

# WORD

From College of Information Science



生命、宇宙、そして万物についての  
究極の疑問の答え

42



香港からの刺客！ 鴛鴦茶即堕ち  
Brainf\*ck compiler for Lua VM  
Goで美少女ゲームをスクレイピング  
段ボール活用事例報告会  
池袋アンダーグラウンド

# 目次

香港からの刺客！ 鴛鴦茶即落ち .....	ひだるま	003
Brainf*ck compiler for Lua VM .....	びしょ～じょ	007
Go で美少女ゲームをスクレイピング .....	ひだるま	021
段ボール活用事例報告会 .....	びしょ～じょ	029
池袋アンダーグラウンド .....	みみずのひもの	036
WORD 編集部への誘い .....	鳥	048
編集後記 .....		049

# 香港からの刺客！ えんおう 鴛鴦茶即墮ち

文 編集部 ひだるま

## 1 Overture

ここは WORD 編集部。秋も深まりまくり寒さが刺さるそんな時期。珈琲を吸っていた私に、扉をピッピッと開けて入ってきた\*1 linerlock がやおらこんなことを言い出した。

「エンオウチャ飲みたいくないですか？」

……はて。このタイミングで「チャ」というのだから「茶」のことなのだろう。では「エンオウ」とは何なのか。現代っ子の私は訊き返す前に Wikipedia 先生に尋ねる。果たして答えが返ってきた。香港では一般的な飲み物で、言ってみれば珈琲紅茶というもののだそう。彼に改めて訊けば、読んでいた小説\*2に登場していた飲み物が飲んでみたくなったとのこと。普段親しんでいる物二つだが、組み合わせるという発想は無かった。とはいえ、美味しい物を組み合わせると美味しい保証はどこにも無い。勿論私は二つ返事で了承し、切らしていた茶葉と必要だという練乳を後日揃えることにしてその場は解散した。

そして iias つくばで「映画魔法つかいプリキュア！ \*3」を鑑賞し、「ポップン Q\*4」予告篇に泣いたりした帰りに KALDI にて茶葉を買い、スーパーで（加糖）練乳\*5を購入した。最近自転車漕ぐのが妙に疲れる。

## 2 作る

さて、鴛鴦茶を淹れるには幾つかバリエーションがある。今回選択したのは液体混合法、ぶっちゃければ珈琲と紅茶を各々淹れてから混ぜる方法である。コーヒーマーカーが折角あるのだから使わない手は無い\*6

\*1 2016 年 11 月現在編集部の扉は開錠の際に五月蠅い。

\*2 「珈琲店タレーランの事件簿 5 この鴛鴦茶がおいしくなりますように」

\*3 もうすぐ中学二年生になる朝日奈みらいは、春休みのある夜、空から謎の物体が近くの公園に落ちていくところを発見します。「もしかして魔法つかいかも！」と期待に胸をふくらませるみらいは、翌朝、ワクワクしながら、幼い頃からずっとかわいがっているクマのぬいぐるみ「モフルン」と一緒に公園に向かいます。

するとそこで目にしたのは…ほうきにまたがって宙に浮く女の子…！ リコと名乗るその女の子は、なんと魔法つかい！！

みらいは、魔法つかいに興味津々でリコを質問攻めに。リコは“ある物”を探しにみらいの住む世界にやってきたらしいのです。さらに、よく見ると二人は同じペンダントをつけていて…。

そんな中、闇の魔法つかい・ドクロクシーの仲間・パッティが二人の前に現れます！「リンクルストーン・エメラルド」を渡せとリコに迫りますが、どうやらリコも、「リンクルストーン・エメラルド」を探しているらしくて…。

パッティの闇の魔法で生まれた怪物ヨクパールに追い詰められたその時…

みらいとリコ、そしてモフルンが手をつなぎ、魔法の言葉『キュアップ・ラパパ！』ととなえると、ペンダントが光輝きます。

そして、伝説の魔法つかい「プリキュア」が誕生するのです…！！

\*4 中学3年生の春、悩みを抱えた少女たちが、奇跡の出会いを体験する。住んでいる地域も、家庭環境も違う5人。

出会うはずのなかった彼女たちが、出会い、そしてぶつかりあう。

それは、それまで目をそらしていた自分の心と向かい合うこと。

\*5 加糖練乳を用いた場合、正しくは鴛鴦走（えんおうそう）というバリエーションになる。

し、二段法<sup>\*7</sup>ではコーヒーマーカーの洗浄が面倒である。原料混合法<sup>\*8</sup>では上手く珈琲の味が出るのか疑わしい。そして後からはっきり認識することになったのだが、何よりこの方法の最大の利点は容易に混合の比率を変えられることである。

## 2.1 用意

今回は以下の環境で試した。

- 紅茶（アールグレイ）定番。KALDI にてセールを行っていた。
- 珈琲（ウガンダ ブルーナイル）コーヒー部員にコスパ豆として人気。酸味、苦味など平均的な感じ<sup>\*9</sup>。
- （加糖）練乳。牛の顔のパッケージでお馴染のアレ。

## 2.2 淹れる

紅茶は茶葉を「ふんわりお茶パック M」で包んでティーポットに放置する簡易的な淹れ方。



珈琲は豆をハンドミルで挽き、コーヒーマーカーに投入。水を入れてスイッチを入れ待つだけだが、高温でしか抽出できないのが難点である。



---

<sup>\*6</sup> 編集部内団体コーヒー部でネルを使わなくなって久しい。楽な手段に人は流されていくのだ。

<sup>\*7</sup> 淹れた紅茶で珈琲を淹れる方法。

<sup>\*8</sup> 珈琲の粉と紅茶の茶葉を同じ容器に淹れて抽出する。

<sup>\*9</sup> 言うまでもないが主観的な評価である。

## 2.3 混ぜる

珈琲と紅茶を器\*<sup>10</sup>に入れ、かき混ぜる。



## 3 飲む

先ずは練乳を混ぜずにそのまま飲んでみる。ふむ、珈琲の落ち着いた薫りと紅茶のスツとした薫りが一体となり、何となくよろしくない。で、では味はどうか。珈琲の苦味、珈琲の酸味、紅茶の酸味……競合してでやがる\*<sup>11</sup>。後味も苦味が妙に残る。

さてさて雲行きが怪しくなってきた。ここに練乳を投入することは果たして正しいのか。しかし未だ「鴛鴦茶を飲む」という実績は解除されていない。意を決してチューブを絞る。ドバドバと流れる練乳。そして……

「珈琲飲みませんか」

「じゃあ鴛鴦茶にしましょう」

数時間後、そこにはドハマリする編集部員の姿がそこにあった。

練乳を投入すると、喧嘩をしていた珈琲と紅茶が和解し、美しい調和が誕生した。珈琲の苦味が心を落ち着かせる。そして練乳の甘味はしつこくなることなく紅茶に流され、スッキリとした後味に。顧客が本当に欲しかったものがそこにあった。

## 4 考察

何故我々が内心侮っていた鴛鴦茶を愛飲するようになってしまったのか。この件に関して、linerlock の言がある。「甘いと美味しい」と。成程一理ある。しかし他にも理由があるのではないか。

工程を思い出してみると。コーヒーマーカー任せで抽出を行ったりお茶パックに入れた茶葉がふやけていくのをただ眺めていたり、それぞれを淹れる過程はかなり単調である。そこに個人の技量は悲しいかな関

---

\*<sup>10</sup> 映っているのは綾地寧々ちゃん。可愛い。

\*<sup>11</sup> 後でマグダレナ スプレモで同様に試したら混ぜる意味合いは薄いもののそこそこ美味しかったので、相性というものがあるのだろう。

係が無い。そこで、珈琲と紅茶と練乳の比率を「自分で」調整できる所に妙があるとは考えられないだろうか。自らの介在する事象というのは往々にして楽しいものなのだから。

また、比率が変わると味が変わるのは自明だろう。淹れる都度違う味というのは、半自動で飲める飲料に慣れた編集部員達にとって変化という楽しみを与えたのではないだろうか。

## 5 おわりに

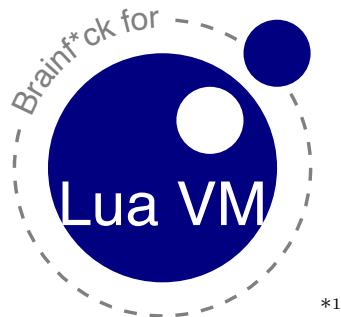
本記事では紅茶の種類がアールグレイ 1 種であったが、こちらを変更することで更なる変化があるのかも知れない。また、プーアル茶を混ぜるというものもあるそうで、まだまだ楽しみは尽きない。

真実はどうあれ、今や日に 1 本練乳のチューブが消費されかねない程、鴛鴦茶は一部部員の中で親しまれている現状をお伝えした。興味を持った諸氏も試しては如何だろうか。



# Brainf\*ck compiler for Lua VM

文 編集部 びしょ〜じょ

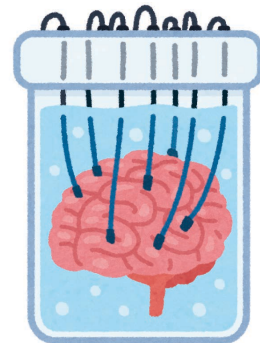


## 1 はじめに

こんにちは、びしょ〜じょです。最近何かしましたか？ 最近ボクは何もしてません。おわり。

## 2 Brainf\*ck?

“ブレインフ コンフラ ク”<sup>\*2</sup> と読む。難解プログラミング言語として極めて有名であり、1つのポインタ、2つのバイトストリーム、8つの命令からなる。



### 2.1 >

ポインタをインクリメントする。C言語における `ptr++` のイメージ。

### 2.2 <

ポインタをデクリメントする。C言語における `ptr--` のイメージ。

### 2.3 +

ポインタの値をインクリメントする。C言語における `(*ptr)++` のイメージ。

<sup>\*1</sup> <https://www.lua.org/images/> から PostScript ファイルが提供されている。

<sup>\*2</sup> いわゆる“猥言”なので伏せ字にするのが一般的。本誌も大学の発行人物なので伏せる。伏せ字の下にある文字は察して欲しい。

## 2.4 -

ポインタの値をデクリメントする。C 言語における `(*ptr)--;` のイメージ。

## 2.5 .

ポインタの値を出力する。C 言語における `putchar(*ptr);` のイメージ。

## 2.6 ,

入力から 1 バイト読み込み、ポインタの指す先に代入する。C 言語における `*ptr = getchar();` のイメージ。

## 2.7 [

ポインタの値が 0 のとき、対応する `]` にジャンプする。C 言語における `while(*ptr){` のイメージ。

## 2.8 ]

ポインタの値が 0 でないとき、対応する `[` にジャンプする。C 言語における (上記の `while(*ptr){` に対応する) `}` のイメージ。

なるほど。原始的な操作しかないが、これでチューリング完全になる。チューリング完全というのは、親や友達に聞いてみよう。

## Hello, World!

とりあえず Hello, World! くらい表示させたいのが人間の性。

Code 1 Hello, World! written in Brainf\*ck

```
1  ++++++[>++++++>++++++>++++<<<-]>.>+ .+++++ .+++> .-----.  
2  <+++++ .----- .+++ .----- .----- .>+.
```

