syngenio

We make IT work.

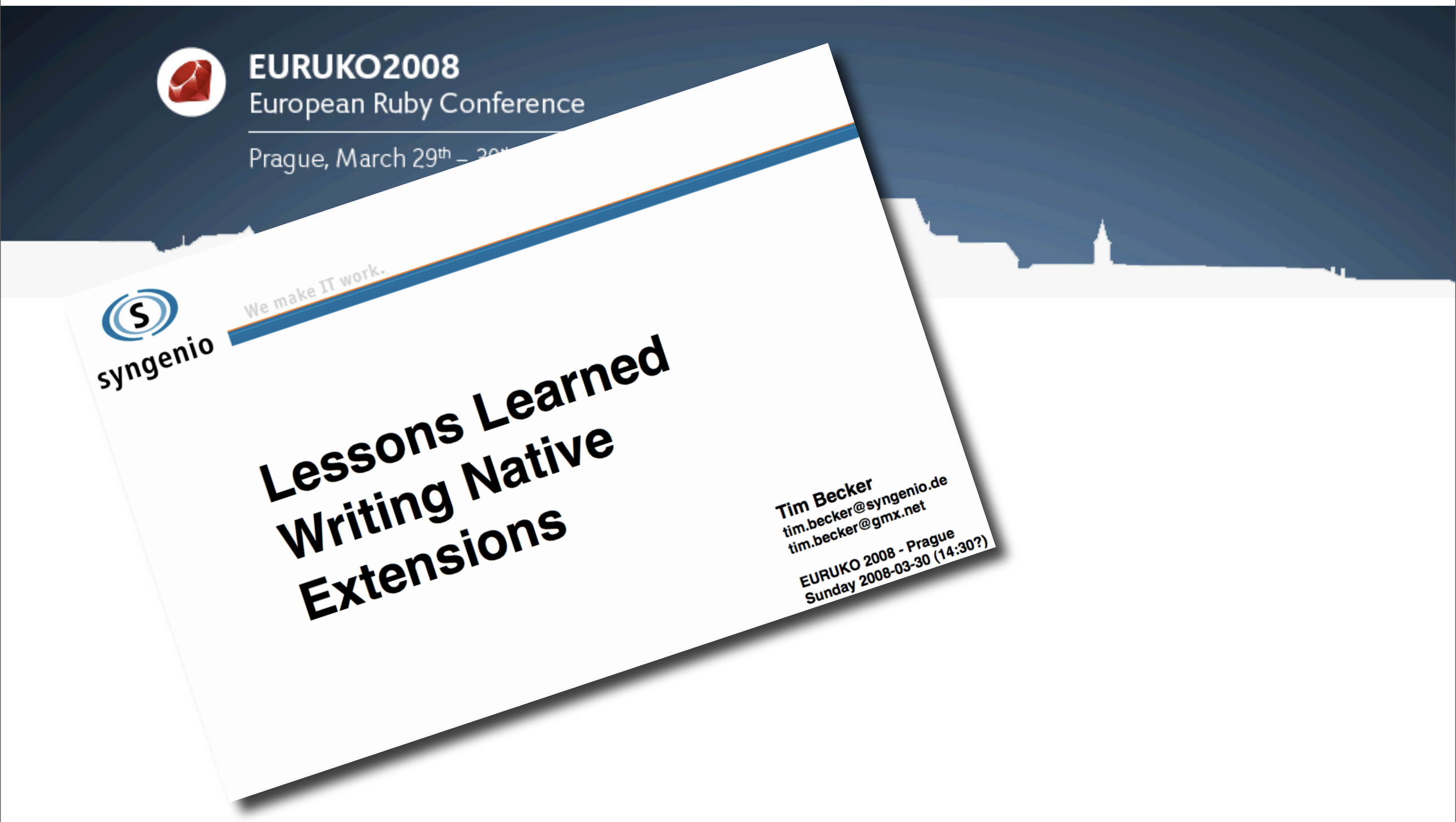
Introducing Weave

Tim Becker

tim.becker@syngenio.de

tim.becker@gmx.net


EURUKO2008
European Ruby Conference
Prague, March 29th – 30th


EURUKO2008
European Ruby Conference
Prague, March 29th –
We make IT work.
syngenio
Lessons Learned
Writing Native
Extensions
Tim Becker
tim.becker@syngenio.de
tim.becker@gmx.net
EURUKO 2008 - Prague
Sunday 2008-03-30 (14:30?)

