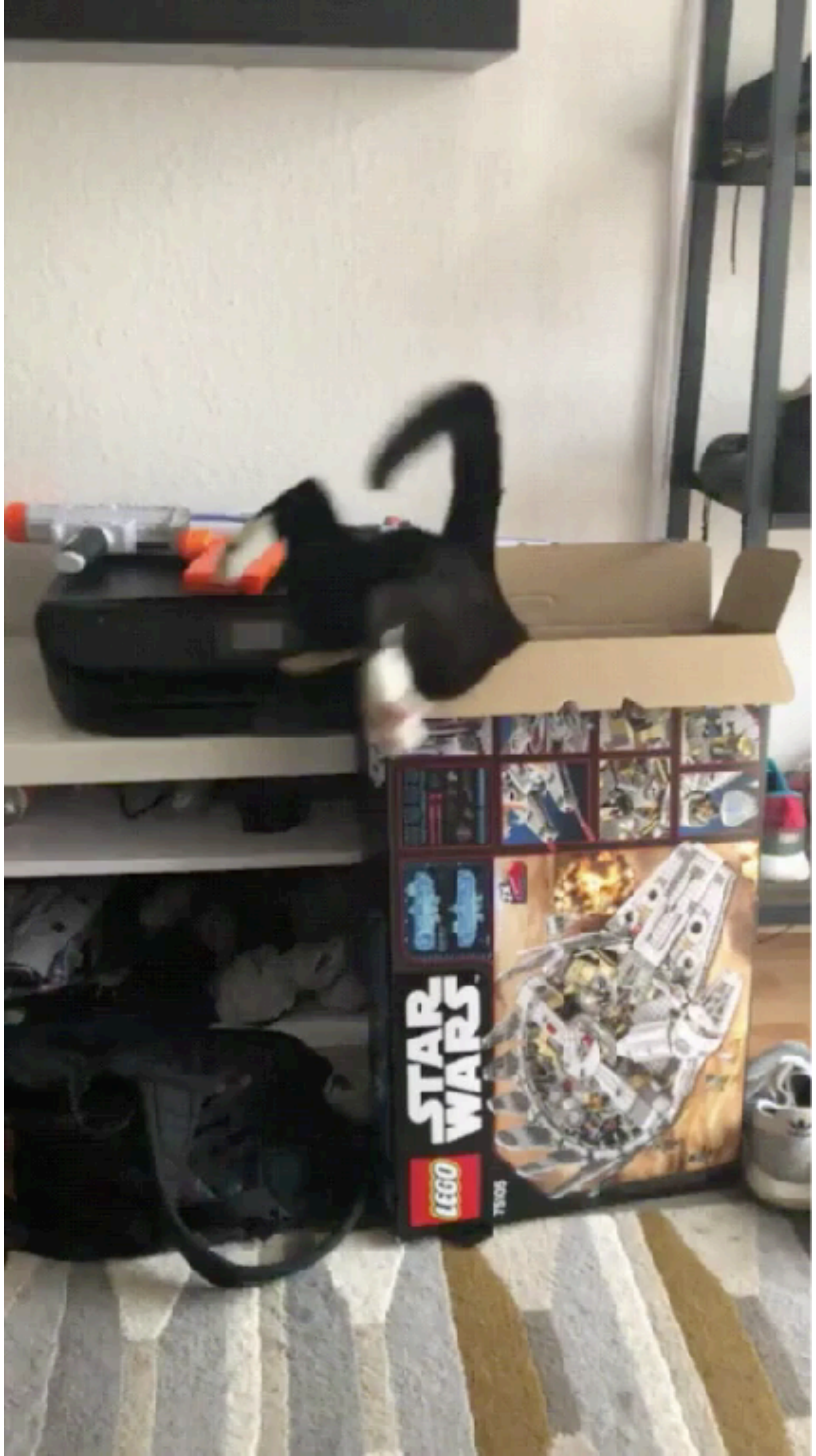
Loop

```
(loop (print (eval (read))))
```

```
(loop {(puts (eval gets.chomp))})
```



```
❯ ruby -e "(loop {(puts (eval gets.chomp))})"
puts "iamhere"
iamhere

wat?
Traceback (most recent call last):
        4: from -e:1:in `<main>'
        3: from -e:1:in `loop'
        2: from -e:1:in `block in <main>'
        1: from -e:1:in `eval'
-e:1:in `eval': undefined method `wat?' for main:Object (NoMethodError)
```


STAR WARS
LEGO
75105

Ruby’s REPLs

- IRB
- PRY
- some more

IRB

- interactive Ruby
- .irbrc file is just Ruby
- add your own helpers & customise the shell
- helpful: http://irb.tools/
- comes with Ruby

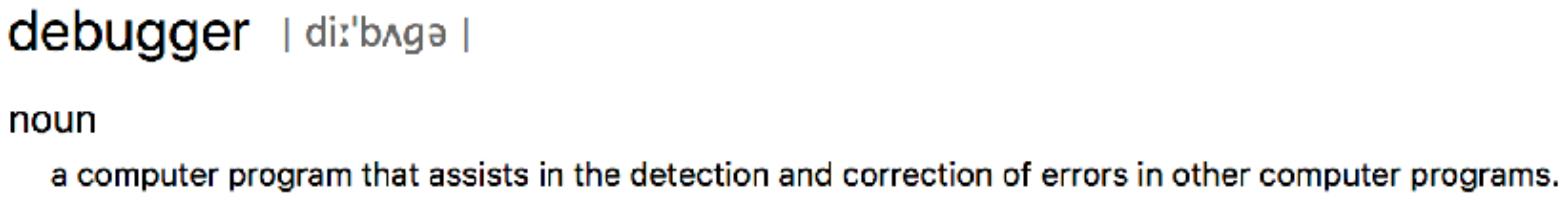
Pry

- http://pryrepl.org/ & on GitHub
- has very good documentation
- Lots of extensions
- Ruby

REPL Driven Development (RDD)

- Use an interactive shell to navigate the code and build understanding
- Explore as much as you can and is relevant for your needs
- quickly change something and see the effects.
- Fast feedback cycle, rapid prototyping and learning

Debugger



debugger | diːˈbʌɡə |

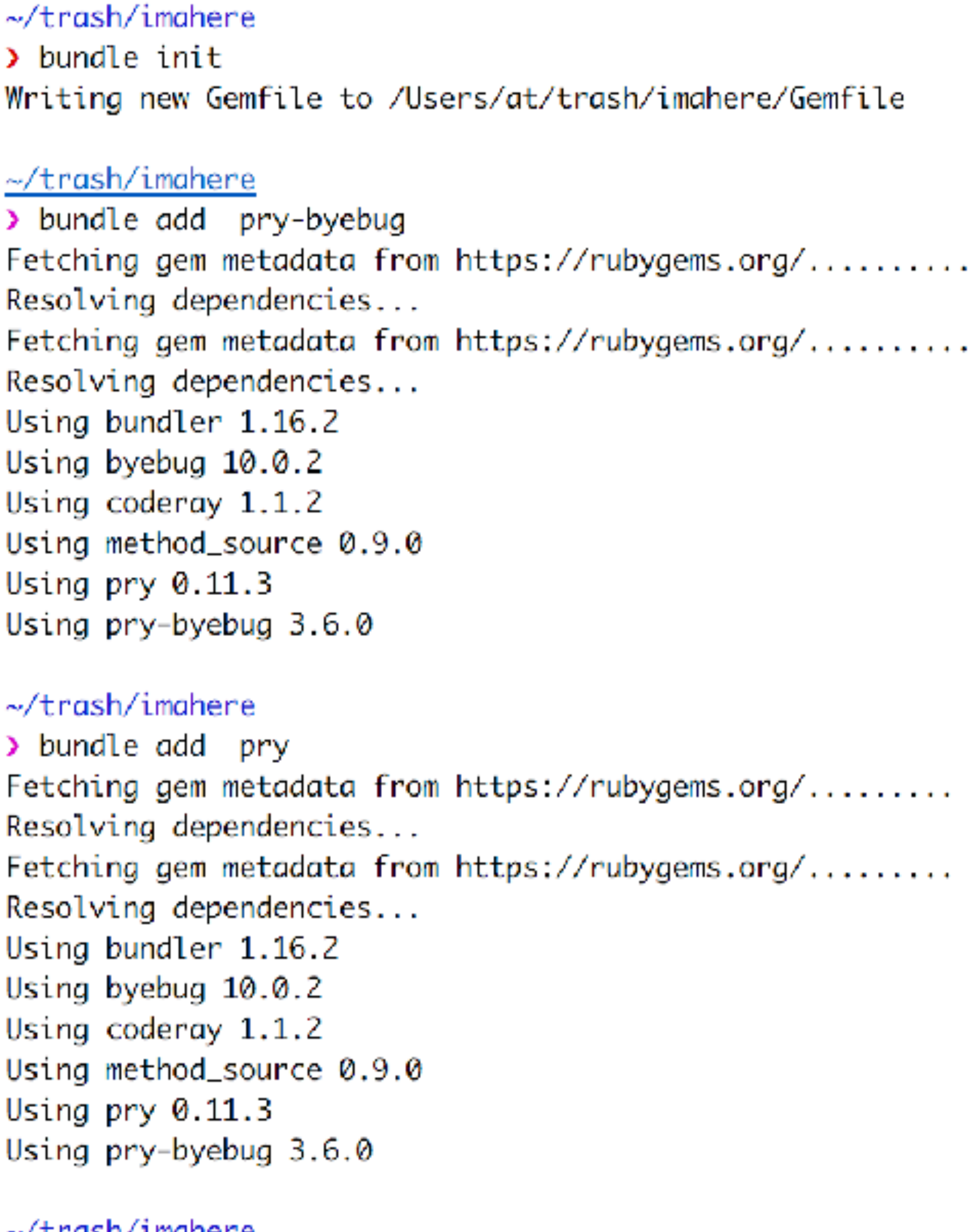
noun

a computer program that assists in the detection and correction of errors in other computer programs.