1 行目が “Hello, ” で、2 行目が “world!” である。とりあえずドットに注目すればなんとなく分かるか…… 分かるか？



### 3 Lua VM



今回使用するのはコレ! Lua Virtual Machine の上で動かしたい。  
 Lua<sup>\*3</sup> について今回は知らなくていいですが、Lua VM については知ってもらいたい。

Lua VM とは、レジスターベースの VM である。Lua インタープリターの正体であり、VM 用の実行形式を受け取りその命令を逐次実行していく。とりあえずレジスターベースということ、レジスターが 256 個ということが伝われば OK です。今回は Lua 5.3 の VM を使います。  
 VM というのは、親や友達に聞いてみよう。

ここで簡単のため新たに記法を 4 つ追加する。

#### R(X)

レジスター  $X$  の値を指す。

#### Kst(X)

Constant list の  $X + 1$  番目の値を指す。

#### RK(X)

$X$  が負の値の場合、Constant list の  $-X$  番目の値を指す。正の数の場合、レジスターの  $X + 1$  番目の値を指す。

#### Upv(X)

$X + 1$  番目の upvalue を指す。

謎の用語が増えましたね。

<sup>\*3</sup> <https://lua.org> つい最近 HTTPS に対応した。めでたい。非常に軽量な実装でありながら、OOP や FP ができるマルチパラダイムなプログラミング言語。いやあ、もっと流行ってほしいんですけど……。

### 3.1 用語

#### Constant list

Lua VM は即値を扱えないので、Constant list という定数のリストを持っている。いくつかの命令がこのリストにアクセスし、定数を持ってくる。

#### Upvalue list

closure の環境を指し、インデックスでアクセスする。`_ENV`がグローバルな値を持つてくれる<sup>\*4</sup> ため、0 番目はだいたい `_ENV`となる。

### 3.2 VM Instructions

Lua 5.3 では VM の命令が 47 個ある<sup>\*5</sup> が、47 個も使わないので必要なものだけを説明していく。もちろんパイナリで表現されてるわけだがよく分からくなりそうなので VM のインストラクションを書いていきます。`opcode operand.....`という形式で紹介する。

#### LOADK A B

`R(A)`に `Kst(B)`を代入する。

#### MOVE A B

`R(A)`に `R(B)`を代入する。`number`型、`string`型、`nil`は値渡しで他はすべて参照渡しとなる。実はオペランドは 3 つなのだが、3 つめは 0 で固定。何かの名残りだろうか。

#### ADD A B C

`R(A)`に `R(B) + R(C)`を代入する。本当は演算子オーバーロードとかできるんですが今回は演算子の引数が数値に限るので省略。

#### SUB A B C

`R(A)`に `R(B) - R(C)`を代入する。演算子の本当の働きについては同上。

#### EQ A B C

`R(B)`と `R(C)`が同値かどうかを調べる。`A == 0 XOR R(B) == R(C)`のときに直後の命令を無視して次の命令から実行を続ける。`+や-`と同じく、演算子の本当の働きについては省略。

---

<sup>\*4</sup> Lua 5.1 までは `_G` という table が持っていたが、Lua 5.2 からは `_G` を内包する `_ENV` が爆誕し、色々テクニカルなことができるようになってしまった。

<sup>\*5</sup> <https://www.lua.org/source/5.3/lopcodes.h.html#OpCode>

**JMP A B**

現在のプログラムカウンターから **B** の相対番地分ジャンプする。つまり、**現在のプログラムカウンター + B + 1** 番目の命令から実行を続ける。**A** は今回使わないので省略 <sup>\*6</sup>。

**CALL A B C**

関数 **R(A)** に **R(A + 1)**, **R(A + 2)**, ..., **R(A + B - 2)** を引数として渡し、その戻り値を **R(A)**, **R(A + 1)**, ..., **R(A + C - 2)** に代入していく。つまり、**R(A)**, **R(A + 1)**, ..., **R(A + C - 2) = R(A)(R(A + 1), R(A + 2), ..., R(A + B - 2))** という感じ。Lua は多値といって関数が複数の値を返すことができるので、複数のレジスターで戻り値を受け取っているわけですね。

**B==1** のとき、関数の引数をとらない。**B==0** のとき、現在使用しているレジスターのトップまでをすべて引数として取る。

**C==1** のとき、関数の戻り値をレジスターに保存しない。**C==0** のとき、関数の戻り値をすべてレジスターに保存する。

**RETURN A B**

関数の呼び出し元に **R(A)**, **R(A + 1)**, ..., **R(A + B - 2)** を返す。

**B==1** のとき、戻り値を返さない。**B==0** のとき、関数の戻り値をすべて返す。

**GETTABUP A B C**

**R(A)** に **Upv(B)[RK(C)]** を代入する。

**GETTABLE A B C**

**R(A)** に **R(B)[RX(C)]** を代入する。

ハア〜〜疲れたな、わかりますよボクも疲れました。

## 4 Brainf\*ck と Lua VM の命令の対応

Lua VM にポインターという概念はない。では、レジスターのインデックス **X** をポインターとして、レジスターの中身 **R(X)** をその値ということでここは一つ手を打とう。注目しているポインター **X** を **looking\_ptr** とでも名付けておく。

また、**.や,** は標準モジュールにある関数で実現する他なく、いちいちロードしてくるのも面倒臭いので後述する 4 つの関数を置いておくスペースを確保する。関数呼び出しにもいくつかスペースを取るため、それ

---

<sup>\*6</sup> 私自身よくわかってないので教えてください。ドキュメントは ANSI C のソースコードです。

用のスペースも取っておく。とりあえず、0～3 番までのレジスターにその関数を突っ込んでおき、253～255 番のレジスターを関数呼び出し用にとっておこう。とするとポインタの指せる範囲は 4～252 番の間のレジスターとなる。

Code 2 part of the implementation

```
ptr_boundary = {start: 4, end: 252}
looking_ptr = ptr_boundary.start
call_ptr = ptr_boundary.end + 1
```

スタックマシンとかのほうが絶対に相性いいよな～そうなんですよレジスターマシンと Brainf\*ck は相性が悪いと思っていた。

## prerequisite

簡単のため、以下のような関数を用意する。

```
function index_of_value(val)
  return Constant listのvalが入ってるインデックス
end
```

これは実装では使わないので軽く考えていただきたく。

それと、ポインタの動く範囲のレジスターの値を 0 で初期化しておきたい。

```
LOADK 4 index_of_value(0)
LOADK 5 index_of_value(0)
.....
LOADK 251 index_of_value(0)
LOADK 252 index_of_value(0)
```

### 4.1 .と,

まず Brainf\*ck の .と, を考えたい。前者はとりあえず `io.write(string.byte(looking_ptr))`、とすると後者は `looking_ptr = string.char(io.read(1))` だろう。つまり必要な関数は

- `io.read`
- `io.write`
- `string.char`

- `string.byte`

の4つとなる。

上から順に0〜3番レジスターに突っ込んでいこう。

```
GETTABUP 0 0 index_of_value("io")
GETTABUP 2 0 index_of_value("string")
GETTABLE 1 0 index_of_value("write")
GETTABLE 0 0 index_of_value("read")
GETTABLE 3 2 index_of_value("byte")
GETTABLE 2 2 index_of_value("char")
```

できるだけ命令数を少なくしたのでちょっと不思議な順番になった。Luaだとこんなふうにかけるか。

```
local read = io
local char = string
local write = read.write
read = read.read
local byte = char.byte
char = char.char
```

## 4.2 >

申し訳ないのですがこれはVMの命令に落ちない。コンパイル時に `looking_ptr` をインクリメントするだけである。

## 4.3 <

無念。同上。`looking_ptr` をデクリメントしていく。

## 4.4 +

これはイケる。

```
ADD looking_ptr looking_ptr index_of_value(1)
```

## 4.5 -

OKだ。

```
SUB looking_ptr looking_ptr index_of_value(1)
```

#### 4.6 [

ポインタの値が 0 のとき、とすると EQ 命令で分岐するとよさそうだ。対応する]に当たるまでは何処にジャンプすればいいのか分からないのでコレだけでは不確定になる。

```
EQ 0 looking_ptr index_of_value(0)
JMP 0 index_of_next_brace_end
```

#### 4.7 ]

[の逆をおこなえばいい。EQ による条件分岐まで戻りたいので少し多めに戻る。

```
EQ 1 looking_ptr index_of_value(0)
JMP 0 index_of_next_brace_start - 2
```

## 5 実装

ここに実装を羅列すると長くなるので、Lua が扱える中間表現にする部分だけを置いておく。完全版は `appendix*7` を参照してください。

以下のような中間表現にしていく。

Code 3 intermediate representation

```
1 {
2   operand1,
3   operand2,
4   operand3,
5   op = opcode
6 }
```

`operand` はそれぞれ `number` 型、`opcode` は `string` 型となる。これが実行順のリストとして table に突っ込まれていく。

---

\*7 <http://appendices.word-ac.net/WORD42/brainfuck-compiler-for-LuaVM/>

せっかくなので MoonScript で実装しよう。MoonScript というのは、親や友達に聞いてみよう。あるいは以下の URL<sup>\*8</sup> <sup>\*9</sup> を参照されたい。

Code 4 parser.moon

```

1  MOVE   = (a, b)  -> {a, b, 0, op: "MOVE"}
2  LOADK  = (a, b)  -> {a, b,  op: "LOADK"}
3  ADD    = (a, b, c) -> {a, b, c, op: "ADD"}
4  SUB    = (a, b, c) -> {a, b, c, op: "SUB"}
5  CALL   = (a, b, c) -> {a, b, c, op: "CALL"}
6  JMP    = (to)     -> {0, to, op: "JMP"}
7  EQ     = (a, b, c) -> {a, b, c, op: "EQ"}
8  RETURN = (a, b, c) -> {a, b, c, op: "RETURN"}
9  GETTABUP = (a, k) -> {a, 0, k, op: "GETTABUP"} -- a = _ENV[K(C)]
10 GETTABLE = (a, b, c) -> {a, b, c, op: "GETTABLE"}
11 nth = (t, c) -> for i = 1, #t do return i if t[i].val == c
12 K = (k) -> k - 1
13 RK = (k) -> k > 0 and -k or K(k)
14
15 ptr_boundary = {start: 4, end: 252}
16 looking_ptr = ptr_boundary.start
17 call_ptr = ptr_boundary.end + 1
18
19 -- position of register
20 READ_PTR = 0
21 WRITE_PTR = 1
22 CHAR_PTR = 2
23 BYTE_PTR = 3
24
25 (input) ->
26   jmp_stack = {}
27   ins_idx = 0
28

