Considering CRYSTAL

@plainprogrammer #OSCON

History & Stats

- 10,000+ commits since September 2012
- First public appearance in 2014
- 11,000+ stars on Github
- 290+ contributors
- **Latest Release:** 0.25.1 (June 2018)

Language Goals

- Similar Syntax to Ruby
- Statically Type-Checked
- Bindable to C
- Compile-time Code Generation (Macros)
- Compile to Efficient Native Code

Examples

```ruby
require “spec”

describe Array do
  describe “#size” do
    it “reports the right number of elements in the array” do
      [1, 2, 3].size.should eq 3
    end
  end
end
```

```
def greet(name, age)
  "I'm #{name}, and I’m #{age} years old.”
end

greet ”OSCON", 20
# I’m OSCON, and I’m 20 years old.


def add(x : Number, y : Number)
  x + y
end

add 1, 2
# 3

add true, false
# Error: no overload matches 'add' with types Bool, Bool
```

```
require “http/server”

server = HTTP::Server.new do |context|
  context.response.content_type = “text/plain”
  context.response.print “Hello world!”
end

address = server.bind_tcp 8080
puts “Listening on http://#{address}”
server.listen
```

```
METHODS = %w(foo bar baz)

{% for name, index in METHODS %}
  def {{name.id}}
    {{index}}
  end
{% end %}


--------------------------------------------------------------


methods.each_with_index do |name, index|
  define_method(name.to_sym) do
    index
  end
end
```

```
channel = Channel(Nil).new

spawn do
  puts “Before send”
  channel.send(nil)
  puts “After send”
end

puts “Before receive”
channel.receive
puts “After receive”

# Before receive
# Before send
# After receive
```

Learn More

https://bit.ly/2NXqEaO (YouTube)

Jon Egeland
”An Introduction to Crystal”

@plainprogrammer #OSCON

Learn More

https://crystal-lang.org/docs/

Crystal's Documentation

Strengths

- Fast
- Friendly
- Testable
- Documented

Weaknesses

- Language Maturity
- Smaller Community
- Libraries
  - Stuff besides REST
  - Integrations

Readiness

USE IT TODAY

- Stuff That Is Easy to Rewrite
- Command-Line Tools
- Some REST Microservices

WAIT AND SEE

- Almost Everything Else
- Monolithic Applications
- Large Codebases

Questions

James Thompson

Principal Software Engineer @ Nav

@plainprogrammer