

coins LT #10

ルアカッ!〜ルアカッドウ!〜

びしょ〜じょ

## 自己紹介

- こんにちは、びしょ～じょです。

♡ 百合とかアイカツとか Arch Linux とか Lua/MoonScript とかが趣味

## Luakatsu

Lua とは

LuaRocks とは

Aikatsu!

Luakatsu!

how to use

実装



## 付録

速度比較

出典など

## Luakatsu

Lua とは

LuaRocks とは

Aikatsu!

Luakatsu!

how to use

実装



## 付録

速度比較

出典など

## ● Lua とは

C/C++への組み込み言語として開発された、マルチパラダイムな動的型付け言語

### ◇ 小さい実装

- ♠ Lua5.2.4 のソースは全部で 2 万行程度 (Ruby 2.2.1 のソースは 20 万行くらい)
- ♣ もちろんこのスライドも  $\text{\LaTeX}$  に Lua が埋め込まれている  $\text{\LaTeX}$

### ♡ 柔軟な記述

- ◇ 手続き型で馴染みやすい
- ♠ 関数がファーストクラス
- ♣ 強力で唯一なデータ構造である **table** によりオブジェクト指向プログラミングもできる。

### ♡ 高速に動作

多くの他のスクリプト言語よりも高速に動作し、JIT コンパイラの *LuaJIT* を用いれば **Java より若干遅い程度の速度**が出せる。

- Lua とは

⇒ *Wikipedia* に詳細な記述がある

人人人人人人人  
> まぁ見てくれ <

### Code 1: example.lua

```
1 (function() io.write(string.format("Hello, %s!\n", "Lua")) end)()
   -- anonymous function
2 t = {"This", "is", "table"} -- global variable
3 for i = 1, #t do           -- 1 origin
4     print(t[i])
5 end
```

```
> lua example.lua
Hello, Lua!
This
is
table
```

## Luakatsu

Lua とは

LuaRocks とは

Aikatsu!

Luakatsu!

how to use

実装



## 付録

速度比較

出典など



## ● LuaRocks とは

Lua のパッケージ管理システム

Ruby でいうところの *RubyGems*、Python でいうところの *Pip*.....という認識。  
パッケージの登録や、他のユーザが登録したパッケージを簡単にインストールして使用することができる。

## Luakatsu

Lua とは

LuaRocks とは

Aikatsu!

Luakatsu!

how to use

実装



## 付録

速度比較

出典など

# ●アイカツ! ~アイドルカツドウ!~

奇跡の賜物

もはや説明は不要

先々週からはアニメの放送時間が 18:30 にずれたので注意

ゆうがた6:30!

Lua + Aikatsu!

Lua + アイカツ! =

Lua + アイカツ! =

型エラー  
Luakatsu

## Luakatsu

Lua とは

LuaRocks とは

Aikatsu!

Luakatsu!

how to use

実装



## 付録

速度比較

出典など

Luakatsu!

- Luakatsu

- ◇ sue445 さんの *Rubicure* に影響されて制作

## ● Luakatsu

◇ sue445 さんの *Rubicure* に影響されて制作

♠ 文字通り Ruby + プリキュア

```
> ruby -rubicure -e 'Cure.marine'
{:human_name=>"来海えりか", :precure_name=>"キュアマリン", :
  cast_name=>"水沢史絵", :created_date=>Sun, 21 Feb 2010, :
  color=>"blue", :transform_message=>"(プリキュアの種、いくで
すう！)\nプリキュア！オープンマイハート！\n海風に揺れる一輪の花！
キュアマリン！\nハートキャッチ、プリキュア！", :extra_names=>["
スーパーキュアマリン"], :attack_messages=>["花よきらめけ！\nプ
リキュア！ブルーフォルテウェイブ！！", "花よ、咲き誇れ！\nプリキュ
ア・ハートキャッチ・オーケストラ！！"], :transform_calls=>["
open_my_heart"]}
```



## ● Luakatsu

◇ sue445 さんの *Rubicure* に影響されて制作

♠ 文字通り Ruby + プリキュア

♣ ではアイカツでこんな感じのやつは? ⇒ ない

♡ *じゃあ作るしか無いよなあ?*

◇ 現在 v2.0 が最新 (黒沢凜君、天羽まどか君に対応)

♠ メジャーバージョンが2のくせにまだまだ機能が少ないのが特徴。

♣ Lua なので、LuaTeX からもちっと *いそいそ* すれば簡単に使える。

★ LuaTeX に入っている Lua の処理系は `~/luarocks` 以下のモジュールを読み込んでくれないのでここで一工夫必要。

参考: <http://bit.ly/1CBXGjy>

♡ Lua に倣い MIT ライセンス

## Luakatsu

Lua とは

LuaRocks とは

Aikatsu!