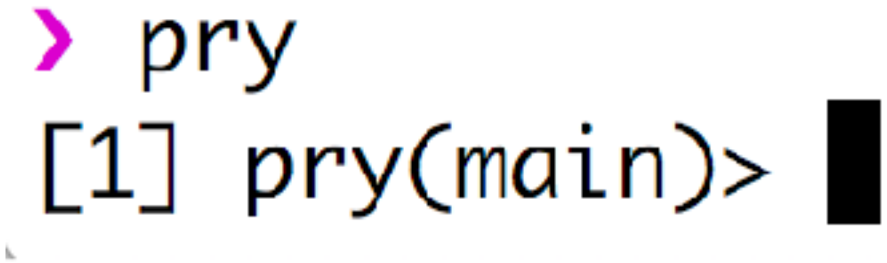
Ruby built in debugger: ruby -r debug

But we will be using byebug with pry.

basics

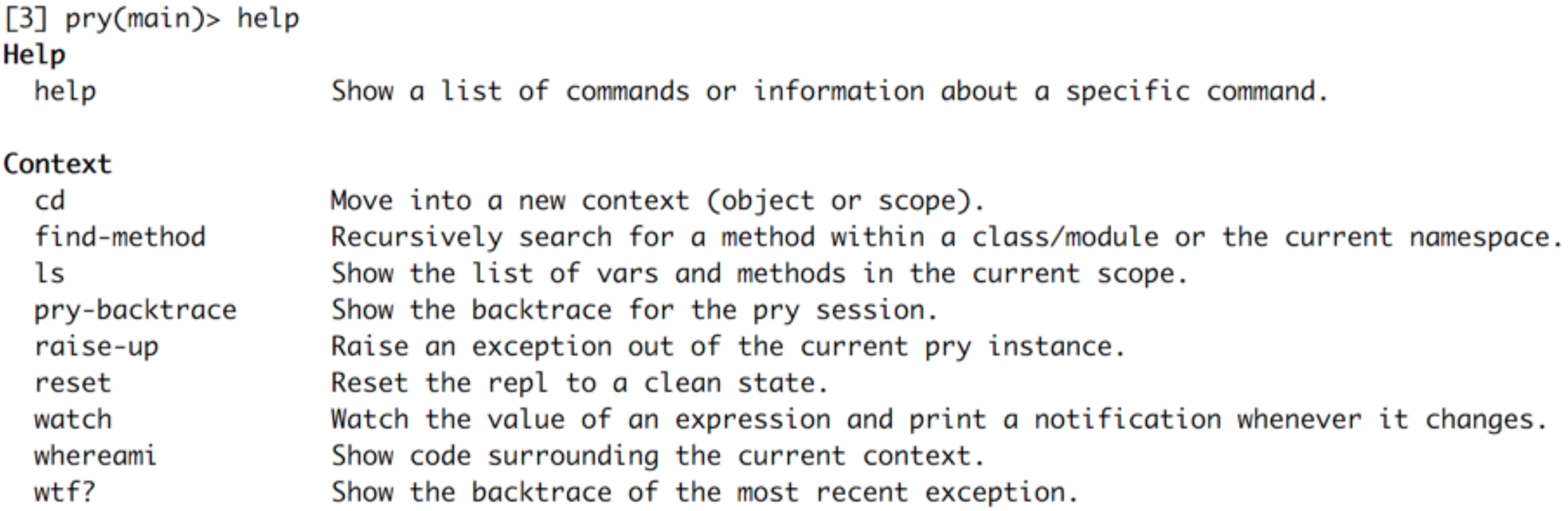


```
~/trash/imahere
❯ bundle init
Writing new Gemfile to /Users/at/trash/imahere/Gemfile

~/trash/imahere
❯ bundle add  pry-byebug
Fetching gem metadata from https://rubygems.org/.........
Resolving dependencies...
Fetching gem metadata from https://rubygems.org/.........
Resolving dependencies...
Using bundler 1.16.2
Using byebug 10.0.2
Using coderay 1.1.2
Using method_source 0.9.0
Using pry 0.11.3
Using pry-byebug 3.6.0

~/trash/imahere
❯ bundle add  pry
Fetching gem metadata from https://rubygems.org/........
Resolving dependencies...
Fetching gem metadata from https://rubygems.org/........
Resolving dependencies...
Using bundler 1.16.2
Using byebug 10.0.2
Using coderay 1.1.2
Using method_source 0.9.0
Using pry 0.11.3
Using pry-byebug 3.6.0

~/trash/imahere
```



```
❯ pry
[1] pry(main)>
```

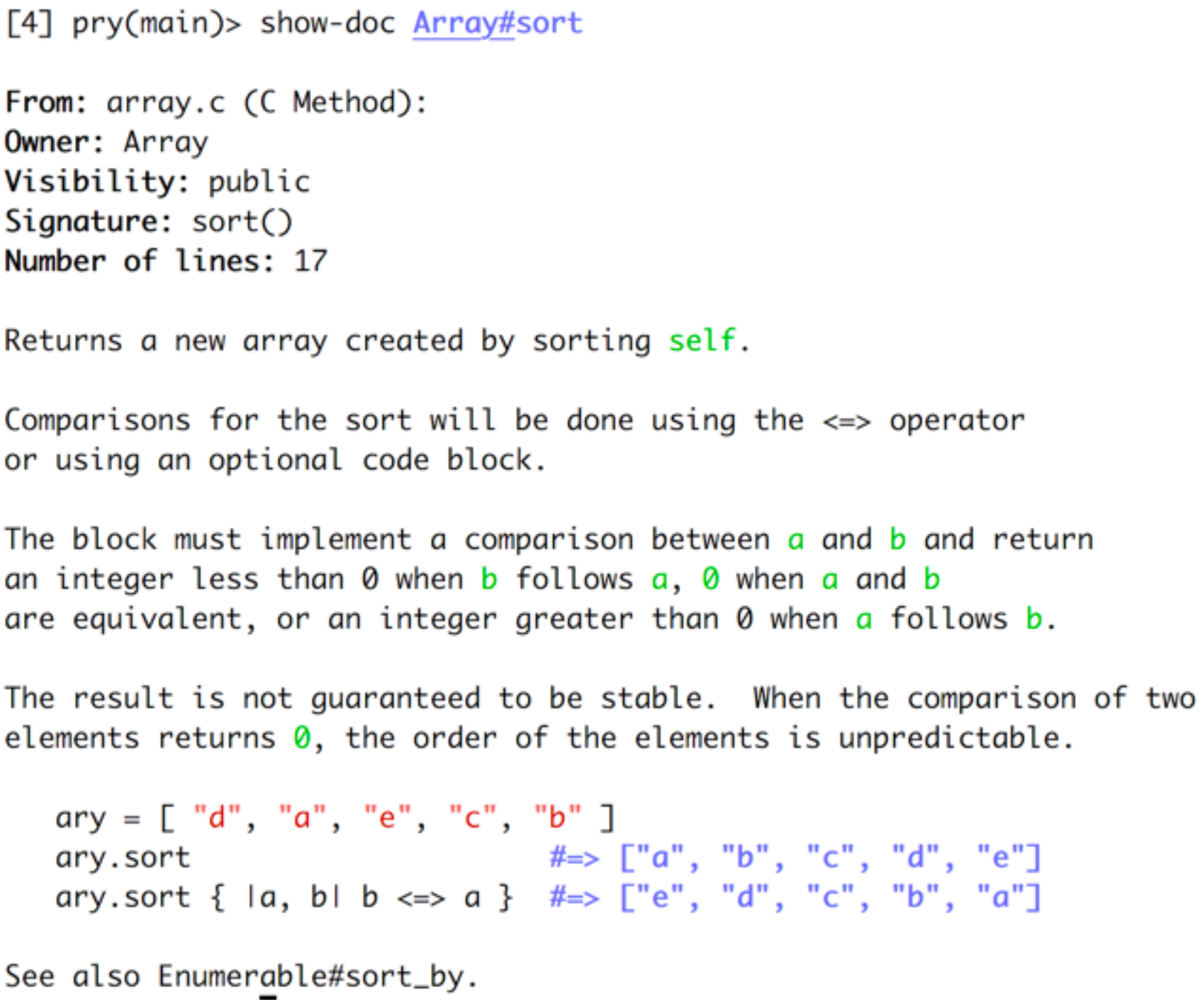
self documented



```
[3] pry(main)> help
Help
  help               Show a list of commands or information about a specific command.

Context
  cd                 Move into a new context (object or scope).
  find-method        Recursively search for a method within a class/module or the current namespace.
  ls                 Show the list of vars and methods in the current scope.
  pry-backtrace      Show the backtrace for the pry session.
  raise-up           Raise an exception out of the current pry instance.
  reset              Reset the repl to a clean state.
  watch              Watch the value of an expression and print a notification whenever it changes.
  whereami           Show code surrounding the current context.
  wtf?               Show the backtrace of the most recent exception.
```

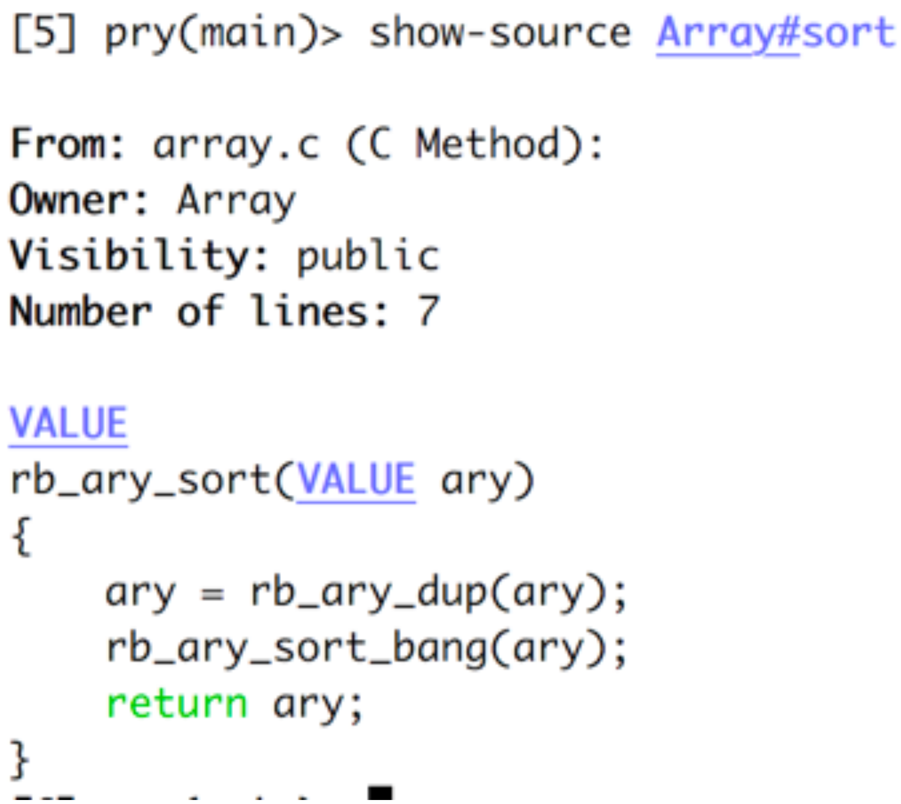
.
.
.

