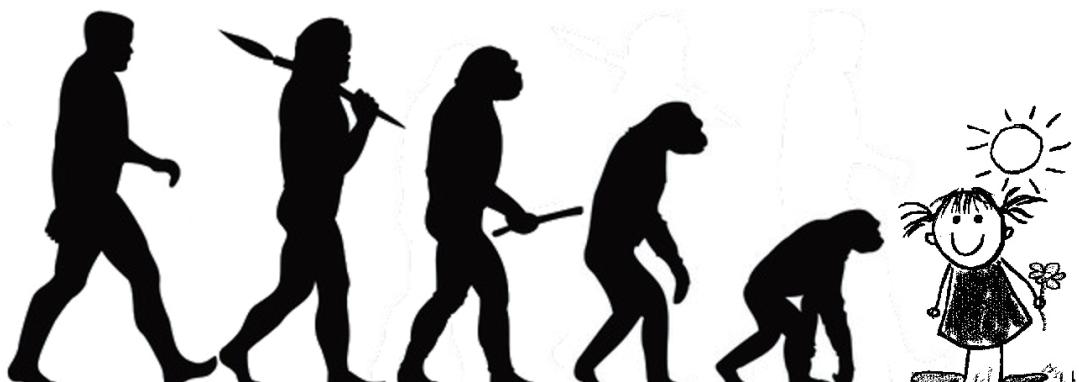


# WORD

From College of Information Science

- クラスファイルを読むために
- Sphinx で妥協する TeX ライフ！ (Install 編)
- 月に代わってお仕置きよ The Rising
- カツジニズムを感じよ
- GeoGuessr 現代～トップランカーを目指す若者たち～
- 地理推定部 入部記
- そうだ 畑 行こう
- WORD 読者アンケート



**36** 学年ビリの WORD 部員が 1 年  
で精神年齢を 20 下げて幼稚園  
に入園した話号

# 目次

クラスファイルを読むために	.....	3
Sphinxで妥協するTeXライフ！（Install編）	.....	9
月に代わってお仕置きよ The Rising	.....	13
カツジニズムを感じよ	.....	26
GeoGuessr現代～トップランカーを目指す若者たち～	.....	31
地理推定部 入部記	.....	41
そうだ 畑 行こう	.....	45
WORD読者アンケート	.....	51
WORD編集部への誘い	.....	60
編集後記	.....	61

# クラスファイルを読むために

文 編集部 azuma962

## 1 はじめに

誰しも一度は、 $\text{\LaTeX}$  というソフトウェアにお世話になった事があるでしょう。

ソースコード 1: 典型的な  $\text{\LaTeX}$  コードの例

```

1 \documentclass{jarticle}
2 \title{title}
3 \author{author}
4 \begin{document}
5 \maketitle
6 hogehoge
7 \end{document}
```

上記のような怪文を書きつらぬ、“platex” やら “dvipdfmx” といった謎の呪文を詠唱すると、イイカンジの PDF ファイルを吐き出してくれて非常に便利ですよね。

$\text{\LaTeX}$  が生成する PDF ファイルは、Word や一太郎といった、一般的なワープロソフトで生成されたものとは違い、独特の瘴気に満ちています。では、あの瘴気の元凶は何なのでしょうか。そう、この記事のタイトルにもある、「クラスファイル」です。

この記事では、そんなクラスファイルを紐解くための前提知識をお伝えします。

## 2 クラスファイルとは

クラスファイルについて述べる前に、 $\text{\LaTeX}$  について軽くおさらいしておきましょう。

コンピューターによる組版の品質の低さに落胆した Donald E. Knuth 教授は組版ソフト  $\text{\TeX}$  を世に送りだしました。 $\text{\LaTeX}$  は、そんな  $\text{\TeX}$  に文書の論理的な構造と視覚的なレイアウトを分離するためのマクロなどを追加したソフトウェアです。

$\text{\LaTeX}$  の強みは、なんといっても  $\text{\TeX}$  講りの組版品質にあります。残念な事に、“\” という、誰も見た事がない記号を多用するせいか、WYSIWYG でないせいか、正確な理由は分かりませんが、アカデミックな場以外では殆ど使われていません\*1。

