



```
group :development do
  gem 'pry-rails'
  gem 'pry-plus'
end
```

# What's Pry?

- “like irb, but better”
- Everything you need to program in Ruby
  - (apart from a text editor)
- 60% of your time is spent *debugging!*

## Brief history

- 2010: John Mair (@banisterfiend) created pry
- 2011: Ryan Fitzgerald (@rrff\_) and I joined in
- 2012: reached 1,000,000 downloads
  - Kyrylo Silin (@kyrylosilin) and rking (@sharpsawDOTorg) started contributing
- 2013: reached 2,000,000 downloads
  - 71 distinct contributors

# Debugging a blog engine

- Introduction
- **How do I use the Base64 library?**
- Why doesn't my method work?
- Where did that nil come from?
- Further reading

## How do I use the Base64 library?

```
$ pry
[1] pry(main)> require 'base64'
=> true
[2] pry(main)> ls Base64
Base64.methods: decode64  encode64
strict_decode64  strict_encode64
urlsafe_decode64  urlsafe_encode64
[3] pry(main)>
```

## How do I use the Base64 library?

```
[2] pry(main)> ls Base64
Base64.methods: decode64  encode64
strict_decode64  strict_encode64
urlsafe_decode64  urlsafe_encode64
[3] pry(main)> Base64.enc<tab>
```

## How do I use the Base64 library?

```
[2] pry(main)> ls Base64
Base64.methods: decode64  encode64
strict_decode64  strict_encode64
urlsafe_decode64  urlsafe_encode64
[3] pry(main)> Base64.encode64
```

## How do I use the Base64 library?

```
[2] pry(main)> ls Base64
Base64.methods: decode64  encode64
strict_decode64  strict_encode64
urlsafe_decode64  urlsafe_encode64
[3] pry(main)> Base64.encode64 "hi"
=> "aGk=\n"
[4] pry(main)>
```

## How do I use the Base64 library?

```
[2] pry(main)> ls Base64
Base64.methods: decode64  encode64
strict_decode64  strict_encode64
urlsafe_decode64  urlsafe_encode64
[3] pry(main)> Base64.encode64 "hi"
=> "aGk=\n"
[4] pry(main)> Base64.decode64
```

## How do I use the Base64 library?

```
[2] pry(main)> ls Base64
Base64.methods: decode64  encode64
strict_decode64  strict_encode64
urlsafe_decode64  urlsafe_encode64
[3] pry(main)> Base64.encode64 "hi"
=> "aGk=\n"
[4] pry(main)> Base64.decode64 _
```

## How do I use the Base64 library?

```
[2] pry(main)> ls Base64
Base64.methods: decode64  encode64
strict_decode64  strict_encode64
urlsafe_decode64  urlsafe_encode64
[3] pry(main)> Base64.encode64 "hi"
=> "aGk=\n"
[4] pry(main)> Base64.decode64 _
=> "hi"
```

## How do I use the Base64 library?

```
[2] pry(main)> ls Base64
Base64.methods: decode64  encode64
strict_decode64  strict_encode64
urlsafe_decode64  urlsafe_encode64
[5] pry(main)>
```

## How do I use the Base64 library?

```
[2] pry(main)> ls Base64
Base64.methods: decode64  encode64
strict_decode64  strict_encode64
urlsafe_decode64  urlsafe_encode64
[5] pry(main)> ? Base64.strict_encode64
```

# How do I use the Base64 library?

```
[5] pry(main)> ? Base64.strict_encode64
```

**From:** ~/ruby-1.9.3/lib/ruby/base64.rb

**Owner:** #<Class:Base64>

**Visibility:** public

**Signature:** strict\_encode64(**bin**)

Returns the Base64-encoded version of **bin**.

This method complies with RFC 4648.

No line feeds are added.

# How do I use the Base64 library?

```
[5] pry(main)> ? Base64.encode64
```

**From:** ~/ruby-1.9.3/lib/ruby/base64.rb

**Owner:** #<Class:Base64>

**Visibility:** public

**Signature:** encode64(**bin**)

Returns the Base64-encoded version of **bin**.

This method complies with RFC 2045.

Line feeds are added every 60 characters.