EURUKO2008

European Ruby Conference

Prague, March 29th – 30th

Tim Becker – „Lessons Learned Writing Native Extensions“

Type-along tutorial on how to write C extension for Ruby. Very interesting, this could actually make me write C code again... He has started talking on cats and tigers and it seems like he wants to teach us how arrays work in C. Booooooooring. Finally he is done with this and is back on the interesting topics like conversion of data types. Overall a really interesting talk. By far the best one today so far. Tim's post with code samples and links.

Les
Writing
Extensions

EURO
Sunday 20

```
VALUE rb_define_class
(const char *name, VALUE super);

void rb_define_method
(VALUE klass, const char *name,
        VALUE (*func)(), int argc);
```

... `Array.sort` is too slow to sort the internet, so I monkeypatched it natively ...


#######
#   # #   # #   # #   # #
#   #   # #   # #   # #
#
#   #   # #   # #   # #
#   # #   # #   # #   # #
#######
_____ _____ _____ _____
\ /		\ /		\ /		\ /	
+---+		+---+		+---+		+---+	
|ESC| ||	[	||	3	||	N		
+---+		+---+		+---+		+---+	
/_____|	/_____|	/_____|	/_____|
#
#
#   # # # #
##
#
#####
#   # #   # #   # #   #
#   #   # #   #
#####
#   #   # #   #   #
#   # #   # #   # #   #
#####
gms_.
_mms d@@@@[
M@@@@W. i@@@@@@ _mWs.
]@@@@@@m@@@@@@@@ms_gW@@@@@
,_. ,@@@@@@@@@@@@@@@@@@@@@@@@@!
g@@@@msW@@@@@@@@********@@@@@@@@@[
M@@@@@@@@@@@*~ ~*@@@@@@s ,___
~M@@@@@@Af VM@@@@@@@@@
i@@@@@f V@@@@@@@@
_____@@@@@f V@@@@@Af
]@@@@@@@@@P Y@@@@[
@@@@@@@@@@` '@@@@W
'~**@@@@@ @@@@@_
@@@@@. ,@@@@@@
Y@@@@b d@@@@@@
(c) <DATE OVERFLOW>
POWERED BY COBOL ON COGS

When you really want to catch a coder by surprise, a monkey doesn’t cut it. What you need is a **Ninja**.

When you really want to catch a coder by surprise, a monkey doesn't cut it. What you need is a **Ninja**.

... silent, untraceable, precise, unpredictable, and always deadly....

When you really want to catch a coder by surprise, a monkey doesn't cut it. What you need is a **Ninja**.

... silent, untraceable, precise, unpredictable, and always deadly....

And unlike monkey patching (...) Ninja-Patching happens when you least expect it.

Monkey Patching Demo

A Web Server Called *Ebb*

Ebb aims to be a small and fast web server specifically for hosting dynamic web applications. It is not meant to be a full featured web server like Lighttpd, Apache, or Nginx. Rather it should be used in multiplicity behind a load balancer and a front-end server. It is not meant to serve static files in production.

At one level Ebb is a minimalist C library that ties together the Mongrel state machine and libev event loop. One can use this library to drive a web application written in C. (Perhaps for embedded devices?) However, most people will be interested in the binding of this library to the Ruby programming language. The binding provides a Rack server interface that allows it to host Rails, Merb, or other frameworks.

A Web Server Called *Ebb*

Ebb aims to be a small and fast web server specifically for hosting dynamic web applications. It is not meant to be a full featured web server like Lighttpd, Apache, or Nginx. Rather it should be used in multiplicity behind a load balancer and a front-end server. It is not meant to serve static files in production.

At one level Ebb is a minimalist C library that ties together the Mongrel state machine and libev event loop. One can use this library to drive a web application written in C. (Perhaps for embedded devices?) However, most people will be interested in the binding of this library to the Ruby programming language. The binding provides a Rack server interface that allows it to host Rails, Merb, or other frameworks.

A Web Server Called *Ebb*

Ebb aims to be a small and fast web server specifically for hosting dynamic web applications. It is not meant to be a full featured web server like Lighttpd, Apache, or Nginx. Rather it should be used in multiplicity behind a load balancer and a front-end server. It is not meant to serve static files in production.