```

<sup>\*8</sup> <http://www.word-ac.net/wordバックナンバー/36-学年ビリのword部員が1年で精神年齢を20下げて>

<sup>\*9</sup> <https://moonscript.org>



```

29  constant = [{
30      val: e
31      type: switch type e
32          when "number" then (math.tointeger e) and 0x13 or 0x03
33          when "string" then (#e > 255) and 0x14 or 0x04
34      }] for e in *{"io", "string", "read", "write", "char", "byte", 1, 0}]
35
36  -- position of constant table
37  ZERO = nth constant, 0
38  ONE = nth constant, 1
39
40  instruction = {
41      GETTABUP 0, -(nth constant, "io")
42      GETTABUP 2, -(nth constant, "string"),
43      GETTABLE WRITE_PTR, 0, -(nth constant, "write") -- R(1) = io.write
44      GETTABLE READ_PTR, 0, -(nth constant, "read") -- R(0) = io.read
45      GETTABLE BYTE_PTR, 2, -(nth constant, "byte") -- R(3) = string.byte
46      GETTABLE CHAR_PTR, 2, -(nth constant, "char") -- R(2) = string.char
47  }
48
49  insertins = =>
50      table.insert instruction, @
51      ins_idx += 1
52
53  -- initialize R(i) to 0
54  insertins (LOADK i, ZERO - 1) for i = ptr_boundary.start, ptr_boundary.end
55
56  for c in input\gmatch'.'
57      switch c
58          when ">"
59              error "pointer limit (ptr <= #{ptr_boundary.end})" if looking_ptr >
                    ptr_boundary.end
60              looking_ptr += 1

```

```

61     when "<"
62         error "pointer limit (ptr >= #{ptr_boundary.start})" if looking_ptr <
            ptr_boundary.start
63         looking_ptr -= 1
64     when "+"
65         -- (*looking_ptr)++
66         insertins ADD looking_ptr, looking_ptr, -ONE
67     when "-"
68         -- (*looking_ptr)--
69         insertins SUB looking_ptr, looking_ptr, -ONE
70     when ",",
71         -- looking_ptr = io.read!
72         insertins MOVE call_ptr , BYTE_PTR
73         insertins MOVE call_ptr + 1, READ_PTR
74         insertins LOADK call_ptr + 2, ONE - 1
75         insertins CALL call_ptr + 1, 2, 2
76         insertins CALL call_ptr, 2, 2
77         insertins MOVE looking_ptr, call_ptr
78     when "."
79         -- io.write string.char *now_ptr
80         insertins MOVE call_ptr, WRITE_PTR
81         insertins MOVE call_ptr + 1, CHAR_PTR
82         insertins MOVE call_ptr + 2, looking_ptr
83         insertins CALL call_ptr + 1, 2, 2
84         insertins CALL call_ptr, 2, 1
85     when "["
86         jmp = JMP!
87         insertins EQ 1, looking_ptr, -ZERO
88         insertins jmp
89         table.insert jmp_stack, {line:ins_idx, jmp} -- determine where to JMP when
            corresponding "]"
90     when "]"
91         jmpback = table.remove jmp_stack

```

```

92     jx = ins_idx - jmpback.line
93     jmpback[1][2] = jx
94     insertins EQ 0, looking_ptr, -ZERO
95     insertins JMP -jx - 4
96 else
97     io.stderr\write "ignore #{c}\n"
98
99 insertins RETURN 0, 1, 0
100 instruction, constant

```

56 行目の `for c in input\gmatch'.'` で 1 文字ずつ拾っていき、`switch` 式で Brainf\*ck の命令を解釈して対応する LuaVM の命令を `instruction` に突っ込んでいく。この `instruction` がコンパイラーに渡され、Lua VM のバイトコードとして吐き出される。

## 6 実行

とりあえず動かしてみたい。ソースコードがあったら人はそう思う。付録のアレを使ってみよう。2.8 で示した Hello, World! を動かしてみる。

Code 5 compile & run ``Hello, World!''

```

$ cat <<BF | bfc4lvm -
+++++++[>+++++++>+++++++>++++<<-]>.>+.+++++.+++>-.-----
<+++++++.-----.+++-----.-----.>+.
BF
$ lua bf.out
Hello, World!

```

ウオッ?! 動いた…… やっぱり作ったものが動く楽しいですね。ここでどういう命令に落ちたかも気になってくるわけだ。`$ luac -o /dev/null -l -l bf.out | less` でな具合で確認してみよう。

Code 6 The instructions for ``Hello, World!''

```

$ luac -o /dev/null -l -l bf.out | less

main <?:0,0> (405 instructions at 0x19609e0)
0 params, 255 slots, 1 upvalue, 0 locals, 8 constants, 0 functions

```

```

1      [-]  GETTABUP    0 0 -1 ; - "io"
2      [-]  GETTABUP    2 0 -2 ; - "string"
3      [-]  GETTABLE    1 0 -4 ; "write"
4      [-]  GETTABLE    0 0 -3 ; "read"
5      [-]  GETTABLE    3 2 -6 ; "byte"
6      [-]  GETTABLE    2 2 -5 ; "char"
7      [-]  LOADK        4 -8   ; 0
8      [-]  LOADK        5 -8   ; 0
9      [-]  LOADK        6 -8   ; 0
10     [-]  LOADK        7 -8   ; 0
11     [-]  LOADK        8 -8   ; 0
12     [-]  LOADK        9 -8   ; 0

.....
.....
253    [-]  LOADK       250 -8 ; 0
254    [-]  LOADK       251 -8 ; 0
255    [-]  LOADK       252 -8 ; 0
256    [-]  ADD         4 4 -7 ; - 1
257    [-]  ADD         4 4 -7 ; - 1
258    [-]  ADD         4 4 -7 ; - 1
259    [-]  ADD         4 4 -7 ; - 1

.....
.....
288    [-]  ADD         8 8 -7 ; - 1
289    [-]  SUB         4 4 -7 ; - 1
290    [-]  EQ          0 4 -8 ; - 0
291    [-]  JMP          0 -26 ; to 266
292    [-]  ADD         5 5 -7 ; - 1
293    [-]  ADD         5 5 -7 ; - 1
294    [-]  MOVE        253 1
295    [-]  MOVE        254 2
296    [-]  MOVE        255 5
297    [-]  CALL        254 2 2

```

298	[~]	CALL	253 2 1
.....			
.....			

フーム、なかなかそれっぽいものが生成されている。

## ところで

Brainf\*ck の言語仕様上、ポインターは 30000 個以上の要素を持つ配列の上を移動できなければならないのだが、Lua VM のレジスターは 255 個しかない。255 - 30000 = -29745 (< 0) のことですが、皆さんは暗算できますか。今日のところはこれにて解散です。

## 7 おわりに

11 月<sup>\*10</sup> といえば、コミックキューンから『シーズンシアターシアトリカル』<sup>\*11</sup> の 1 巻が発売じゃないですか!!!!!! 『ひだまりスケッチ』9 巻や『北陸とらいあんぐる』<sup>\*12</sup> とかうわあああああああ『エクレア あなたに響く百合アンソロジー』<sup>\*13</sup> も発売じゃん!!!!!! 2016 年 11 月に人類が滅亡したりしなかったりするなこれは……。パソコンカタカタしとる場合じゃないぞ、おい……。

12 月の話もしていいですか?! 12 月の話もしたいな〜でもな〜12 月発売の百合漫画群の話をすると来年 1 月の話もしたくなるので、今日はこのへんで。

---

<sup>\*10</sup> この記事は 11 月に書かれました。

<sup>\*11</sup> <https://www.amazon.co.jp/gp/product/404068785X/> 『ふ〜ふ』や『ゆりどりみどり』などのラブラブ百合を描いてる源久也先生じゃないですか!!! コミックキューン最高!!!

<sup>\*12</sup> <https://www.amazon.co.jp/gp/product/4040685822/>

<sup>\*13</sup> <https://www.amazon.co.jp/gp/product/404892432X/> とうとう出てしまった電撃コミックス発の百合アンソロジー。『やがて君になる』の作者、仲谷鳩先生が表紙を描き、執筆陣にも十二分な破壊力を持った、中身を見ずとも完全勝利であることがわかる一冊。制作発表した日に数回心肺が停止した。

# Go で美少女ゲームをスクレイピング

文 編集部 ひだるま

## 1 はじめる前に

本記事に登場するキャラクターは全て低身長の18歳以上です\*<sup>1</sup>。

また、本記事のリンク先が18歳以上を対象としたものである場合も特に記述はしていないのでご了承下さい。

## 2 はじめに

皆さんはえっちいゲームの情報をどこから入手するでしょうか。人伝ひとづて、Twitter、美少女ゲーム雑誌、公式サイトなど色々ありますが、そこそこ早くて統一的に見られるサイトが嬉しいですね。該当するのがDMM.R18\*<sup>2</sup>やSofmap\*<sup>3</sup>などダウンロード販売や流通のサイトです。その中でも筆者が使っているのはGetchu屋のサイト、Getchu.comです。謂わば集合知であるErogameScape\*<sup>4</sup>はSQLによる情報入手ができますが、SQLを叩ける場所が許可制なので\*<sup>5</sup>今回はスルーします。

毎度Webブラウザでアクセスするのも手間です。そこで必要最低限の情報をターミナル画面で得られるようにしたいと思います。幸いにして、Getchu.comには月毎に発売するタイトルをまとめたページがあるので、そこから情報を取得することにしました。

今回は、以下の環境で実装、実行しました。実行の際はインターネット環境が要るので悪しからず。

OS	Ubuntu 16.04
言語	Golang 1.6.1
スクレイピングライブラリ	scrape
言語環境	UTF-8
(Web 解析)	Firebug (FireFox)

とはいっても、Go言語は1つのビルド環境で様々な実行環境に対応可能かつ容易かつバージョン間の差異が(今のところは)ほとんど無いので、別環境でも上手く動作するかと思います。

\*<sup>1</sup>ちっちゃくないもんっ！

\*<sup>2</sup><http://dlsoft.dmm.co.jp/>

\*<sup>3</sup><https://www.sofmap.com/>

\*<sup>4</sup>[https://erogamescape.dyndns.org/~ap2/ero/toukei\\_kaiseiki/](https://erogamescape.dyndns.org/~ap2/ero/toukei_kaiseiki/)

\*<sup>5</sup>localhostはOKなので実はそんなに困らないのですが。

### 3 Web スクレイピング

(Web) スクレイピングとは、Wikipedia の記述<sup>\*6</sup>を要約すると「情報を Web サイトから抽出する技術のこと」です。今回は Getchu.com のサイトから html を取得し、必要な情報を抽出して標準出力に出力します。

Go 言語においては恐らく goquery ライブラリ<sup>\*7</sup>の方がよく見ますが、jQuery ライクと言われても jQuery 分らないし……という訳で、曰く

"A simple, higher level interface for Go web scraping."

