How to Use “Kagemusha"

2007-09-29 Tatsuhiro Ujihisa
at Ruby Kansai Workshop#19

- Tatsuhiro Ujihisa
- ruby, haskell, lisp, javascript
- Fuzzy Rough Sets Theory
  - using ruby and haskell
- writer of Kagemusha README

What

- “Kagemusha is a library of helper functions for testing Ruby scripts."

  -- Kagemusha README

Who

- Yuya Kato
- My teacher of Ruby

```
class Fixnum
  def to_s
    "ujihisa"
  end
end

puts 23412
#=> ujihisa
```

Let's try on IRB

Open Class

- global side-effects

```
"abc" #=> 1
:abc  #=> nil
3     #=> "a"
3 + 5 #=> "aa"

etc...
It's crazy.
```

»Open Class with Scope?

Kagemusha

```
require 'rubygems'
require 'kagemusha'

m = Kagemusha.new Fixnum
m.def :to_s do
  "ujihisa"
end
m.swap do
  # write what you want to do
  puts 1
end
```

```ruby
class Fixnum
  def to_s
    "ujihisa"
  end
end

puts 23412
#=> ujihisa
```

```ruby
require 'rubygems'
require 'kagemusha'

m = Kagemusha.new Fixnum
m.def :to_s do
  "ujihisa"
end
m.swap do
  # write what you want to do
  puts 1
end
```

Remember these:

- requires
- Kagemusha.new
- Kagemusha#def, defs
- Kagemusha#swap

That's All.

Open Class	Kagemusha

	`require 'rubygems'`
	`require 'kagemusha'`

`class Fixnum`	`m = Kagemusha.new`

`def add(i)`
`  self + i`
`end`
	`m.def :to_s do |i|`
	`  self + i`
	`end`

`def self.add(i, j)`
`  i + j`
`end`
	`m.defs :add do |i, j|`
	`  i + j`
	`end`

Other styles

```
Kagemusha.new(A) do |m|
  m.def(:f) {|c| puts c }
  m.swap do
    a = A.new
    a.f 'blocked style'
  end
end
```

```
Kagemusha.new(A).
  def(:f) {|c| puts c }.
  swap do
    a = A.new
    a.f 'chained style'
  end
```

Exercise

- Kernel.rand is always 0.0
- Fixnum#+ means minus
- Hash#map returns not array but hash