I love the console!

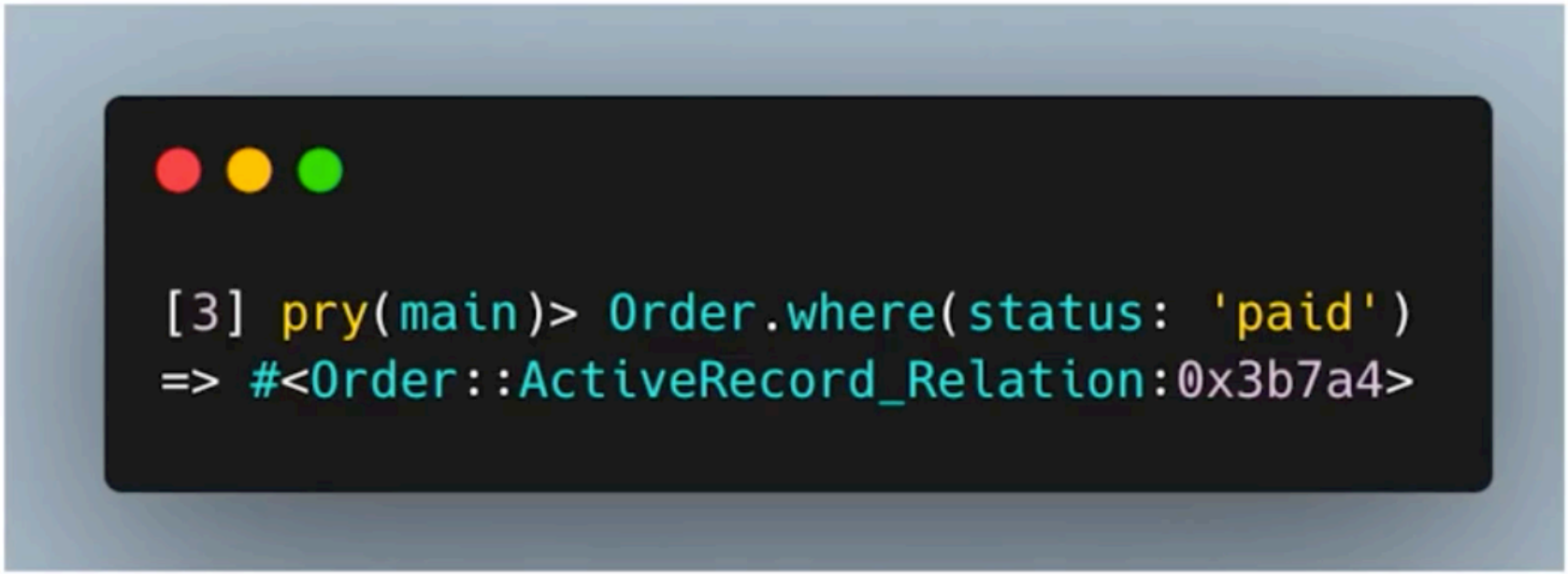
@SwetaSanghavi

Deep Dive Into Our Console

Deep Dive Into Our Console

Tactics to Turbocharge your use of the Tool ⚡


$ rails c


[3] pry(main)> Order.where(status: 'paid')
=> #<Order::ActiveRecord_Relation:0x3b7a4>

We can define and evaluate Ruby methods and classes

```
irb(main):020:0> def full_name(first_name, last_name)
irb(main):021:1>   first_name + " " + last_name
irb(main):022:1>   end
=> :full_name
irb(main):023:0> full_name("Olivia", "Rodrigo")
=> "Olivia Rodrigo"
```

The console: the underdog

- Evaluate a piece of code in a specific context
- Context is configurable
- Requires little set up
- Quick feedback about Rails functionality and the data in our application

Let's find out!

@SwetaSanghavi

What Do you Get?

`irb`

Core Ruby Libraries

What Do you Get?

`irb`

Core Ruby Libraries

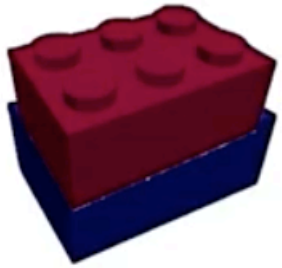
`bundle console`

Core Ruby Libraries
+
Gems from the Gemfile

What Do you Get?

