dRuby

その導入編を松江で

seki@druby.org
twitter: @m_seki
hatena: id:m_seki

Agenda

- 私について
- dRubyの紹介
- 演習

重要なことを先に

- 先に

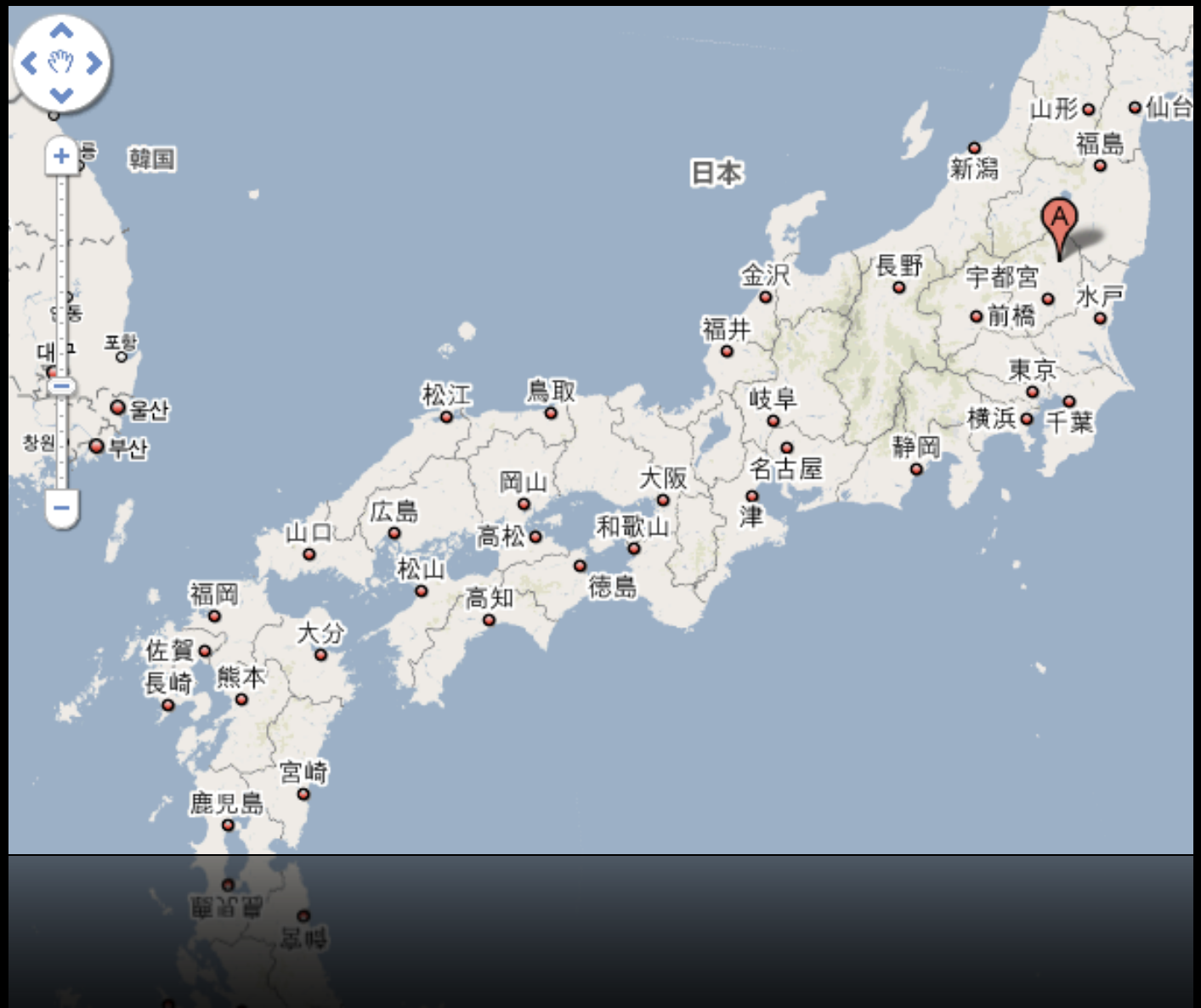
重要

- まだ初刷買えます!

私について

- 地理的な関係
- しごと関係
- Rubyとの関係

栃木県那須塩原市


韓国
日本
山形
仙台
福島
新潟
金沢
長野
宇都宮
水戸
前橋
福井
東京
岐阜
横浜
千葉
静岡
名古屋
津
大阪
和歌山
松江
鳥取
岡山
広島
高松
山口
松山
徳島
高知
福岡
大分
佐賀
長崎
熊本
宮崎
鹿児島
울산
부산
포항
창원
A

toRuby50回記念

- とちぎRubyの勉強会
- 毎月第一水曜日、西那須野公民館
- 50回記念イベント2011-2-26開催!