---

\*1 実際には、様々な出版社で使われています。詳しくは <<http://oku.edu.mie-u.ac.jp/~okumura/texwiki/?TeX>> で作られた本>を参照して下さい。

クラスファイルは、そんな L<sup>A</sup>T<sub>E</sub>X の文書の見栄えを定義するファイルです。名前の莊厳さに対して、その内実は、L<sup>A</sup>T<sub>E</sub>X の命令やマクロ定義の集合体です。言葉の響きに惑わされないようにしましょう。

### 3 クラスファイルを覗いてみよう

クラスファイルに慣れるために、まずはクラスファイルを覗きます。今回覗くクラスファイルは、日本の L<sup>A</sup>T<sub>E</sub>X 使いなら必ず知っている “jarticle” です。では、早速好みのテキストエディタで開いてみましょう<sup>\*2</sup>。

コメントの殆ど無い怪文が出てきたかと思います。我々は、ジュラシックパークの保守を担当するエンジニア達のように、コメントの無いプログラムを解析しなくてはならないのでしょうか。絶望の闇に取り込まれる前に、ファイルの冒頭に書かれたコメントを読んでみましょう。

ソースコード 2: jarticle.cls の冒頭のコメント

```
1 %%  
2 %% This is file 'jarticle.cls',  
3 %% generated with the docstrip utility.  
4 %%  
5 %% The original source files were:  
6 %%  
7 %% jclasses.dtx (with options: 'article,yoko')  
8 %%
```

この記述から、docstrip なるプログラムが jclasses.dtx なるファイルから自動生成したものである事が分かります。docstrip というプログラム名から、jclasses.dtx にはドキュメントが記載されている事が想像できるでしょう。ドキュメントも無しに L<sup>A</sup>T<sub>E</sub>X のコードを覗くのは苦痛以外の何物でもありません。jarticle.cls を覗くのはやめにして、jclasses.dtx<sup>\*3</sup> を覗きましょう。

### 4 dtx ファイルとは

jclasses.dtx を覗く前に、まず dtx ファイルについて知っておく必要があります。

dtx ファイルとは、クラスファイルとドキュメントを一体にしたファイルです。Javadoc や Doxygen<sup>\*4</sup> 似ていますね。dtx ファイルの優秀な所は、一つのファイルから、数多く<sup>\*5</sup> のクラスファイルとクラスファイルに関するドキュメントの二種類のファイルが生成できる所にあります。

---

<sup>\*2</sup> “kpsewhere jarticle.cls” を実行する事で、目的のクラスファイルの場所を見つける事ができます。

<sup>\*3</sup> jclasses.dtx は <http://ascii.asciiimw.jp/pb/ptex/archives/ascii-ptex/platex/1998-09-01/jclasses.dtx> からダウンロードする事ができます。

<sup>\*4</sup> Javadoc、Doxygen 共に、コメントとしてドキュメントが埋め込まれたソースコードからドキュメントを生成するためのソフトです。

<sup>\*5</sup> この記事では扱いませんが、jclasses.dtx から、jarticle.cls だけではなく、jbook.cls なども生成する事ができます。

クラスファイルに関するドキュメントを生成する手順は、通常の L<sup>A</sup>T<sub>E</sub>X のコードと同じです。

ソースコード 3: クラスファイルに関するドキュメントを出力する手順

```
1 $ plateax jclasses.dtx
2 $ plateax jclasses.dtx
3 $ dvipdfmx jclasses.dvi
```

ソースコード 3 にて、コンパイルを 2 回行っていますが、これは 2143 行目と 2503 行目の相互参照を解決するために必要な作業です。クラスファイルを生成する手順は、少しだけ特殊です。

ソースコード 4: jarticle.cls を出力する手順

```
1 $ cat << _EOF_ > jarticle.ins
2 \input docstrip.tex
3 \keepsilent
4 \generateFile{jarticle.cls}{t}{\from{jclasses.dtx}{article}}
5 \endbatchfile
6 _EOF_
7 $ plateax jarticle.ins
```

複雑そうに見えますが、行っている事は単なる L<sup>A</sup>T<sub>E</sub>X のコマンドのバッチ処理です。ここで、jarticle.ins が行う事について簡単に解説しておきます。