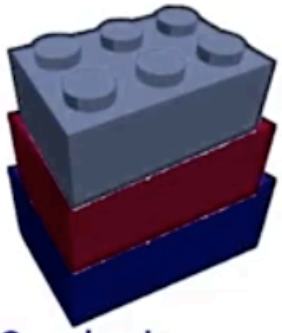
`irb`

Core Ruby Libraries



`bundle console`

Core Ruby Libraries
+
Gems from the Gemfile



`rails console`

Core Ruby libraries
+
Gems from the Gemfile
+
Rails Environment

@SwetaSanghavi

Storing Context

@SwetaSanghavi

```
irb(main):001:0> full_name
(irb):1:in `<main>': undefined local variable or method `full_name' for
main:Object (NameError)
    from
/Users/swetasanghavi/.rbenv/versions/3.0.2/lib/ruby/gems/3.0.0/gems/irb-
1.3.5/exe/irb:11:in    `<top (required)>'
    from /Users/swetasanghavi/.rbenv/versions/3.0.2/bin/irb:23:in `load'
    from /Users/swetasanghavi/.rbenv/versions/3.0.2/bin/irb:23:in `<main>'
```

We can define methods and classes in our console session

```
irb(main):020:0> def full_name(first_name, last_name)
irb(main):021:1>   first_name + " " + last_name
irb(main):022:1>   end
=> :full_name
irb(main):023:0> full_name("Olivia", "Rodrigo")
=> "Olivia Rodrigo"
```

IRB uses top-level Subsessions to save and differentiate context

```
Loading development environment (Rails 7.0.3)
[1] pry(main)>
```

We can spin up new subsessions with `irb`

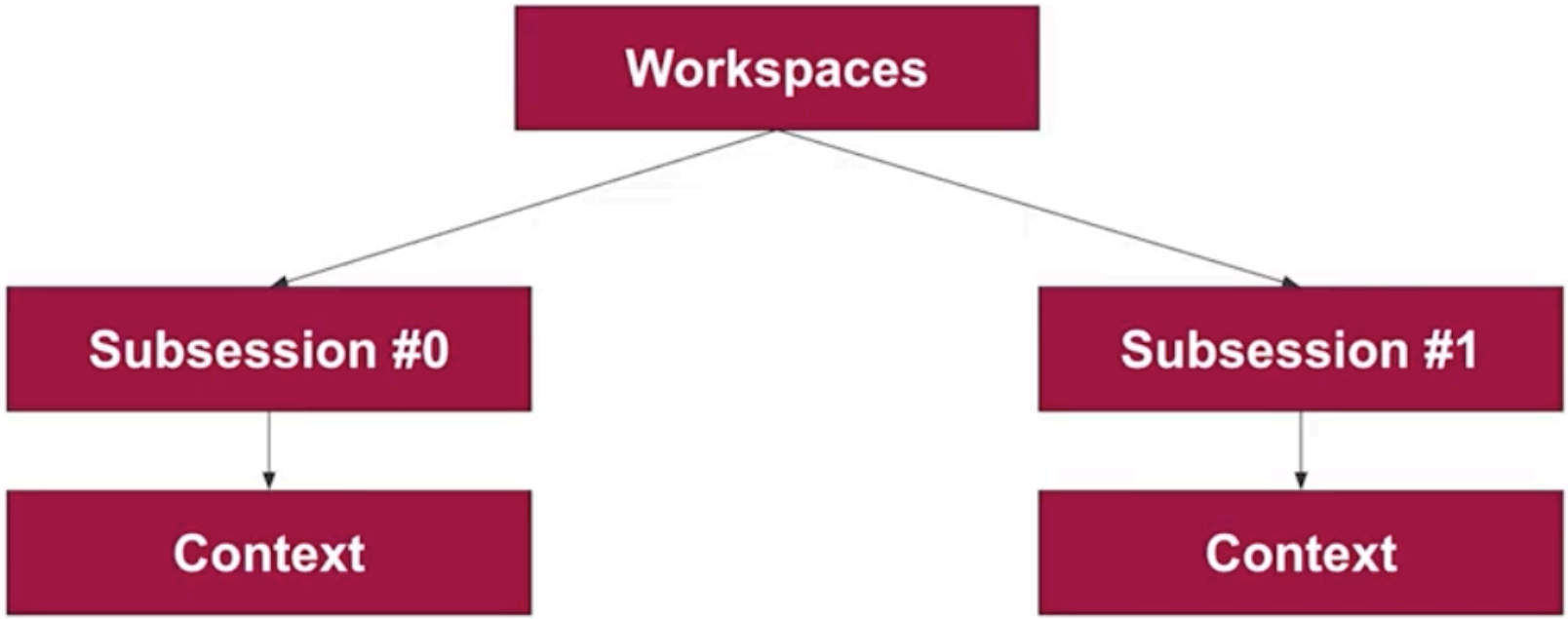
```
irb(main):024:0> irb
irb#1(main):001:0> jobs
=>
#0->irb on main (#<Thread:0x00007f8fb707bc70
sleep_forever>: stop)
#1->irb#1 on main (#<Thread:0x00007f8fb695f108
/Users/swetasanghavi/.rbenv/versions/3.0.2/lib/ruby/3.0.0/i
rb/ext/multi-irb.rb:192 run>: running)
```