Luakatsu!

how to use

実装



## 付録

速度比較

出典など

## 簡単インストール

```
> luarocks --local install luakatsu
```

## ワンライナー

```
> lua -luakatsu -e "Aikatsu()"
私のアツいアイドル活動、アイカツ！ 始めます！ フゥッ
> date -s "04/01"
> lua -luakatsu -e "Aikatsu()"
アメージング！ ショータイム！ 俺のホットなティーチャー活動、アイカツ！ スタート
だぜ、イエア！
```

## Luakatsu!

how to use

### Code 2: luakatsu\_example.lua

```
1  require "luakatsu"
2
3  print(Aikatsu.Kii.name)
4  print(Aikatsu.groups.Powapuri.name)
5
6  for _, v in pairs(Aikatsu.Yurika.favorite_foods) do -- iterator
7      print(v)
8  end
9
10 print("Version: ", Aikatsu.version)
11 Aikatsu.Ichigo() -- not needed `print` function
```

## Luakatsu!

### how to use

```
> lua luakatsu_example.lua
Version: v2.0-0
冴草 きい
Powa2 × PuRiRiN
牛丼
ラーメン
name 星宮いちご
actor 諸星 すみれ
birthday 03/15
blood_type 0
favorite_foods のり弁
special_ability ご飯をきっちりよそう
favorite_brand Angely Sugar
type Cute
signature_songs 輝きのエチュード, Growing for a dream, Dance in the rain
sing 霧島 若歌
affiliation Soleil, STAR ANIS, 2wingS
school スターライト学園
```

⇒ 幼女先輩各位にも簡単に触ってもらえるようなインターフェイス

## Luakatsu

Lua とは

LuaRocks とは

Aikatsu!

Luakatsu!

how to use

**実装**



## 付録

速度比較

出典など

以下が未実装

- ◇ 各話タイトル
- ♠ ステージ
- ♣ ドレス (カード)
- ♡ アイカツ格言
- ◇ .....などなど



以下が未実装

♠ 各話タイトル

⇐

♣ ステージ

⇐ まあいいかな

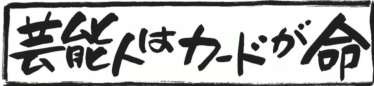
♡ ドレス (カード)

◇ アイカツ格言

♠ .....などなど

以下が未実装

- ♣ 各話タイトル
- ♡ ステージ
- ◇ ドレス (カード)
- ♠ アイカツ格言
- ♣ .....などなど

⇐ 

⇐ 肝心

Luakatsu!

実装

以下が未実装

♡ 各話タイトル

◇ ステージ

♠ ドレス (カード)

♣ アイカツ格言

♡ .....などなど

(だめじゃん)

\ ドッ /

\ ワハハハハハ /

⇐ 

⇐ 肝心

実態はAikastuというtableをグローバル変数として定義し、そこにいろいろ突っ込んでいる。アイドルの情報も全部table。

### 星宮いちご (Aikatsu.Ichigo) —その実態—

```
{{"name", "星宮 いちご"}, {"actor", "諸星 すみれ"}, {"birthday", "03/15"}, {"zodiac_sign", "Pices"}, {"blood_type", "O"}, {"favorite_foods", {"のり弁"}}, {"special_ablity", {"ご飯をきっちりよそう"}}, {"favorite_brand", "Angely Sugar"}, {"type", "Cute"}, {"signature_songs", {"輝きのエチュード", "Growing for a dream", "Dance in the rain"}}, {"sing", "霧島 若歌"}, {"affilication", {"Soleil", "STAR ANIS", "2wingS"}}, {"school", "スターライト学園"}}
```

しかし上記のように`{{str, str}, {str, str} ...}`みたいに定義しても当然`t.k`のように参照はできない

Q. そもそもなんで上記のような定義なのか

A. Lua の連想配列部分 (`{k1 = v1, k2 = v2, ...}`) は順序情報を保存しないので、小技を用いてチョチョイと.....。

Q. 別によくない?

A. でもプロフィールを見るときは最初に名前が出ると嬉しいやんけ

⇒ **metatable** という機能を使う

*metatable* .....**table**に付与する特別な**table**。演算子オーバーロードとかもできる。

```
setmetatable(t, {__index = ..., __call = ...})
```