# How do I use the Base64 library?

```
[5] pry(main)> ? Base64.strict_encode64
```

**From:** ~/ruby-1.9.3/lib/ruby/base64.rb

**Owner:** #<Class:Base64>

**Visibility:** public

**Signature:** strict\_encode64(**bin**)

Returns the Base64-encoded version of **bin**.

This method complies with RFC 4648.

No line feeds are added.

# How do I use the Base64 library?

- Evaluate ruby code.
- Colour
- Tab-completion
- \_ (the last output)
- ls (list methods)
- ? (show-doc)

# Debugging a blog engine

- Introduction
- How do I use the Base64 library?
- **Why doesn't my method work?**
- Where did that nil come from?
- Further reading

# Why doesn't my method work?

```
[1] pry(main)> cat basic_auth.rb
require 'base64'
module BasicAuth
  def self.encode(username, password)
    s = [username, password].join(":")
    "Basic #{Base64.strict_encode64 s}"
  end

  def self.decode(header)
    base64 = header[/Basic (.*)/, 1]
    Base64.decode64 base64.split(":")
  end
end
```

# Why doesn't my method work?

```
[1] pry(main)> cat basic_auth.rb
[2] pry(main)> require _file_
=> true
[3] pry(main)>
```

# Why doesn't my method work?

```
[1] pry(main)> cat basic_auth.rb
[2] pry(main)> require _file_
=> true
[3] pry(main)> BasicAuth.encode 'hi', 'mum'
```

# Why doesn't my method work?

```
[1] pry(main)> cat basic_auth.rb
[2] pry(main)> require _file_
=> true
[3] pry(main)> BasicAuth.encode 'hi', 'mum'
```

# Why doesn't my method work?

```
[1] pry(main)> cat basic_auth.rb
[2] pry(main)> require _file_
=> true
[3] pry(main)> BasicAuth.encode 'hi', 'mum'
=> "Basic aGk6bXVt"
[4] pry(main)> BasicAuth.decode _
```

# Why doesn't my method work?

```
[1] pry(main)> cat basic_auth.rb
[2] pry(main)> require _file_
=> true
[3] pry(main)> BasicAuth.encode 'hi', 'mum'
=> "Basic aGk6bXVt"
[4] pry(main)> BasicAuth.decode _
NoMethodError: undefined method `unpack' for
["aGk6bXVt"]:Array
[5] pry(main)>
```

# Why doesn't my method work?

```
[5] pry(main)> wtf?  
0: ~/ruby-1.9.3/lib/ruby/base64.rb:58 in  
`decode64'  
1: ~/basic_auth.rb:10 in `decode'  
2: (pry):3 in `__pry__'  
[6] pry(main)> $ BasicAuth.decode
```

# Why doesn't my method work?

```
[5] pry(main)> wtf?  
0: ~/ruby-1.9.3/lib/ruby/base64.rb:58 in  
`decode64'  
1: ~/basic_auth.rb:10 in `decode'  
2: (pry):3 in `__pry__'  
[6] pry(main)> $ BasicAuth.decode  
[7] pry(main)> edit
```

# Why doesn't my method work?

```
[6] pry(main)> $ BasicAuth.decode  
[7] pry(main)> edit <alt+_>
```

# Why doesn't my method work?

```
[6] pry(main)> $ BasicAuth.decode  
[7] pry(main)> edit <alt+_> (on Linux)  
                  <esc>_ (on a Mac)
```

# Why doesn't my method work?

```
[6] pry(main)> $ BasicAuth.decode  
[7] pry(main)> edit BasicAuth.decode
```

```
basic_auth.rb          basic_auth.rb

1 require 'base64'
2 module BasicAuth
3   def self.encode(username, password)
4     s = [username, password].join(":")
5     "Basic #{Base64.strict_encode64 s}"
6   end
7
8   def self.decode(header)
9     base64 = header[/Basic (.*)/, 1]
10    Base64.decode64 base64.split(":")
11  end
12 end
13

From: /Users/cirwir
Number of lines:
Owner: #<Class:E
Visibility: public

def self.decode()
  base64 = header
  Base64.decode64
end
[8] pry(main)> edit-method BasicAuth.decode
```

```
basic_auth.rb          basic_auth.rb
1 require 'base64'
2 module BasicAuth
3   def self.encode(username, password)
4     s = [username, password].join(":")
5     "Basic #{Base64.strict_encode64 s}"
6   end
7
8   def self.decode(header)
9     base64 = header[/Basic (.*)/, 1]
10    Base64.decode64(base64).split(":")
11  end
12 end
13

From: /Users/cirwir
Number of lines:
Owner: #<Class:E
Visibility: public

def self.decode()
  base64 = header[/Basic (.*)/, 1]
  Base64.decode64(base64).split(":")
end
[8] pry(main)> edit-method BasicAuth.decode
```

