Ruby - Eine Einführung

Matthias Beyer

Furtwangen University

matthias.beyer@hs-furtwangen.de
mail@beyermatthias.de

23. April 2014

Overview

Was, wie, wo, wann, wer?

- Mitte 1990
- Yukihiro Matsumoto
- Mehr-Paradigmen-Sprache
- Scripting Sprache
- Dynamische Typisierung
- GC

Beeinflusst von

- Smalltalk
- Perl
- Python
- Lisp
- CLU
- Eiffel
- Ada
- Dylan
- JS

Wikipedia

Beeinflusst

- Groovy

Wikipedia

Funktional

```
1 def eins
    1
3 end
```

Funktional

```
1 def mul(a, b)
    a * b
3 end
```

Funktional

```
1 def mul a, b
    a * b
3 end
```

OOP

```
1 1.to_s
```

OOP

```
1 class Integer
    def mul(other)
3     self * other
    end
5 end
```

Arrays

```
1 [1, 2, 3]
```

Arrays

```
1 ['Hallowelt', 2, 3]
```

Hashes

```
1 { :a => 'a' }
```

Hashes

```
1 { :a => 'a', 'a' => :a }
```

Iterieren

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

Iterieren

```
[1, 2, 3].each_with_object(2) { |x, o| puts(x * o) }
[1, 2, 3].each_with_index { |x, i| puts(x * i) }
```

Iterieren

```
{ :a => 'a' }.each_key { |k| puts k }
{ :a => 'a' }.each_value { |v| puts v }
{ :a => 'a' }.each_pair { |k, v| puts k, v }
```

Blöcke und Procs

```ruby
[1, 2, 3].each { |x| puts x }

class Array
  def iterate(with)
    self.each do |x|
      self[i] = with.call(x)
  end
end

p = Proc.new do |x|
  puts x
end

[1, 2, 3].iterate(p)
```

Blöcke und Procs

```
[1, 2, 3].iterate(lambda { |n| n**2 })
```

Klassen - Basis

- Class
- Object
- Module
- Proc
- Method
- Exception (und Subklassen)
- Process
- Signal
- Thread
- TrueClass
- FalseClass
- NilClass

Klassen - Basis

- TrueClass
- FalseClass
- NilClass
- Comparable
- Enumerable

Klassen - Rechnen

- Numeric
- Integer
- Fixnum
- Bignum
- Float
- Range
- Rational
- Complex
- String

Klassen - Datenstrukturen

- Struct
- Array
- Hash
- Queue

Klassen - Andere

- File
- Dir
- Time
- Date
- Random
- Regex

Firmen

- Motorola
- Siemens
- Lucent

Apps

- Web
  - Rails
  - Sinatra
- Forschung
  - Simulationen (Motorola)
  - Robotik (Siemens)
  - Wireless Telefonie (Lucent)

The end