

Five Rooby Tips



Justin Burris

neo

The Five Tips

1. The Quick Draw
2. The Reductionist
3. The Expressive Regular
4. The Exorcist
5. The Hired Gun

Tip One : The Quick Draw

ruby 2.0+

options and keyword arguments

Tip One : The Quick Draw

Options arguments

```
def hash_args(options: {})  
  options[:one_of_many]  
end
```

```
hash_args({one_of_many: "one of all",  
          the_underrated_one: "I'll never be returned"})  
=> "one of all"
```

Tip One : The Quick Draw

Keyword arguments

```
def hash_args(options: {})  
  options[:one_of_many]  
end
```

```
def keyword_args(one_of_many: nil, the_underrated_one: nil)  
  one_of_many  
end
```

```
keyword_args({one_of_many: "one of all",  
              the_underrated_one: "I'll never be returned"})  
=> "one of all"
```

Tip Two : The Reductionist

ruby 1.8.7+

map & reduce

Tip Two : The Reductionist

Map

```
secret_value = 10  
[1,2,3].map do |number|  
  number * secret_value  
end
```

```
=> [10,20,30]
```

Tip Two : The Reductionist

Reduce

```
[1,2,3].reduce(0) |accumulator, number|  
  accumulator + number  
end
```

=> 6

Tip Two : The Reductionist

Reduce

```
[1,2,3].reduce(0) | accumulator, number |  
  accumulator + number  
end
```

=> 6

bonus!

If you have a symbol, reudce is a breeze

```
[1,2,3].reduce(0, :+)
```

=> 6

**Map is also known as
collect**

Reduce is also known as
inject

Tip Three : The Expressive Regular

ruby 1.8.6+

Generate complex regular expressions with less escaping

Tip Three : The Expressive Regular

Using // to deliniate a regular expression

`/some_text/`

Tip Three : The Expressive Regular

Using // to deliniate a regular expression

/some_text/

/http:\\/\\/www.mysite.com\\/\\/ # ugh what?

Tip Three : The Expressive Regular

Using %r and {}

```
%r{http://mysite.com/} # My sanity, it has returned!
```

Tip Four : The Exorcist

ruby 1.9.1+

daemon and fork

Tip Four : The Exorcist

Sending a process to the background

`Process.daemon`

But what if you want a

PID?

Tip Four : The Exorcist

Use Fork!

```
if pid = fork
  File.write('/var/run/super_awesome_ruby_program.pid', pid)
  Process.detach(pid)
  exit
end
```

Tip Five : The Hired Guns

- [rubular](#)
- [pry-remote](#)
- [graphite](#)
- [remark.js](#)

Rubular

a Ruby regular expression editor

Your regular expression:

/ /

Your test string:

Rubular is a Ruby-based regular expression editor. It's a handy way to test regular expressions as you write them.

To start, enter a regular expression and a test string. Or you can [try an example](#).

Wrap words ☒ Show invisibles ☐ Ruby version

make permalink

clear fields

Regex quick reference

[abc]	A single character of: a, b, or c	.	Any single character	(...)	Capture everything enclosed
[^abc]	Any single character except: a, b, or c	\s	Any whitespace character	(a b)	a or b
[a-z]	Any single character in the range a-z	\S	Any non-whitespace character	a?	Zero or one of a
[a-zA-Z]	Any single character in the range a-z or A-Z	\d	Any digit	a*	Zero or more of a
^	Start of line	\D	Any non-digit	a+	One or more of a
\$	End of line	\w	Any word character (letter, number, underscore)	a{3}	Exactly 3 of a
\A	Start of string	\W	Any non-word character	a{3,}	3 or more of a

Tip Five : The Hired Guns

pry-remote

Dig into those pesky errors, in your console

<https://github.com/Mon-Ouie/pry-remote>

Tip Five : The Hired Guns

pry-remote

Dig into those pesky errors, in your console

<https://github.com/Mon-Ouie/pry-remote>

```
require 'pry-remote'
```

```
class FooController
  def bar_action
    binding.remote_pry
  end
end
```

www.website.com/foo/bar_action

Tip Five : The Hired Guns

pry-remote

Dig into those pesky errors, in your console

<https://github.com/Mon-Ouie/pry-remote>

```
require 'pry-remote'

class FooController
  def bar_action
    binding.remote_pry
  end
end
```

`www.website.com/foo/bar_action`

```
[pry-remote] Waiting for client on drb://localhost:9876
$ pry-remote
```

Tip Five : The Hired Guns

pry-remote

Dig into those pesky errors, in your console

<https://github.com/Mon-Ouie/pry-remote>

```
require 'pry-remote'

class FooController
  def bar_action
    binding.remote_pry
  end
end

www.website.com/foo/bar_action

[pry-remote] Waiting for client on drb://localhost:9876
$ pry-remote
```

Combine with `better_errors` for even more flexibility in debugging

Tip Five : The Hired Guns

Graphite

Log everything over UDP!

<http://graphite.wikidot.com/>

Tip Five : The Hired Guns

Graphite

Log everything over UDP!

<http://graphite.wikidot.com/>

In real time!

Tip Five : The Hired Guns

Remark.js

<https://github.com/gnab/remark>

<http://remarkjs.com/>

Thanks!

github.com/justinbburris/slides

Twitter

@justinbburris

GitHub

/justinbburris