comes with ruby docs



```
[4] pry(main)> show-doc Array#sort

From: array.c (C Method):
Owner: Array
Visibility: public
Signature: sort()
Number of lines: 17

Returns a new array created by sorting self.

Comparisons for the sort will be done using the <=> operator
or using an optional code block.

The block must implement a comparison between a and b and return
an integer less than 0 when b follows a, 0 when a and b
are equivalent, or an integer greater than 0 when a follows b.

The result is not guaranteed to be stable.  When the comparison of two
elements returns 0, the order of the elements is unpredictable.

   ary = [ "d", "a", "e", "c", "b" ]
   ary.sort                     #=> ["a", "b", "c", "d", "e"]
   ary.sort { |a, b| b <=> a }  #=> ["e", "d", "c", "b", "a"]

See also Enumerable#sort_by.
```

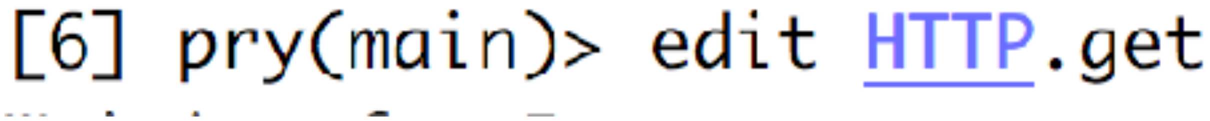
comes with source view



```
[5] pry(main)> show-source Array#sort

From: array.c (C Method):
Owner: Array
Visibility: public
Number of lines: 7

VALUE
rb_ary_sort(VALUE ary)
{
    ary = rb_ary_dup(ary);
    rb_ary_sort_bang(ary);
    return ary;
}
```

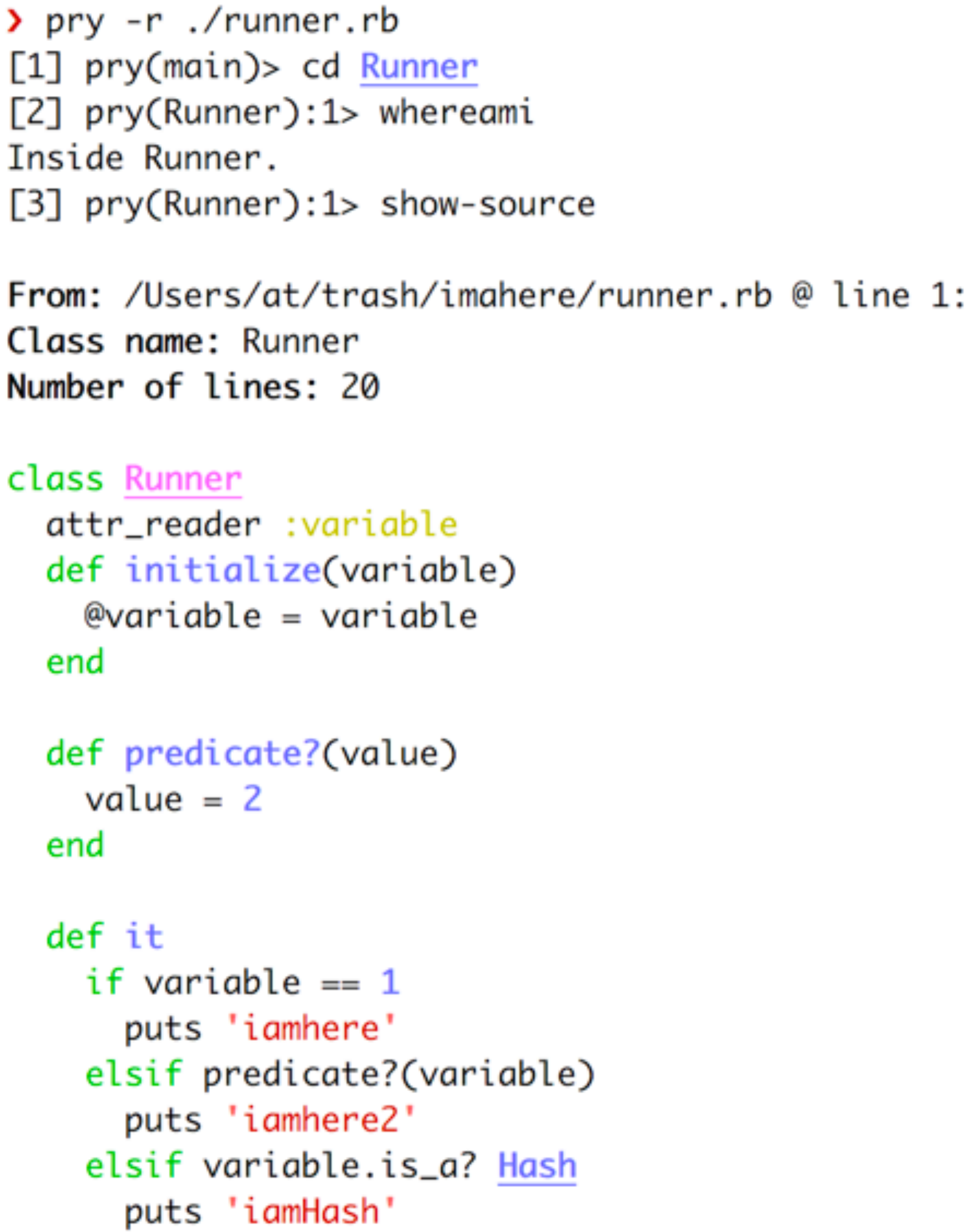
edit all code

(except binaries and compiled stuff)