Setting variables in a subsession

```
irb#1(main):003:0> fg 0
=> #<IRB::Irb: @context=#<IRB::Context:0x00007f8fb69d9fe8>,
@signal_status=:IN_EVAL, @scanner=#
<RubyLex:0x00007f8fb6a42bb0>>
irb(main):025:0> display_name = full_name("Olivia",
"Rodrigo")
=> "Olivia Rodrigo"
```

Once we switch subsessions, variables are not defined in the context

```
irb(main):026:0> fg 1
=> #<IRB::Irb: @context=#<IRB::Context:0x00007f8fb695ec08>,
@signal_status=:IN_EVAL, @scanner=#
<RubyLex:0x00007f8fb694b680>>
irb#1(main):004:0> display_name
(irb#1):4:in `<main>': undefined local variable or method
`display_name' for main:Object (NameError)
Did you mean?  display
irb#1(main):005:0> full_name("Grace", "Hopper")
=> "Grace Hopper"
```


Workspaces
Subsession #0
Subsession #1
Context
Context


Main
Top Level Context
Instance
Variables
Instance
Methods
Self

Execution Context

OR

The Binding Object

Parsing Ruby

@SwetaSanghavi

Console Evaluates Ruby

```
irb(main):020:0> def full_name(first_name, last_name)
irb(main):021:1>   first_name + " " + last_name
irb(main):022:1>   end
=> :full_name
irb(main):023:0> full_name("Olivia", "Rodrigo")
=> "Olivia Rodrigo"
```


Abstract Syntax Trees

- Parsers process code into an abstract syntax tree
- Low-level tree representation of the program's mechanics
- Simple data structures

In Ruby, Ripper outputs the trees for parsing

- Ripper library introduced to show us how Ruby is parsing Ruby
- Ripper provides an interface for parsing our program and tokenizing into a symbolic expression tree
- Returns abstract syntax trees or simple lexical analysis

Rippers #sexp method outputs a symbolic expression tree

```
Ripper#sexp(src, filename='-', lineno =1)
```

```
irb#1(main):013:0> Ripper.sexp("def full_name(first_name,
last_name) first_name + " " + last_name end")
```

```
irb#1(main):013:0> Ripper.sexp("def full_name(first_name,
last_name) first_name + " " + last_name end")
=>
[:program,
 [[:def,
   [:@ident, "full_name", [1, 4]],
   [:paren, [:params, [[:@ident, "first_name", [1, 14]],
[:@ident, "last_name", [1, 26]]], nil, nil, nil, nil, nil,
nil]],
   [:bodystmt, [[:binary, [:var_ref, [:@ident,
"first_name", [1, 37]]], :+, [:unary, :+@, [:var_ref,
[:@ident, "last_name", [1, 53]]]]]], nil, nil, nil]]]]
```