# Why doesn't my method work?

```
[7] pry(main)> edit BasicAuth.decode  
[8] pry(main)>
```

# Why doesn't my method work?

```
[7] pry(main)> edit BasicAuth.decode  
[8] pry(main)> <ctrl+r>
```

# Why doesn't my method work?

```
[7] pry(main)> edit BasicAuth.decode  
[8] pry(main)>  
(reverse-i-search)`':
```

# Why doesn't my method work?

```
[7] pry(main)> edit BasicAuth.decode  
[8] pry(main)> BasicAuth.encode 'hi', 'mum'  
(reverse-i-search)`enc'
```

# Why doesn't my method work?

```
[7] pry(main)> edit BasicAuth.decode  
[8] pry(main)> BasicAuth.encode 'hi', 'mum'  
=> "Basic aGk6bXVt"  
[9] pry(main)>
```

# Why doesn't my method work?

```
[7] pry(main)> edit BasicAuth.decode  
[8] pry(main)> BasicAuth.encode 'hi', 'mum'  
=> "Basic aGk6bXVt"  
[9] pry(main)> BasicAuth.decode _
```

# Why doesn't my method work?

```
[7] pry(main)> edit BasicAuth.decode
[8] pry(main)> BasicAuth.encode 'hi', 'mum'
=> "Basic aGk6bXVt"
[9] pry(main)> BasicAuth.decode _
=> ["hi", "mum"]
[10] pry(main)>
```

## Why doesn't my method work?

- wtf? (show backtrace, also `_ex_.backtrace`)
- \$ (show-source)
- edit (uses `$EDITOR`)
- <Alt+\_>, <esc>\_ (copy last word down)
- <ctrl+r> (search history)
- `_file_` (last shown file, also `_dir_`)
- `_out_` (array of all output values, also `_in_`)

# Debugging a blog engine

- Introduction
- How do I use the Base64 library?
- Why doesn't my method work?
- **Where did that nil come from?**
- Further reading

# Where did that nil come from?

```
$ ruby post.rb
post.rb:11:in `make_safe': undefined method
`gsub' for nil:NilClass (NoMethodError)
  from post.rb:7:in `safe_title'
  from post.rb:19:in `<main>'
$
```

```
post.rb                                     post.rb
1
2 class Post
3   def initialize(params)
4     @params = params
5   end
6
7   def safe_title
8     make_safe @params[:title]
9   end
10
11  def make_safe(str)
12    str.gsub(/[^w]/, '-')
13  end
14
15
16 new_post = Post.new('title' => 'new post',
17                      'body' => 'your text here')
18
19 puts new_post.safe_title
20
```

[2] pry(main)

cirwin-mn:~ 1890557.mob

1890557.p12

Desktop

Documents

Downloads

Dropbox

Library

cirwin-mn:~ Line 18, Column 15

cirwin-mn:~ cirwin\$ subl post.rb

cirwin-mn:~ cirwin\$

```
post.rb
1 require 'pry'
2 class Post
3   def initialize(params)
4     @params = params
5   end
6
7   def safe_title
8     make_safe @params[:title]
9   end
10
11  def make_safe(str)
12    str.gsub(/[^\\w]/, '-')
13  end
14 end
15
16 new_post = Post.new('title' => 'new post',
17                      'body' => 'your text here')
18 binding.pry # |Added for debugging
19 puts new_post.safe_title
20
```