1 行目では、docstrip.tex を読み込んでいます。これが、jarticle.cls で見た “docstrip” とやらの正体です。

2 行目のコマンドは、クラスファイル生成時に進捗状況を表示しないための「おまじない」です。

3 行目で、jclasses.dtx から jarticle.cls を作成します。\\generateFile がポイントです。

ソースコード 5: generateFile の引数について

```
1 \generateFile
2 {出力ファイル名}
3 {出力先にファイルが既に存在した時に強制的に上書きする(f)か確認する(t)
   )か}
4 {\from{入力ファイル}{オプション}}
```

4 行目は、バッチファイルの終端を示しています。

ソースコード 4 では、バッチファイル名として jarticle.ins を採用しています。ファイル名は何でも構いませんが、拡張子は “ins” にする事が推奨されています。

## 5 dtx ファイルを覗いてみよう

dtx ファイルの使い方について理解した所で、jclasses.dtx を覗いてみましょう。と言っても、このファイルは 4000 行にも及ぶ非常に巨大なものです。そのため、この記事では、dtx ファイルと格闘する上で最低限必要な点を挿い摘んでお伝えします。

### 5.1 一般的なマクロ定義の例

ソースコード 6: \parskip 及び \parindent の定義

```
729 % \begin{macro}{\parskip}
730 % \begin{macro}{\parindent}
731 % /\parskip/ は段落間に挿入される、縦方向の追加スペースです。
732 % /\parindent/ は段落の先頭の字下げ幅です。
733 %   \begin{macrocode}
734 \setlength{\parskip}{0\p@ + \p@}
735 \setlength{\parindent}{1\Cwd}
736 %   \end{macrocode}
737 % \end{macro}
738 % \end{macro}
```

\begin{macro}{\hoge} は、\hoge マクロの定義を説明する項である事を示します。上記コード片は、\parskip マクロ及び、\parindent マクロの説明です。コメント行%で始まっている行のみが、クラスファイルの内容として出力されます。

\begin{macrocode} と \end{macrocode} だけ、インデントがなされています。これは、「\begin{macrocode} と \end{macrocode} の前に 4 つの半角スペースを入れなくてはならない」という制限に従った結果です。半角スペースを挿入しなくても、クラスファイルの生成は問題無く行われますが、ドキュメントの生成及び、後述する、CheckSum の計算に失敗します。

### 5.2 コメント

ソースコード 7: dtx ファイル内でもコメントとして扱われる記述の例

```
1 % \iffalse meta-comment
2 %% File: jclasses.dtx
3 %
4 % Copyright 1995, 1996, 1997, 1998 ASCII Corporation.
5 %
6 % This file is part of the pTeX2e system.
```

```

7 % -----
8 %
9 % \fi

```

dtx ファイルからドキュメントを生成する際、コメント行<sup>\*6</sup>に記述された内容も出力するために、dtx ファイル内の “%” をほぼ全て<sup>\*7</sup>除去してから処理を行います。

そのため、dtx ファイルにコメントを挿入する際には少々グロッキーな手段を取る必要があります。その手段の一つ<sup>\*8</sup>がこれ、\iffalse です。LaTeX は\iffalse で囲まれた部分を処理しないため、結果的にコメントとなります。

dtx ファイルでは、ドキュメントを記述している場所に限り、コメントを表す記号として、“^^A”を使用することができます。ですが、この記述方法は殆ど使われていません。一応記述例を示しておきますが、クラスファイルを読むにあたって、この記述に遭遇する事は無いでしょう。

ソースコード 8: dtx ファイル内でのみコメントとして扱われる記述の例

```

1 % \section{めう}
2 % ^^A ここはコメントとして扱われます
3 % めうはめうめうめう ^^A ここもコメントとして扱われます
4 %   \begin{macrocode}
5 %^^A ここはコメントとして扱われません
6 %   \end{macrocode}

```

### 5.3 エラー検出機能

dtx ファイルからドキュメントを生成する機構には、簡単なエラー検出機能が搭載されています。ファイル内の “\” の数を数えるだけという、無いよりはマシという代物です。

ソースコード 9: jclasses.dtx の 12 行目から引用

```

1 % \CheckSum{4399}
2 %% \CharacterTable
3 %% {Upper-case    \A\B\C\D\E\F\G\H\I\J\K\L\M\N\O\P\Q\R\S\T\U\V\W\X\Y\
4 %%           Z
5 %% Lower-case    \a\b\c\d\e\f\g\h\i\j\k\l\m\n\o\p\q\r\s\t\u\v\w\x\y\
6 %%           z