という scrape ライブラリ<sup>\*8</sup>を利用しました。

### 4 ディレクトリ構成

```
$GOPATH
├── bin
├── pkg
├── src
│   ├── github.com
│   ├── golang.org
│   └── getchu.....ここに作りました。
│       └── getchu.go
```

### 5 コード

後述の Listing 1 の上から流していきます。

#### 5.1 ライブラリ

3 行目からです。

- html 取得までに必要な http、url、cookiejar
- 年月の引数及びデフォルト値設定に必要な flag、time
- 取得した html から情報の抽出を行うのに用いる scrape、atom、html
- 情報を変換、出力するのに用いる ioutil、fmt、transform、japanese、strings

を利用しています。

---

<sup>\*6</sup><https://ja.wikipedia.org/wiki/ウェブスクレイピング>

<sup>\*7</sup><https://github.com/PuerkitoBio/goquery>

<sup>\*8</sup><https://github.com/yhat/scrape>



## 5.2 時刻フォーマット

20 行目からです。const に予め時刻フォーマット用の値、stdLongYear と stdNumMonth を用意しておきます。Go 言語におけるフォーマットの指定方法であって、2006 や 1 自体をデフォルト値に設定している訳ではないので注意しましょう。

## 5.3 グローバル変数

25 行目からです。入手したい年月を引数に指定するので、格納するための変数 year と month を用意します。

## 5.4 エンコード関数

30 行目からです。eucjpToUtf8 は EUC-JP の文字列を受け取って UTF-8 の文字列に変換して返します。Getchu.com は EUC-JP で構成されるサイトなので、用意しておかないと折角取ってきた情報が全く読めないなんてことになります。

## 5.5 引数の展開

40 行目からです。flag.StringVar で登録した引数を flag.Parse で展開します。それぞれ short flag (一文字) 版も用意しています。デフォルト値に現在の年月を指定するために、time を使用しています。Format に const で宣言したフォーマットを指定しています。

## 5.6 html の取得 with Cookie

46 行目からです。ただ html をリクエストしても、望んだえっちな情報は得られません。年齢確認ページに飛ばされてしまいます。Web ブラウザならそれで「はい」をクリックして情報に辿りつけば良いのですが<sup>\*9</sup>、今回は一発でファイルを取得するためクエリに年齢確認済みを示す Cookie<sup>\*10</sup> をくっつけておきます。元となる URL が getchuURL、これに cookiejar<sup>\*11</sup> を利用して Cookie をセットし、展開した引数を格納し、client.Do にて実際にページ要求を行っています。

## 5.7 必要情報の抽出

71 行目からです。ようやく html を取得しても、生の html をそのまま読むくらいなら Web ブラウザで普通に閲覧した方が余程見やすいです。必要な情報が載っているタグを見極め、抽出しましょう。

先程取得し、パースした html は html 型になっていますから、抽出もそれに合わせて行います。matcher に限定するタグを設定していきます。atom には各タグなどの const が詰まっています。さしあたって必要なのは発売日<sup>\*12</sup>、タイトル名、ブランド名として、生 html と睨めっこした成果が Listing 1 になります。

matcher を設定したら、html から情報を抽出していきます。複数のタグを同時にかつ個別に取得する良い方法が分からなかったので一旦 map に格納して最後に一緒に呼び出すことにしました。格納する際にテキス

---

<sup>\*9</sup> 本記事は 18 歳以上の読者を想定しています。

<sup>\*10</sup> getchu\_ad 「a」 lt\_flag なのが数時間悩んだポイント。

<sup>\*11</sup> 実際の所、よく分からないで使っています。

<sup>\*12</sup> 月末金曜発売とは限らないのです。

ト部分を抽出して `string` 型にしていますね。

Listing 1: getchu.go

```
1 package main
2
3 import (
4     "flag"
5     "fmt"
6     "io/ioutil"
7     "net/http"
8     "net/http/cookiejar"
9     "net/url"
10    "strings"
11    "time"
12
13    "github.com/yhat/scrape"
14    "golang.org/x/net/html"
15    "golang.org/x/net/html/atom"
16    "golang.org/x/text/encoding/japanese"
17    "golang.org/x/text/transform"
18 )
19
20 const (
21     stdLongYear = "2006"
22     stdNumMonth = "1"
23 )
24
25 var (
26     year  string
27     month string
28 )
29
30 func eucjpToUtf8(str string) (string, error) {
31     ret, err := ioutil.ReadAll(transform.NewReader(strings.NewReader(str
```

```
        ), japanese.EUCJP.NewDecoder()))
32     if err != nil {
33         return "", err
34     }
35     return string(ret), err
36 }
37
38 func main() {
39     t := time.Now()
40     flag.StringVar(&year, "year", t.Format(stdLongYear), "sell_year")
41     flag.StringVar(&year, "y", t.Format(stdLongYear), "sell_year_short")
42     flag.StringVar(&month, "month", t.Format(stdNumMonth), "sell_month")
43     flag.StringVar(&month, "m", t.Format(stdNumMonth), "sell_month_short")
44     flag.Parse()
45
46     getchuURL := "http://www.getchu.com/all/price.html?genre=pc_soft&
        gage=adult"
47     jar, _ := cookiejar.New(nil)
48     var cookies []*http.Cookie
49     cookie := &http.Cookie{
50         Name:   "getchu_adalt_flag",
51         Value:   "getchu.com",
52         Path:    "/",
53         Domain:  "www.getchu.com",
54     }
55     cookies = append(cookies, cookie)
56     u, _ := url.Parse(getchuURL)
57     jar.SetCookies(u, cookies)
58     client := &http.Client{
59         Jar: jar,
60     }
61 }
```

```
62     postData := strings.NewReader("&year=" + year + "&month=" + month)
63     req, _ := http.NewRequest("POST", getchuURL, postData)
64     req.Header.Add("Content-Type", "application/x-www-form-urlencoded")
65     resp, err := client.Do(req)
66     if err != nil {
67         panic(err)
68     }
69     defer resp.Body.Close()
70
71     root, err := html.Parse(resp.Body)
72     if err != nil {
73         panic(err)
74     }
75
76     matcherGame := func(n *html.Node) bool {
77         if n.DataAtom == atom.A && n.Parent != nil && n.Parent.
78             Parent != nil {
79             return scrape.Attr(n.Parent, "align") == "left"
80         }
81         return false
82     }
83
84     matcherSellday := func(n *html.Node) bool {
85         if n.DataAtom == atom.Td {
86             return scrape.Attr(n, "class") == "black" && scrape.
87                 Attr(n, "align") == "center"
88         }
89         return false
90     }
91
92     matcherBrand := func(n *html.Node) bool {
93         if n.DataAtom == atom.Td {
94             return scrape.Attr(n, "class") == "black" && scrape.
```

```

    Attr(n, "align") == "left"
93         }
94         return false
95     }
96
97     games := scrape.FindAll(root, matcherGame)
98     selldays := scrape.FindAll(root, matcherSellday)
99     brands := scrape.FindAll(root, matcherBrand)
100
101     selldayList := make(map[int]string)
102     for i, sellday := range selldays {
103         selldayList[i] = scrape.Text(sellday)
104     }
105
106     brandList := make(map[int]string)
107     for i, brand := range brands {
108         brandList[i], err = eucjpToUtf8(scrape.Text(brand))
109         if err != nil {
110             panic(err)
111         }
112     }
113
114     for i, game := range games {
115         game, err := eucjpToUtf8(scrape.Text(game))
116         if err != nil {
117             panic(err)
118         }
119         fmt.Printf("%s: %s | %s\n", selldayList[i], game, brandList[i]
120             ])
121     }
}
```

## 6 実行ファイル作成

```
$ go install getchu
```

のようにして実行ファイルを作成します。\$GOPATH/bin/にあるはずです。PATH を通してあげましょう。

Mac 用のバイナリを作りたい場合、

```
$ GOOS=darwin GOARCH=amd64 go build -o getchu ./src/getchu/getchu.go
```

の様にすれば作れる、はずです。

## 7 おわりに

```
$ getchu -y 2016 -m 3
```

03/17: きんとうか 初回限定版 | GRISEDGE

03/24: こいのす☆イチャコライズ<早期予約特典+予約特典:おっぱい色紙 3 枚付> | eRONDO

03/24: はるるみなもに! <早期予約キャンペーン特典付き> 初回限定版 | クロシェット

03/24: 春音アリス\*グラム <早期予約キャンペーン特典付き> | NanaWind

03/24: 神頼みしすぎて俺の未来がヤバイ。 初回版 | HULOTTE (ユロット)

03/24: 想いを捧げる乙女のメロディー | ensemble

03/24: 働くオタクの恋愛事情 | あかべえそふとすりい

03/24: 魔女と剣と千の月 | KAI

```
echo ^^c2^^a9 Getchu.com 1999-2016
```

```
^^c2^^a9 Getchu.com 1999-2016
```

のように、情報を取得することができました。

また、<http://laughingandgrief.hatenablog.com/entry/2016/11/16/165025> にてターミナル越しに getchu 検索をできるようにした例を載せているので、よろしければご覧ください。

今回は筆者が使えないので使っていませんが、チャネルやゴルーチンといった Go 言語らしい機能を使って何か書けるようになればいいなあと考えていたり。

今回参考にしたサイトは以下になります。

- <http://akb428.hatenablog.com/?page=1437903964> Ruby による先達様
- <https://gist.github.com/Rabbit52/a8a44c3c4cd514052952> Cookie の埋め込み
- <http://qiita.com/uchiko/items/1810ddacd23fd4d3c934> 文字コード変換

# 段ボール活用事例報告会

文 編集部 びしょ〜じょ

## 1 はじめに

こんにちは、びしょ〜じょです。お宅に段ボールは溜まってませんか? そうですね、百合漫画を amazon で2冊以上頼むと箱が大量に増えますね。捨てるのも面倒だし何かに使いたい。と。