index d0ed3  
--- a/post.  
+++ b/post.  
@@ -4,7 +4,  
end  
  
def safe  
- make\_s  
+ make\_s  
end  
  
def make

[2] pry(main)  
cirwin-mn:~ 1890557.mob  
1890557.p12  
Desktop  
Documents  
Downloads  
Dropbox  
Library  
cirwin-mn:~ Line 18, Column 25

cirwin-mn:~ cirwin\$ subl post.rb  
cirwin-mn:~ cirwin

## Where did that nil come from?

```
$ subl post.rb  
$ ruby post.rb
```

# Where did that nil come from?

```
$ subl post.rb
```

```
$ ruby post.rb
```

```
From: post.rb @ line 18 :
```

```
15: new_post = Post.new(  
16:                      'title' => 'new post',  
17:                      'body'  => 'your text here')  
=> 18: binding.pry # Added for debugging  
19: puts new_post.safe_title
```

```
[1] pry(main)>
```

# Where did that nil come from?

```
$ subl post.rb
```

```
$ ruby post.rb
```

```
From: post.rb @ line 18 :
```

```
15: new_post = Post.new(  
16:                      'title' => 'new post',  
17:                      'body'  => 'your text here')  
=> 18: binding.pry # Added for debugging  
19: puts new_post.safe_title
```

```
[1] pry(main)> play -l 19
```

## Where did that nil come from?

```
[1] pry(main)> puts new_post.safe_title  
NoMethodError: undefined method `gsub' for nil  
[2] pry(main)> cd new_post
```

# Where did that nil come from?

```
[1] pry(main)> puts new_post.safe_title  
NoMethodError: undefined method `gsub' for nil  
[2] pry(main)> cd new_post  
[3] pry(#<Post>):1> ls
```

# Where did that nil come from?

```
[1] pry(main)> puts new_post.safe_title
NoMethodError: undefined method `gsub' for nil
[2] pry(main)> cd new_post
[3] pry(#<Post>):1> ls
Post#methods: make_safe  safe_title
instance variables: @params
locals: __dir__ __ex__ __file__ __in__ __out__ __pry__
[4] pry(#<Post>):1>
```

# Where did that nil come from?

```
[1] pry(main)> puts new_post.safe_title
NoMethodError: undefined method `gsub' for nil
[2] pry(main)> cd new_post
[3] pry(#<Post>):1> ls
Post#methods: make_safe  safe_title
instance variables: @params
locals: __dir__ __ex__ __file__ __in__ __out__ __pry__
[4] pry(#<Post>):1> $safe_title
```

# Where did that nil come from?

```
[1] pry(main)> puts new_post.safe_title
NoMethodError: undefined method `gsub' for nil
[2] pry(main)> cd new_post
[3] pry(#<Post>):1> ls
Post#methods: make_safe  safe_title
instance variables: @params
locals: __dir__ __ex__ __file__ __in__ __out__ __pry__
[4] pry(#<Post>):1> $ Post#safe_title
```

# Where did that nil come from?

```
[4] pry(#<Post>):1> $ safe_title  
From: post.rb @ line 7:  
Number of lines: 3  
Owner: Post  
Visibility: public
```

```
def safe_title  
  make_safe @params[:title]  
end  
[5] pry(#<Post>):1>
```

# Where did that nil come from?

```
[4] pry(#<Post>):1> $ safe_title  
From: post.rb @ line 7:  
Number of lines: 3  
Owner: Post  
Visibility: public
```

```
def safe_title  
  make_safe @params[:title]  
end  
[5] pry(#<Post>):1> @params  
{"title"=>"new post", "body"=>"your text here"}  
[6] pry(#<Post>):1>
```