@SwetaSanghavi

```
irb#1(main):013:0> Ripper.sexp("def full_name(first_name,
last_name) first_name + " " + last_name end")
=>
[:program,
 [[:def,
   [:@ident, "full_name", [1, 4]],
   [:paren, [:params, [[:@ident, "first_name", [1, 14]],
[:@ident, "last_name", [1, 26]]], nil, nil, nil, nil, nil,
nil]],
   [:bodystmt, [[:binary, [:var_ref, [:@ident,
"first_name", [1, 37]]], :+, [:unary, :+@, [:var_ref,
[:@ident, "last_name", [1, 53]]]]]], nil, nil, nil]]]]
```

```
irb#1(main):013:0> Ripper.sexp("def full_name(first_name,
last_name) first_name + " " + last_name end")
=>
[:program,
 [[:def,
   [:@ident, "full_name", [1, 4]],
   [:paren, [:params, [[:@ident, "first_name", [1, 14]],
[:@ident, "last_name", [1, 26]]], nil, nil, nil, nil, nil,
nil]],
   [:bodystmt, [[:binary, [:var_ref, [:@ident,
"first_name", [1, 37]]], :+, [:unary, :+@, [:var_ref,
[:@ident, "last_name", [1, 53]]]]]], nil, nil, nil]]]]
```

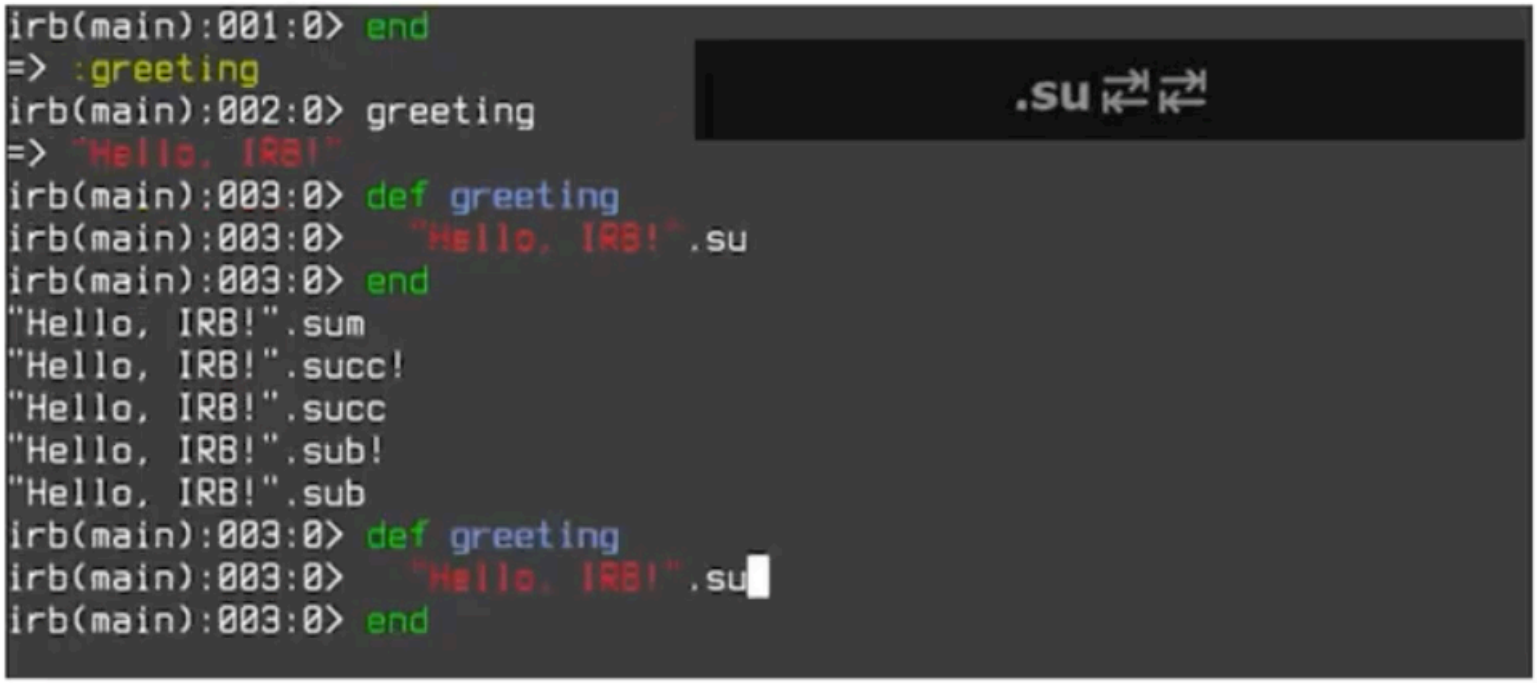
@SwetaSanghavi

```
irb#1(main):013:0> Ripper.sexp("def full_name(first_name,
last_name) first_name + " " + last_name end")
=>
[:program,
 [[:def,
   [:@ident, "full_name", [1, 4]],
   [:paren, [:params, [[:@ident, "first_name", [1, 14]],
[:@ident, "last_name", [1, 26]]], nil, nil, nil, nil, nil,
nil]],
   [:bodystmt, [[:binary, [:var_ref, [:@ident,
"first_name", [1, 37]]], :+, [:unary, :+@, [:var_ref,
[:@ident, "last_name", [1, 53]]]]]], nil, nil, nil]]]]
```