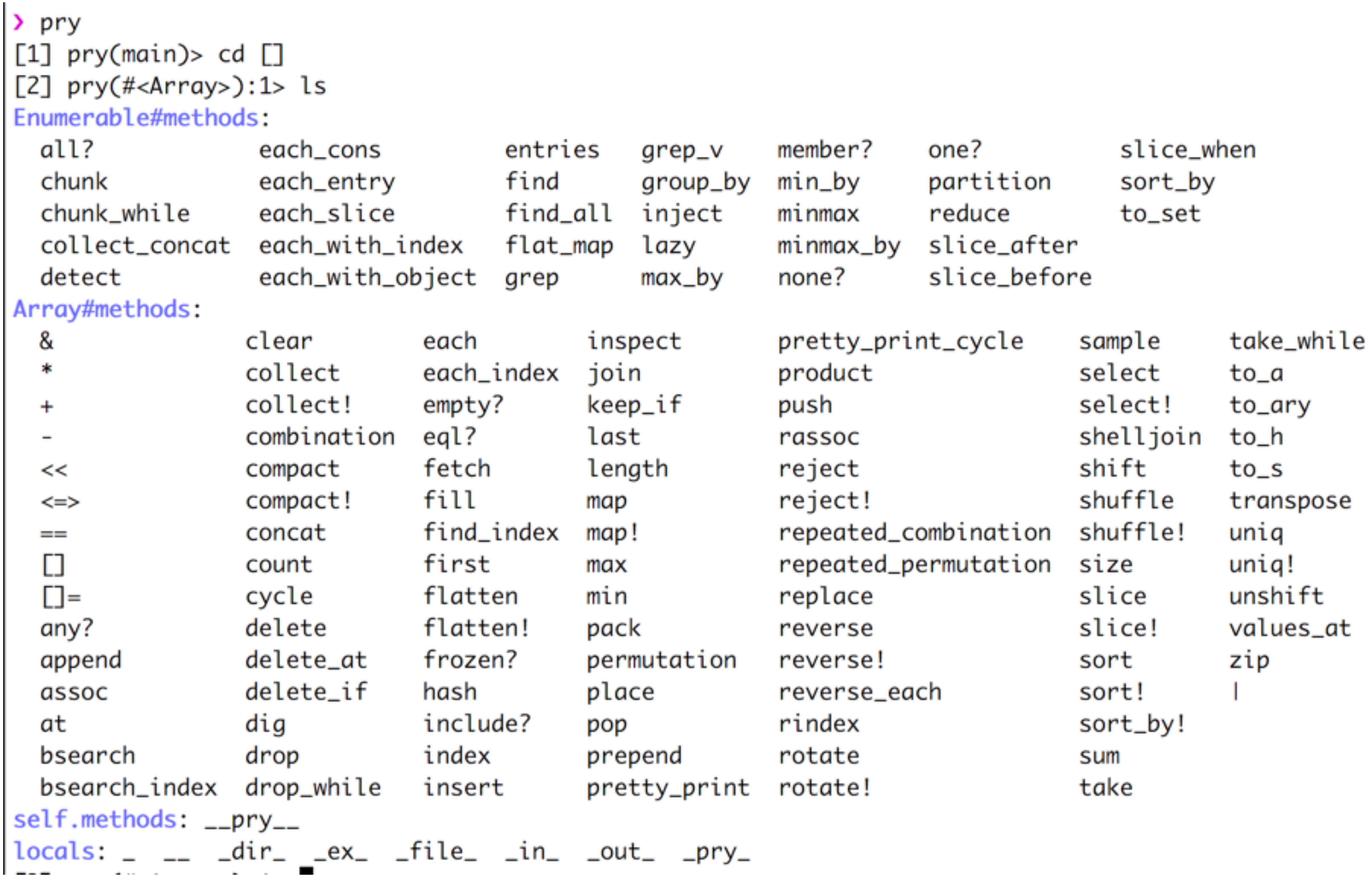
```
[6] pry(main)> edit HTTP.get
```

navigate through code



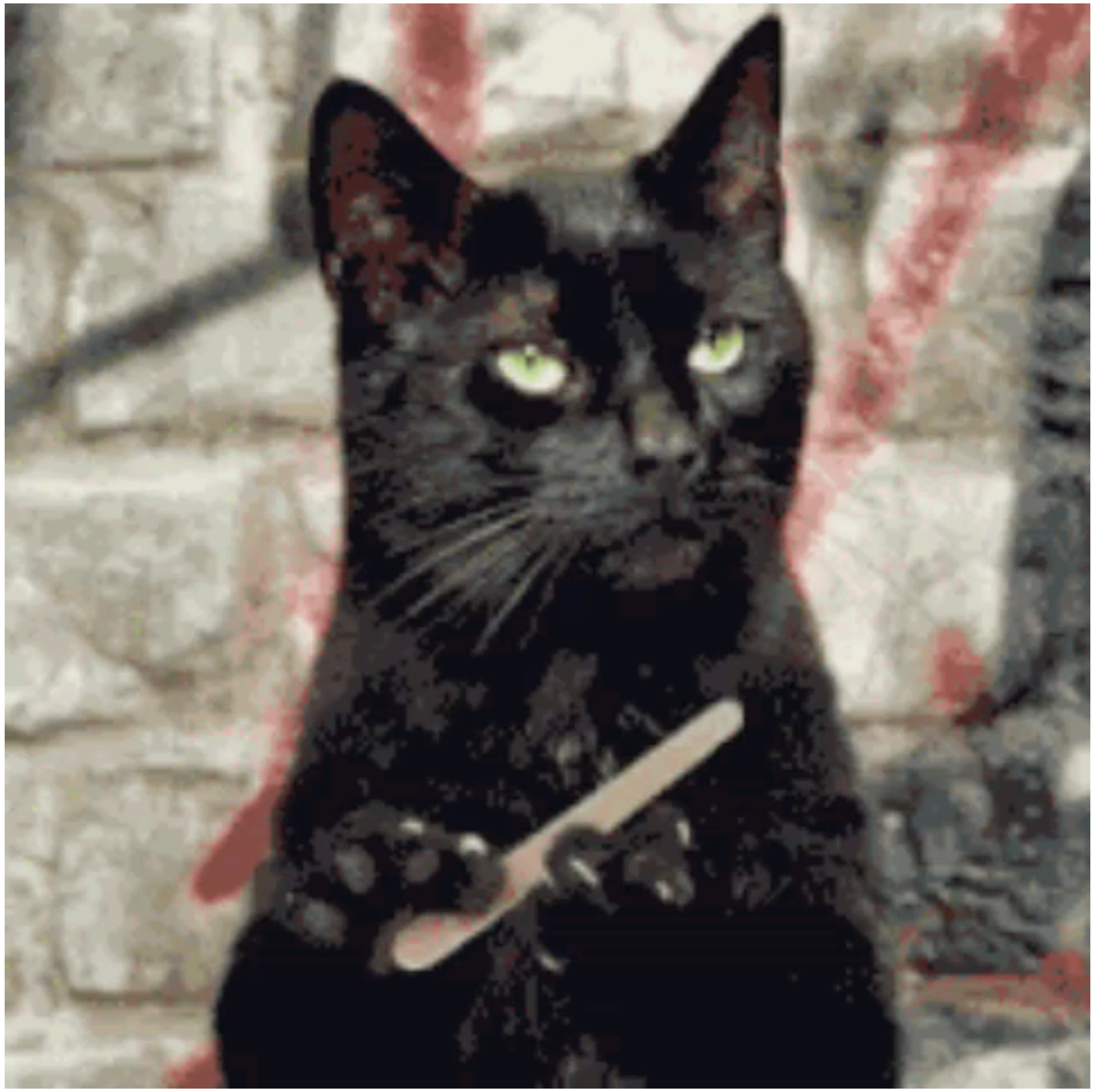
```
❯ pry -r ./runner.rb
[1] pry(main)> cd Runner
[2] pry(Runner):1> whereami
Inside Runner.
[3] pry(Runner):1> show-source

From: /Users/at/trash/imahere/runner.rb @ line 1:
Class name: Runner
Number of lines: 20

class Runner
  attr_reader :variable
  def initialize(variable)
    @variable = variable
  end

  def predicate?(value)
    value = 2
  end

  def it
    if variable == 1
      puts 'iamhere'
    elsif predicate?(variable)
      puts 'iamhere2'
    elsif variable.is_a? Hash
      puts 'iamHash'
```

find methods and variables



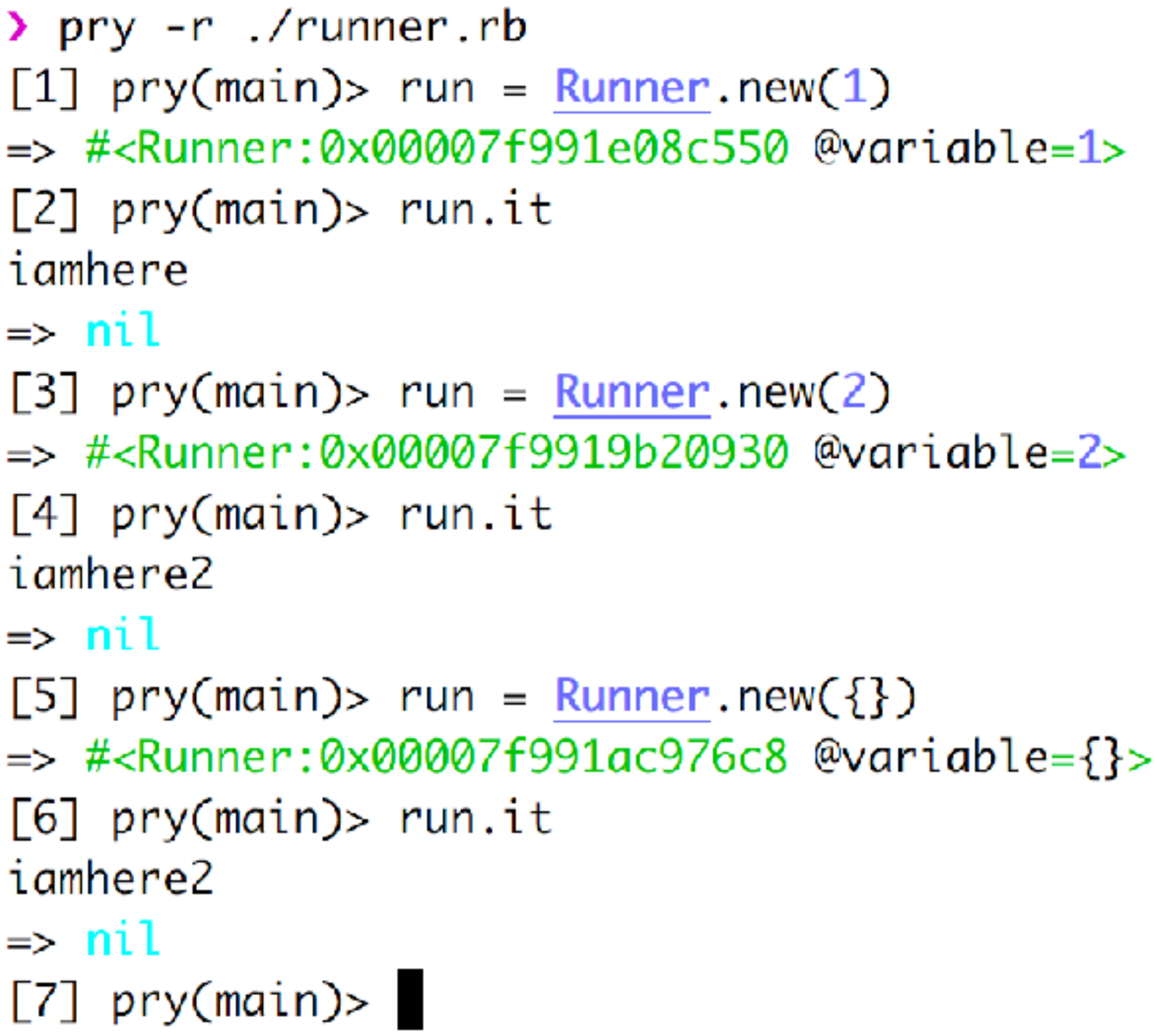
```
❯ pry
[1] pry(main)> cd []
[2] pry(#<Array>):1> ls
Enumerable#methods:
  all?            each_cons         entries   grep_v    member?    one?          slice_when
  chunk           each_entry        find      group_by  min_by     partition     sort_by
  chunk_while     each_slice        find_all  inject    minmax     reduce        to_set
  collect_concat  each_with_index   flat_map  lazy      minmax_by  slice_after
  detect          each_with_object  grep      max_by    none?      slice_before
Array#methods:
  &              clear        each        inspect       pretty_print_cycle    sample     take_while
  *              collect      each_index  join          product               select     to_a
  +              collect!     empty?      keep_if       push                  select!    to_ary
  -              combination  eql?        last          rassoc                shelljoin  to_h
  <<             compact      fetch       length        reject                shift      to_s
  <=>            compact!     fill        map           reject!               shuffle    transpose
  ==             concat       find_index  map!          repeated_combination  shuffle!   uniq
  []             count        first       max           repeated_permutation  size       uniq!
  []=            cycle        flatten     min           replace               slice      unshift
  any?           delete       flatten!    pack          reverse               slice!     values_at
  append         delete_at    frozen?     permutation   reverse!              sort       zip
  assoc          delete_if    hash        place         reverse_each          sort!      |
  at             dig          include?    pop           rindex                sort_by!
  bsearch        drop         index       prepend       rotate                sum
  bsearch_index  drop_while   insert      pretty_print  rotate!               take
self.methods: __pry__
locals: _  __  _dir_  _ex_  _file_  _in_  _out_  _pry_
```

How does this help me?