サンライズ出雲

- 東北新幹線→サンライズ出雲
- 今年は早めに予約
  - シングルとれた!!

しごと

- サラリーマン
- 自称アーティスト

ちゃんとした会社員

- わりと大きな組織
- 製品のためのプログラミング
- eXtreme Programmingだったもの
- ソフトウェアの品質系学会
  - 多くの場合、悪役として招待される

自称アーティスト

- 自分のため・自己中心
- アーティストとしてのプログラマ
- わかって欲しい人にウケるため
- オープンソースへの愛や正義には興味がない

Rubyと私

- 20世紀の終わり頃から
- ERB, dRuby, Rinda, RWiki, ...
- 2000年のPerl/RubyConference
- 書籍・雑誌

ERB

- 文書にRubyスクリプトを埋め込む
- Railsの実習で触った?
- 非常にこだわりがあるライブラリ
  - しかし利用者にはほとんど伝わらない

dRuby

- 今日のテーマです

今日は

- dRubyの基本的な機能を紹介し、実際に動かして実験します
- 細かいどうでもよい点（しかし私がていねいに設計した点）にはなるべく触れません
- 雰囲気を味わってください

今日は

- 二人で短いプログラムを写経
- 一度にたくさんのプログラムを起動します
  - やったことありますか？

RMI

- Remote Method Invocation
- 別のプロセス／マシンのオブジェクトにメッセージを送り、メソッドを起動する仕組み

いったい何がすごいの

- プロセスの境界
  - OSによって保護された空間
- プロセスの壁を越えるにはどうする?

サーバ

- 誰かに奉仕するプロセス
- たとえばWebサーバ
  - それが本業であるプログラム
  - そうなるべく最初から設計されてる
  - お願いを待ち受ける仕組を持つ