At one level Ebb is a minimalist C library that ties together the Mongrel state machine and libev event loop. One can use this library to drive a web application written in C. (Perhaps for embedded devices?) However, most people will be interested in the binding of this library to the Ruby programming language. The binding provides a Rack server interface that allows it to host Rails, Merb, or other frameworks.

A Web Server Called *Ebb*

Ebb aims to be a small and fast web server specifically for hosting dynamic web applications. It is not meant to be a full featured web server like Lighttpd, Apache, or Nginx. Rather it should be used in multiplicity behind a load balancer and a front-end server. It is not meant to serve static files in production.

At one level Ebb is a minimalist C library that ties together the Mongrel state machine and libev event loop. One can use this library to drive a web application written in C. (Perhaps for embedded devices?) However, most people will be interested in the binding of this library to the Ruby programming language. The binding provides a Rack server interface that allows it to host Rails, Merb, or other frameworks.

Ruby people like meta.

Ruby people like meta.

Ruby people like metameta.

Ruby people like meta.

Ruby people like metameta.

Ruby people like self-referentiality.

Ruby people like meta.

Ruby people like metameta.

Ruby people like self-referentiality.

Ruby people like meta.

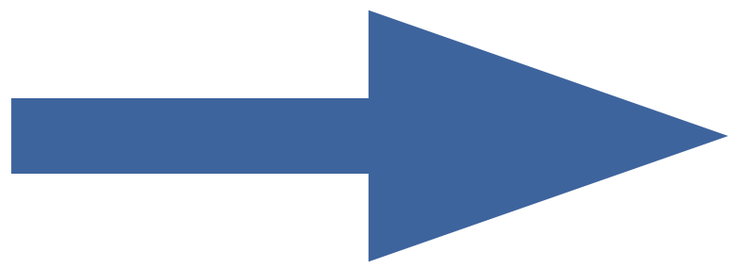
Ruby people like metameta.

Ruby people like self-referentiality.

Ruby people like meta.

Ruby people like metameta.

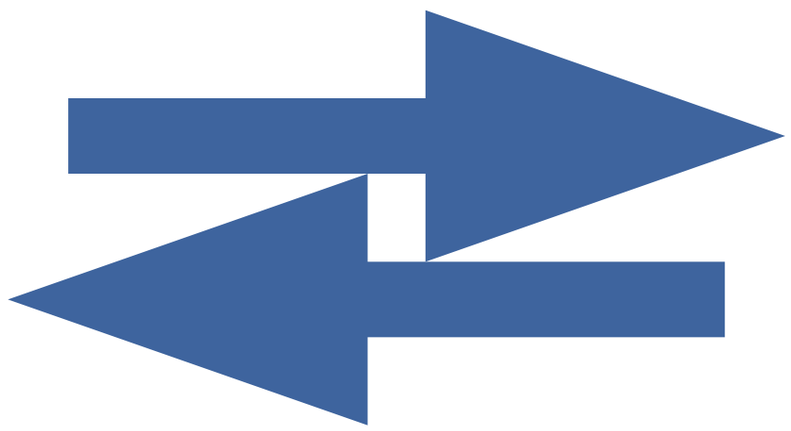
Ruby people like self-referentiality.



Ruby people like meta.

Ruby people like metameta.

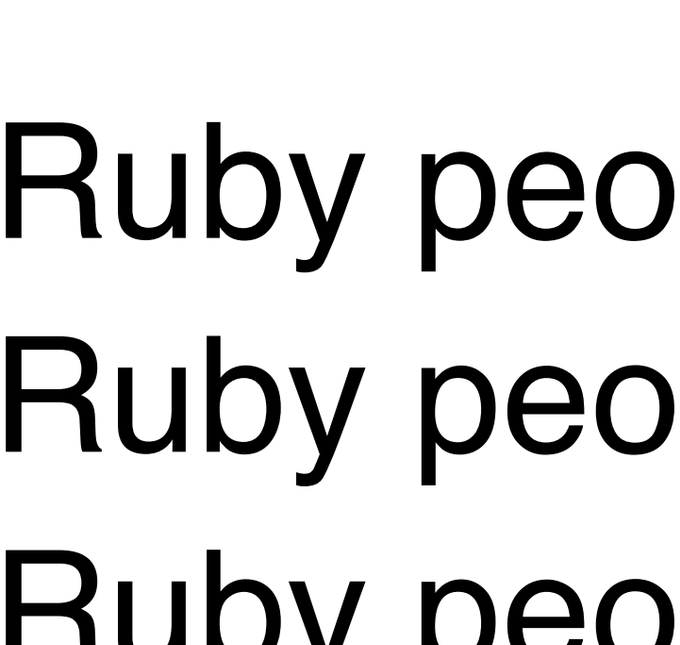
Ruby people like self-referentiality.