## 2 準備

まず準備します。とにかく箱がほしい。この時期<sup>\*1</sup>に手に入りそうな箱というと、そうですね、『【Amazon.co.jp 限定】アイカツ! あかり Generation Blu-ray BOX6(豪華版)(初回生産限定)(描き下ろし B1 サイズ布ポスター付き)』<sup>\*2</sup>を amazon で注文してみましようか。他にも発売日が近い百合漫画を適当に購入して<sup>\*3</sup><sup>\*4</sup><sup>\*5</sup>『できる限り商品をまとめて発送(最小限の配送料・手数料がかかります)』にチェックを入れて注文します。

わ〜い箱がいっぱいだあ



図 1: 百合漫画などが梱包されていた箱達.png

これだけあれば充分でしょう。

---

<sup>\*1</sup> この記事を書いたのは 2016 年 7 月 5 日です。

<sup>\*2</sup> <https://www.amazon.co.jp/gp/product/B0181W8HV2/>

<sup>\*3</sup> <https://www.amazon.co.jp/gp/product/4040682904/>

<sup>\*4</sup> <https://www.amazon.co.jp/gp/product/4040686047/>

<sup>\*5</sup> <https://www.amazon.co.jp/gp/product/4048659979/>



### 3 活用事例 1: キーボード & ディスプレイの高さ調節

難易度: ★

大事ですね。

#### 3.1 作り方

1. 段ボールを引っ張ってくる
2. 適当に折って高さを調節
3. 輪ゴムで縛る
4. 完成

工程 3. が重要なんやな。これで滑り止めるわけです。



図 2: 大事です。

## 4 活用事例 2: 隙間を埋める

難易度: ★

居間に除湿機を設置して、ホースを経由して風呂場に排水している人は多いと思います。しかしホースを扉で挟んでしまうと排水が滞るので、若干扉と壁に隙間が開くわけですね。居間にエアコンがあり、扉を一枚挟んだ風呂場のある方にはエアコンがないので温度差がある、という社会構造は T 市<sup>\*6</sup>の住人はよく体験していると思います。隙間は積極的に埋めていきましょう。

### 4.1 作り方

1. 段ボールを引っ張ってくる
2. 適当にカット
3. 貼り付ける
4. 完成

やるだけ。



図 3: 寒かったり暑かったりするので、隙間は埋めたほうが良いです.png

---

<sup>\*6</sup>I 県 T 市。T 大学や T 山があり、私も住んでいる。

## 5 活用事例 3: 天空要塞水切りカゴの礎

難易度: ★★

台所が異常に狭いので水切りカゴを空中に床に貼り付けておきたいというお客様はいらっしゃいませんか。  
そのままだと傾くのでスツとやる。



図 4: ススツ…?



図 5: [margin-temae: 1danbouru;](#)

## 6 活用事例 4: ファンシーな小物入れ

難易度: ★★★

かわいいは作れる! 本棚に埋め込めるぞ。

### 6.1 作り方

1. 段ボールを用意します
2. 箱状にする
3. ガムテープで固定
4. 完成♡

本棚に埋め込みたいときは工程 2. が難しいかもしれませんねえ…。



図 6: ドキュート

## 7 活用事例 5: 爆安通気性最高 PC ケース

難易度: 高め

なんと PC ケースができてしまうんですね～5000 円くらい浮くのでピカピカ光るファンを買い放題です。

### 7.1 作り方

1. 段ボールを持ってくる
2. H170M Pro4 を購入
3. Intel Pentium G4400 を購入
4. ZOTAC GeForce GTX 750 Ti を購入
5. 電源とメモリ、CPU ファンを適当に購入
6. PC を組み立てる
7. 寸法を測る

8. 段ボールから切り出す
9. ファンの位置などを考え、穴を開ける
10. テープで固定
11. できあがり

マザボのピンに段ボールが刺さってショートしてしまい、いつまで経っても PC が起動せず首を傾げるなどのハートフルストーリーがあります。

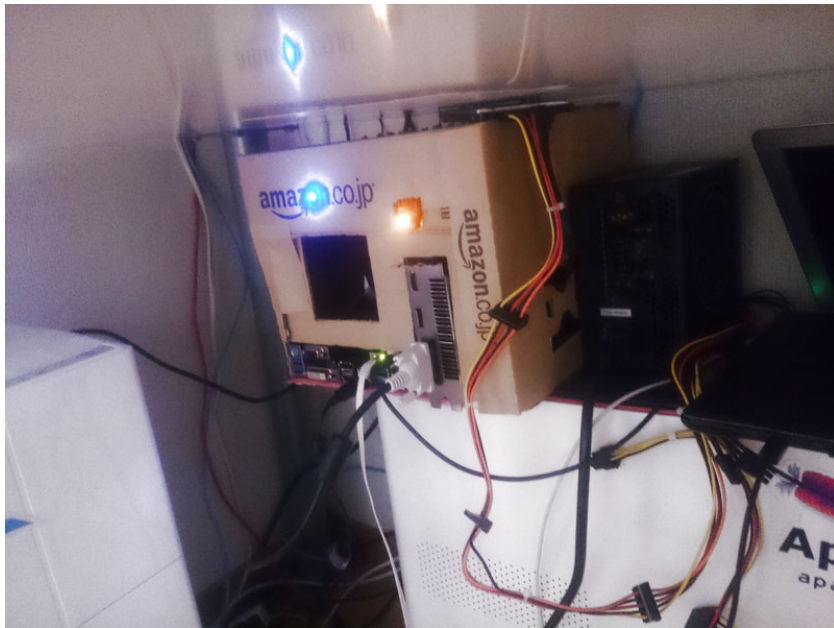


図 7: amazOn セレクト

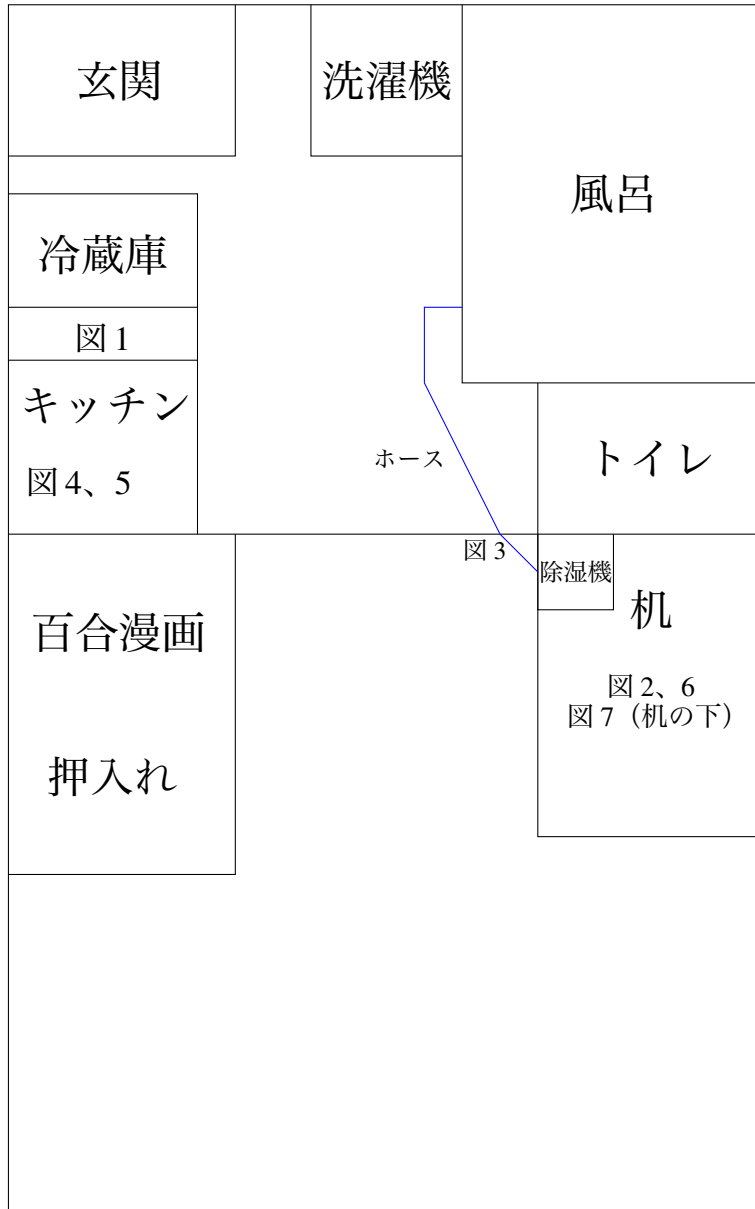
## 8 おわりに

人生ケチらずに行こう!! もっといっぱい百合漫画を買いましょう。段ボール? 何の話ですか?

## 付録

### 部屋の見取り図

本記事を 120%楽しむための、我がアパートの一室です。



## 池袋アンダーグラウンド

文 編集部 みみずのひもの

### 1 Introduction

事の始まりは、とある金曜日の黄昏時だ。私と本誌編集者びしょ〜じょ（22 歳男性）は中央体育館第二トレーニング場で一汗流し\*1これから中華料理専門店の笑飯店に行って腹ごなしでしようやという話が決まっていた頃合いだ。彼は何気なく提案した。

「みみずさ〜ん、明日ドッツのライブ行かないッスか？」

「……………」（以降「ドッツ」）\*2 とは本誌 OB の ranha 氏が最近ドハマリしている、原宿系新人アイドルグループのことだ。今年の 9 月 4 日に原宿ストロボカフェで鮮烈なデビューを飾ったドッツは、「目を隠す」「本名非公開」「アイドルの仮名が毎月変わる」など、秘密工作員も顔負けの匿名性重視アイドルとして売り出されている。どうやらそういうコンセプトらしい。当初は初音ミクくんの楽曲カバーを歌っていたが、近頃オリジナル楽曲も発表しているという、リアルな駆け出しアイドルだ。

「うーん、ドッツかあ……」

笑飯店に到着し、新メニューの砂肝辛口炒めをもきゅもきゅと噛み締めながら、私はあまり乗り気ではない旨を口から漏らした。

「びしょ〜じょくん、それ入場料いくら？」

「事前予約すれば 3000 円ッス」

「う〜〜〜ん……」

現在の通貨でおおよそ 1 焼肉、1 春秋航空国際便\*3に相当するが、ライブイベントとしては安い部類に入らるろう。ちなみにびしょ〜じょは事前に ranha 氏とのやり取りで予約済みだそう。さすがこの男、その場の勢いに天命を委ねている。