ふつうは

- 聞く耳を持たないプロセス
- 自分から周りの声を聴こうとしなくてはわからない

dRubyは

- そんなプロセスにも待ち受ける仕組みを提供する
- みんなをサーバにするライブラリ

Ruby風

- Webサービスとの違い
  - dRubyはRubyに特化している
  - 徹底的にRubyの常識に従う

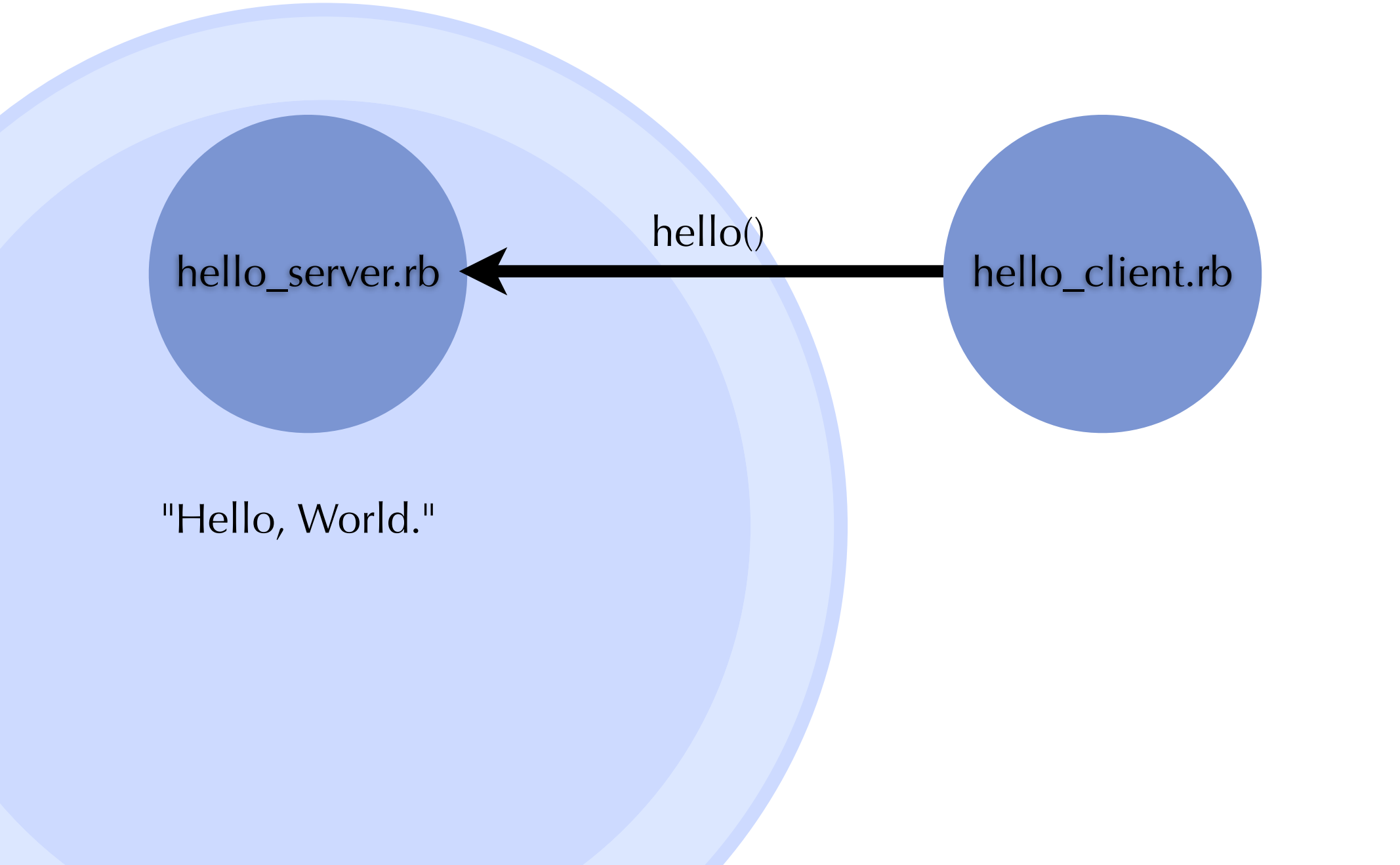
演習

- 1台のマシンの中の複数のプロセスが協調して動く様子を体験します

演習1

- Hello, World.
- いくつかの約束事を覚えます

演習1



hello_server.rb

```
require 'drb/drb'

class Hello
  def hello
    puts('Hello, World.')
  end
end

DRb.start_service('druby://localhost:54000', Hello.new)
while true
  sleep 1
end
```

require

```
require 'drb/drb'

class Hello
  def hello
    puts('Hello, World.')
  end
end

DRb.start_service('druby://localhost:54000', Hello.new)
while true
  sleep 1
end
```

DRb.start_service

```
require 'drb/drb'

class Hello
  def hello
    puts('Hello, World.')
  end
end

DRb.start_service('druby://localhost:54000', Hello.new)
while true
  sleep 1
end
```

URI

```
require 'drb/drb'

class Hello
  def hello
    puts('Hello, World.')
  end
end

DRb.start_service('druby://localhost:54000', Hello.new)
while true
  sleep 1
end
```

終了させない

```
require 'drb/drb'

class Hello
  def hello
    puts('Hello, World.')
  end
end

DRb.start_serv                                    new)
while true
  sleep 1
end
```

本来は sleep でよい

いくつかの環境向けのトリック

約束事

- require 'drb/drb'
- DRb.start_service
- URI
- 終了させない

hello_client.rb

```
require 'drb/drb'

DRb.start_service
ro = DRbObject.new_with_uri('druby://localhost:54000')
ro.hello
```

hello_client.rb

```
require 'drb/drb'

DRb.start_service
ro = DRbObject.new_with_uri('druby://localhost:54000')
ro.hello
```

hello_client.rb

```
require 'drb/drb'

DRb.start_service
ro = DRbObject.new_with_uri('druby://localhost:54000')
ro.hello
```

hello_client.rb

```
require 'drb/drb'

DRb.start_service
ro = DRbObject.new_with_uri('druby://localhost:54000')
ro.hello
```

約束事

- require 'drb/drb'
- DRb.start_service
- DRbObject.new_with_uri

```ruby
require 'drb/drb'

class Hello
  def hello
    puts('Hello, World.')
  end
end

DRb.start_service('druby://localhost:54000', Hello.new)
while true
  sleep 1
end
```

```ruby
require 'drb/drb'

DRb.start_service
ro = DRbObject.new_with_uri('druby://localhost:54000')
ro.hello
```

元はこれ

```
class Hello
  def hello
    puts('Hello, World.')
  end
end

ro = Hello.new
ro.hello
```

演習1

- hello_client.rbをなんども実行できますか？
- hello_server.rbを終了させてhello_client.rbを実行させるとどうなりますか？

