Pry

Pry as a REPL replacement

```
gem install pry
```

```
▶ pry
[1] pry(main)> 2 + 2
=> 4
[2] pry(main)> "A sample string".reverse
=> "gnirts elpmas A"
[3] pry(main)>
```

Exit

- exit - Continues execution
- exit! - Exits
- CTRL-C - Abort current execution

Navigating long output

```
(1..10000).to_a
```

```
=> [1,
 2,
 3,
 4,
 5,
 6,
 7,
 8,
 9,
 10,
 11,
 12,
 13,
 14,
 15,
 16,
 17,
 18,
 19,
 20,
 21,
 22,
 23,
:
```

Navigating long output

The `vim` text editor uses these keyboard shortcuts as well.

- q - Exits
- / - search
- n - next result
- N - previous result

pry-coolline

Syntax highlighting as you type.

```
gem install pry-coolline
```

IRON HACK

pry-doc

gem install pry-doc

```
[6] pry(main)> a_string = "Name"
=> "Name"
[7] pry(main)> ls a_string
Comparable#methods: <  <=  >  >=  between?
String#methods:
  %    ascii_only?  chomp      delete         encode         hash     lstrip     reverse     setbyte      start_with?  to_c    tr_s!
  *    b            chomp!     delete!        encode!        hex      lstrip!    reverse!    shell_split  strip        to_f    unpack
  +    bytes        chop       downcase       encoding       include? match      rindex      shellescape  strip!       to_i    upcase
  <<   bytesize     chop!      downcase!      end_with?      index    next       rjust       shellsplit   sub          to_r    upcase!
  <=>  byteslice    chr        dump           eql?           insert   next!      rpartition  size         sub!         to_s    upto
  ==   capitalize   clear      each_byte      force_encoding inspect  oct        rstrip      slice        succ         to_str  valid_encoding?
  ===  capitalize!  codepoints each_char      freeze         intern   ord        rstrip!     slice!       succ!        to_sym
  =~   casecmp      concat     each_codepoint getbyte        length   partition  scan        split        sum          tr
  []   center       count      each_line      gsub           lines    prepend    scrub       squeeze      swapcase     tr!
  []=  chars        crypt      empty?         gsub!          ljust    replace    scrub!      squeeze!     swapcase!    tr_s
[8] pry(main)> ? a_string.gsub

From: string.c (C Method):
Owner: String
Visibility: public
Signature: gsub(*arg1)
Number of lines: 34

Returns a copy of str with the all occurrences of
pattern substituted for the second argument. The pattern is
typically a Regexp; if given as a String, any
regular expression metacharacters it contains will be interpreted
```

ls

```
[3] pry(main)> a_string = "Thing"
=> "Thing"
[4] pry(main)> ls a_string
Comparable#methods: <  <=  >  >=  between?
String#methods:
  %    []=          center      concat     each_char       force_encoding  insert   next       rindex      shellescape  strip      to_c    tr_s
  *    ascii_only?  chars       count      each_codepoint  freeze          inspect  next!      rjust       shellsplit   strip!     to_f    tr_s!
  +    b            chomp       crypt      each_line       getbyte         intern   oct        rpartition  size         sub        to_i    unpack
  <<   bytes        chomp!      delete     empty?          gsub            length   ord        rstrip      slice        sub!       to_r    upcase
  <=>  bytesize     chop        delete!    encode          gsub!           lines    partition  rstrip!     slice!       succ       to_s    upcase!
  ==   byteslice    chop!       downcase   encode!         hash            ljust    prepend    scan        split        succ!      to_str  upto
  ===  capitalize   chr         downcase!  encoding        hex             lstrip   replace    scrub       squeeze      sum        to_sym  valid_encoding?
  =~   capitalize!  clear       dump       end_with?       include?        lstrip!  reverse    scrub!      squeeze!     swapcase   tr
  []   casecmp _    codepoints  each_byte  eql?            index           match    reverse!   setbyte     start_with?  swapcase!  tr!
```

show-source

```
[17] pry(main)> show-source say

From: /Users/rafael/tmp/sample.rb @ line 7:
Owner: Object
Visibility: private
Number of lines: 4

def say(text)
  puts "Text"
  binding.pry
end
```

*show-method is an alias for show-source. You can use either indifferently

show-source

```
[12] pry(main)> a_string = "Thing"
=> "Thing"
[13] pry(main)> show-source a_string.gsub

From: string.c (C Method):
Owner: String
Visibility: public
Number of lines: 5

static VALUE
rb_str_gsub(int argc, VALUE *argv, VALUE str)
{
    return str_gsub(argc, argv, str, 0);
}
```

‘Debugging’ with pry

```ruby
require 'pry'

class A
  attr_accessor :name
end

a = A.new
a.name = "My name"

sum = '1.'.to_i + '3'.to_i
binding.pry

puts sum

100.times do
  binding.pry
end
```

