HOW TO USE PRY TO KILL HEADCRABS AND OTHER BUGS

A LITTLE ABOUT ME...

JASON CARTER

SOFTWARE ENGINEER AT MAVENLINK

“Mavenlink delivers cloud-based software and services that transform how businesses do work with distributed teams, contractors, and clients.”

(but really we're a kickass software development team that practices extreme programming, pair programming, and test driven development)

We're hiring in SLC!

BUGS HAPPEN...

» test gaps

» human error

» unforeseen side effects

» black mesa incidents


WE'VE ALL SPENT
HOURS TRACKING ONE
DOWN...

...AND WE'VE ALL USED PRY

WHAT IS PRY

“Pry is a powerful alternative to the standard IRB shell for Ruby. It features syntax highlighting, a flexible plugin architecture, runtime invocation and source and documentation browsing.”

--http://pryrepl.org/

PRY IS A REPL

```
loop { p eval gets }
```

Takes user input, evaluates it, prints the result, and loops again

IN A RAILS APP

```
gem 'pry'
gem 'pry-byebug'
```

pry is the base gem that replaces irb
pry-byebug allows us to stop execution in a rails app

THE BASICS

» Make sure you're running rails s not unicorn

» Toss a binding.pry in your code somewhere

» Call that code somehow!

HOW I USED PRY...

```
def ravenholm
  binding.pry
  freeman = GordonFreeman.new(weapon: crowbar)
  while headcrabs_alive do
    binding.pry
    freeman.attack
  end
  binding.pry
end
```

LETS LEARN SOME (HOPEFULLY) NEW STUFF

THE `help` COMMAND

Type help to get a handy list of pry commands

You can also type help before a command to learn more about it

```
Ruby:(2.2.3) object:(main) >> help wtf?

Usage: wtf[?|!]

Show's a few lines of the backtrace of the most recent exception (also available
as `_ex_.backtrace`). If you want to see more lines, add more question marks or
exclamation marks.

wtf?
wtf?!???!?!?

# To see the entire backtrace, pass the `-v` or `--verbose` flag.
wtf -v

    -v, --verbose      Show the full backtrace
    -h, --help         Show this message.
```

RUNNING SHELL COMMANDS

Simply prepend your command with .

```
Ruby:(2.2.3) object:(main) >> . ps aux | grep ruby
```

```
Ruby:(2.2.3) object:(#<HalfLife2>) >> ["gravity gun", "shotgun", "crowbar"]
[
    [0] "gravity gun",
    [1] "shotgun",
    [2] "crowbar"
]
Ruby:(2.2.3) object:(#<HalfLife2>) >> weapon_types = _
[
    [0] "gravity gun",
    [1] "shotgun",
    [2] "crowbar"
]
Ruby:(2.2.3) object:(#<HalfLife2>) >> weapon_types
[
    [0] "gravity gun",
    [1] "shotgun",
    [2] "crowbar"
]
```

GETTING SOME 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.

whereami

```
Ruby:(2.2.3) object:(#<HalfLife2>) >> whereami

From: /Users/jasoncarter/Documents/pry-talk-code.rb @ line 32 HalfLife2#play_ravenholm:

    28: def play_ravenholm
    29:   freeman = GordonFreeman.new("crowbar")
    30:   while headcrabs_alive do
    31:     binding.pry
 => 32:     freeman.attack
    33:   end
    34: end
```

find-method

```
Ruby:(2.2.3) object:(#<HalfLife2>) >> find-method -c true
HalfLife2
HalfLife2#headcrabs_alive:   true
Object
Object#__binding__:     # the singleton class gives false positives for `true` and `false`).
Object#DelegateClass:   klass.define_singleton_method :public_instance_methods do |all=true|
                      klass.define_singleton_method :protected_instance_methods do |all=true|
Ruby:(2.2.3) object:(#<HalfLife2>) >> find-method headcrabs_alive
HalfLife2
HalfLife2#headcrabs_alive
```

cd,ls

```
Ruby:(2.2.3) object:(#<HalfLife2>) >> cd freeman
Ruby:(2.2.3) object:(#<GordonFreeman>:1) >> ls
GordonFreeman#methods: attack  switch_weapon  throw_grenade  weapon  weapon=
self.methods: __pry__
instance variables: @grenades  @weapon
locals: _  __  _dir_  _ex_  _file_  _in_  _out_  _pry_
Ruby:(2.2.3) object:(#<GordonFreeman>:1) >> @weapon
"crowbar"
```