使いどころ

- 空間
  - 一つのプロセスの扱える量を超える
- 時間
  - プロセスの寿命を超える (例 CGI)
- 機能による分割

演習2

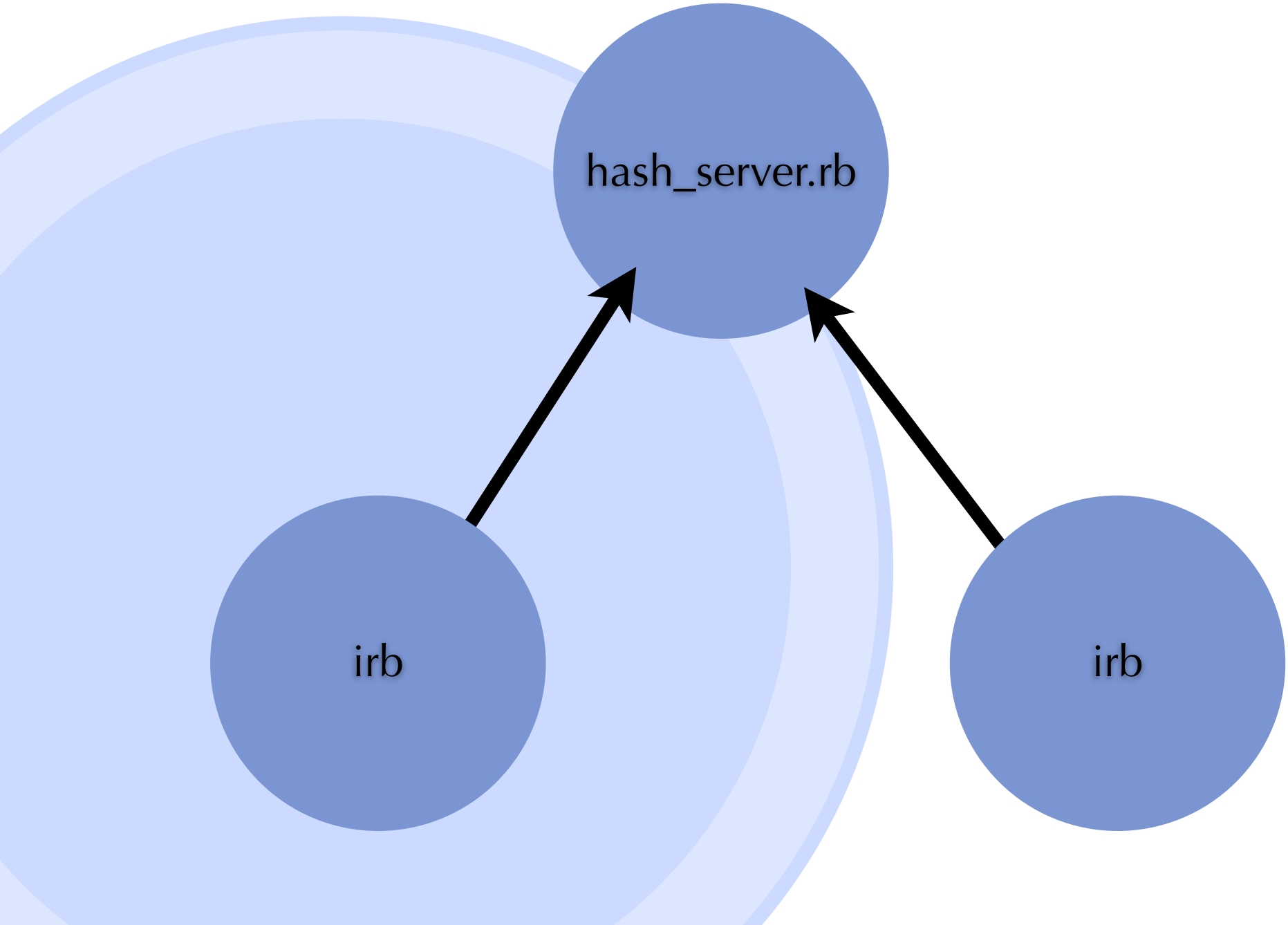
- 共有されたHashにいろいろ出し入れしてみる
- 複数のプロセスがオブジェクトを交換する様子を感じて

KVS...

```
require 'drb/drb'
require 'pp'

front = Hash.new
DRb.start_service('druby://localhost:54300', front)
while true
  sleep 10
  pp front
end
```