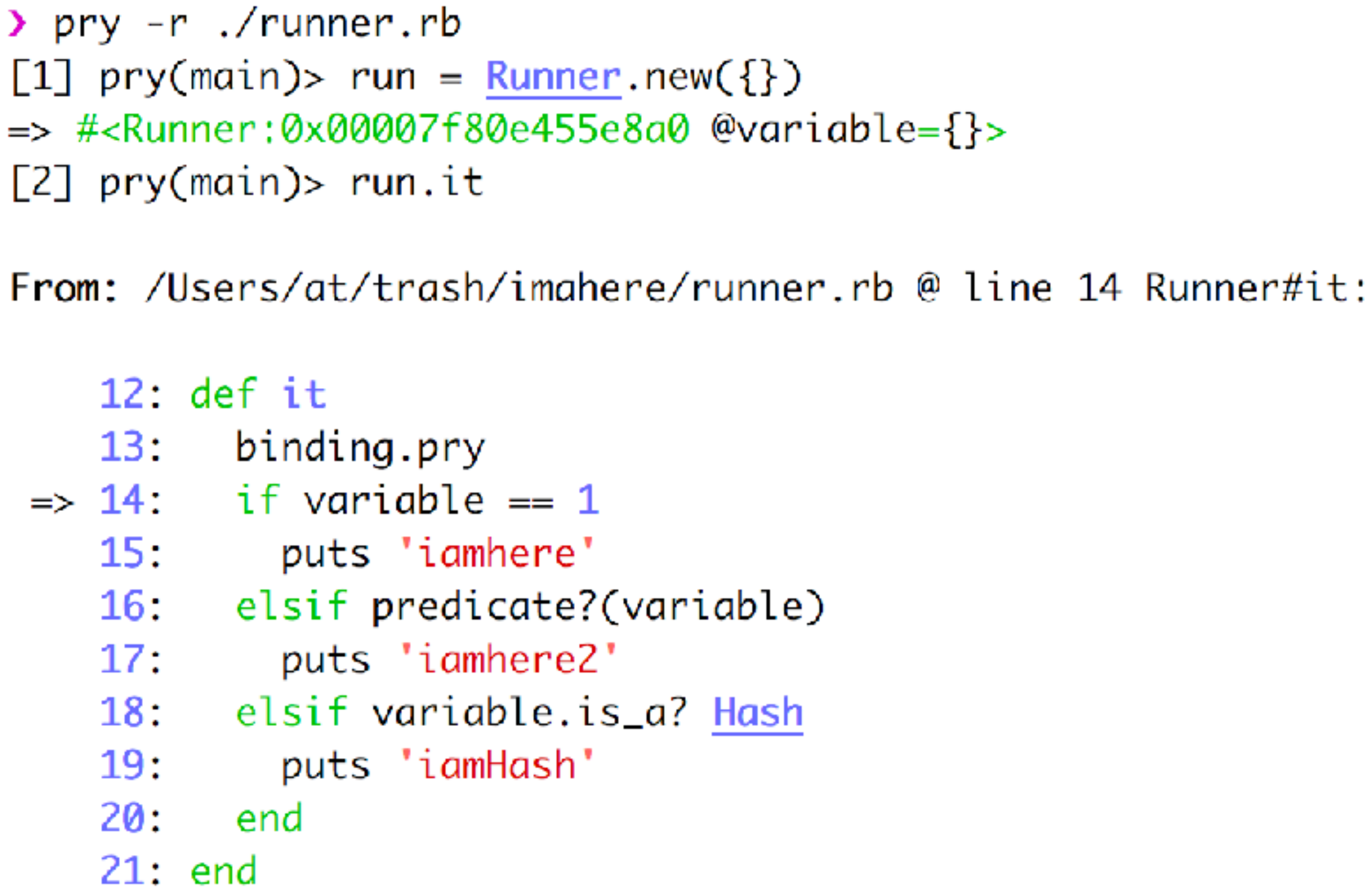
Huston we have a bug

```ruby
class Runner
  attr_reader :variable
  def initialize(variable)
    @variable = variable
  end

  def predicate?(value)
    variable = 2
  end

  def it
    if variable == 1
      #some code
      puts 'iamhere'
    elsif predicate?(variable)
      #some other code
      puts 'iamhere2'
    elsif variable.is_a? Hash
      #some more other code
      puts 'iamHash'
    end
  end
end
```

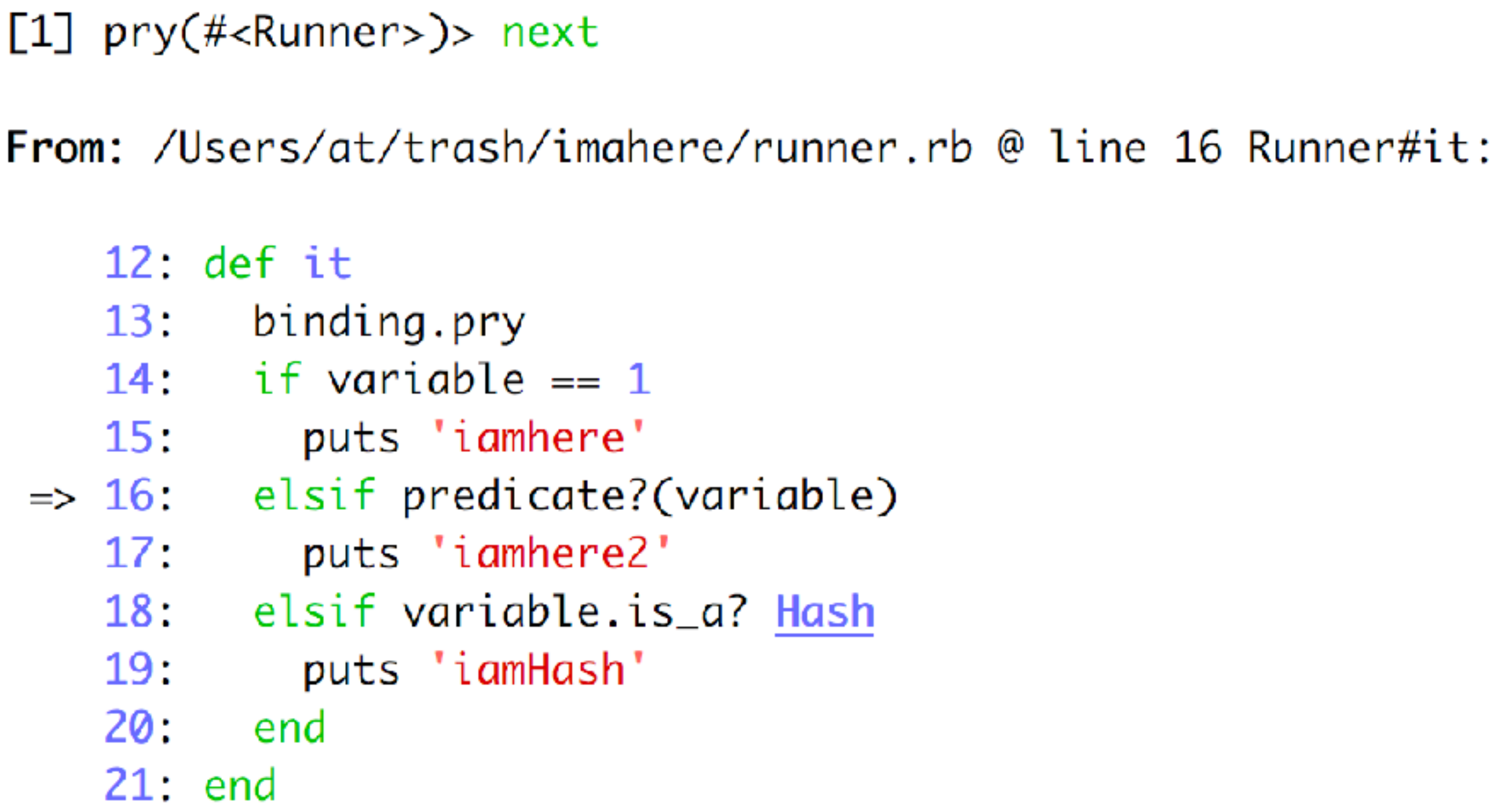


```
❯ pry -r ./runner.rb
[1] pry(main)> run = Runner.new(1)
=> #<Runner:0x00007f991e08c550 @variable=1>
[2] pry(main)> run.it
iamhere
=> nil
[3] pry(main)> run = Runner.new(2)
=> #<Runner:0x00007f9919b20930 @variable=2>
[4] pry(main)> run.it
iamhere2
=> nil
[5] pry(main)> run = Runner.new({})
=> #<Runner:0x00007f991ac976c8 @variable={}>
[6] pry(main)> run.it
iamhere2
=> nil
[7] pry(main)>
```