# Where did that nil come from?

```
[4] pry(#<Post>):1> $ safe_title  
From: post.rb @ line 7:  
Number of lines: 3  
Owner: Post  
Visibility: public
```

```
def safe_title  
  make_safe @params[:title]  
end  
[5] pry(#<Post>):1> @params  
{"title"=>"new post", "body"=>"your text here"}  
[6] pry(#<Post>):1> edit --ex
```

```
post.rb
1 require 'pry'
2 class Post
3   def initialize(params)
4     @params = params
5   end
6
7   def safe_title
8     make_safe @params[:title]
9   end
10
11  def make_safe(str)
12    str.gsub(/[^w]/, '-')
13  end
14 end
15
16 new_post = Post.new('title' => 'new post',
17                      'body' => 'your text here')
18 binding.pry # Added for debugging
19 puts new_post.safe_title
20
```

```
post.rb
1 require 'pry'
2 class Post
3   def initialize(params)
4     @params = params
5   end
6
7   def safe_title
8     make_safe @params[:title]
9   end
10
11  def make_safe(str)
12    str.gsub(/[^w]/, '-')
13  end
14 end
15
16 new_post = Post.new('title' => 'new post',
17                      'body' => 'your text here')
18 binding.pry # Added for debugging
19 puts new_post.safe_title
20
```

```
post.rb
1 require 'pry'
2 class Post
3   def initialize(params)
4     @params = params
5   end
6
7   def safe_title
8     make_safe @params['title']
9   end
10
11  def make_safe(str)
12    str.gsub(/[\w]/, '-')
13  end
14 end
15
16 new_post = Post.new('title' => 'new post',
17                      'body' => 'your text here')
18 binding.pry # Added for debugging
19 puts new_post.safe_title
20
```

# Where did that nil come from?

```
[5] pry(#<Post>):1> @params
{“title”=>”new post”, “body”=>”your text here”}
[6] pry(#<Post>):1> edit --ex
[7] pry(#<Post>):1>
```

# Where did that nil come from?

```
[7] pry(#<Post>):1> .git diff  
diff --git a/post.rb b/post.rb  
index d0ed356..057d37c 100644  
--- a/post.rb  
+++ b/post.rb  
@@ -4,7 +4,7 @@ class Post  
end
```

```
def safe_title  
-  make_safe @params[:title]  
+  make_safe @params['title']  
end
```

```
[8] pry(#<Post>):1>
```

# Where did that nil come from?

```
[7] pry(#<Post>):1> .git diff
diff --git a/post.rb b/post.rb
index d0ed356..057d37c 100644
--- a/post.rb
+++ b/post.rb
@@ -4,7 +4,7 @@ class Post
end

def safe_title
-  make_safe @params[:title]
+  make_safe @params['title']
end
[8] pry(#<Post>):1> cd
```

# Where did that nil come from?

```
[8] pry(#<Post>):1> cd ..  
[9] pry(main)> whereami
```

# Where did that nil come from?

```
[8] pry(#<Post>):1> cd ..
[9] pry(main)> whereami
From: post.rb @ line 18 :

    15: new_post = Post.new(
    16:                 'title' => 'new post',
    17:                 'body'  => 'your text here')
=> 18: binding.pry # Added for debugging
    19: puts new_post.safe_title
[10] pry(main)>
```

# Where did that nil come from?

```
[9] pry(main)> whereami
From: post.rb @ line 18 :

  15: new_post = Post.new(
  16:           'title' => 'new post',
  17:           'body'  => 'your text here')
=> 18: binding.pry # Added for debugging
  19: puts new_post.safe_title
[10] pry(main)> puts new_post.safe_title
new-post
=> nil
[11] pry(main)>
```

# Where did that nil come from?

```
[10] pry(main)> puts new_post.safe_title  
new-post  
=> nil  
[11] pry(main)> puts new_post.safe_title;
```

## Where did that nil come from?

```
[10] pry(main)> puts new_post.safe_title  
new-post  
=> nil  
[11] pry(main)> puts new_post.safe_title;  
new-post  
[12] pry(main)>
```

# Where did that nil come from?

```
[10] pry(main)> puts new_post.safe_title  
new-post  
=> nil  
[11] pry(main)> puts new_post.safe_title;  
new-post  
[12] pry(main)> <ctrl+d>
```

# Where did that nil come from?

```
[10] pry(main)> puts new_post.safe_title  
new-post  
=> nil  
[11] pry(main)> puts new_post.safe_title;  
new-post  
[12] pry(main)>  
new-post  
$
```

## Where did that nil come from?

- binding.pry (open pry right here, right now)
- cd (change self)
- whereami
- edit --ex
- play -l
- . (run's shell commands)
- <ctrl-d> (cd .. or exit)
- ; (at end of line, hides output)