```

<sup>\*6</sup> ここでは、“%” で始まる行を指します。

<sup>\*7</sup> \begin{macrocode} と \end{macrocode} で囲まれている部分の “%” だけは除去されません。

5	<i>Digits</i>	\0\1\2\3\4\5\6\7\8\9				
6	<i>Exclamation</i>	\!	<i>Double quote</i>	\"	<i>Hash (number)</i>	\#
7	<i>Dollar</i>	\\$	<i>Percent</i>	\%	<i>Ampersand</i>	\&
8	<i>Acute accent</i>	\'	<i>Left paren</i>	\(	<i>Right paren</i>	\)
9	<i>Asterisk</i>	\*	<i>Plus</i>	\+	<i>Comma</i>	\,
10	<i>Minus</i>	\-	<i>Point</i>	\.	<i>Solidus</i>	\/
11	<i>Colon</i>	\:	<i>Semicolon</i>	\;	<i>Less than</i>	\<
12	<i>Equals</i>	\=	<i>Greater than</i>	\>	<i>Question mark</i>	\?
13	<i>Commercial at</i>	\@	<i>Left bracket</i>	\[	<i>Backslash</i>	\`
14	<i>Right bracket</i>	\]	<i>Circumflex</i>	\^	<i>Underscore</i>	\_
15	<i>Grave accent</i>	\`	<i>Left brace</i>	\{	<i>Vertical bar</i>	\
16	<i>Right brace</i>	\}	<i>Tilde</i>	\~{}		

jclasses.dtx の冒頭に記述されている\CheckSum{4399}は、そんなエラー検出機構に、正常な“\”の数を伝えるためのものです。

\CheckSum の下に、\CharacterTable という、アルファベット一覧らしき何かがありますね。これは、dtx ファイルをダウンロードする際に、\CheckSum で指定された値が化けてないかを判別するためのものです。このテーブルが壊れているかどうかを確認する事で、\CheckSum で指定された値が化けているか否かを判別する事ができるらしいのですが……歴史が垣間見えますね。

## 6 おわりに

この記事では、クラスファイル、もとい dtx ファイルを解読するための前提知識をお伝えしました。おわりに、dtx ファイルの解読の強い味方となる文献を紹介します。困った時は、彼らに頼ってみて下さい。きっと、手助けをしてくれます。

- 奥村晴彦・黒木裕介(2013)「[改訂第6版] L<sup>A</sup>T<sub>E</sub>X2e 美文書作成入門」技術評論社
- Some TeX Developments | Coding in the TeX world  
<http://www.texdev.net/>
- DocStrip Documentation  
<http://texdoc.net/texmf-dist/doc/latex/base/docstrip.pdf>

皆様も是非、dtx ファイルと格闘してみてください。格闘する中で、L<sup>A</sup>T<sub>E</sub>X との正しい付き合い方を見つける事ができるはずです。

# Sphinx で妥協する TeX ライフ！ (Install 編)

文 編集部 hidaruma

## 1 はじめに

某 SoftwareDesign でも Sphinx の連載記事が載っていました。こちらより丁寧な解説なので、バックナンバーを買ったりすれば今からでも間に合います。まあ日本語公式サイト<sup>\*1</sup>と情報量ほぼ一緒だった気もしなくもないですが。

はい、そんなわけでこちらは SphinxJP のサイトから多くの情報を落としノイズを加筆した、マニアックな貴方のための Sphinx 入門記事第2回です。Sphinx 入門では HTML への変換から入るのが常道ですが、本記事では TeX<sup>\*2</sup>への変換を主に扱います。希望があれば HTML などに関して言及するのも客かではなくなくもなくってよ！

例によってこの記事は Sphinx を使って書いています。

## 2 前回までの TeX ライフ！

情報系に使い勝手が良さそうな（偏見）な文法中心に reST 記法を勉強しました<sup>\*3</sup>。あと先号に「Django のドキュメントも Sphinx」と書きましたが、記事執筆時点<sup>\*4</sup>でなんか公式サイトのデザイン変わってました。まあ Python 系プロジェクトのドキュメントの部分が Sphinx を利用していることが多いのは変わりないようです。Atom エディタでの rst プレビューですが、以前紹介した package が更新されないなーとか思ってたら新しいのが生えてました。「rst-preview-pandoc」。これで幸せになります。