@SwetaSanghavi

```
irb#1(main):013:0> Ripper.sexp("def full_name(first_name,
last_name) first_name + " " + last_name end")
=>
[:program,
 [[:def,
   [:@ident, "full_name", [1, 4]],
   [:paren, [:params, [[:@ident, "first_name", [1, 14]],
[:@ident, "last_name", [1, 26]]], nil, nil, nil, nil, nil,
nil]],
   [:bodystmt, [[:binary, [:var_ref, [:@ident,
"first_name", [1, 37]]], :+, [:unary, :+@, [:var_ref,
[:@ident, "last_name", [1, 53]]]]]], nil, nil, nil]]]]
```

Interactivity



gif from ruby/reline Github

gif from ruby/reline Github

GNU Readline

- Provides in-line editing and history capabilities
- Used across shells, the MySQL command-line tool, and the GNU Debugger among others
- Initial release in 1989

GNU Readline

- Provides in-line editing and history capabilities
- Used across shells, the MySQL command-line tool, and the GNU Debugger among others
- Initial release in 1989

Readline line editor was abstracted from GNU

- C library that ships in standard Ruby
- Provides interface for GNU Readline
- Facilitates completion
- Accesses input history from the Ruby interpreter

Reline

- Readline could only be used when GNU Readline is installed before Ruby builds
- Itoyanagi Sakura built a pure Ruby implementation of GNU Readline called Reline
- IRB started using Reline in Ruby 2.7
- In Ruby 3.0, Reline allows IRB to accommodate multiline editing command and auto-completion

Turbocharge ⚡

@SwetaSanghavi

Superpowers of Rails C

- Performance Improvement
- Debugging
- Product Development

Improving Performance

Rails semantic scopes can hide complexity

```
ProjectLead.sweet_spot.where(ended_at: Time.current)
```

Ruby's Benchmark module outputs time used to execute code

```
irb(main):006:0> Benchmark.measure { |x|
ProjectLead.sweet_spot.where(ended_at: Time.current) }
=> #<Benchmark::Tms:0x00007fcf595a08b8 @cstime=0.0,
@cutime=0.0, @label="", @real=0.00052169300033710 9,
@stime=0.0, @total=0.0, @utime=0.0>
```

