Be a language designer

Yukihiro "Matz" Matsumoto
@yukihiro_matz

Athens

April Fool's Day

Euruko

European Ruby Conference

European Ruby Conference

We are here because we're interested in Ruby

Because we love Ruby

Because we're programmers

Well, most of us

As programmers

We program

We program in Ruby

Programming language

We tell computers how to work

In programming languages

Computers don't understand human languages

Just bits

0s and 1s

1001101110001...

We don't understand bits

Do you?

1001101110001...

Let's compromise

We define artificial languages

They try to understand us

Powered by Rabbit 1.0.4

We communicate

using programming languages

We tell computers how to work

Programming languages are medium

Is programming easy?

In elementary level, Yes

```
puts "Hello World"
```

In the real world, No

Why?

Because we face complexity

Real world is very complex

Real world software reflects real world

Business rules

Laws

Traditions

Cultures

Human mind

What is complexity?

Complexity: being complex

Complexity: too many things being involved

Computers don't complain about complexity

Physics don't complain about complexity

"Error: Too many oxygen molecules in this room"

Our bodies don't complain about complexity

60 trillion cells in a human body

Complexity is in the eyes of beholders

Complexity is caused by the limitation of our brains

your ±7 item slots

How do we fight against software complexity?

Framework?

Library?

DSL?

Language?

Language!

Not only because I push Ruby

But because languages matter

Programming languages are medium

They form our mind

Sapir-Whorf hypothesis

"Human thoughts are influenced by (spoken) languages"

Languages influence human thought, more than you think

More so in programming by programming languages

Example: Recursion

BASIC do not allow recursion

How do we battle against complexity using programming languages?

Languages can be weapons

It's a matter of mind

Languages can be weapons against complexity

How do you feel about languages?

Given

Designing a language

Freedom

Freedom gives you power

Freedom gives you understanding

Understanding of your mind

Understanding of human nature

Understanding of programming

I encourage you to design your language

Designing own language?

How many language designers you know?

Powered by Rabbit 1.0.4

me (Ruby)

Guido (Python)

Larry (Perl)

Rasmus (PHP)

?

Join them?

I am serious

I encourage you to design your language

Language geeks

Language geeks: geeks who love programming languages
(often too much)

"language"

more broader sense

Every structure of communication is a language

In that sense, API is a language

Method set is a language

User interface is a language

Mostly everything you design is a language

Languages consists of syntax and vocabularies

Syntax: how you form your communication

Syntax: structure of your language

Vocabularies: elements of your language

Vocabularies: class, methods, functions,...

Syntax and vocabularies

I'd add one more thing

Model

- Syntax

- Vocabularies

- Model

Model: that something formed by the combination of syntax and vocabularies

Model: the designer's intention

Model: the illusion the designer tries to put in your brain

Example

Ruby the object-oriented programming language

Ruby the object-oriented programming language

Object-oriented language

Scripting language

Block building language

Ruby is like LEGO

Ruby provides blocks to combine

Classes and Objects

Blocks

Ruby blocks are fundamentally function objects as arguments

But it only takes one

It was abstraction of loops, originally

```ruby
100.times do
  puts "I love programming"
end
```

It can be used as callbacks

Like Tk event handler

```
Tk.after(100) do
  arc.move 100, 100
end
```

It can be used as scopes

Like RSpec description

```
describe Bowling, "#score" do
  it "returns 0 for all gutter game" do
    bowling = Bowling.new
    20.times { bowling.hit(0) }
    bowling.score.should eq(0)
  end
end
```

It can be used as rules

Like Rakefiles

```
task :default => [:test]
task :test do
  ruby "test/unittest.rb"
end
```

Classes, objects and blocks

You can build your own blocks

Programming is a process of designing your own DSL

-- Dave Thomas

Ruby is a kit to build your DSL

This is a model

This is an illusion

Rails the web domain specific language

Rails = Ruby + Something

Rails syntax = Ruby syntax

Rails vocabularies = Ruby + Web

Rails model = opinionated

MVC the Rails way

ActiveRecord the Rails way

Rails tries to guide you to build Web application

This is a model

This is an illusion

How can we create our own language?

Patterns

Find repeated patterns
in your work

You can find patterns
everywhere in the real world

Your work reflects the real world

You can find patterns
everywhere in your work

Create a block corresponding to the pattern

Try to be concise

Abstraction

Hide detail

Reduce concept

Maximize your ±7 item slots

Opinion

Have strong belief how your work should be

You can learn it from patterns

Your enemy

Perfection

Real world is full of exceptions

One size does not fit all

Don't try to do everything at once

It leads to leaky abstraction

Pareto principle

80:20 rule

Try your best on 80%

Limitation is you friend

Function arguments vs blocks

Multiple inheritance vs Mix-in

Stick with your domain

Then enhance it

Be a designer

Because programming is designing essentially

Designer is a decision maker

Decision is freedom

Freedom is power

Power is motivation

Motivation gives you more power

Be conscious you are a designer

Be a language designer

Language in broad sense

Because language is a good abstraction of design

DSL

Frameworks

Patterns

Be an illusionist

Because you can enhance your ability by (good) illusion

Min-in your intention to your design

Brain wash your users, including yourself

Just like I did

Thank you

FAQ

Ruby2.0?

FAA

Yes

FAQ

Ruby2.1?

FAA

Christmas!

FAQ

mruby?

FAA

Yes!

Any questions?

Thank you