「場所？」

「池袋」

まさかのブクロ。未だにブクロと聞くと、ウエストゲートパークだとかカラーギャングのメッカというイメージがまず浮かんでしまうあたり、完全に頭が古典的概念に囚われるお爺ちゃんだ。とはいえ乙女ロードが存在したり、実はデュラララ!!の聖地らしいし、昔アイドルマスターのスタンプラリーで池袋中を歩き回った

---

\*1 筑波大生は講習会さえ受講すれば、無料でトレーニングルームを利用することができる。高価なトレーニング機器が使い放題なのでオススメ。同様にプールも申請さえすれば利用可能である。

\*2 公式 HP:<http://dots.tokyo>  
公式 Twitter 広報:<https://twitter.com/dotstokyo>

\*3 爆安 LCC 春秋航空のタイムセール期間に 3000 円を投げ込むと、重慶や武漢などといった中国の田舎まで片道で移動できる。ただし帰りは別途必要だし、空港使用料も含まれていない。そうそうに安くてうまい話はないという教訓だ。



記憶などを掘り下げると、近年はヲタクシティ化しているという事実が容易に類推できた。

「今月忙しいんだよなあ……」

「でもみみずさんどうせ今週暇でしょ？」

図星だ。来週再来週と怒涛の激務に忙殺される予定だったが、奇妙にも今週は暇だったのだ。そして外付け HDD<sup>\*4</sup> 買ったり、PSP の充電器買ったり、スマホのフィルム買ったりなどで、一応都内まで出て買い物をする予定もこさえていた。その時の私は奇遇にも、興味の方が面倒臭さを僅かに上回ったのだろう。

「じゃあ一応事前予約して、かつ朝起きてたら行ってみるか……」

「アーイ」

こうして私は、人生で初めて地下アイドルの現場へと身を投じることとなったのである……。

## 2 Preliminaries

地下？ アイドル？ 本記事を目にされた賢明なる WORD 読者の皆様方は、あまり馴染みのないカルチャーを目の当たりにし混乱している頃合いだろうか。あるいは、ドルヲタクっさ！ などと悪態を付きながら、次の記事にページを手繰り始めた頃合いだったかもしれない。確かにそのお気持ちもごもつともだがちゃんとこれから順を追って説明するので、お時間さえよろしければ、どうかもうしばらくこの駄文にお付き合い願いたい。

何故比較的オーソドックスな硬派を自称していた編集部員達が、アイドルという大衆的文化の闇に飲み込まれてしまったのか……。それを語るためにはおよそ 4 年の歳月に及ぶ史実をここに記すことになるが、話せば長くなるので要点のみを端的にまとめるとしよう。

古来 (A.D.2012) から編集部員とはアイドルゲームやアニメに傾倒する生物であった。東でアイドルマスター<sup>\*5</sup> のライブがあればヲタク棒<sup>\*6</sup> を必死に振り、西でラブライブ！<sup>\*7</sup> のグミが発売されればヲタクが編集部員に無償配布し、北に筑波山があればアイカツ<sup>\*8</sup> と称して登頂し、南の西友にプリパラ筐体が爆誕すれば<sup>\*9</sup> 早速遊びに向かったのである。テレビに出てくるようなアイドルはよく知らないけど、3DCG やアニメーションで動くアイドルに向けてならいくらでも応援できる……編集部にはそういったアイドルを賛美する土壌が既に備わっていたと言えよう。

<sup>\*4</sup> おじいちゃんなので俗世に疎かったが、外付け HDD3TB でも 1 万円切ることを知った。これで Macbook の Time Machine でバックアップできるね。

<sup>\*5</sup> 全部で 9 人だったり 10 人だったり 13 人だったり 37 人だったり 183 人だったりする所属人数が不定期なアイドル事務所のアイドル候補生を、プロデュースしたりしなかったりするゲームだったりアニメだったりする神話。今年で 11 年目に入る。

<sup>\*6</sup> サイリウムやペンライトのことを指す。

<sup>\*7</sup> 廃校寸前の高校を救済すべく、聖ホノカスを中心とした 9 名の使徒が奮闘する過程を描いた神話。サン邪淫サンシャインという続編では、信奉者高海が当該神話を沼津の高校で再現せんと奔走する。

<sup>\*8</sup> 「これ、アイカツか？」「うん、これもまたアイカツだね！」で集約されるように、任意の行動をアイカツへと帰着させる神話。崖登りもアイカツの 1 つだが、筑波山山頂に居住するドレスデザイナーは未だ発見されていない。

<sup>\*9</sup> 自身の魂のステージを神アイドルまで高める神話。今現在ではつくば市内に 4 箇所所在するゲーム筐体だが、初のプリパラ筐体は西武デパート内子供服売場にしか存在しなかった。

風向きが変化したのは A.D.2014 の終わり頃、渡来人兼当時編集部員 shrinkbr が i ☆ Ris というアイドルグループを推し始めたところからだ。i ☆ Ris は先述のプリパラというアニメ・ゲーム作品でメインキャラクターの声優及び歌唱を担当しているアイドルグループだ。すなわちプリパラというコンテンツには、現実のアイドル文化とアニメ文化の融合という実験的要素も含まれている。ある日プリパラのライブで i ☆ Ris のメンバー芹澤優さんとの運命的な出会いを果たしてしまった彼は、月日を重ねるごとに一挙一足がドルヲタ（アイドルヲタク）のそれと化していった。ある日、彼は編集部員達へと文化を伝来しにやってきた。



図 1: 色とりどりのサイリウムを持ってヲタ芸をしている、アイドルファン（オタク）のイラストです。

「タイガー！ ファイヤー！」

「やあ shrinkbr くん、それは何の呪文だい？」

「ああ、これは MIX と呼ばれるものさ！」

ドルヲタがアイドル楽曲の間奏に咆哮する雄叫び——「MIX」の存在は瞬く間に編集部員の間で広まった。今までアイドルへのコールと言えば手拍子だとか「オオ〜ッ」だとか「フッフー」しか知らなかった部員たちにとって、間奏で勝手に訳の分からない呪文を喧しく叫ぶ行為は新鮮であり、多少なりとも驚きをもって迎え入れられたのである。

またある時も、彼は覚えてたの文化を携えて編集部員にやってきた。

「俺と一緒に人生歩もう！ 世界で一番愛してる！ ア・イ・シ・テ・ル〜！」

「やあ shrinkbr くん、それは新しい MIX かい？」

「うんにゃ、これはガチ恋口上と呼ばれるものさ！」

自分の推しアイドルに向けて、己がガチで恋している様子を口上として体現化した魂の叫び——「ガチ恋口上」も有志によって速やかに暗唱された。他にもその後推しジャンプ<sup>\*10</sup>、リフト・モッシュ、咲クラブ<sup>\*11</sup>……などの厄介行為の数々は次々と編集部員に伝来されていった。本来ドルヲタがアイドルに注目されるべく生み出した苦肉の必殺技は、あまりにも刺激的で、そして魅力的だったため、自分でも実際にやってみたいという気持ちが非常に強まってしまうのだ。この時点で既に、我々は普通のコールでは満足できない身体になっていたのだろう。

A.D.2016 に突入したあたりから、ドルヲタの行動そのものを研究しようという動きが広まった。研究に惜しみのない我々編集部員は、これらの新しい概念・行動がアイドル現場で発掘される度、動画等でそれを細

<sup>\*10</sup> 推しの歌唱中指を指しながらジャンプを繰り返す行為。またの名をマサイとも。

<sup>\*11</sup> 落ちサビ（C メロ）のしっとりとした曲調の場面で、花が咲くように腕を広げながらゆっくりとクラブする行為。現場によってチンパンジークラブとして嫌厭されることもあるので注意。

かく検証し、実戦での投入を想定して日々練習に練習を重ねていた。「タイガー！ ファイヤー！」のように生易しかったコールも、この頃には

「ア〜〜ッッ！（パンパパンパン） ッシャ〜イグゾォ!!! ッタイギァア!!! ッファイアア!!!」

という聞くに耐えない、大変陰しい咆哮へと変貌していた。

特に今年の 8 月にお台場で開催された「TOKYO IDOL FESTIVAL 2016」というアイドルの大祭典に参加したドルヲタ編集部員たちは、客席の最前線に紛れ込んでコールを打ち、現場のヲタクとほぼ同型のコールを再現するにまで至ったのだ。ちょっとした興味からドルヲタの真似をしているうちに、我々はドルヲタに引けを取らない存在になってしまったのである。

そして現在に至る。結果的にドルヲタの深淵を覗いてしまった者達は、次々とドルヲタへと堕ちてしまった。大学入学前からアイドルグループ、Wake Up, Girls! を愛して止まなかった **コンプラ** くんは、いつの間にやら *BELLING* 少女ハートという、モッシュ、リフト、ダイブ等諸々過激なパフォーマンスを生業とする<sup>\*12</sup>アイドルグループの熱狂的シンパと化していた。かつてラブライブ！ 最初期の CD を布教のため無償で頒布していた心優しき ranha さんも、先述のドッツというアイドルグループの立ち上げから現在までを進行形で見守り続けている最古参ヲタクになってしまった。shinkbr は公務員の平均月収額分ぐらい i ☆ Ris の新盤 CD を購入した。こうして皆何かに堕ちていったのだ、誰も彼もが……。



図 2: アイドルの CD についてくる握手券を目的に、大量の CD を買って山積みしている男性のイラストです。

このようにしてアイドル現場にいる面白いドルヲタを観察しようという当初の目標はいつしか崩壊し、自らが面白ドルヲタとなりかけていた編集部員達。しかし未だ朱に染まっていなかったのが、私とびしょ〜じょの二人だ。我々には今や面白人間になってしまった ranha さんを観察する義務がある。池袋へと足を向け 3000 円を投じたのは、そういった使命感にかられての行動だったのかもしれない……。

### 3 Survey

翌朝 8 時、妙に目覚めは快適だった。同じバスに乗っていたにもかかわらず何故か巡り会えなかった私とびしょ〜じょの両名は、ひとまずつくば号で東京駅に降り立ち、そこから丸ノ内線で池袋まで辿り着いた。霧雨が降りしきる中我々は人混みをかき分けかき分け、会場となる池袋ルード K3 へと足を運んだのである。

まず入り口を見て驚いた。閑散とした住宅地にひっそりと佇む雑居ビル……何の変哲もないビルの 1 階に、地下へと通じる入口がただ鎮座していた。

<sup>\*12</sup> 本人曰く「ヲタクが過激なだけでアイドルはそうでない（ちょっとライブ中に毒霧するぐらい）」らしい



図 3: 父さん母さん、僕は大人の階段を下ります



図 4: 後ろから見るヲタクの姿は勇ましい

謙虚な見た目からは想像もつかないほど奥まで伸びる深淵は、まるで都内にぽっかりと開通した洞窟だった。そうだ、我々が今から潜入するのは地下アイドル現場の最前線。あれこそが現<sup>うつ</sup>世と隠<sup>かく</sup>り世を隔<sup>はざかい</sup>てる端境なのだ。ここまで来たら引き返す選択も脳裏にはよぎらない。入り口に掲げられた「GET IT ON!」の言葉に導かれるがまま、我々は階段に足を踏み入れた。