演習2



演習2

- どんなオブジェクトが入るか試して
  - String, Integer, IO

演習3

- 待ち行列を使った同期
- RubyのQueue
  - スレッド同期の仕組み

非同期と同期

- それぞれ勝手に動くプログラムを協調させる
- Queueは待合せと情報の交換を同時に行う

Queue

- FIFOバッファ
- pushとpop
- 空のときにpopするとブロックする
- データが届いたらまた動き出す

Queue

- スレッドの視点に立つと‥
  - 「待つ」のは取り出す係
  - 並べるのはオブジェクト
  - 並んで待つのではないことに注意

演習3

- データがない間、停まってるか？

```ruby
require 'drb/drb'
require 'thread'

DRb.start_service('druby://localhost:54320', Queue.new)
while true
  sleep 1
end
```

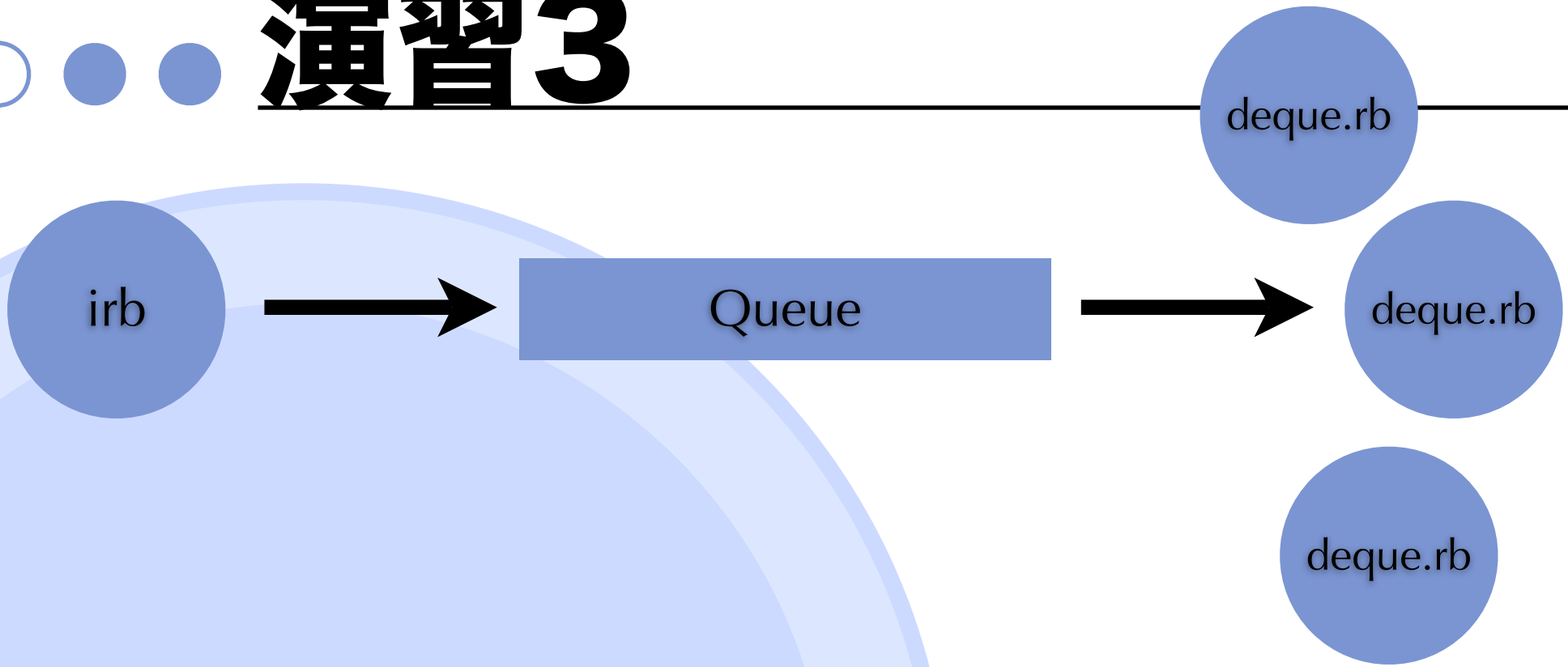
```ruby
require 'drb/drb

DRb.start_service
queue = DRbObject.
while true
  p queue.pop
  sleep(rand)
end
```

Queueからデータを一つpopして印字する

ランダムに少しsleepする

演習3



- deque.rbを増やしたらどうなる?

重要なことをもう一度

- もう一度

重要

- まだ初刷買えます!

まとめ

- 初刷五周年
- dRubyの雰囲気を味わえた？

ふりかえり