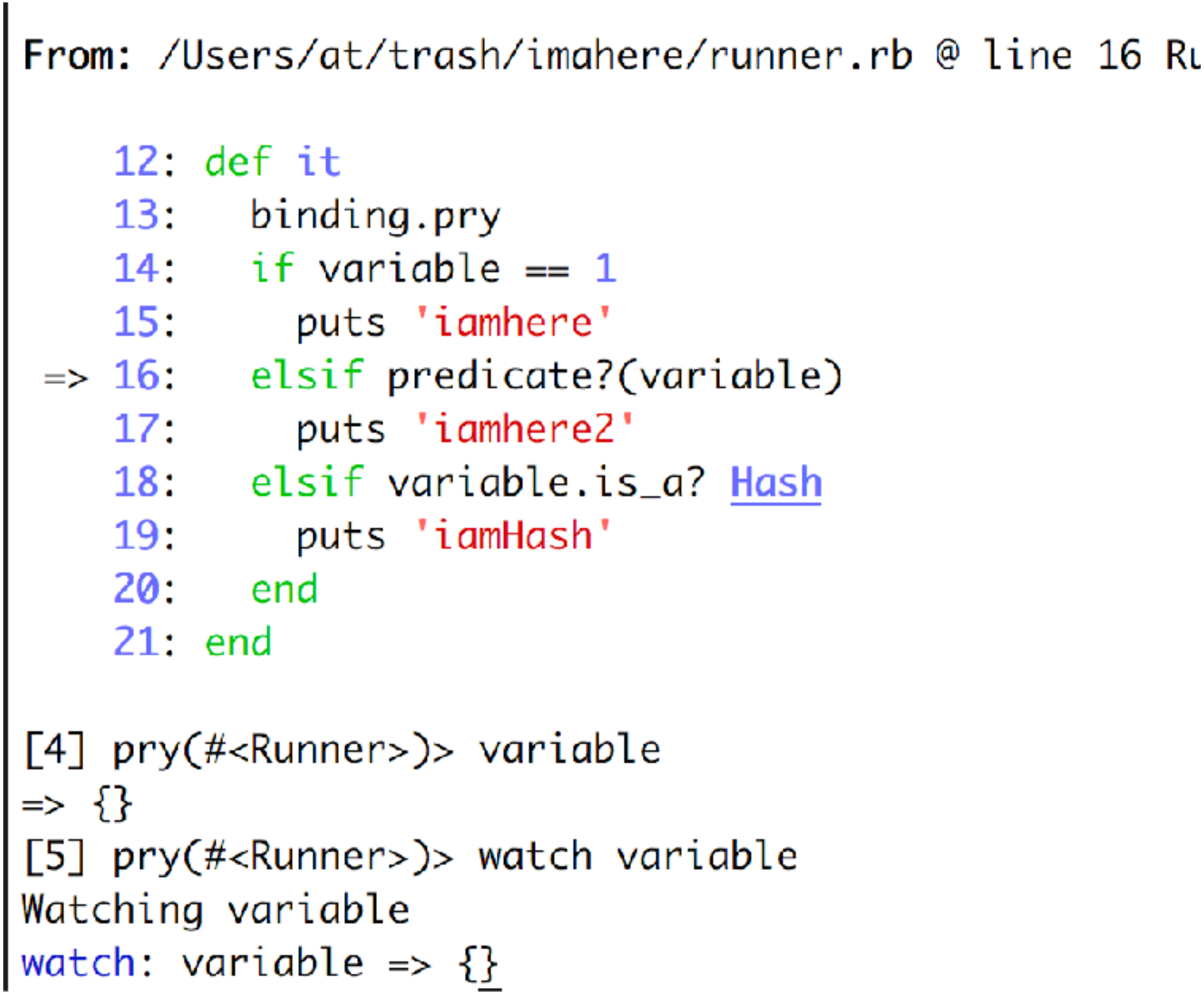
breakpoint



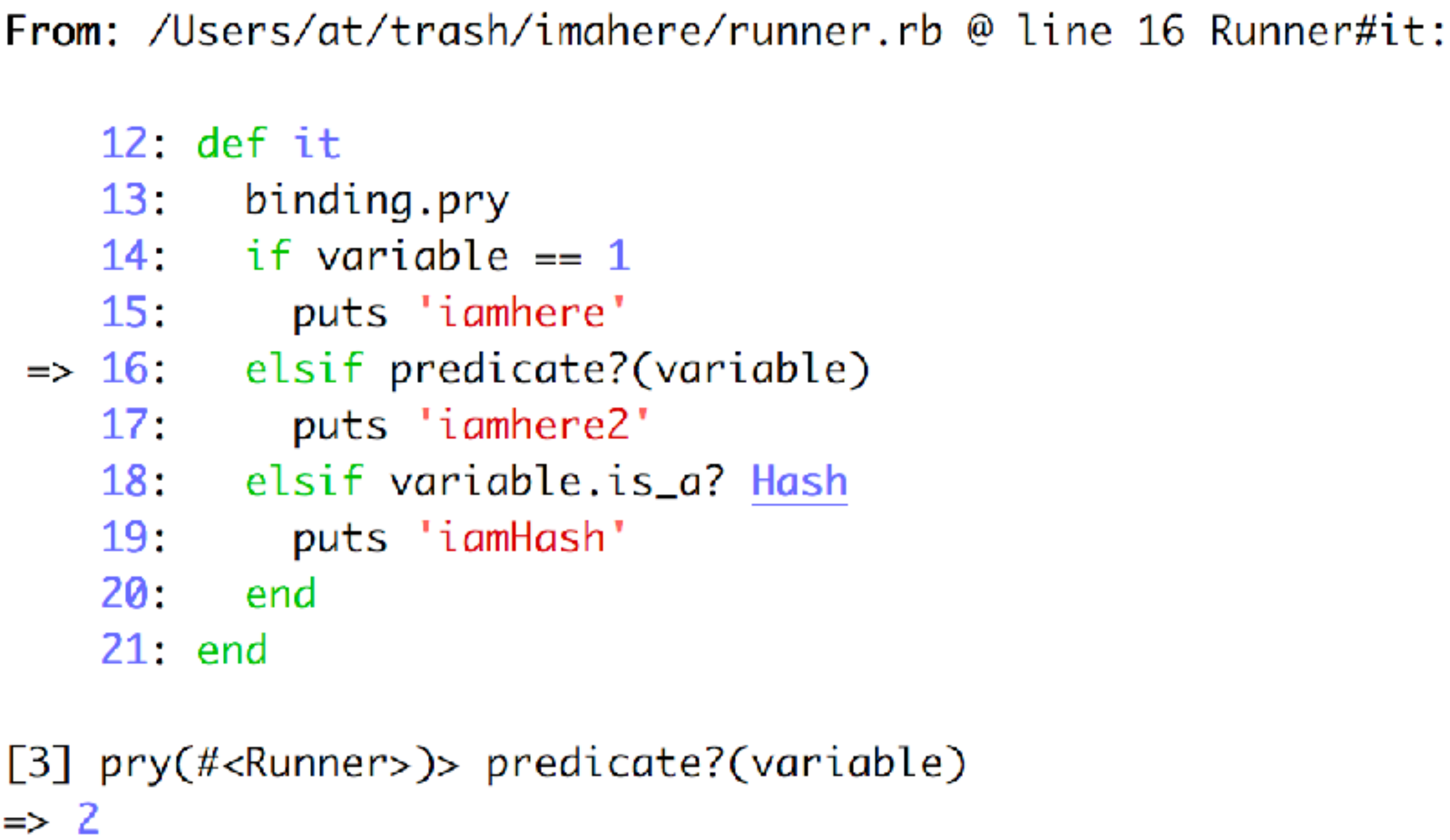
```
❯ pry -r ./runner.rb
[1] pry(main)> run = Runner.new({})
=> #<Runner:0x00007f80e455e8a0 @variable={}>
[2] pry(main)> run.it

From: /Users/at/trash/imahere/runner.rb @ line 14 Runner#it:

    12: def it
    13:   binding.pry
 => 14:   if variable == 1
    15:     puts 'iamhere'
    16:   elsif predicate?(variable)
    17:     puts 'iamhere2'
    18:   elsif variable.is_a? Hash
    19:     puts 'iamHash'
    20:   end
    21: end
```



```
[1] pry(#<Runner>)> next

From: /Users/at/trash/imahere/runner.rb @ line 16 Runner#it:

    12: def it
    13:   binding.pry
    14:   if variable == 1
    15:     puts 'iamhere'
 => 16:   elsif predicate?(variable)
    17:     puts 'iamhere2'
    18:   elsif variable.is_a? Hash
    19:     puts 'iamHash'
    20:   end
    21: end
```

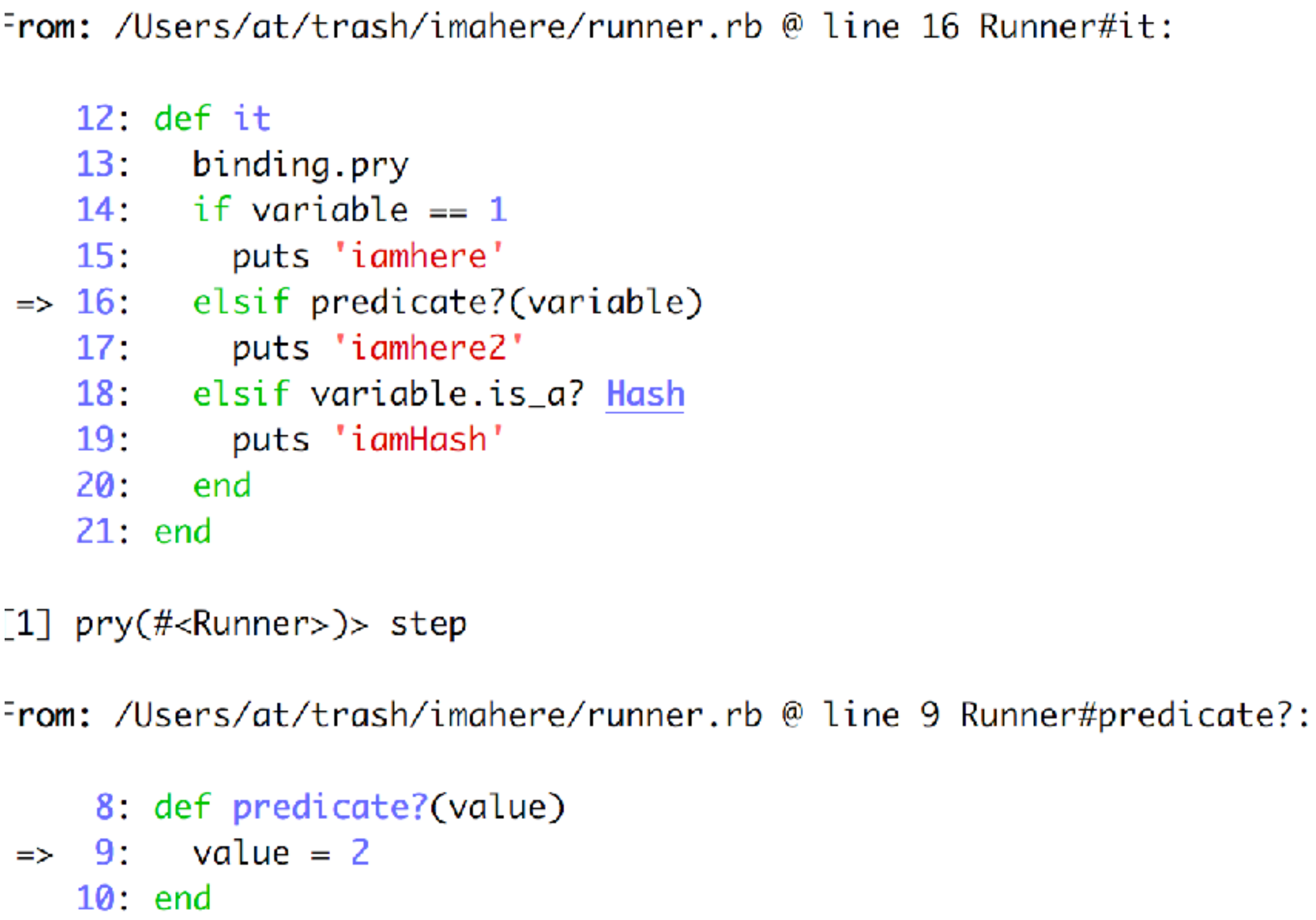


```
From: /Users/at/trash/imahere/runner.rb @ line 16 Ru

    12: def it
    13:   binding.pry
    14:   if variable == 1
    15:     puts 'iamhere'
 => 16:   elsif predicate?(variable)
    17:     puts 'iamhere2'
    18:   elsif variable.is_a? Hash
    19:     puts 'iamHash'
    20:   end
    21: end

[4] pry(#<Runner>)> variable
=> {}
[5] pry(#<Runner>)> watch variable
Watching variable
watch: variable => {}
```