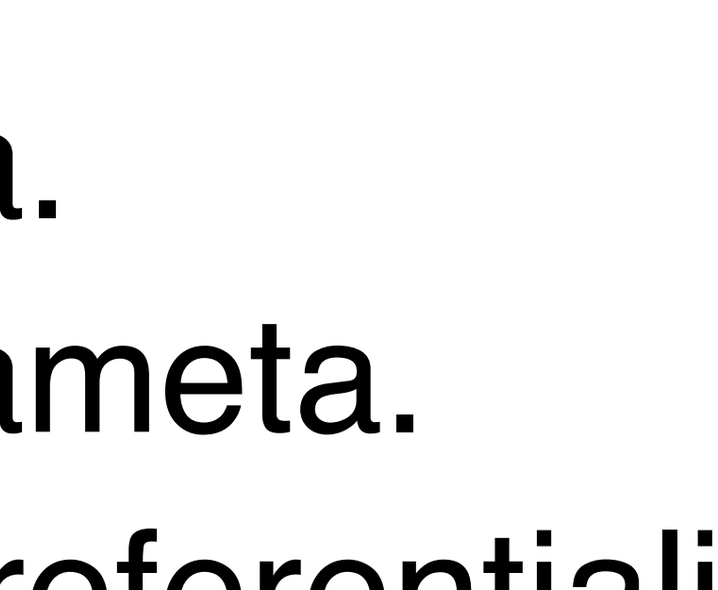


Ruby people [illegible]meta.

Ruby peop[illegible]etameta.

Ruby people [illegible]self-referentiality.

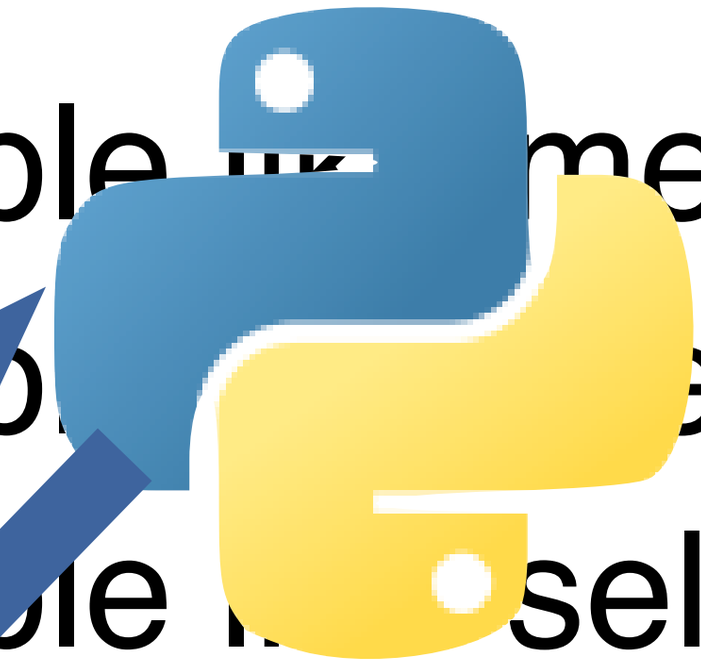


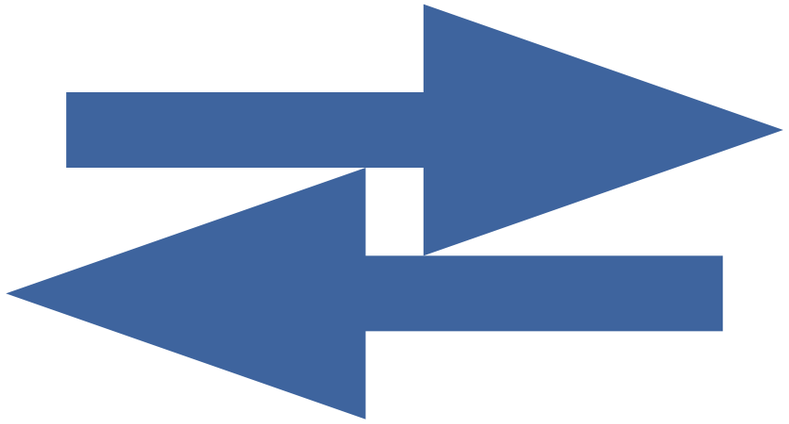


Ruby people like metaprogramming.