`to_sql` outputs the SQL underlying your AR query

```
[2] pry(main)> ProjectLead.sweet_spot.where(ended_at:
Time.current).to_sql
=> "SELECT \"project_leads\".* FROM \"project_leads\" WHERE
(usd_amount > 10000 and project_leads.backer_count > 100)
AND \"project_leads\".\"ended_at\" = '2022-05-15
20:05:45.465750'"
```

```
[5] pry(main)> ProjectLead.sweet_spot.where(ended_at:
Time.current).explain
  ProjectLead Load (6.1ms)  SELECT "project_leads".* FROM
"project_leads" WHERE (usd_amount > 10000 and
project_leads.backer_count > 100) AND
"project_leads"."ended_at" = $1  [["ended_at", "2022-05-15
20:29:57.209094"]]
=> EXPLAIN for: SELECT "project_leads".* FROM
"project_leads" WHERE (usd_amount > 10000 and
project_leads.backer_count > 100) AND
"project_leads"."ended_at" = $1 [["ended_at", "2022-05-15
20:29:57.209094"]]
                                                QUERY PLAN
-----------------------------------------------------------
-------------------------------------------
 Index Scan using index_project_leads_on_ended_at on
project_leads  (cost=0.14..8.16 rows=1 width=4823)
   Index Cond: (ended_at = '2022-05-15
20:29:57.209094'::timestamp without time zone)
   Filter: ((usd_amount > '10000'::numeric) AND
(backer_count > 100))
(3 rows)
```

```
[5] pry(main)> ProjectLead.sweet_spot.where(ended_at:
Time.current).explain
  ProjectLead Load (6.1ms)  SELECT "project_leads".* FROM
"project_leads" WHERE (usd_amount > 10000 and
project_leads.backer_count > 100) AND
"project_leads"."ended_at" = $1  [["ended_at", "2022-05-15
20:29:57.209094"]]
=> EXPLAIN for: SELECT "project_leads".* FROM
"project_leads" WHERE (usd_amount > 10000 and
project_leads.backer_count > 100) AND
"project_leads"."ended_at" = $1 [["ended_at", "2022-05-15
20:29:57.209094"]]
                                                    QUERY PLAN
------------------------------------------------------------------
-----------------------------------------------
 Index Scan using index_project_leads_on_ended_at on
project_leads  (cost=0.14..8.16 rows=1 width=4823)
   Index Cond: (ended_at = '2022-05-15
20:29:57.209094'::timestamp without time zone)
   Filter: ((usd_amount > '10000'::numeric) AND
(backer_count > 100))
(3 rows)
```

