

Mad Science: Polyglot Bridges

RubyNation 2015

Brock Wilcox
awwaiid@thelackthereof.org



polyglot



Web

Videos

Images

Books

News

More ▾

Search tools

About 4,800,000 results (0.27 seconds)

pol·y·glot

/ˈpālē,glät/

adjective

1. knowing or using several languages.
"a polyglot career woman"

noun

1. a person who knows and is able to use several languages.



Translations, word origin, and more definitions

"or using" ... hmmm

JVM

Java

JRuby

Scala

Clojure

JavaScript

Opal

CoffeeScript

ClojureScript

C

Libraries

Languages!

Use case: graph some stuff from Perl6/Rakudo

```
> sudo apt-get install matplotlib
```

```
# Python
```

```
import matplotlib.pyplot as plt
```

panda install Inline::Python

Perl6 (Rakudo) → Python

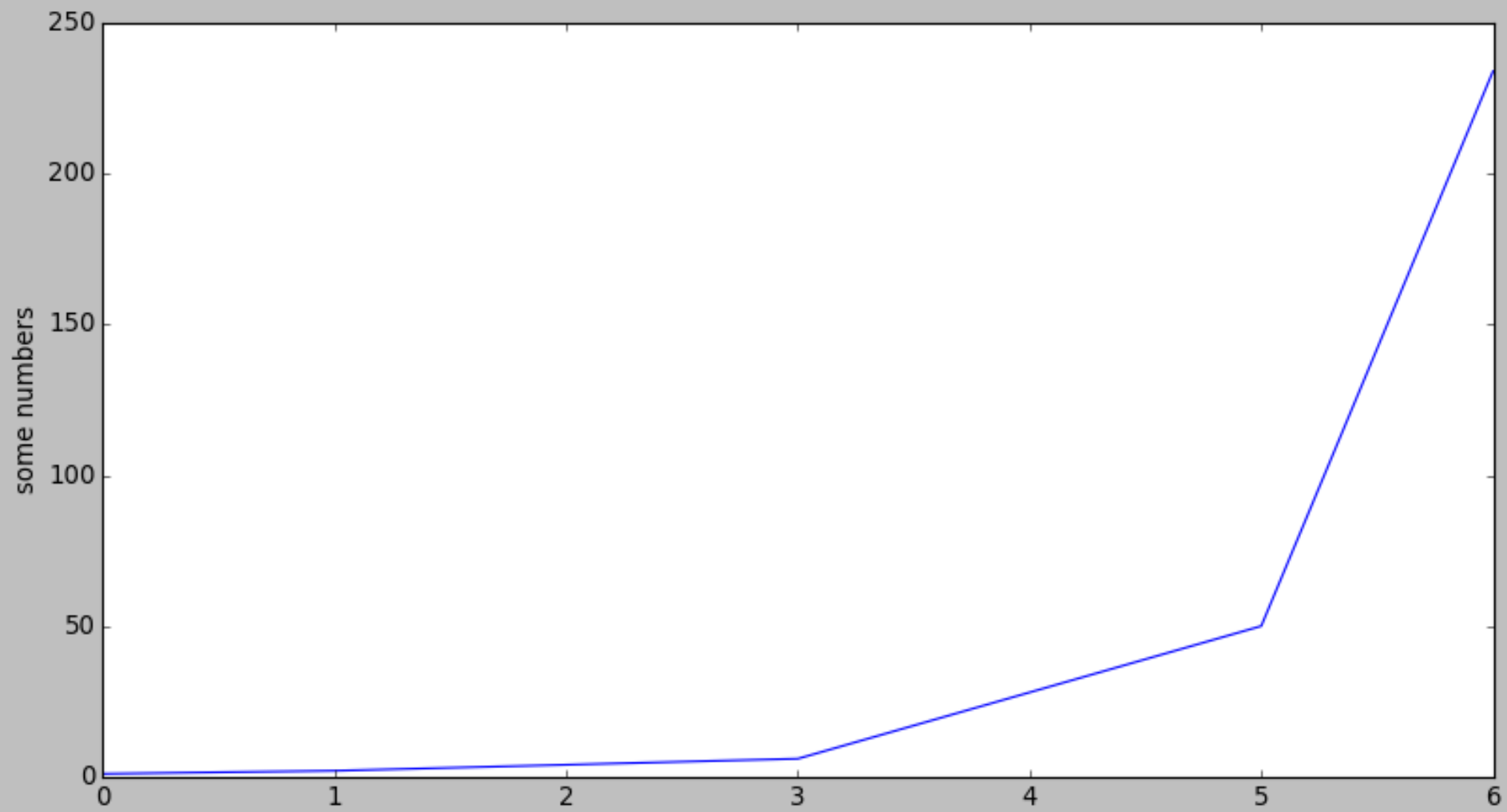
```
#!/usr/bin/env perl6

use lib 'lib';
use Matplotlib;

my $plt = Matplotlib.new;
$plt.plot([1,2,4,6,28,50,234]);
$plt.ylabel('some numbers');
$plt.show;
```



~



Ruby?

rubypython

Description

RubyPython is a bridge between the Ruby and Python interpreters. It embeds a running Python interpreter in the Ruby application's process using FFI and provides a means for wrapping, converting, and calling Python objects and methods.