Moving in the code

gem install pry-byebug MRI 2.0+

Moving in the code

```
From: /Users/rafael/tmp/sample.rb @ line 9 Object#say:

     7: def say(text)
     8:   puts "Text"
 =>  9:   binding.pry
    10: end

[1] pry(main)> n

From: /Users/rafael/tmp/sample.rb @ line 13 :

     8:   puts "Text"
     9:   binding.pry
    10: end
    11:
    12: salute
 => 13: salute
```

Moving in the code

- step: Step execution into the next line or method. Takes an optional numeric argument to step multiple times. Aliased to s

- next: Step over to the next line within the same frame. Also takes an optional numeric argument to step multiple lines. Aliased to n

- finish: Execute until current stack frame returns. Aliased to f

- continue: Continue program execution and end the Pry session. Aliased to c

Caller

```
[2] pry(main)> caller
=> ["/Users/rafael/.rvm/gems/ruby-2.1.2/gems/pry-0.10.0/lib/pry/pry_instance.rb:353:in `eval'",
 "/Users/rafael/.rvm/gems/ruby-2.1.2/gems/pry-0.10.0/lib/pry/pry_instance.rb:353:in `evaluate_ruby'",
 "/Users/rafael/.rvm/gems/ruby-2.1.2/gems/pry-0.10.0/lib/pry/pry_instance.rb:321:in `handle_line'",
 "/Users/rafael/.rvm/gems/ruby-2.1.2/gems/pry-0.10.0/lib/pry/pry_instance.rb:241:in `block (2 levels) in eval'",
 "/Users/rafael/.rvm/gems/ruby-2.1.2/gems/pry-0.10.0/lib/pry/pry_instance.rb:240:in `catch'",
 "/Users/rafael/.rvm/gems/ruby-2.1.2/gems/pry-0.10.0/lib/pry/pry_instance.rb:240:in `block in eval'",
 "/Users/rafael/.rvm/gems/ruby-2.1.2/gems/pry-0.10.0/lib/pry/pry_instance.rb:239:in `catch'",
 "/Users/rafael/.rvm/gems/ruby-2.1.2/gems/pry-0.10.0/lib/pry/pry_instance.rb:239:in `eval'",
 "/Users/rafael/.rvm/gems/ruby-2.1.2/gems/pry-0.10.0/lib/pry/repl.rb:77:in `block in repl'",
 "/Users/rafael/.rvm/gems/ruby-2.1.2/gems/pry-0.10.0/lib/pry/repl.rb:67:in `loop'",
 "/Users/rafael/.rvm/gems/ruby-2.1.2/gems/pry-0.10.0/lib/pry/repl.rb:67:in `repl'",
 "/Users/rafael/.rvm/gems/ruby-2.1.2/gems/pry-0.10.0/lib/pry/repl.rb:38:in `block in start'",
 "/Users/rafael/.rvm/gems/ruby-2.1.2/gems/pry-0.10.0/lib/pry/input_lock.rb:61:in `call'",
 "/Users/rafael/.rvm/gems/ruby-2.1.2/gems/pry-0.10.0/lib/pry/input_lock.rb:61:in `__with_ownership'",
 "/Users/rafael/.rvm/gems/ruby-2.1.2/gems/pry-0.10.0/lib/pry/input_lock.rb:79:in `with_ownership'",
 "/Users/rafael/.rvm/gems/ruby-2.1.2/gems/pry-0.10.0/lib/pry/repl.rb:38:in `start'",
 "/Users/rafael/.rvm/gems/ruby-2.1.2/gems/pry-0.10.0/lib/pry/pry_instance.rb:346:in `repl'",
 "/Users/rafael/.rvm/gems/ruby-2.1.2/gems/pry-byebug-1.3.3/lib/pry-byebug/processor.rb:115:in `block in resume_pry'",
 "/Users/rafael/.rvm/gems/ruby-2.1.2/gems/pry-byebug-1.3.3/lib/pry-byebug/processor.rb:21:in `block in run'",
 "/Users/rafael/.rvm/gems/ruby-2.1.2/gems/pry-byebug-1.3.3/lib/pry-byebug/processor.rb:20:in `catch'",
 "/Users/rafael/.rvm/gems/ruby-2.1.2/gems/pry-byebug-1.3.3/lib/pry-byebug/processor.rb:20:in `run'",
 "/Users/rafael/.rvm/gems/ruby-2.1.2/gems/pry-byebug-1.3.3/lib/pry-byebug/processor.rb:114:in `resume_pry'",
 "/Users/rafael/.rvm/gems/ruby-2.1.2/gems/pry-byebug-1.3.3/lib/pry-byebug/processor.rb:82:in `at_line'",
 "/Users/rafael/.rvm/gems/ruby-2.1.2/gems/byebug-2.7.0/lib/byebug/context.rb:82:in `at_line'",
 "/Users/rafael/tmp/sample.rb:13:in `<main>'"]
```