## 3 Requirement

Python は 3.x 系を前提として書いていきます。2.x 系でも動きますが pip3 コマンドなどは適宜読み替えてください。

---

<sup>\*1</sup> <http://sphinx-users.jp/>

<sup>\*2</sup> 本記事内では主に  $\text{\LaTeX}2\epsilon$  を指します。

<sup>\*3</sup> 詳しくは先号「WORD35 号」より、「Sphinx で妥協する TeX ライフ！ (reST 記法編)」をどうぞ。reST 記法のうち、僕の利用頻度の高いものをベースに紹介しました。今回説明無しに特殊な語彙が出ていた場合、まずは前回記事を参照ください。

<sup>\*4</sup> 2015/05

## 4 Setup TeX & pip

Python のパッケージ管理ツールである pip と, TeX 環境である TeXLive を用意しておきましょう。TeXLive を入れる理由は pdf への変換でお世話になるからです。reST の処理系である docutils には rest2pdf という直接 pdf を吐く方法も用意されていますが、Sphinx の恩恵を受けるならば |LaTeX| を経由するべきです。

### 4.1 Debian 系

Ubuntu とかは Python3 系はデフォルトで入っているはずなので省略。texlive は full で入れるのがオススメです。パッケージの依存関係で死ぬ人間を減らすためには結局全部入れちゃったほうがいいんです。

```
$ sudo apt-get install python3-pip  
$ sudo apt-get install texlive-full
```

### 4.2 RedHat 系

```
$ sudo yum install python3-pip  
$ sudo yum install texlive-full
```

### 4.3 Mac

MacPort で入れる方法が SphinxJP のサイトに載っていますので、ここでは HomeBrew と pip を使う方法を紹介します。

```
$ brew update  
$ brew install ghostscript  
$ brew install imagemagick  
  
$ brew tap phinze/cask  
$ brew install brew-cask  
$ brew cask install mactex  
  
$ brew install python3
```

### 4.4 Windows

TeXLive を入れておきましょう。GUI のインストーラは [Installing TeXLive over the Internet](#) から `install-tl-windows.exe` または `install-tl.zip` をどうぞ。

## 5 Install Sphinx

### 5.1 Not Windows

pip 環境と TeX 環境ができたらあとは差がないはず。

```
$ pip3 install sphinx
```

記事執筆時点では 1.3.1 が最新版として入ります。docutils などの依存パッケージもこれで一緒に入ってくれます。これで準備は整いました。

## 5.2 Windows

Sphinx が提供しているスタンドアロンインストーラーがあるようなのでこちらを利用してください<sup>\*5</sup>

## 6 ドキュメントを作成する

`mkdir` などで作業するディレクトリを作成したら、作業ディレクトリに移動後 `sphinx-quickstart` コマンドを叩きましょう。対話形式でドキュメントの概要を設定していきます<sup>\*6</sup>。ここで設定したものは一部のディレクトリ設定などを除けば、生成された `conf.py` を後から編集して変更できます。

```
$ mkdir newdoc
$ cd ./newdoc
~/newdoc # sphinx-quickstart

Welcome to the Sphinx 1.3.1 quickstart utility.

Please enter values for the following settings (just press Enter to
accept a default value, if one is given in brackets).

Enter the root path for documentation.
```

デフォルトのまま進めていくと、Sphinx はドキュメントを変換する際、最初に `index.rst` と設定ファイルの `conf.py` を参照します。ちょっととした文章なら `index.rst` に全文を書いてしまいましょう。

reST で書かれた文書がこのようにできたら、

```
$ make latexpdfja
```

を叩きましょう。`build/latex` 以下に `tex`, `dvi`, `pdf` が出力されます。`make latex` を叩くと `tex` ファイルが生成されます。慣れてくると、これをちょっと手直しして `platex`, `dvipdfmx` と叩くようになります。更に慣れてくると自分で拡張などを書いて `make latexpdfja` で最初から調整された `pdf` を吐かせるようになります。