Ruby people like metametaprogramming.

Ruby people like self-referentiality.

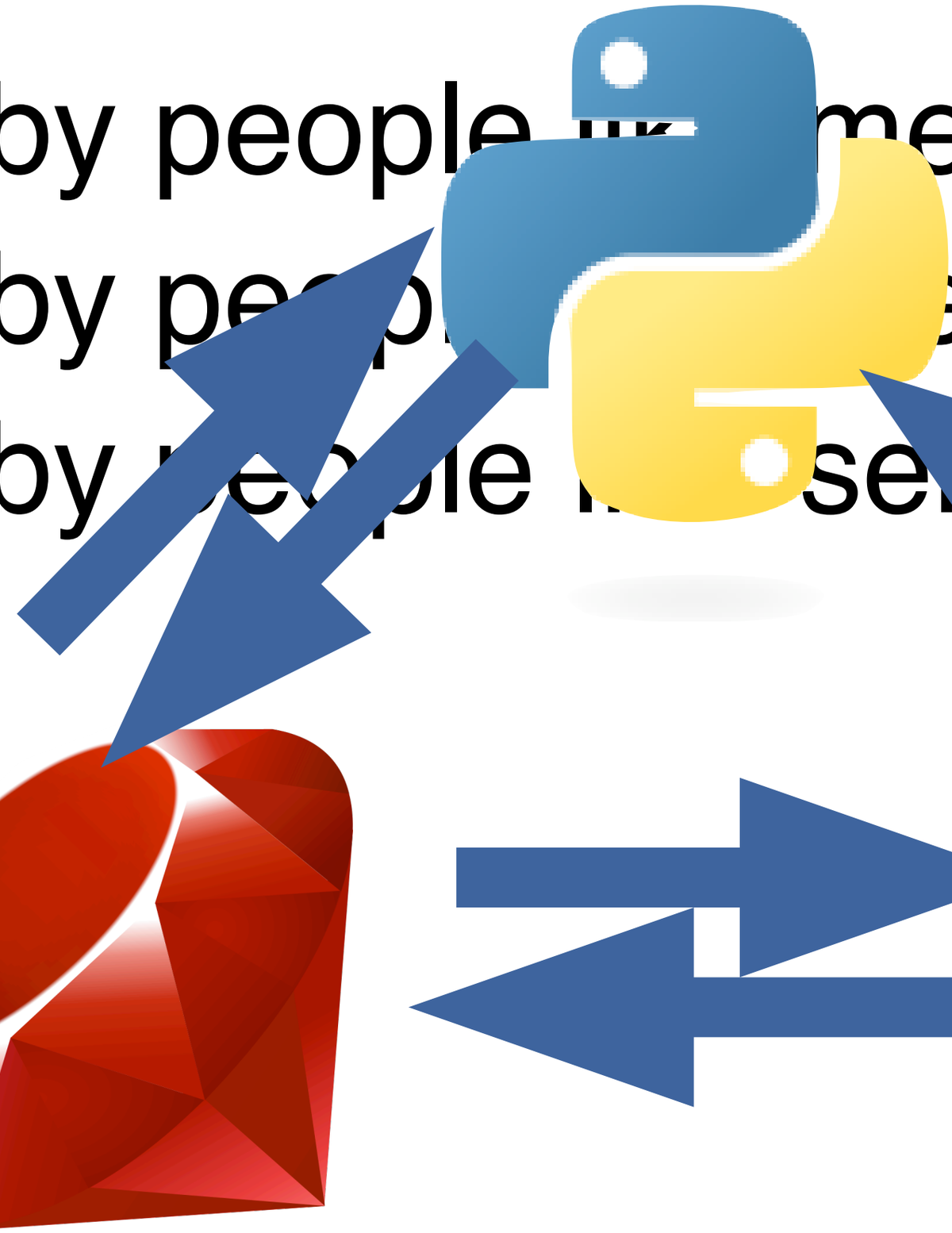


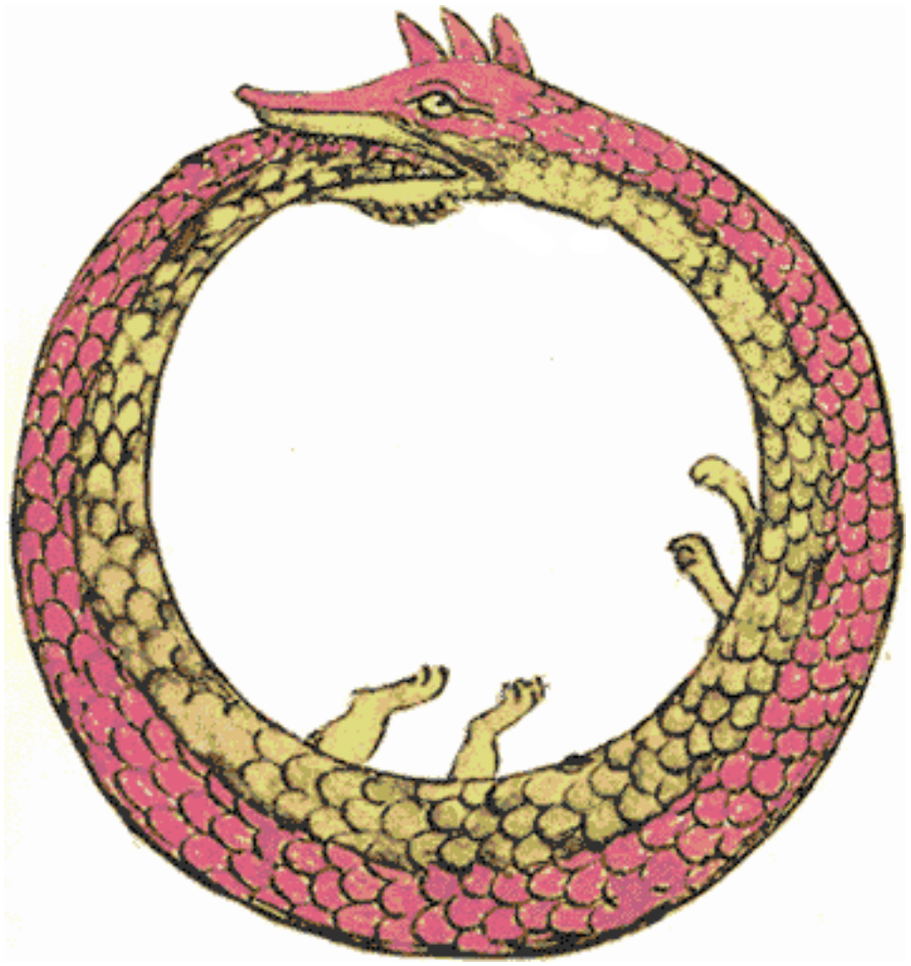


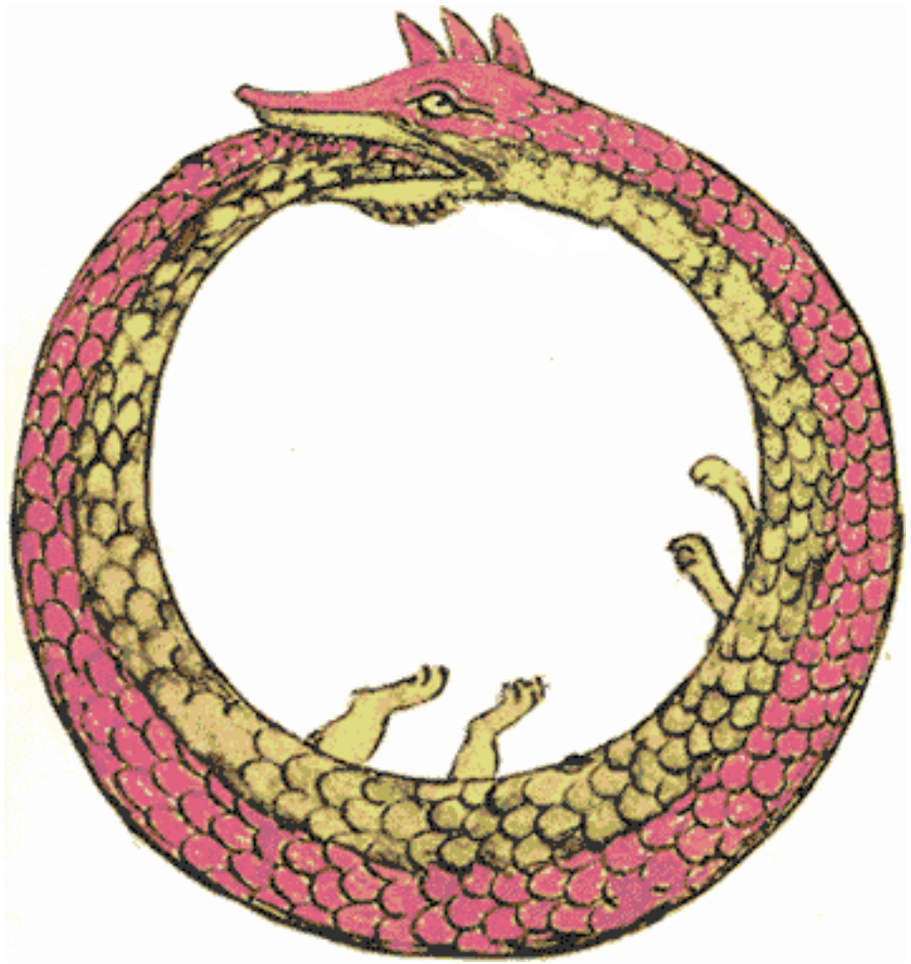
Ruby people

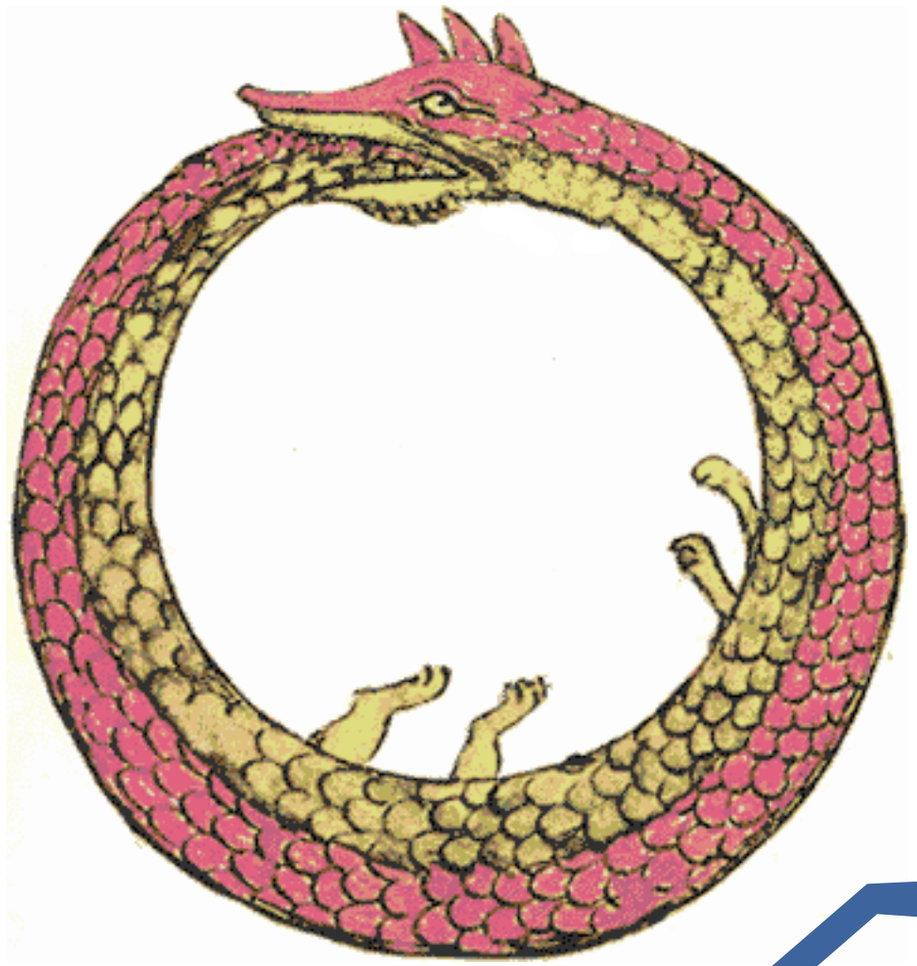
Ruby people