ls FLAGS

```
-m, --methods                Show public methods defined on the Object
-M, --instance-methods       Show public methods defined in a Module or Class
-p, --ppp                    Show public, protected (in yellow) and private (in green) methods
-q, --quiet                  Show only methods defined on object.singleton_class and object.class
-v, --verbose                Show methods and constants on all super-classes (ignores Pry.config.ls.ceiling)
-g, --globals                Show global variables, including those builtin to Ruby (in cyan)
-l, --locals                 Show hash of local vars, sorted by descending size
-c, --constants              Show constants, highlighting classes (in blue), and exceptions (in purple).
                             Constants that are pending autoload? are also shown (in yellow)
-i, --ivars                  Show instance variables (in blue) and class variables (in bright blue)
-G, --grep                   Filter output by regular expression
-h, --help                   Show this message.
```

ls -q

```
Ruby:(2.2.3) object:(#<GordonFreeman>:1) >> ls
FPSGuy#methods: jump  run  walk
GordonFreeman#methods: attack  switch_weapon  throw_grenade  weapon  weapon=
self.methods: __pry__
instance variables: @grenades  @weapon
locals: _  __  _dir_  _ex_  _file_  _in_  _out_  _pry_
Ruby:(2.2.3) object:(#<GordonFreeman>:1) >> ls -q
GordonFreeman#methods: attack  switch_weapon  throw_grenade  weapon  weapon=
self.methods: __pry__
instance variables: @grenades  @weapon
locals: _  __  _dir_  _ex_  _file_  _in_  _out_  _pry_
```

ls -G

```
Ruby:(2.2.3) object:(#<GordonFreeman>:1) >> ls -G grenade
GordonFreeman#methods: throw_grenade
instance variables: @grenades  @weapon
```

pry-backtrace

```
Ruby:(2.2.3) object:(#<HalfLife2>) >> pry-backtrace
Backtrace:
--
/Users/jasoncarter/.rvm/gems/ruby-2.2.3@mavenlink/gems/pry-byebug-3.4.0/lib/byebug/processors/pry_processor.rb:115:in `block in resume_pry'
/Users/jasoncarter/.rvm/gems/ruby-2.2.3@mavenlink/gems/pry-byebug-3.4.0/lib/byebug/processors/pry_processor.rb:28:in `block in run'
/Users/jasoncarter/.rvm/gems/ruby-2.2.3@mavenlink/gems/pry-byebug-3.4.0/lib/byebug/processors/pry_processor.rb:27:in `catch'
/Users/jasoncarter/.rvm/gems/ruby-2.2.3@mavenlink/gems/pry-byebug-3.4.0/lib/byebug/processors/pry_processor.rb:27:in `run'
/Users/jasoncarter/.rvm/gems/ruby-2.2.3@mavenlink/gems/pry-byebug-3.4.0/lib/byebug/processors/pry_processor.rb:111:in `resume_pry'
/Users/jasoncarter/.rvm/gems/ruby-2.2.3@mavenlink/gems/pry-byebug-3.4.0/lib/byebug/processors/pry_processor.rb:63:in `at_line'
/Users/jasoncarter/.rvm/gems/ruby-2.2.3@mavenlink/gems/byebug-9.0.5/lib/byebug/context.rb:96:in `at_line'
/Users/jasoncarter/Documents/pry-talk-code.rb:46:in `play_ravenholm'
/Users/jasoncarter/Documents/pry-talk-code.rb:56:in `<main>'
```

» these get big in a rails app 😅

watch

```
Ruby:(2.2.3) object:(#<GordonFreeman>:1) >> watch @weapon
Watching @weapon
watch: @weapon => "crowbar"
Ruby:(2.2.3) object:(#<GordonFreeman>:1) >> cd ..
Ruby:(2.2.3) object:(#<HalfLife2>) >> freeman.switch_weapon("Gravity Gun")
watch: @weapon => "Gravity Gun"
"Gravity Gun"
```

EDIT ON THE FLY

!	Clear the input buffer.
amend-line	Amend a line of input in multi-line mode.
edit	Invoke the default editor on a file.
hist	Show and replay readline history.
play	Playback a string variable, method, line, or file as input.
show-input	Show the contents of the input buffer for the current multi-line expression.

hist

```
Ruby:(2.2.3) object:(#<HalfLife2>) >> hist
 1: cd freeman
 2: ls
 3: ls -q
 4: help ls
 5: ls -G weapon
```

hist FLAGS

-a, --all	Display all history
-H, --head	Display the first N items
-T, --tail	Display the last N items
-s, --show	Show the given range of lines
-G, --grep	Show lines matching the given pattern
-c, --clear	Clear the current session's history
-r, --replay	Replay a line or range of lines
--save	Save history to a file
-e, --exclude-pry	Exclude Pry commands from the history
-n, --no-numbers	Omit line numbers
-h, --help	Show this message.