受付で 3000 円+ドリンク代 600 円を支払い入場した。受付のお姉さんに「どのアイドルさん目的で来られましたか？」と聞かれて、まだ実物すら見たこと無いドッツの名を本当に挙げてよいのか葛藤し、「あつ、ドッ……ドッ……」と挙動不審になってしまった経緯について多くは語らない。重々しい扉を一枚押し広げると、フロアから流れ出た大音響が耳に飛び込んでくる。最前線でただ 1 人ジャージャーするヲタク、後ろで腕を組みじっと地蔵（コールやクラップを打つことも無く、地蔵のようにじっと静止する様）を決め込むヲタク、ステージ上で必死に歌とダンスを披露するアイドル、暇そうなバーテン。地下アイドル現場という未踏の魔境が、そこには実在していた。

何故だろう。私自身、何も初めてアイドルのライブに来たわけではない。しかし今まで味わったことのないこの新<sup>フレッシュ</sup>鮮な感覚は、一体何だろうか。そうだ……根本的に異質なのは客層だ。合計で 20 人もいないこの空間では、決して全ての人間が 1 つの主体性に染まることはない。各々が推しアイドルというアイデンティティを確立した個別のドルヲタなのだ。故にあまり人気のないアイドルだと、客はいれども本当に自分たちの事を知るファンは 3,4 人しかいない状況が容易に発生する。今まで数百から数千人規模のライブ会場しか見たことがなかった私にとって、このファンの声援が文字通りダイレクトに直結する環境は未踏の領域だった。

となればアイドル側も攻め方が変わってくる。歌唱中に客とハイタッチを交わしたり視線を交わして手を降ったりと、少人数ならではの積極的なやり取りをファンとガッツリ交わしてくようになるのだ。ヲタクは少人数でも力の限り声援をねじ込み、それに呼応するかのようにアイドルはファンサービスで答える。互いに循環し合うモデルがそこには完結していた。

とはいえまずは様子見だ。この地下空間に順応するため、アイドルとヲタクの挙動を見極めなければならない。私は手持ちの鞆をロッカーに預け、バーテンにカールスベルグを一杯注文し、がっしりと腕を組んで会場後ろに仁王立ちしながら舞台上のアイドルを眺め始めた。なお会場内は事前に注意があったアイドルを除いて撮影可能だったので、掲載写真は全て合法である。読者の皆様はご安心頂きたい。

## リトルモンスター

「続いての出演は初登場、リトルモンスターさんです！」という紹介で、ステージに現れたのは小学生二人組だった。あっけにとられたが、紛うこと無く小学校低学年の二人組だった。

### Tips

この瞬間私の脳裏によぎったのは、先日テレビ東京で紹介されていた「あいちゃん6さい」という業の深いアイドルだ。彼女はその名の通り6歳児のアイドルだが、ファン層はもちろん一般男性が中心となる。持ち歌は童謡のカバーを中心としているらしく、となりのトトロでお馴染みの楽曲「さんぽ」などが代表曲だ。だが「歩くの大好きー」に対する合いの手として、大の大人が口を揃えて「俺もー！」とがつつく光景を目にしてみると、数多の厳しいヲタクを目の当たりにした私でも陰しい表情を隠し切れない。今は分かり合えないが、きっとその界限にドハマリさえしてしまえば楽しいのだろう。

あいちゃん6さいの再来らしきアイドルの登場に私の意識は一瞬跳んでいたが、どうやら呆然としていたのは客層だけではなかった。ステージに登壇したものの、リトルモンスターの二人は初舞台の緊張からか動作が完全に停止してしまった。このままでは完全に放送事故だが、我々にはどうすることもできない。さて、どうしたものだろうか。

その時、私の脇をスッと通り抜ける影があった。やや背が低めで凄みのある妙齢の女性だ。彼女はスマホを片手に最前へと接近し、ガンガン写真を取り始めた。おっ、これはがつつきヲタかな？ と一瞬訝しんだが、何やら様子が異なる。彼女は舞台上のリトルモンスターに向かって強い眼力を送りながら手招きをし、「前に出てこい」というジェスチャーを交わしたのだ。



そしてこの僅かなやり取りから確信が得られた。そうだ間違いない、この女性はママだ。舞台上の二人もママを目視で確認し安心できたようで、一歩前に出てから自己紹介を始めた。何でもこの二人は姉妹で、姉が8歳で妹が7歳らしい。

それにしてもこの二人、どういった経緯でアイドルを始めたのだろうか。自分から希望したのか、それとも母親が自分からプロデューサーに売り込んだのだろうか。これってひょっとして闇の深いとされるジュニアアイドル……いやいや、きっとアイカツか何かを見ていて「私もアイドルになりたい！」と思った娘の願いを叶えてあげた優しいお母さんなのだろう。うん、そうに違いない。

図 5: 陰しい

さて、出鼻はくじかれてしまったがアイドルの本質は歌とダンスだ。一体どんなパフォーマンスを見せて

くれるのだろうか。スピーカーから聞き馴染みのある曲のイントロが流れ出した。

姉「アーアイ」

妹「アーアイ」

姉「アーアイ」

妹「アーアイ」

二人「おさーるさーんだよー」

……正直厳しかった。

別に舞台上で小学生二人組がアイアイを歌唱するという状況自体に異論を挟む気は毛頭ないが、それを拝聴する男子大学院生という構図に己の限界を感じてしまったのだ。違うんだ、私は単に面白ヲタくを見たかっただけで、お遊戯会を見るために 3000 円費やした訳じゃ……この時点でアイドルを応援するヲタくとしての私は死んでも同然であった。私は無気力にただ手拍子を打ち続けた。

彼女らは 2 曲目に「secret base〜君がくれたもの〜」を歌い終えてから、足早に退場していった。

## メトロポリス

その後も愛知県出身の中学生 3 人組のユニット「ハッピーパブリカ」や、京都から来た中学 2 年生のアイドル「松井小雛」などが出演した。ひょっとしてこの劇場、小中学生のアイドルしか出演しないのではないだろうかという疑念は、次に現れた小学生 1 人・中学生 2 人の 3 人組によってついに確信へと変貌した。

だがこのメトロポリスというアイドル、前座とは打って変わって歌とダンスがべらぼうに上手い。正直今回の出演者の中で一番アイドルをしていたし、格の違いというものを目にしてしまった。

それもそのはずだった。まずメトロポリスの公式 HP <sup>\*13</sup> をご覧頂きたい。この「rocks」というドメインは初めて聞いたが、何でもロックな gTLD として音楽業界、アイドル業界、さらには岩業界で人気沸騰中のドメインだそう。そして近未来というコンセプトに沿いつつ、日本語・英語切り替え可能な web ページは海外展開も視野に入れている。またオリジナル楽曲の 1 つ「テスラにおねがい!」は巷では有名(らしい)ボカロ P かめりあ/Camellia 氏が作曲を担当しているということで、作曲方面にも力が入っている。他にもライブ動画が見られる YouTube 公式チャンネルや、イベントスケジュール広報用の Ameba ブログ、各人の Twitter アカウント等、イベント情報が得やすい形となっているのはファンにとってもありがたい。素人目線ながらこういった痒いところに手が届くビジネスモデルには感心させられてしまう。

ちなみに海外展開は実際に成功しているようだ。直接リンクは貼らないが、頭頂部薄めの厳しいメリケンドルヲタが、メトロポリス所属アイドルの等身大立て看板やチェキをコレクションしている……という大変険しい様子が、なんとメトロポリス公式の配信動画内でビデオレターとして紹介されている。お国は違えども「推し」という精神的感情は万国共通であることが分かる、貴重な映像資料だ。

---

<sup>\*13</sup> <http://metropolis.rocks>

.....

さて、当日の出来事を思いながら書いている本人もそろそろ疲れてきたので本命のドッツの話をしてしよう。無論 ranha 氏に引きつられた私とびしょ〜じょ両名は最前キープヲタクと化していた。周囲を見渡したが、明確なドッツガチ勢はドッツイラスト T シャツを着た ranha 氏と、ドッツイラストパーカーを身にまとったヲタク\*14 程度しか見当たらず、他はファインダー越しのアイドル撮影を目的としたカメコが大半だ。大変厳しい局面を強いられはいるが、その程度で怯む ranha 氏ではない。彼は後方で地蔵を決め込んだヲタクを最前に誘導するなどして、積極的にドッツの世界へと引き込んでいったのである。こうしてドッツのパフォーマンスが始まった……。

1 曲目をそつなく終えた時点で応援していたのは、実質 4 人だった。ranha さん、ドッツパーカーヲタク、びしょ〜じょ、そして私だ。これは手厳しい、コールすらもアウェー中のアウェーなのか。頼むドッツの面々よ、ここでぶち上がる曲を一発流し込んで、会場のボルテージを最高潮まで盛り上げてくれ……。

しかし続けて始まったドッツのパフォーマンスは我々の予想の斜め上を行くものであった。



図 6: 文庫本を手にするドッツ



図 7: 文庫本を音読し、読み終えたページを破り捨て始めるドッツ



図 8: 突如奇声を上げ地面に突っ伏すドッツ



図 9: ステージ上を蚯蚓のように蠕動するドッツ

騒音楽曲\*15 が喧しく鳴り響く中、横たわったドッツの面々はずいに動かなくなった。あ、ここまで前振りでここから歌い始めるんですかね……などと悠長に考えていたが、会場に静寂が訪れると彼女らはすっと立

\*14 どちらも公式グッズではない点にご留意して頂きたい。双方共に自作である。

\*15 「にせんねんもんだい」というガールズノイズバンドのカバーだったらしい。この曲もシューゲイザーの一種と書いたところ、これはノイズバンドというツッコミを ranha 氏から受けたので私はもれなく頭を抱えた。

ち上がった。以上が2曲目だったらしい。

後ほど聞くとところによると、彼女らが得意とする楽曲ジャンルをシューゲイザーと言うそうだ。そもそもドッツが世界初のシューゲイザーアイドルとして売り出されている事実すら知りえなかった私にとって、彼女らの未知の表現技法は、まるでインディアンにとってのコロンブスであり、エリオットにとってのE.T.であり、山本寛にとっての岡田斗司夫<sup>\*16</sup>だったのである。何が言いたいかという、それほど衝撃的だったということだ。