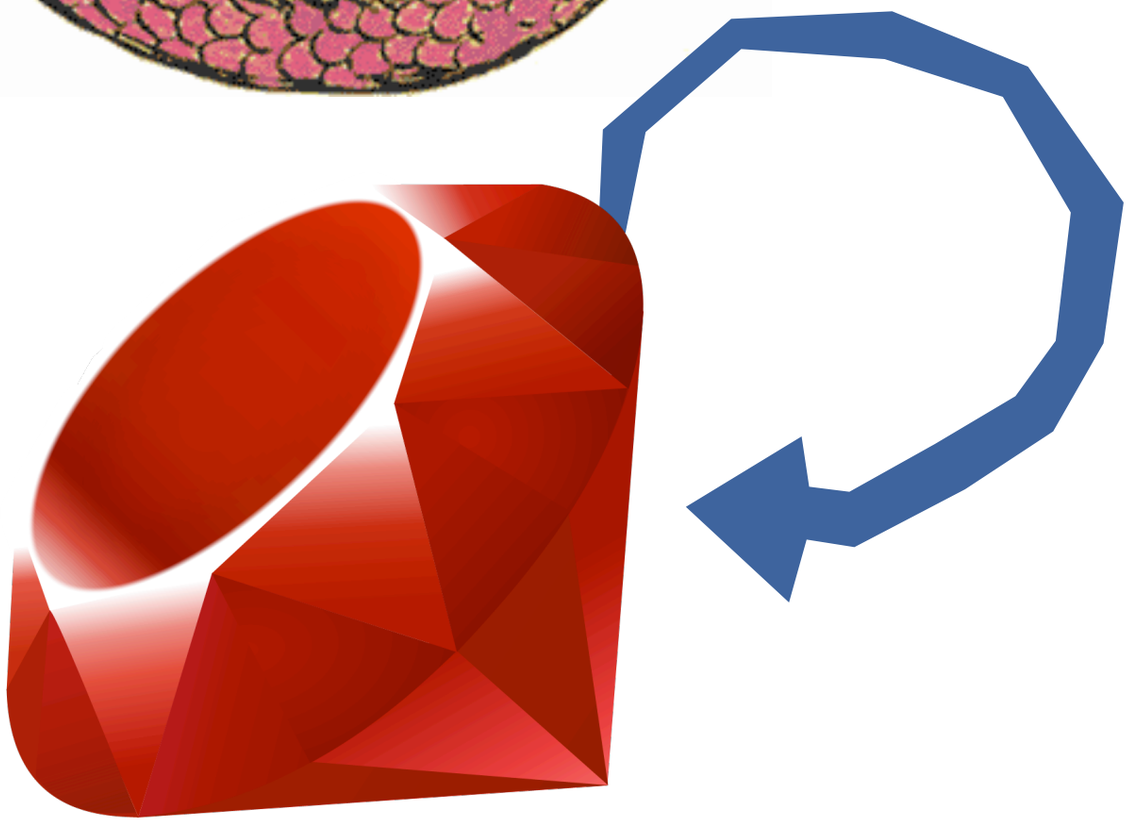
Ruby people











```
VALUE rb_define_class
(const char *name, VALUE super);

void rb_define_method
(VALUE klass, const char *name,
        VALUE (*func)(), int argc);
```

```
module Weave
  def self.rb_define_method(
        klass, name, sym, arity)
    # (...)
  end
  def self.rb_define_singleton_method(
        object, name, sym, arity)
    # (...)
  end
end
```

Weave:

Weave:

monkey patching

Weave:

native

monkey patching

Weave:

live

native

monkey patching

Weave:

red hot

live

native

monkey patching

Monkey Weaving Demo

Links

Ninja Patching:
http://avdi.org/devblog/2008/04/01/announcing-ninja-patching/

Ebb:

http://ebb.rubyforge.org/

Slides:

http://blog.kuriositaet.de/

Weave:

http://weave.rubyforge.org/

Thank you!

Questions?

No, it has no practical use.