DEFINE METHODS ON THE FLY!

```
Ruby:(2.2.3) object:(#<GordonFreeman>:1) >> def stand_stoicly
>>  puts "..."
>>end
:stand_stoicly
Ruby:(2.2.3) object:(#<GordonFreeman>:1) >> ls
FPSGuy#methods: jump  run  walk
GordonFreeman#methods: attack  switch_weapon  throw_grenade  weapon  weapon=
self.methods: __pry__  stand_stoicly
instance variables: @grenades  @weapon
locals: _  __  _dir_  _ex_  _file_  _in_  _out_  _pry_
Ruby:(2.2.3) object:(#<GordonFreeman>:1) >> cd ..
Ruby:(2.2.3) object:(#<HalfLife2>) >> freeman.stand_stoicly
...
```

!,show-input,amend-line

```
Ruby:(2.2.3) object:(#<GordonFreeman>:1) >> def say_name
>>  puts "Master Chief"
>>  show-input
1: def say_name
2:   puts "Master Chief"
>>  amend-line 2 puts "Gordon Freeman"
1: def say_name
2:   puts "Gordon Freeman"
>>end
:say_name
Ruby:(2.2.3) object:(#<GordonFreeman>:1) >> show-input

Ruby:(2.2.3) object:(#<GordonFreeman>:1) >> def pick_up_the_can
>>  !
Input buffer cleared!
```

A LITTLE TIME FOR INTROSPECTION

```
ri                  View ri documentation.
show-doc            Show the documentation for a method or class.
show-source         Show the source for a method or class.
stat                View method information and set _file_ and _dir_ locals.
```

show-source

```
Ruby:(2.2.3) object:(#<GordonFreeman>:1) >> show-source switch_weapon

From: /Users/jasoncarter/Documents/pry-talk-code.rb @ line 27:
Owner: GordonFreeman
Visibility: public
Number of lines: 3

def switch_weapon(weapon)
  @weapon = weapon
end
```

show-source

```
Ruby:(2.2.3) object:(#<GordonFreeman>:1) >> show-source GordonFreeman#attack

From: /Users/jasoncarter/Documents/pry-talk-code.rb @ line 36:
Owner: GordonFreeman
Visibility: public
Number of lines: 3

def attack
  puts "Attacking with #{@weapon}"
end
```

show-source

```
Ruby:(2.2.3) object:(#<GordonFreeman>:1) >> show-source GordonFreeman

From: /Users/jasoncarter/Documents/pry-talk-code.rb @ line 18:
Class name: GordonFreeman
Number of lines: 22

class GordonFreeman < FPSGuy

  attr_accessor :weapon

  def initialize(weapon)
    @weapon = weapon
    @grenades = 4
  end

  def switch_weapon(weapon)
    @weapon = weapon
  end

  def throw_grenade
    puts "Throwing grenade!"
    @grenades -= 1
  end

  def attack
    puts "Attacking with #{@weapon}"
  end
end
```

TRAVERSE AROUND LIKE A PRO

```
break                   Set or edit a breakpoint.
continue                Continue program execution and end the pry session.
down                    Move current frame down.
finish                  Execute until current stack frame returns.
frame                   Move to specified frame #.
next                    Execute the next line within the current stack frame.
step                    Step execution into the next line or method.
up                      Move current frame up.
```

break

```
Examples:

  break SomeClass#run          Break at the start of `SomeClass#run`.
  break Foo#bar if baz?        Break at `Foo#bar` only if `baz?`.
  break app/models/user.rb:15 Break at line 15 in user.rb.
  break 14                     Break at line 14 in the current file.

  break --condition 4 x > 2   Add/change condition on breakpoint #4.
  break --condition 3          Remove the condition on breakpoint #3.

  break --delete 5             Delete breakpoint #5.
  break --disable-all          Disable all breakpoints.

  break                        List all breakpoints.
  break --show 2               Show details about breakpoint #2.

    -c, --condition        Change condition of a breakpoint.
    -s, --show             Show breakpoint details and source.
    -D, --delete           Delete a breakpoint.
    -d, --disable          Disable a breakpoint.
    -e, --enable           Enable a disabled breakpoint.
        --disable-all      Disable all breakpoints.
        --delete-all       Delete all breakpoints.
    -h, --help             Show this message.
```