それにしても、彼女らはこの曲で何を表現していたのだろうか。生憎にも私は音楽・舞台芸術を評価する美的センスを有していなかったため、彼女らの行為を深く分析することはできなかった。文庫本を破り捨てる行為は、文字に支配された近代文明からの脱却を表現しているのだろうか。ステージ上に這いつくばって蠕動する行為には、人間が人間であることへの批判を込めているのだろうか。今村昌平の短編映画「おとなしい日本人 (2002) <sup>\*17</sup>」のように、彼女らは人間であることを辞めたがっているのだろうか……？

そういった諸々の意味について第一線ヲタクである ranha 氏にお伺いを立てたところ、一言「知らん」と返された。ドッツ応援番長たる ranha 氏でも理解に至れない表現を、私のようなニワカが解釈に到れるはずもないので、意味についてはこれ以上考えないことにする。ちなみに ranha 氏はシューゲイザーの本質を理解すべく、現在シューゲイザー楽曲を何百曲と聴き漁っているそうだ。

こうして3曲目にはドッツオリジナル楽曲「ねえ」が始まった。先程の残虐性とは打って変わって、ポップでキャッチーな印象を受ける。まるで河原で殴り合ったヤンキー達が直後和解した心情をそのまま歌にしたようだ。ドッツの面々が仲睦まじく肩を組みながら歌唱する場面では、我々も隣のヲタクと肩を組み合った。



図 10: ドッツのパフォーマンスにヲタクもニコリ

さて一通り曲を聞き終えた後の感想をここで述べたいところだが、手前の稚拙な表現能力ではドッツの魅力を十分の一も伝えられないので、上記に挙げた写真と事実のみから読者の皆様には何があったのか類推して頂きたい。それでもドッツについてもっと知りたいというあなたは、毎週のように開催されているドッツの現場に足を運んで頂きたいと思う。

<sup>\*16</sup> オタキングがオタク・イズ・デッドを宣言してむせび泣いたら、感銘を受けたヤマカンが号泣しながらアニメ・イズ・デッドと宣言らしい。詳細な経緯は調べよう。

<sup>\*17</sup> 戦後故郷に帰ってきた軍人・勇吉は戦地のあまりの悲惨さから自身が人間であることに絶望し、蛇として生きることを決める。手足を使わず身をくねらせて移動し、時折ペロペロと舌を出す勇吉に周囲の人間は困惑するが……という話。



## pixie さち

ドッツのステージが終わり、次に現れたのは女子小学生 1 人だった。何でも「pixie」というの二人組アイドルグループのうち、さちくんの方だけ今日は来たらしい。ああ、また小学生アイドルか……と私はもはや食傷気味だったが、会場のヲタク達の反応は違った。後列でドッツをぼんやり眺めていたヲタク達は、一斉に前へと詰め寄ったのだ。

まずステージ上手にいたヲタクは、突然上着を脱いで pixie ちせサイン入り T シャツを露わにし、ピンクの名前入りヲタク棒を振りかざした。一方ステージ下手にいたヲタクは終始無言のカメコだったものの、白 T シャツに刻まれた「ゆうな」と「さち」のサインは紛うこと無くその筋であることが窺えた。開演当初から常に最前をキープしていたこの 2 人の目的はここにあり、彼らにとってのライブはここが始まりだったのだ。

結論から言うと、さちくんのステージパフォーマンスは非常に高かった。先述のあいちゃんやらリトルなんちゃらなどの低年齢童謡歌唱アイドルとは違って、しっかりとオリジナル楽曲を用意しており、歌いながらダンスをしても息切れないという高いポテンシャルを我々に見せつけた。また沼津出身のアイドルということで、最近ラブライブ！ サンシャイン！！ とコラボした事でお馴染みの沼津ローカルパン「のつぽパン」をファンにプレゼントするなどの、サービス精神も十分すぎるほどだった。



図 11: おじさまから絶大な人気を誇る魔性のアイドル

それにしても pixie さちを推すヲタクは誰も彼も大変に陰しく、そして執拗に力が籠りすぎていた。pixie ガチ勢の多くは 30 代後半が多く見られたが、恐らく既婚者であれば彼女らほどの歳の娘が既に存在している頃合いだろう。これは勝手な想像だが、もし同年代の娘が存在していれば自身がドルヲタに堕ちていることも多分無いだろう。だとすれば独身の彼らにとってのアイドルとは自分の娘のようなものであり、現場とは有料の学芸会なのだろうか。そして現場に通い続けることで、ヲタクは彼女らの成長を我が娘のように見守り続けているのだろうか……。

こうした構図は外野から見れば異質な、狂気の沙汰に見えるのかもしれない。だがアイドルは脂ぎった世代のヲタクを確保し、ヲタクは全力で推しを応援するという、ローティーンアイドル業界の図式は双方ウィンウィンの関係で、事実このようにして成り立っている。それに対して外部の人間がとやかく言う必要もないだろう。

彼女は何故アイドルを始め、一体いつまでアイドルを続けるのだろうか……私は複雑な感情を胸に秘めながら、舞台上を去りゆく pixie さちに手を振った。

## その後

その後も、8歳と5歳の姉妹アイドル「ぼん bom しすたあ〜ず」や、仙台のご当地アイドル「杜の都センセーション」、そしてエイベックスと謎の繋がりがあることから *CRAZY GONNA CRAZY* や *EZ DO DANCE* などの TRF 名曲をガンガンカバーし会場を盛り上げた「原宿物語〜Cloth〜」などが登壇した。やっぱり小室ソングは最高やな。

こうして私の地下アイドル現場初体験は、興奮冷めやらぬ中終わりを迎えたのである。

## 4 Conclusion

今回訪れた地下アイドル現場は若干異質だった気がする。とは言え、いい年をしたおっさんが遥かに年下のアイドルを応援する姿は色々と胸にグッと来るものがあったし、アイドル側も数々のおっさんを骨抜きにすべく日夜鍛錬を欠かさない様子が垣間見れたのは関心の一言に尽きる。3000円という私財を投じた価値は十分あっただろう。そしてそんな修羅場をくぐり抜けて我を通し続ける博士課程の ranha さんは凄い、改めてそう思えるのであった。

それにしても今回はドルヲタの深淵を覗きすぎるのは危険だということが、改めて浮き彫りになる良い教訓だった。アイドルライブを楽しむのはあくまで趣味の範囲に留めるのが一番。何事も DD（誰でも大好き）の中庸が一番良いに決まっている。私はのめり込まず、あくまでも観測者の立場を貫くのが性に合っているのだ。そう思いながら私は池袋を後にし、つくばへの帰路についたのであった……。

(追記)

あれから1ヶ月後。気がつくと、私はまたしても地下現場に居た。手にはヲタク棒が握られ、周囲ではヲタクがコールを打ち、ステージ上ではアイドルが踊っている。どういうことだ、私はついさっきドルヲタに堕ちないことを心に決めたはずじゃ……？ 私が私自身では無いような奇妙な感覚に支配されている。どうやら現在出演中のアイドルグループは名を**マボロシ可憐 gene** と言い、私はその中でも**夏野香波** (通称かなみん) さんを猛プッシュしているようだ。まるで訳が分からない。

「マボロシ可憐 gene、物販待機列はこちらになりまーす」

ライブ終了後、私の意思に反して肉体は物販列の方へぞろぞろと移動し始める。現金を取り出そうとした財布には、過去に撮影された夏野香波さんとのチェキが既に何枚か挟まれていた。いつの間にこんなものを？

しばらくすると自分の順番が回ってくる。目の前に現れたのは無論夏野さんだ。

「へ〜【みみずのひもの】さんって、'93年生まれなんですか。同い年ですねー！」

「いやはや、かなみんは同い年に見えないほどお若いすなあ〜」

ツーショットチェキを撮影し、意気揚々と会話し始めた【みみずのひもの】は、既に照れ隠しのあまりヲタク語調になってしまっている。彼奴は恐ろしいまでに滑稽な姿を彼女に晒していた。

「そういえば、かなみんは日本酒を嗜むとお聞きしましたが？」

「そーなの！ 私日本酒には目がなくて！」

「なんと奇遇な。よろしければ今度差し入れでお持ち致しますぞ」

「えーっ！ やだー！ 本当？ 嬉しいー！」

おい待て、何を勝手に約束しているんだ【みみずのひもの】よ、これでは完全にドルヲタ丸出しじゃないか、ちょっと待ってくれ、おい……。

自室で目を覚ました私は、嫌な冷や汗で全身を湿らせていた。どうやらこの記事を仕上げている最中に寝落ちてしまったのだろう。いやはや恐ろしい、あれはドルヲタに堕ちてしまった後の悪夢だろうか。

まずは落ち着こう。私は冷たい飲み物でカラカラになった喉でも潤そうと、冷蔵庫の中身を物色し始める。

冷蔵庫ドア脇のポケットには、綺麗にラッピングされた日本酒が鎮座していた。

## WORD 編集部への誘い

文 編集部 鳥

我々 WORD 編集部は情報科学類の学類誌「WORD」を発行している情報科学類公認の団体です。WORD は「情報科学」から「カレーの作り方」までを手広くカバーする総合的な学類誌です。年に数回発行しており、主に第三エリア 3A、3C 棟や図書館前で配布しています。

編集部の拘束時間には週一回の編集会議と、年に数回の赤入れや製本作業等発行に伴う作業があります。日常的に活動する必要はありませんので、他サークルの掛け持ちの障壁にはなりません。実際、多くの編集部員が他サークルと掛け持ちで在籍しています。

WORD 編集部には、情報科学類生や情報メディア創成学類生などが在籍しています。例えば、以下のよう  
な人達が在籍しています。

**natto** 当たり屋

**azuma962** APL プログラマの末裔

れつくす ホワイトハッカー日本代表

リザウド Lisp 大好きクラブ会員

びしょ〜じょ 百合仙人

下記に当てはまる方や、WORD に興味を持った方は是非、情報科学類学生ラウンジ隣の編集部室（3C212）へいつでも見学に来てください。時間を問わず常に開いています。

- AC 入試で入学した方
- それ以前に AC な方
- 印刷、組版や製版に興味がある方
- ネットワークの管理を経験したことがある方
- APL や Lisp が書ける方

その他質問がある方は、word@coins.tsukuba.ac.jp か、Twitter アカウントをお持ちの方は@word\_tsukuba までお気軽にお問い合わせください。



情報科学類誌

WORD

From College of Information Science

生命、宇宙、そして万物について  
の究極の疑問の答え号

発行者

情報科学類長

編集長

吉本 祐樹

筑波大学情報学群

制作・編集

情報科学類 WORD 編集部  
(第三エリアC棟212号室)

2017年1月20日 初版第1刷発行 (192部)