



Crystal

とみたまさひろ

2015-08-23

Crystal



CRYSTAL

The Programming Language

Language Goals

- ✓ Ruby-inspired syntax.
- ✓ Statically type-checked but without having to specify the type of variables or method arguments.
- ✓ Be able to call C code by writing bindings to it in Crystal.
- ✓ Have compile-time evaluation and generation of code, to avoid boilerplate code.

<http://crystal-lang.org/>



Ruby風の言語

Ruby風？



Crystal	Ruby
静的	動的
型あり	型なし
コンパイル言語	スクリプト言語

Crystal プログラム



ふつーに Ruby

```
class Hoge
  def foo
    p "abc"
  end
end
Hoge.new.foo #=> "abc"
```

実行



```
% ruby hoge.rb  
"abc"  
% crystal hoge.rb  
"abc"
```

ベンチマーク



```
# フィボナッチ数列のN番目の値
def fib(n)
  if n < 2
    n
  else
    fib(n-1) + fib(n-2)
  end
end
p fib(35)
```

速い！

```
% time ruby fib.rb
9227465
real    0m2.275s
user    0m2.268s
sys     0m0.004s
```

```
% time crystal fib.rb
9227465
real    0m0.299s
user    0m0.236s
sys     0m0.056s
```


コンパイルして実行ファイルを生成



```
% crystal build fib.rb --release  
% ./fib  
9227465
```

さらに速い！

```
% time crystal fib.rb
9227465
real    0m0.299s
user    0m0.236s
sys     0m0.056s

% time ./fib
9227465
real    0m0.090s
user    0m0.088s
sys     0m0.000s
```



RubyスクリプトをCrystalで実行 するだけで速くなる!?



そんなうまい話はない



Crystal \neq Ruby

文字と文字列



- "A" - 文字列
- 'A' - 文字

"A"+"B" #=> "AB"

'A'+ 'B' #=> *undefined method '+' for Char*

多倍長整数がない



整数演算の謎に見える挙動

```
0x7FFFFFFF.class #=> Int32
0x80000000.class #=> Int64
0x7FFFFFFF+1      #=> -2147483648
0x7FFFFFFFi64+1  #=> 2147483648
256*256*256      #=> 16777216
256**3           #=> 1.67772e+07
256*256*256*256 #=> 0
256**4           #=> 4.29497e+09
```



Crystal は Crystal として使おう



拡張子は .rb じゃなくて .cr



Crystalの特徴(Rubyと比較して)

型



```
a = [1, 2, 3]  #=> Array(Int32)
a.push 4      #=> OK
a.push "a"
  #=> no overload matches 'Array(Int32)#push'
  #   with types String (コンパイル時エラー)

a = []        #=> Syntax error
a = [] of Int32 #=> OK
```

メソッド



```
def hoge(a, b)
  a + b
end
```

```
hoge("abc", "xyz") #=> "abcxyz"
```

```
hoge(1, 2) #=> 3
```

```
hoge("abc", 2) #=> コンパイル時エラー
```

コンパイル時エラー



```
% crystal build hoge.cr  
Error in ./hoge.cr:7: instantiating 'hoge(String, Int32)'
```

```
hoge("abc", 2)  
^~~~~
```

```
in ./hoge.cr:2: no overload matches 'String#+' with types Int32  
Overloads are:  
- String#+(other : self)  
- String#+(char : Char)
```

```
a + b  
  ^
```

オーバーロード



```
def hoge(a, b)
  a + b
end
```

```
def hoge(a : String, b : Int)
  a * b
end
```

```
hoge("abc", "xyz")  #=> "abcxyz"
```

```
hoge(1, 2)          #=> 3
```

```
hoge("abc", 2)     #=> "abcabc"
```

変数に型はない



```
var = 123  
var = 'a'
```

複数の型の可能性



```
var = rand < 0.5 ? 123 : "abc"  
var.size # Int がないのでエラー  
var + 1 # String がないのでエラー
```


ちゃんと型チェックすればエラーにならない



```
var = rand < 0.5 ? 123 : "abc"  
if var.is_a? Int  
  var + 1  
else  
  var.size  
end
```



静的

静的



- eval がない
- クラス定義/メソッド定義は後勝ち

メソッド定義は後勝ち



```
if rand < 0.5
  def hoge      # ← rand の値に関係なく無視される
    123         #
  end          #
else
  def hoge      # ← 常にこっちが有効
    "abc"       #
  end          #
end
```

直感に反したり



```
def hoge
  123
end
if false
  def hoge
    "abc"
  end
end
hoge #=> "abc"
```



その他

名前付き引数



Rubyより自然かも

```
def hoge(n=123, s="abc")
  p [n, s]
end
hoge                #=> [123, "abc"]
hoge(789)           #=> [789, "abc"]
hoge(s: "xyz")     #=> [123, "xyz"]
```

マクロ



```
macro define_method(name, content)
  def {{name}}
    {{content}}
  end
end

define_method foo, 123
foo #=> 123
```


プロジェクト雛形作成(Rails風?)



```
% crystal init app my_project
  create  my_project/.gitignore
  create  my_project/LICENSE
  create  my_project/README.md
  create  my_project/.travis.yml
  create  my_project/Projectfile
  create  my_project/src/my_project.cr
  create  my_project/src/my_project/version.cr
  create  my_project/spec/spec_helper.cr
  create  my_project/spec/my_project_spec.cr
Initialized empty Git repository in /tmp/my_project/.git/
```

ライブラリ依存関係解決(Bundler風)



```
my_project% cat Projectfile
deps do
  github "manastech/crystal-mysql"
end

my_project% crystal deps install
Cloning into '.deps/manastech-crystal-mysql'...
remote: Counting objects: 55, done.
remote: Total 55 (delta 0), reused 0 (delta 0), pack-reused 55
Receiving objects: 100% (55/55), 7.03 KiB | 0 bytes/s, done.
Resolving deltas: 100% (19/19), done.
Checking connectivity... done.
```

ドキュメント作成(YARD風)



```
my_project% crystal doc  
my_project% ls doc  
Myproject.html  css  index.html  js  list.html  main.html
```

テスト(RSpec風)



```
my_project% crystal spec
.  
  
Finished in 0.38 milliseconds  
1 examples, 0 failures, 0 errors, 0 pending
```