break

```
Ruby:(2.2.3) object:(#<HalfLife2>) >> break HalfLife2#headcrabs_alive

  Breakpoint 1: HalfLife2#headcrabs_alive (Enabled)

  50: def headcrabs_alive
51:   true
52:   # when are there not headcrabs?
53: end


Ruby:(2.2.3) object:(#<HalfLife2>) >> continue
Attacking with crowbar

  Breakpoint 1. First hit

From: /Users/jasoncarter/Documents/pry-talk-code.rb @ line 50 HalfLife2#headcrabs_alive:

 => 50: def headcrabs_alive
    51:   true
    52:   # when are there not headcrabs?
    53: end
```

up, down

```
Ruby:(2.2.3) object:(#<HalfLife2>) >> continue
Attacking with crowbar

  Breakpoint 1. Hit 2 times.

From: /Users/jasoncarter/Documents/pry-talk-code.rb @ line 50 HalfLife2#headcrabs_alive:

 => 50: def headcrabs_alive
    51:   true
    52:   # when are there not headcrabs?
    53: end

Ruby:(2.2.3) object:(#<HalfLife2>) >> up

From: /Users/jasoncarter/Documents/pry-talk-code.rb @ line 44 HalfLife2#play_ravenholm:

    42: def play_ravenholm
    43:   freeman = GordonFreeman.new("crowbar")
 => 44:   while headcrabs_alive do
    45:     binding.pry
    46:     freeman.attack
    47:   end
    48: end

Ruby:(2.2.3) object:(#<HalfLife2>) >> down

From: /Users/jasoncarter/Documents/pry-talk-code.rb @ line 50 HalfLife2#headcrabs_alive:

 => 50: def headcrabs_alive
    51:   true
    52:   # when are there not headcrabs?
    53: end
```

CONFIGURATION

» Pry is configured through a .pryrc

» Can also be configured at runtime

.PRYRC

» Our .pryrc is configured by ansible-workstation

» Submit a pr maybe?

I PROPOSE

```
Pry.commands.alias_command 'c', 'continue' rescue nil
Pry.commands.alias_command 's', 'step' rescue nil
Pry.commands.alias_command 'n', 'next' rescue nil
Pry.commands.alias_command 'r!', 'reload!' rescue nil
```

SOME RAD STUFF IN THERE ALREADY

```
command "copy", "Copies any supplied string to the system clip board"  do |string|
  IO.popen('pbcopy', 'w') { |f| f << string.to_s }
end

command "sql",
        "Send any supplied SQL statement to the currently connected ActiveRecord database.",
        requires_gem: ['activerecord'] do |query|
  ActiveRecord::Base.connection.select_all(query)
end

command "caller_method", "Reveal the caller of the current method." do |depth|
  depth = depth.to_i || 1
  if /^(.+?):(\d+)(?::in `(.*)')?/ =~ caller(depth+1).first
    file = Regexp.last_match[1]
    line = Regexp.last_match[2].to_i
    method = Regexp.last_match[3]
    output.puts [file, line, method]
  end
end

command "array_toy",
        "Returns an Array object keyed from 1 to 10. This is helpful for experimenting with the Array library.",
        keep_retval: true do
  Array.new(10) { |i| i+1 }
end

command "hash_toy",
        "Returns a hash object keyed from 'a' to 'j'. This is helpful for experimenting with the hash library.",
        keep_retval: true do
  Hash[("a".."j").to_a.zip((1..10).to_a)]
end

command "local_methods", "Shows the local methods of the current object", keep_retval: true  do |object|
  case object.class
  when Class
    object.public_methods.sort - Object.public_methods
  when Module
    object.public_methods.sort - Module.public_methods
  else
    object.public_methods.sort - Object.new.public_methods
  end
end
```

SOME HELPFUL ALIASES

!!!	Alias for `exit-program`
!!@	Alias for `exit-all`
$	Alias for `show-source`
?	Alias for `show-doc`
@	Alias for `whereami`
breakpoint	Alias for `break`
breaks	Alias for `breakpoints`
clipit	Alias for `gist --clip`
file-mode	Alias for `shell-mode`
history	Alias for `hist`
quit	Alias for `exit`
quit-program	Alias for `exit-program`
reload-method	Alias for `reload-code`
show-method	Alias for `show-source`

RESOURCES

```
» pry github

» pry wiki

» pry-byebug github
```


NOW GO KILL SOME
BUGS...