RubyPython uses FFI to marshal the data between the Ruby and Python VMs and make Python calls. You can:

- Inherit from Python classes.
- Configure callbacks from Python.
- Run Python generators (on Ruby 1.9.2 or later).

```
gem install rubypython # !!!
```

(small patch to fix lib)

Ruby (YARV) → Python

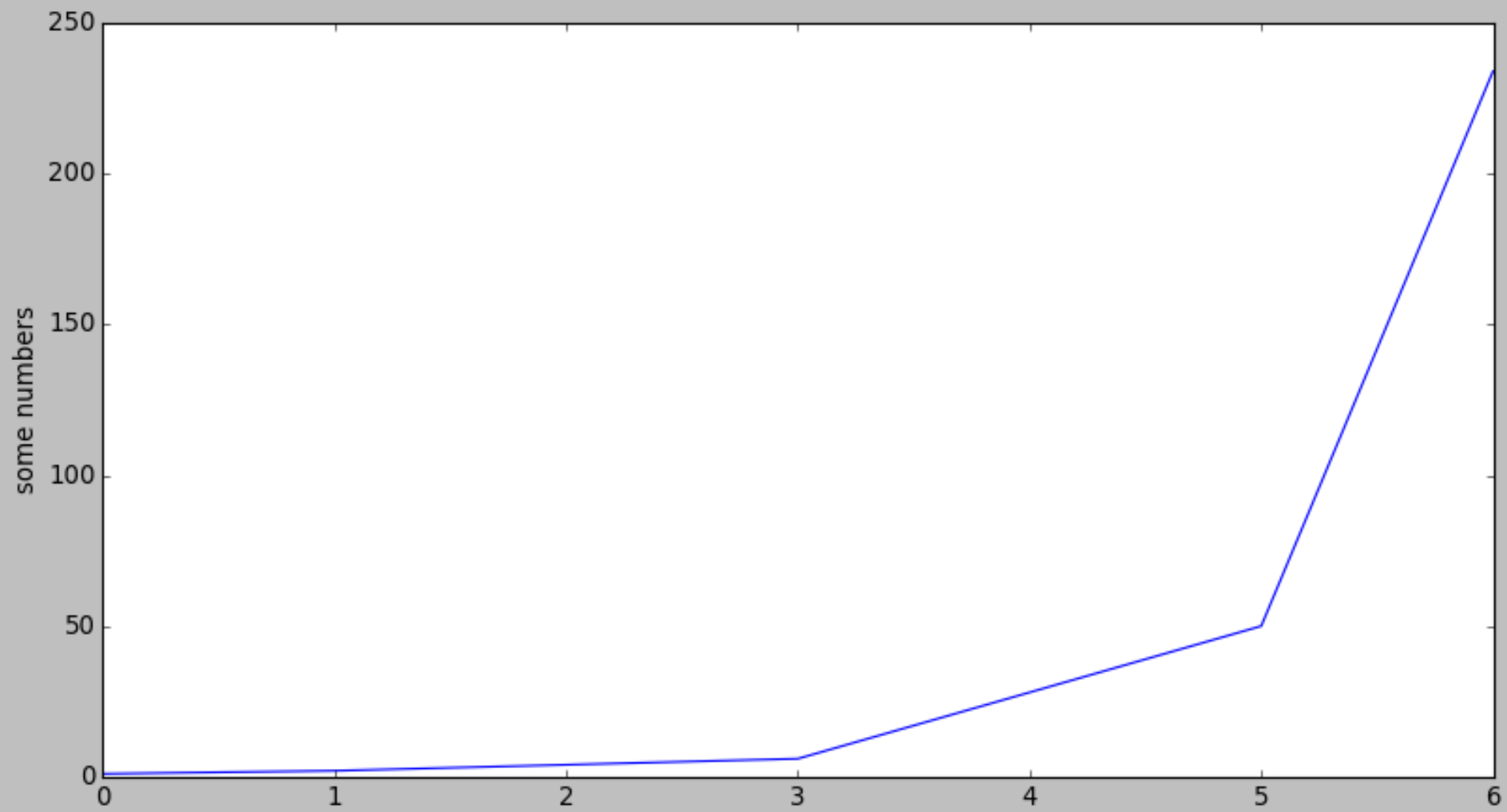

```
#!/usr/bin/env ruby

require 'rubypython'
RubyPython.start
plt = RubyPython.import('matplotlib.pyplot')

# ... and now we have all the power!

plt.plot([1, 2, 4, 6, 28, 50, 234])
plt.ylabel('some numbers');
plt.show

~
~
~
```



```
#!/usr/bin/env ruby

require 'rubypython'
RubyPython.start
plt = RubyPython.import('matplotlib.pyplot')
np = RubyPython.import('numpy')

# Based on "Stove Ownership" from XKCD by Randall Monroe
# http://xkcd.com/418/

plt.xkcd()

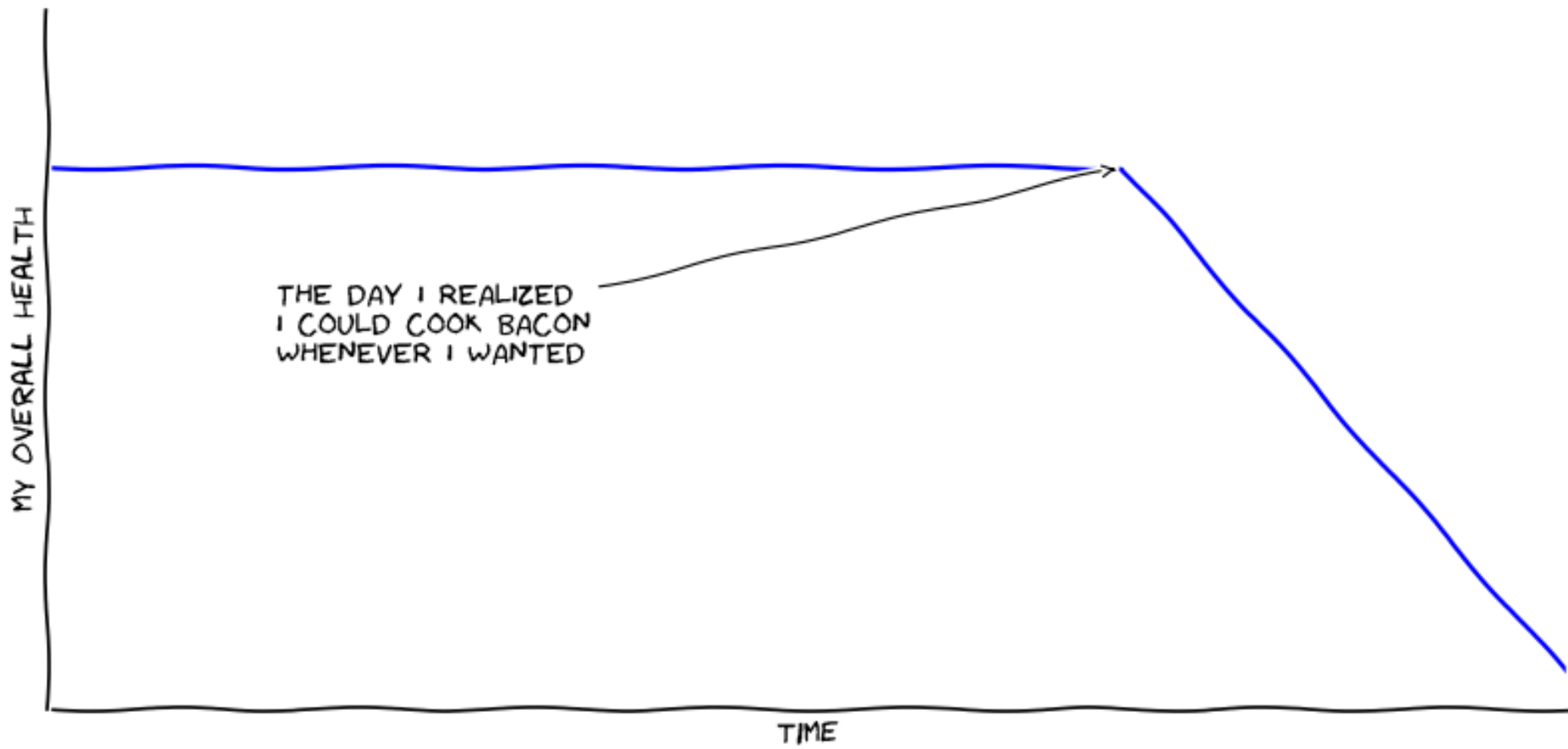
fig = plt.figure()
ax = fig.add_axes([0.1, 0.2, 0.8, 0.7])
ax.spines['right'].set_color('none')
ax.spines['top'].set_color('none')
plt.xticks([])
plt.yticks([])
ax.set_ylim([-30, 10])

data = np.ones(100)
data[(70..99).to_a] -= np.arange(30)

plt.annotate!(
  "THE DAY I REALIZED\nI COULD COOK BACON\nWHENEVER I WANTED",
  xy: [70, 1], arrowprops: {arrowstyle: '->'}, xytext: [15, -10])

plt.plot(data)

plt.xlabel('time')
plt.ylabel('my overall health')
fig.text(
  0.5, 0.05,
  '"Stove Ownership" from xkcd by Randall Monroe',
  ha: 'center')
```



"STOVE OWNERSHIP" FROM XKCD BY RANDALL MONROE



FFI... so sweet!

Ruby (JVM) → Python

Also works!

Exercise for the reader:

Pick a model. Visualize it!


```
require 'rubypython'  
RubyPython.start  
plt = RubyPython.import('matplotlib.pyplot')  
  
# The 'left' slices it down to year-month  
order_counts = Order  
  .group("left(created_at, 7)")  
  .count  
  .values  
  
plt.plot(order_counts)  
  
plt.show
```

THE END