

## Ruby入門

日本Rubyの会  
吉田和弘

1

## 最近のRuby

- Ruby on Rails
- YARV -> RubyVM (Ruby 1.9)
- Ruby 1.8.6-p36
- JRuby, IronRuby, etc.
- Ruby会議2007

2

## Rubyの特徴

- スクリプト言語
- オブジェクト指向言語
- 動的言語

3

## スクリプト言語

- インタプリタ
- 実行時にも構文解析できる
  - require
  - eval
- 文法要素の多くが値を持つ
  - 制御構文も式
- トップレベルがある
  - 最初から式を書ける

4

## オブジェクト指向言語

- 統一データモデル
- 全ての値はObject
  - 1+1
  - 1.+(1)
- Mix-in
- 特異メソッド/クラス

5

## 動的言語

- リフレクション
  - クラス情報を実行時に操作できる
- 動的なメソッド呼び出し
  - send
  - method\_missing
- オープンクラス
  - いつでも/どこでも 再定義できる
- 全ては実行文

6

```
puts "Hello"
# Hello

puts 1+1
# 2
```

7

```
def hello(name)
  "Hello from #{name}"
end

puts hello("kaz")
# Hello from kaz
```

8

## クラスとインスタンス

```
class C
  def hello(name)
    "Hello from #{name}"
  end
end

obj = C.new
puts obj.hello("kaz")
# Hello from kaz
```

9

## インスタンス変数

```
class C
  def initialize(name)
    @name = name
  end
  def hello
    "Hello from #{@name}"
  end
end

obj = C.new("kaz")
puts obj.hello # Hello from kaz
```

10

## モジュールとmix-in

```
module M
  def hello
    "Hello from #{@name}"
  end
end

class C
  def initialize(name)
    @name = name
  end
  include M
end
```

11

## 継承

```
class B < C
  def bye
    ...
  end
end

obj = B.new("kaz")
obj.hello
obj.bye
```

12

## Rubyの本質

13

## 特異メソッド

```
obj = C.new("kaz")
def obj.bye
  ...
end

obj.bye
```

14

## 特異クラス

```
obj = C.new("kaz")
class << obj
  def bye
    ...
  end
end

obj.bye
```

15

## 継承関係

```
class C ... end
class C < Object ... end

obj = C.new(...)
p obj.class # C
p C.superclass # Object

# obj: C < Object
```

16

## モジュールの探索

```
module M ... end
class C
  include M
end

# obj: C < M < Object
```

17

## 特異クラスの探索

```
obj = C.new("...")
class << obj ... end # (obj)

p obj.class # C

# obj: (obj) < C < Object
```

18

## クラスメソッド

```
class C
  def self.hello
    puts "Hello"
  end
end

C.hello
```

19

```
class C
  class << self
    def hello
      puts "Hello"
    end
  end
end

end
```

20

## "Class"クラス

```
class C
  p self # C
end

p C.class # Class

# C: Class
# C: Class < Module < Object
```

21

## クラスの特異メソッド

```
def C.hello
  puts "Hello"
end

# C: (C) < (Object) < Class
```

22

## 関数ぽいメソッド

```
puts "Hello"

p self # main
p self.class # Object
```

23

## "Kernel"モジュール

```
class Object
  include Kernel
end

module Kernel
  def puts(...)
    ...
  end
end
```

24