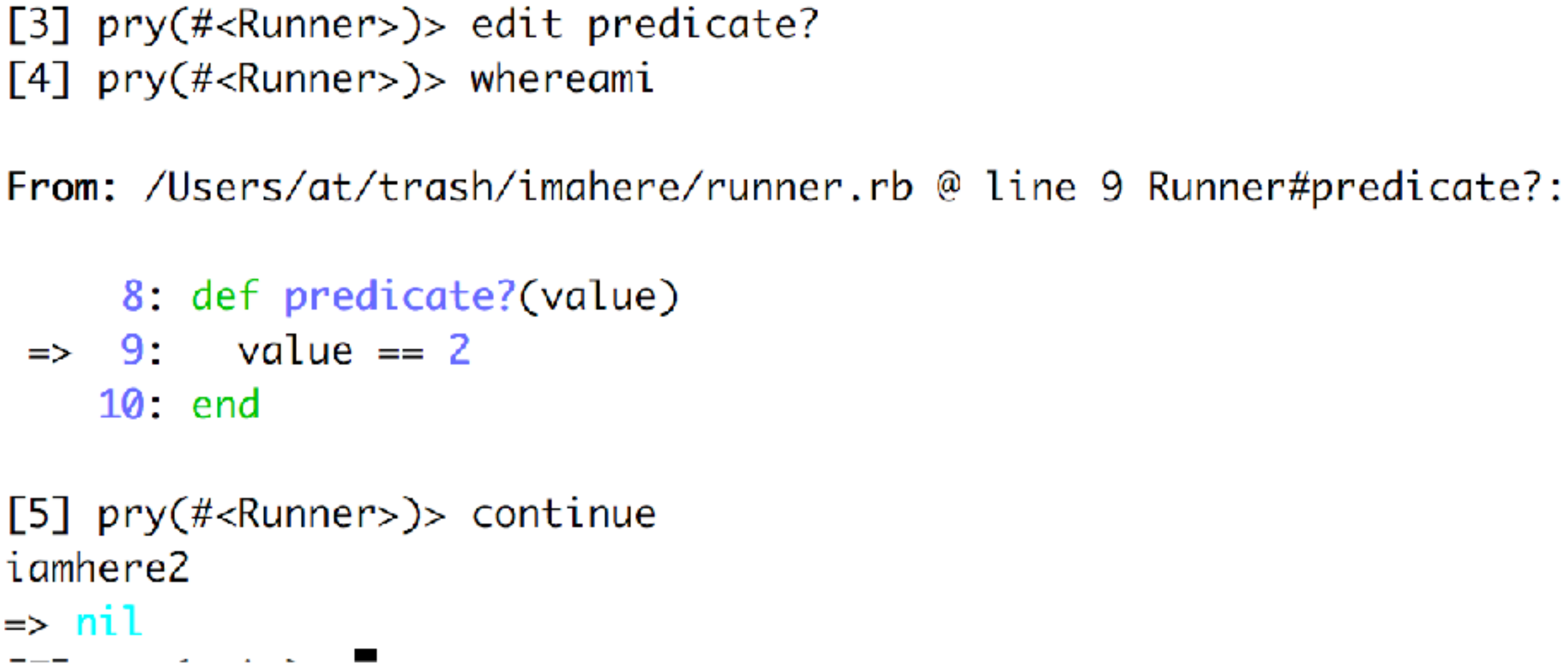


```
From: /Users/at/trash/imahere/runner.rb @ line 16 Runner#it:

    12: def it
    13:   binding.pry
    14:   if variable == 1
    15:     puts 'iamhere'
 => 16:   elsif predicate?(variable)
    17:     puts 'iamhere2'
    18:   elsif variable.is_a? Hash
    19:     puts 'iamHash'
    20:   end
    21: end

[3] pry(#<Runner>)> predicate?(variable)
=> 2
```

???

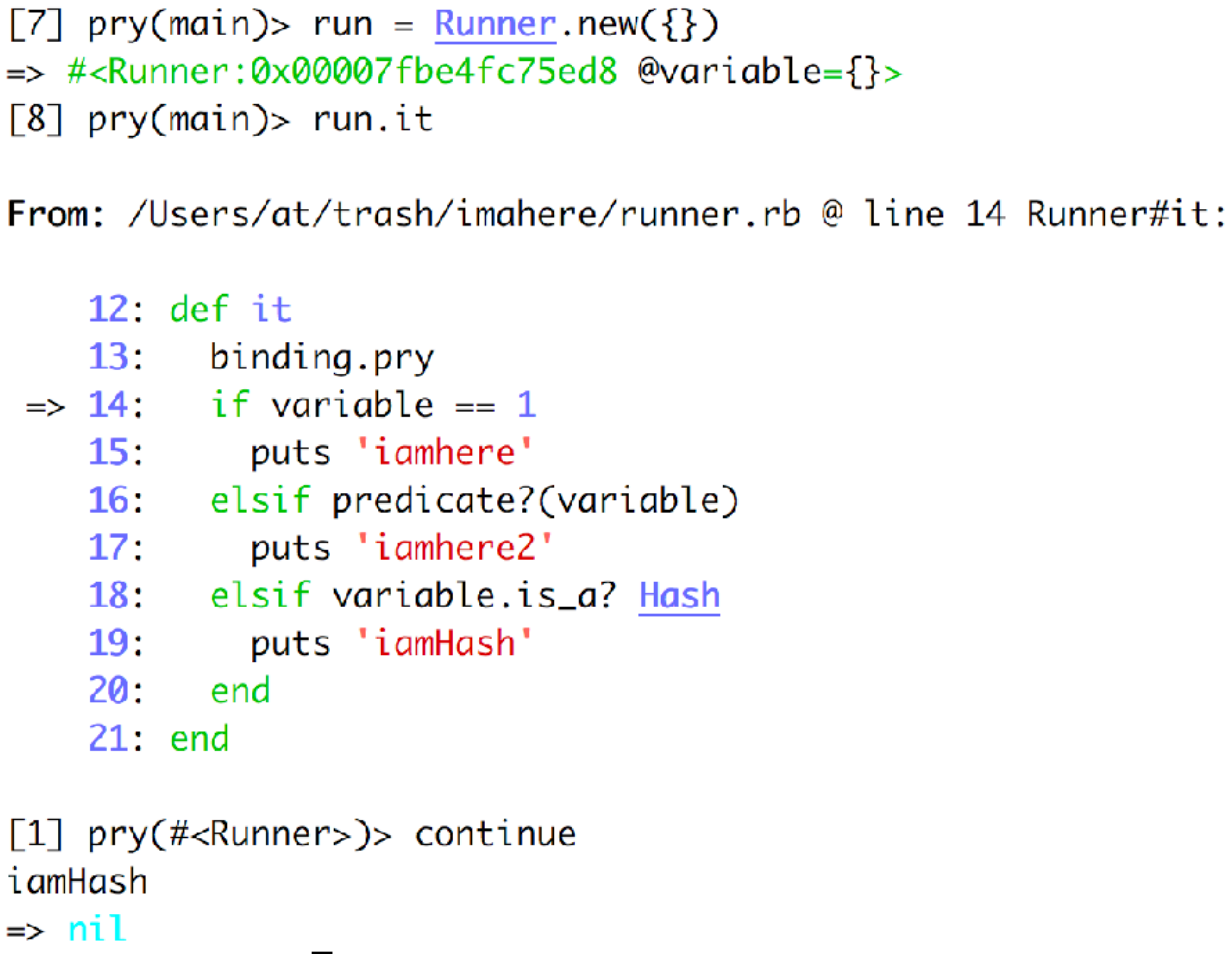


```
From: /Users/at/trash/imahere/runner.rb @ line 16 Runner#it:

    12: def it
    13:   binding.pry
    14:   if variable == 1
    15:     puts 'iamhere'
 => 16:   elsif predicate?(variable)
    17:     puts 'iamhere2'
    18:   elsif variable.is_a? Hash
    19:     puts 'iamHash'
    20:   end
    21: end

[1] pry(#<Runner>)> step

From: /Users/at/trash/imahere/runner.rb @ line 9 Runner#predicate?:

     8: def predicate?(value)
 =>  9:   value = 2
    10: end
```



```
[3] pry(#<Runner>)> edit predicate?
[4] pry(#<Runner>)> whereami

From: /Users/at/trash/imahere/runner.rb @ line 9 Runner#predicate?:

     8: def predicate?(value)
 =>  9:   value == 2
    10: end

[5] pry(#<Runner>)> continue
iamhere2
=> nil
```