# Debugging a blog engine

- Introduction
- How do I use the Base64 library?
- Why doesn't my method work?
- Where did that nil come from?
- **Further reading**

## pry-rescue/pry-stack\_explorer

- pry-rescue
  - Automatically opens pry wherever you have an unhandled exception or test failure.
  - cd-cause lets you pry into any exception that happens inside the console
- pry-stack\_explorer
  - lets you move up/down the callstack as though you had a binding.pry at every level

## better\_errors

- BetterErrors.use\_pry!
- Opens a web page on unhandled exceptions
- Lets you debug your program after it crashes

## oops!

Application Frames

All Frames

A

```
#<Class:0x007fb27ad0870>@_app_views_main_index_html_erb_2234720...
app/views/main/index.html.erb, line 2
```

- block in ActionView::Template#render
   
(gem) actionpack-3.2.8/lib/action\_view/template.rb, line 145
- ActiveSupport::Notifications.instrument
   
(gem) activesupport-3.2.8/lib/active\_support/notifications.rb, line 125
- ActionView::Template#render
   
(gem) actionpack-3.2.8/lib/action\_view/template.rb, line 143
- block (2 levels) in ActionView::TemplateRenderer#render\_template
   
(gem) actionpack-3.2.8/lib/action\_view/renderer/template\_renderer.rb, line 47
- block in ActionView::TemplateRenderer#instrument
   
(gem) actionpack-3.2.8/lib/action\_view/renderer/abstract\_renderer.rb, line 38
- block in ActiveSupport::Notifications.instrument
   
(gem) activesupport-3.2.8/lib/active\_support/notifications.rb, line 123
- ActiveSupport::Notifications::Instrumenter#instrument
   
(gem) activesupport-3.2.8/lib/active\_support/notifications/instrumenter.rb, line 20
- ActiveSupport::Notifications.instrument
   
(gem) activesupport-3.2.8/lib/active\_support/notifications.rb, line 123
- ActionView::TemplateRenderer#instrument
   
(gem) actionpack-3.2.8/lib/action\_view/renderer/abstract\_renderer.rb, line 38
- block in ActionView::TemplateRenderer#render\_template
   
(gem) actionpack-3.2.8/lib/action\_view/renderer/template\_renderer.rb, line 46
- ActionView::TemplateRenderer#render\_with\_layout
   
(gem) actionpack-3.2.8/lib/action\_view/renderer/template\_renderer.rb, line 64
- ActionView::TemplateRenderer#render\_template

app/views/main/index.html.erb

\_app\_views\_main\_index\_html\_erb\_2

```
1 <h1><%= @message %></h1>
2 <% raise "oops!" %>

>> __FILE__
=> "/Users/charlie/code/better_errors/tmp/my_app/app/views/main/index.html.erb"
>> @message
=> "hello world"
>> "this is cool".upcase
=> "THIS IS COOL"
>> Rails.root
=> #<Pathname:/Users/charlie/code/better_errors/tmp/my_app>
>> @_controller.class
=> MainController
>> |
```

A This is a live shell. Type in here.

## Local Variables

|                    |     |
|--------------------|-----|
| local_assigns      | {}  |
| output_buffer      | nil |
| _old_virtual_path  | nil |
| _old_output_buffer | nil |

## Instance Variables

|         |               |
|---------|---------------|
| @config | {}            |
| @routes | nil           |
| message | "hello world" |

# pry-debugger

- break
  - Adds a breakpoint
- step/next
  - Step into or over the next line
- continue
  - Stop stepping through

## Recommended plugins

- **pry-plus**
  - pry-debugger, pry-rescue, pry-stack\_explorer
  - pry-doc, documentation for C methods
  - pry-docmore, documentation for ruby syntax
  - bond, tab completion for Hash#[], etc.
  - jist, Integration with gist.github.com
- **pry-rails**
  - makes “rails c” use pry!

- 60% of your time is spent debugging
- use binding.pry

# Questions?!

- Pry (@pryrepl)
  - <http://pryrepl.org>
  - irc://irc.freenode.net/#pry
  - type “help” in pry :)
- Conrad Irwin (@ConradIrwin)
  - <http://cirw.in>
  - irc://irc.freenode.net/cirwin