**setmetatable**関数で**metatable**をセット

◇ \_\_index

- ♠ **t.k**が**未定義の時**、**t**に設定されている**\_\_index table**を**参照する**、という機能
- ♣ 連想配列部分にも **value** を突っ込んでいくと**table**がでかくなってしまう。
- ♡ {"k1", "v1"}のような**table**を**k1 = v1**のように変換した**table**を**\_\_index**としてもとの**table**に設定する。

◇ \_\_call

- ♠ **table**を**function**のように扱った時、**\_\_call**で定義した**function**が実行される、という機能
- ♣ **t.\_\_call()**とかしなくて良いので、インターフェイスがすっきりする。
- ♡ **table**の配列部分は順序が保持されるので、{"key", "value"}を順番に出力

## metatableその他一部

### ◇ 四則演算

`__add` (+)、`__sub` (-)、`__mul` (×)、`__div` (÷)

### ♠ 比較

`__eq` (==)、`__lt` (<)、`__le` (<=)

### ♣ シフト (Lua 5.3 から)

`__shl` (<<)、`__shr` (>>)

### ♡ 特殊

`__newindex` ..... `table`への(配列部分、連想配列部分問わず)値の追加時に発火する関数

## Luakatsu

Lua とは

LuaRocks とは

Aikatsu!

Luakatsu!

how to use

実装



## 付録

速度比較

出典など



- ◇ Luakatsu は アイカツ!の Lua 実装
- ♠ Lua の連想配列部分の順序を保持しないところを *metatable* でカバー
- ♣ この辺の **table** を生成する関数くらいしか頑張ったところがない(おわり)。
- ♡ あとはアイドル情報を魂を込めて一人ひとり手打ちしたくらい(公式などからうまくスクレイピングなどで引っ張れる形の情報がなくてつらい)。
- ◇ Lua $\text{\LaTeX}$  めっちゃしんどかった。

# みんなでアイカツ!



あかりのスターライトスクールドレス!

ピンクパレードコーデ



ご清聴ありがとうございました。

## Luakatsu

Lua とは

LuaRocks とは

Aikatsu!

Luakatsu!

how to use

実装



## 付録

速度比較

出典など

## Luakatsu

Lua とは

LuaRocks とは

Aikatsu!

Luakatsu!

how to use

実装



## 付録

速度比較

出典など

## ● 速度比較

38 番目のフィボナッチ数を求めるまでの速度を計測

実行環境は ArchLinux(Linux Kernel 4.0.0)、Xeon E3-1270

pseudocode

```
def fib(n)
  if n < 2 then
    return n
  end

  return fib(n - 1) + fib(n - 2)
end

fib(38)
```

Code 3: fibs/fib.lua

```
1 local function fib(n)
2   if n < 2 then return n end
3   return fib(n - 1) + fib(n - 2)
4 end
5
6 fib(38)
```

Code 4: fibs/fib.rb

```
1 def fib n
2   return n if n < 2
3
4   return fib(n - 1) + fib(n - 2)
5 end
6
7 fib 38
```

Code 5: fibs/fib.py

```
1 def fib(n):
2     if n < 2: return n
3     return fib(n - 1) + fib(n - 2)
4
5 fib(38)
```

## Code 6: fibs/fib.java

```
1 class fib{  
2     static int fib(int n){  
3         if(n < 2) return n;  
4         return fib(n - 1) + fib(n - 2);  
5     }  
6  
7     public static void main(String[] arg){  
8         fib(38);  
9     }  
10 }
```



Code 7: fibs/fib.c

```
1 int fib(int n){  
2     if(n < 2) return n;  
3     return fib(n - 1) + fib(n - 2);  
4 }  
5  
6 int main(){  
7     fib(38);  
8 }
```

## 速度比較 I

timeコマンドにて計測。

```
for i in `seq 1 10`; time COMPILER file
```

Compiler/ Interpreter	Version	time
Lua	5.2.4	5.6319
Ruby	2.2.1p85	5.5308
Python	3.4.3	15.6391
LuaJIT	2.0.3	0.347
Java	1.8.0_40	0.2204
GCC	4.9.2	0.3356

Java、お前速いんだな。

## Luakatsu

Lua とは

LuaRocks とは

Aikatsu!

Luakatsu!

how to use

実装



## 付録

速度比較

出典など

## ● 出典など

♠ さなフォン飾 (タイトル、その他)  
[http://sana.s12.xrea.com/2\\_sanafonkazari.html](http://sana.s12.xrea.com/2_sanafonkazari.html)

♣ Rubicure  
<https://github.com/sue445/rubicure>

♡ アイカツ格言

◇ ゆうがた 6:30!

アイカツ! 第 126 話

♠ 芸能人はカードが命

アイカツ! 第 1 話

♣ みんなでアイカツ!

アイカツ! 第 102 話