It is still failing??? Why???



```
[7] pry(main)> run = Runner.new({})
=> #<Runner:0x00007fbe4fc75ed8 @variable={}>
[8] pry(main)> run.it

From: /Users/at/trash/imahere/runner.rb @ line 14 Runner#it:

    12: def it
    13:   binding.pry
 => 14:   if variable == 1
    15:     puts 'iamhere'
    16:   elsif predicate?(variable)
    17:     puts 'iamhere2'
    18:   elsif variable.is_a? Hash
    19:     puts 'iamHash'
    20:   end
    21: end

[1] pry(#<Runner>)> continue
iamHash
=> nil
```

Advanced Wizardry

commands

Modify the user input

The syntax is more intuitive than building a plugin

They are local to your Pry console and do not require monkey patches

Help to form your workflow

```
Pry::Commands.command /^$/, "repeat last” do
 _pry_.run_command Pry.history.to_a.last
end


Pry.commands.alias_command ‘c', 'continue'
```

Plugins

Allow to extend Pry

Use the power of other libraries and Gems

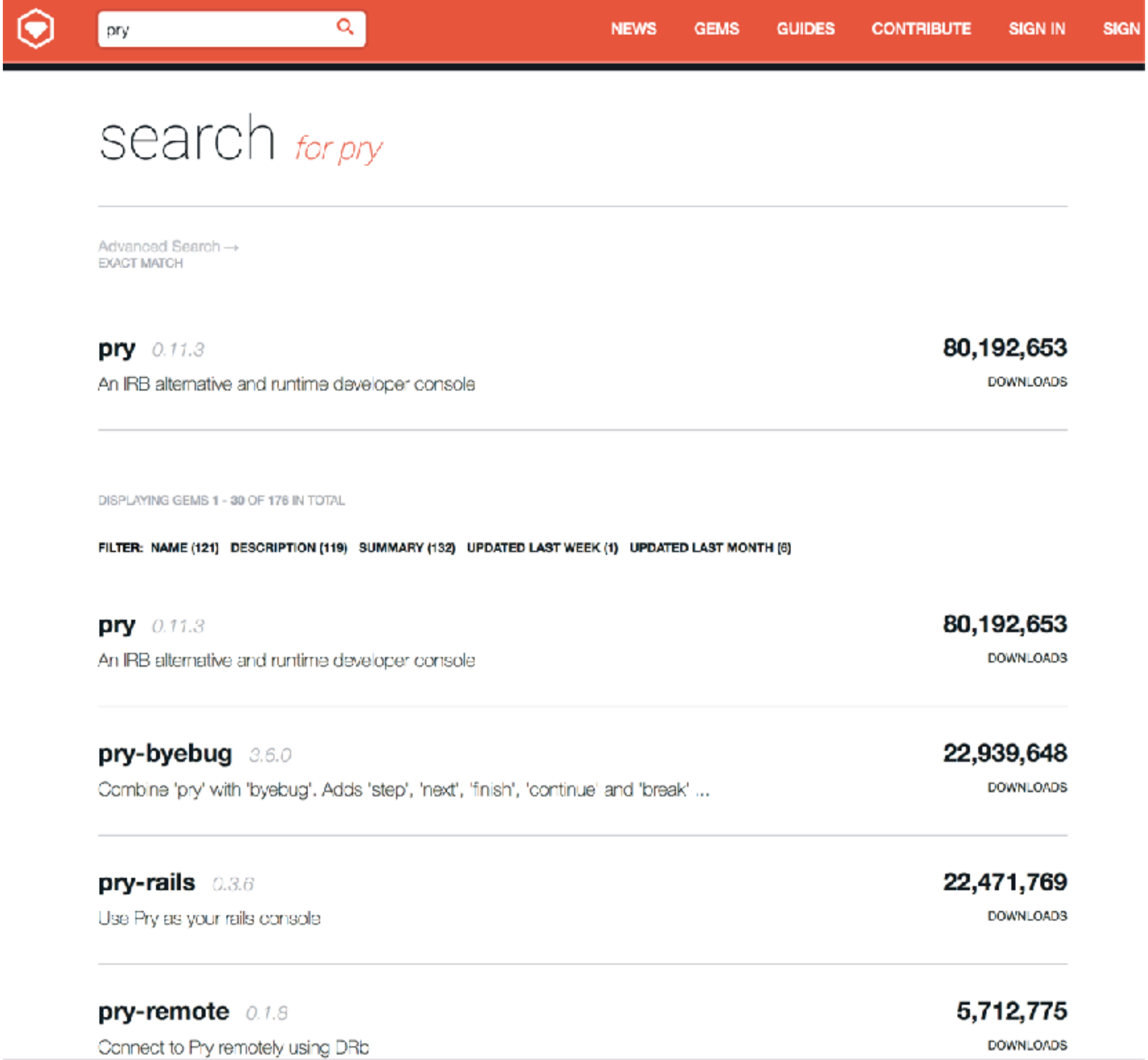
Help to form your workflow

pry-**byebug**

pry-**exception_explorer**

pry-**macro**

pry-**rails**



search *for pry*

Advanced Search →
EXACT MATCH

pry 0.11.3 **80,192,653** DOWNLOADS
An IRB alternative and runtime developer console

DISPLAYING GEMS 1 - 30 OF 176 IN TOTAL

FILTER: NAME (121) DESCRIPTION (119) SUMMARY (132) UPDATED LAST WEEK (1) UPDATED LAST MONTH (6)

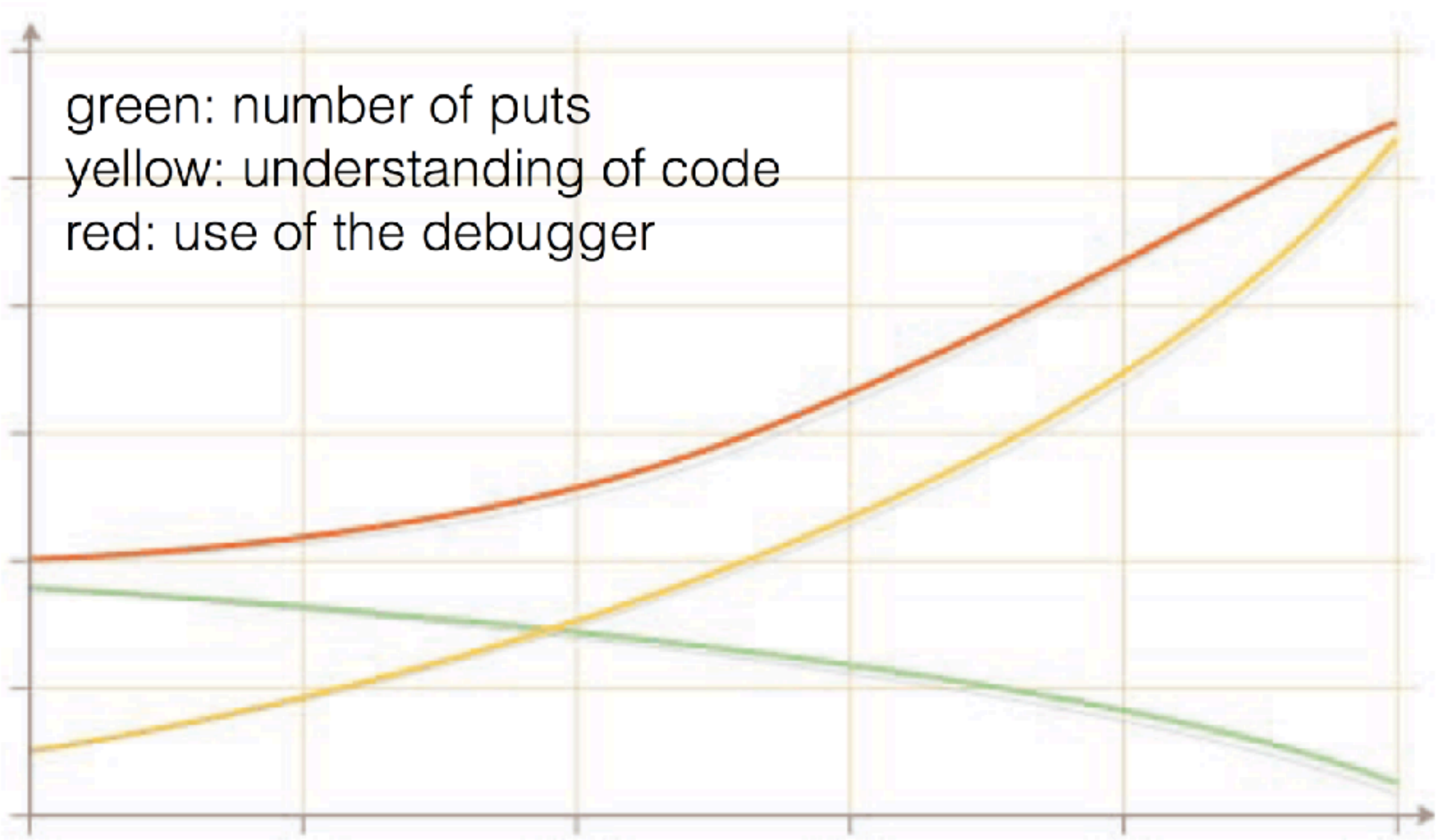
pry 0.11.3 **80,192,653** DOWNLOADS
An IRB alternative and runtime developer console

pry-byebug 3.6.0 **22,939,648** DOWNLOADS
Combine 'pry' with 'byebug'. Adds 'step', 'next', 'finish', 'continue' and 'break' ...

pry-rails 0.3.6 **22,471,769** DOWNLOADS
Use Pry as your rails console

pry-remote 0.1.8 **5,712,775** DOWNLOADS
Connect to Pry remotely using DRb

Graph



Warning: Fictional chart

a REPL you shall have

```
#!/usr/bin/env ruby
require 'pathname'
ENV['BUNDLE_GEMFILE'] ||= File.expand_path('../../
Gemfile', Pathname.new(__FILE__).realpath)
require 'rubygems'
require 'bundler/setup'
Bundler.require(:default)
$LOAD_PATH << 'lib'
require 'user'
Pry.start
```

or

```
$ pry -r ./yourthing.rb
```

Andy Tiefenthaler

@pxlpnk

Software & Security Engineer 💻 Product Developer 💡

Help startups to achieve minimum viable security
www.occamslabs.com

Product and Software at:
www.guardrails.io