

# Ruby Intro

2015-08-10

李磊

# 大纲

- 关于Ruby
- Syntax
- Meta Programming
- Tools & Libs

# 关于Ruby

- 泛型，解释型，反射，垃圾回收
- 为程序员而生，不是计算机
- 比Perl更强大，比Python更OOP
- 一切皆是object
- 动态类型
- Lightweight
- Cross Platform

一切都是对象

```
10.times do |t|  
  puts "#{t} hello world"  
end
```

```
"hello world".is_a?(String)  
# => true
```

```
"hello world".gsub(/world/, 'lei').upcase  
# => "HELLO LEI"
```

```
def say_hi_to(name)
  puts "Hi, " + name.capitalize
end
say_hi_to('lei')
# => "Hi, Lei"
```



```
"hello world".methods.grep(/case/)
# => [:casecmp, :upcase, :downcase, :swapcase, :upcase!, :downcase!, :swapcase!]
```

# Syntax

```
a = 1
b = 2.34
name = "lei"
name2 = :lei
persons = ["rose", "jack", "andy"]
info = { "name" => "lei", "age" => 20 }
info = { name: "lei", age: 20 }

re = /hello/i
range = 1..10

calculator = proc { |x, y| x + y }

def say_hi_to(name)
end

class Bird
end

module BirdHelpers
end
```

# Syntax

```
if var1
  # do something
elsif bvar2
  # do something
else
  # do something
end
```

# Syntax

```
unless var1  
  # do something  
elsif var2  
  # do something  
end
```

# Syntax

```
a = 1 unless defined? a  
b = 2 if a == 1
```

# Syntax

```
case var3
when 1
  # do something
when 2
  # do something
when /hello/
  # do something
else
  # do something
end
```

# Syntax

```
for x in [1, 2, 3] do  
  puts x  
end
```

```
[1, 2, 3].each do |x|  
  puts x  
end
```

# Syntax

```
x = 10
while x > 0 do
  puts x
  x -= 1
end
```



# Syntax

```
begin
  # do something
  puts 1
rescue => e
  # exception handling
  puts 2
else
  # if there's no exception...
  puts 3
ensure
  # whether or not exception happens, do this...
  puts 4
end
```

# Syntax

```
false, nil
```

```
# every else is true
```

# Syntax

```
# String vs Symbol  
a = "hello"  
b = :hello
```

# 变量

- local variable 局部变量
  - `a = 1`
- class variable 类变量
  - `@@name = "female"`
- instance variable 实例变量
  - `@category = "fruit"`
- global variable 全局变量
  - `$mail_config = {}`
- constant 常量
  - `ApplicationController`

# 括号和分号都可以省略

```
def say_hi_to name  
  puts "Hi, " + name.capitalize  
end  
say_hi_to 'lei'
```

# Class & Inheritance

```
class Bird
  def fly
    puts 'flying'
  end
end
```

```
class Owl < Bird
end
```

```
owl = Owl.new
owl.fly
# => flying
```

# Reopen

```
class String
  def first_4
    self[0..3] + "..."
  end
end

puts "hello".first_4
# => hell...
```

# Module

```
module StringHelpers
  def first_4
    self[0..3] + "..."
  end

  def sanitize
    self.gsub(/[^a-z0-9]+/i, '_')
  end
end

class String
  include StringHelpers
end

puts "hello world".sanitize
# => hello_world
```



# Module

module vs class

# Module

```
module A
  def hello
    puts "hello from A"
  end
end
```

```
class B
  include A
  def hello
    super
    puts "hello from B"
  end
end
```

```
b = B.new
b.hello
# => hello from A
# => hello from B
```

# Module

```
module A
  def hello
    puts "hello from A"
  end
end
```

```
class B
  def hello
    super
    puts "hello from B"
  end
end
```

```
class C < B
  include A
  def hello
    super
    puts "hello from C"
  end
end
```

```
c = C.new
c.hello
#=> A, B, C
```

# Block

```
proc = Proc.new do |name|  
  puts "hello " + name  
end  
  
proc.call('lei')  
# => hello lei
```

# Block

```
proc = Proc.new do |name|  
  puts "hello " + name  
end
```

```
proc.call('lei')  
# => hello lei
```

```
def say_hi_to name, &block  
  puts "hi " + name  
  yield if block_given?  
end
```

```
say_hi_to('lei') do  
  puts "i am 31"  
end  
# => hi lei  
# => i am 31
```

# Block

```
File.open('/tmp/app.log', 'a+') do |f|  
  f.puts "one line"  
end
```

# Reflection

```
"hello world".respond_to? :length  
"hello world".class  
"hello world".methods  
String.included_modules  
String.ancestors  
String.superclass
```

# Meta Programming

What's Meta Programming?



# Meta Programming

```
# Rspec test case sample:  
Site.should_receive('get_original_servers').with("http://  
www.google.com").and_return(original_servers)  
  
response.body.should == original_servers.to_json
```

# Meta Programming

```
# Cucumber test case sample:  
it "signs in successfully" do  
  Capybara.app_host = "http://localhost"  
  visit new_user_session_path  
  page.should have_content('Login')  
  fill_in "user_email", with: user.email  
  fill_in "user_password", with: user.password  
  click_on 'Login'  
  page.should have_content('Settings')  
end
```

# Meta Programming

```
# Puppet configuration sample:  
class helloworld::motd {  
  file {  
    '/etc/motd':  
      owner   => 'root',  
      group   => 'root',  
      mode    => '0644',  
      content => "hello, world!\n",  
    }  
  }  
}
```

# Meta Programming

```
# Homebrew formula for wget:
class Wget < Formula
  homepage "https://www.gnu.org/software/wget/"
  url "https://ftp.gnu.org/gnu/wget/wget-1.15.tar.gz"
  sha256 "52126be8cf1bddd7536886e74c053ad7d0ed2aa89b4b630f76785bac21695fcd"

  def install
    system "./configure", "--prefix=#{prefix}"
    system "make", "install"
  end
end
```

# Tools & Libs

- irb
- yaml

# Q & A

谢谢!