PRY CHEAT SHEET

```
pry(main)> █
```

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

Install Pry

The pry-rails gem configures Pry properly for Rails apps

```
gem 'pry-rails', :group => :development
```

Recommended Plugins

Pry's plugin architecture allows for useful add-ons

```
gem 'pry-byebug', :group => :development
gem 'pry-rescue', :group => :development
```

Configure Pry

Pry is configured with a `.pryrc` file

Setting the Default Editor

```
Pry.config.editor = 'vim'
```

Defining Shortcuts

```
if defined?(PryByebug)
  Pry.commands.alias_command 's', ‘step'
  Pry.commands.alias_command 'n', ‘next'
end

class Object
  def me
    User.find_by(email: 'you@example.com')
  end
end
```

Using Pry

Invoke Pry by adding a breakpoint in the Ruby code

```
binding.pry
```

Conditional Invokation

Control when Pry takes over by firing it conditionally

```
so_many_things.each do |thing|
  binding.pry if thing.some_property.nil?
end
```

Context Commands

Get your bearings by seeing the surrounding code

`whereami`	Show code surrounding the current context
`cd`	Move into a new context (object or scope)
`ls`	Show the list of variables and methods in the current scope

Explore Objects

Inspect Ruby objects and their methods

Changing Context

Inspect a different object by changing context with `cd`

`cd Post`	Change context to the Post object
`nesting`	Show nested contexts
`cd ..`	Move up a context
`cd /`	Return to the root context

PRY CHEAT SHEET

```
pry(main)>
```

Listing Methods

Append options to `ls` to narrow things down

`-m, --methods`	Show public methods defined on the Object (default)
`-M, --module`	Show 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 locals, including those provided by Pry (in red)
`-c, --constants`	Show constants, highlighting classes (in blue), and exceptions (in purple)
`-i, --ivars`	Show instance variables (in blue) and class variables (in bright blue)
`-G, --grep`	Filter output by regular expression
`-h, --help`	Show additional options

Step Through Code

Navigate through code execution (requires pry-byebug)

`step`	Step execution **into** the next line or method
`next`	Step **over** to the next line within same frame
`finish`	Execute until current stack frame returns
`continue`	Continue execution and end Pry session

Edit Source Code

Edit the source of files, objects and methods in your app

`edit app/models/post.rb`	Edit a file
`edit Post`	Edit an object
`edit Post#a_method`	Edit a method
`edit -c`	Edit current context

Shell Mode

Run shell commands from within a Pry session

`shell-mode`	Enable shell mode
`.cd app/models`	Change directory
`.ls app/controllers`	List directory
`.cat app/models/post.rb`	Print contents of a file
`.vim app/models.post.rb`	Open file in Vim

Debug An Error

Figure out what's going wrong

`wtf?`	Print most recent error's stack trace (add additional !'s and ?'s to print more lines :D)
`show-method`	Print the source of a method
`show-doc`	Prints the documentation for an object

Exit Pry

Bug fixed? Get out of the Pry session

`exit`	Exit the session and continue execution
`Ctrl + d`	Keyboard shortcut for exit
`!!!`	Exit the session but halt further execution
`disable-pry`	Continue execution and prevent Pry from running again until server is restarted