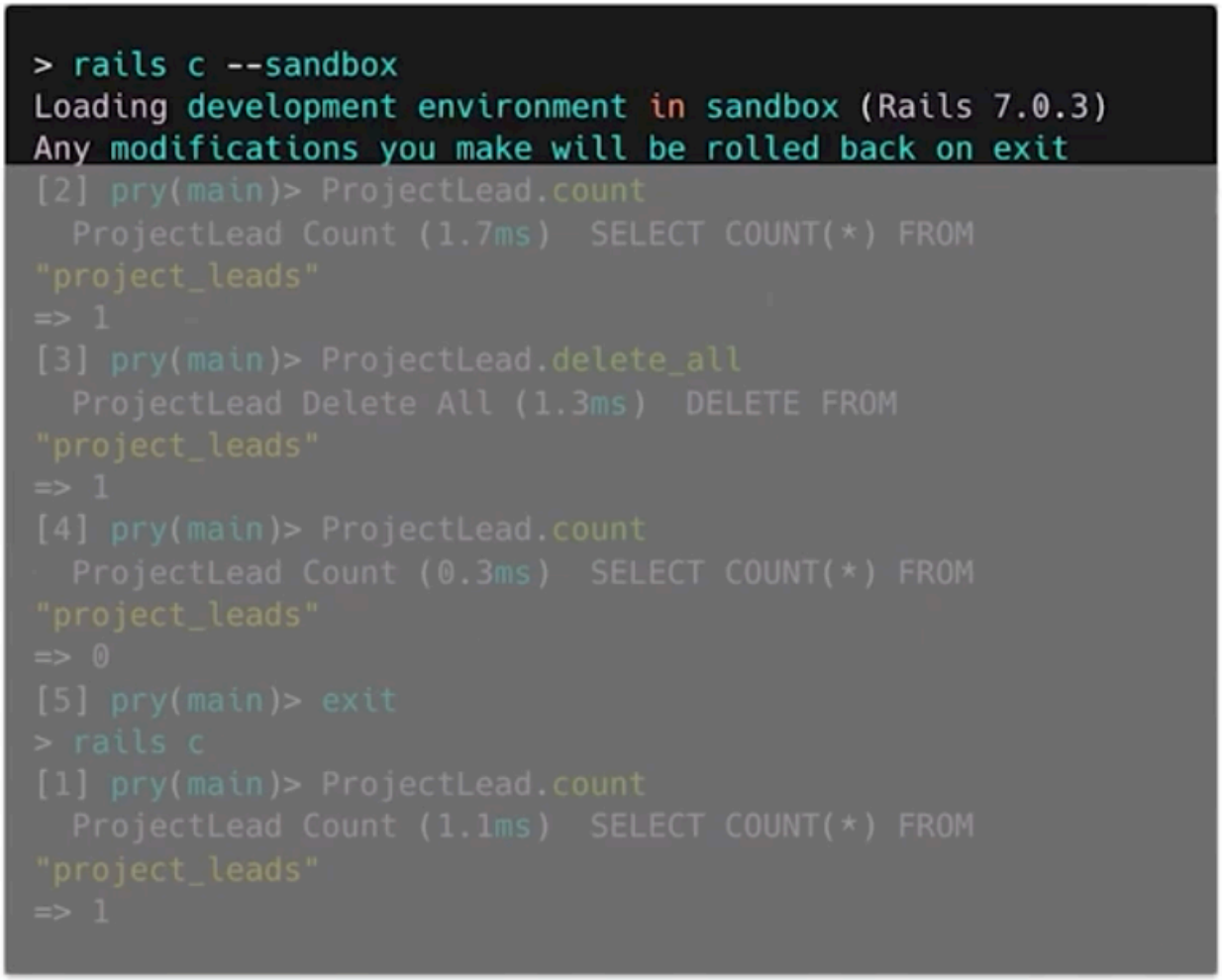
@SwetaSanghavi

Console in Production

```
heroku run rails c -a production
```

Console in Sandbox Mode

```
rails console --sandbox
```


> rails c --sandbox
Loading development environment in sandbox (Rails 7.0.3)
Any modifications you make will be rolled back on exit
[2] pry(main)> ProjectLead.count
ProjectLead Count (1.7ms) SELECT COUNT(*) FROM "project_leads"
=> 1
[3] pry(main)> ProjectLead.delete_all
ProjectLead Delete All (1.3ms) DELETE FROM "project_leads"
=> 1
[4] pry(main)> ProjectLead.count
ProjectLead Count (0.3ms) SELECT COUNT(*) FROM "project_leads"
=> 0
[5] pry(main)> exit
> rails c
[1] pry(main)> ProjectLead.count
ProjectLead Count (1.1ms) SELECT COUNT(*) FROM "project_leads"
=> 1

Console allows you access production specific context

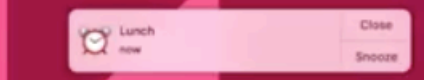
- Data Fixes
- Debugging

Use your console for fast experiments

```
[3] pry(main)> [].present?
=> false
[4] pry(main)> [].first
=> nil
```

Informing product decisions and impact with simple queries as a superpower

@SwetaSanghavi



Cheatsheet

@SwetaSanghavi

Clearing the Console

```
CMD + K
```

Reload!

```
[2] pry(main)> reload!
Reloading...
=> true
[4] pry(main)> reload!; DataUpdater.new.call
Reloading...
```

Getting the value of the last evaluated item

```
[5] pry(main)> ProjectLead.last
  ProjectLead Load (0.7ms)  SELECT "project_leads".* FROM
"project_leads" ORDER BY "project_leads"."id" DESC LIMIT $1
[["LIMIT", 1]]
=> #<ProjectLead:0x00007fb3fd88d480
[10] pry(main)> lead = _
```

Reverse Command Search

```
CTRL + R
```

CMD + K	Clear your console before browser testing to confirm the user interaction maps to the rails logs
reload!	When you change your code while in a console session, reload! to get code changes
var = _	Last evaluated item in the console is automatically stored in the underscore
CTRL + R	Search back through the consoles input history to find previously used commands

@SwetaSanghavi

Welcome to my shout outs slide!

Like pairing? Like extreme programming? Like product ownership and experimentation? Like Crowdfunding? Like Tabletop games?

We're hiring!

BACKERKIT

CROWDFUNDED PROJECTS YOU CAN TRUST

WNB.rb

A virtual community for women and non-binary Rubyists.

@SwetaSanghavi

Thanks for ⚡ turbocharging ⚡ with me!

swetagsanghavi@gmail.com

@SwetaSanghavi