---

<sup>\*5</sup> <https://www.tug.org/texlive/acquire-netinstall.html>

<sup>\*6</sup> [http://sphinx-users.jp/gettingstarted/install\\_windows\\_standalone.html](http://sphinx-users.jp/gettingstarted/install_windows_standalone.html)

## 7 conf.py をいじる

先ほど生成された pdf のタイトル部分が “hogehoge Documentation!” とかになっていてちょっと困った方もいるかと思います。このあたりの設定は先程書いたように index.rst と一緒に生成された conf.py からいじることができます。今回は Install 編ということをいじり方については次回以降に譲ることにします。

## 8 おまけ

Sphinx では docutils から幾つかの記法がリッチになっています<sup>7</sup>。例えば

```
.. image:: ./image.png
```

とすると、パスにある画像を挿入しますが、対応している画像形式が docutils の画像挿入よりも多種類へ対応、オプションで図番号の挿入などが容易に行えます。

また、よく使うもので言えば code ディレクティブの代わりに code-block ディレクティブを用いることによってハイライトに対応するコードの拡充や枠囲いなどが追加されています。

```
.. code-block:: rst
```

コード

コードのベタ書きではなく読み込みを行う場合は

```
.. literalinclude:: /path/code.py
```

のようにすると良いでしょう。include ディレクティブは docutils にありますが、より細かいオプションが設定できます。

レポート作成においてはこの 3 つあたりがよく使うディレクティブになるでしょう。

---

<sup>7</sup> Sphinx1.3 から、オプションによってこのやり取りを省略できるようになりました。慣れてきたら公式サイトの方でやり方を見てみましょう。

# 月に代わってお仕置きよ

## *The Rising*

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。最近は百合漫画を読んでいます。ちなみに私はプログラミング歴よりも百合愛好歴のほうが長いです。以上です。

今回は MoonScript の作者 leafo 氏が作成した、MoonScript の処理系 `moon`、`moonc` を元に<sup>\*1</sup> MoonScript について紹介していく。

#### 1-1. 用語

この記事での用語

- `.lua` ファイル ..... Lua が記述されているファイル。拡張子が `.lua` なので。
- `.moon` ファイル ..... MoonScript が記述されているファイル。拡張子が `.moon` なので。

### 2. Let's MoonScript

#### 2-1. そもそも Lua とは何か

```

1 -- This is comment line.
2 --[
3   This is BLOCK comment.
4 --]
5
6 local str = "Hello, Lua"
7 print(str) --> Hello, Lua
8
9 ----- fibonacci number
10 function fib(n)
11   return n < 2 and n or fib(n - 1) + fib(n - 2)
12 end

```

<sup>\*1</sup> <https://github.com/leafo/moonscript>

```
13 print(fib(10)) --> 55
14
15
16 ----- pseudo class
17 local Node = {}
18
19 setmetatable(Node,
20 {
21     -- constructor
22     __call = function(self, val, left, right)
23         local n = {
24             val = val,
25             left = left,
26             right = right
27         }
28         return setmetatable(n, {__index = Node})
29     end
30 }
31 )
32
33 -- class method
34 function Node:depth()
35     if self.left and self.right then
36         return 1 + math.max(self.left:depth(), self.right:depth())
37     else
38         local ch = self.left or self.right
39         if ch then
40             return 1 + ch:depth()
41         else
42             return 0
43         end
44     end
45 end
46
47 local l = Node(1, nil, nil)
48 local r = Node(2, nil, nil)
49 local root = Node(0, l, r)
50
51 print(root.val, root.left.val, root.right.val) --> 0, 1, 2
```

手続き型言語のように、あるいはオブジェクト指向言語や関数型言語のような記述が可能なマルチパラダイムの動的型付け言語。`local` プレフィックスなしに変数や関数を宣言するとグローバル変数になったり `and` や `or` の使い方が特徴的。みんな大好き Wireshark のプラグインとかアイマス 2 に一役買っているとかヤマハルーターのルーティングスクリプトとか `LuaTeX`<sup>\*2</sup> とかで活躍を見せるなどなど Wikipedia に書いてありました。

また、処理系の中でも `LuajIT` は強力な JIT コンパイラで、他の Lua 処理系どころか Java で同じ実装をしたプログラムよりやや遅い程度であり、スクリプト言語の中では非常に高速である。

OK 諸君、それではこの説明である程度 `Lua` は理解した前提で話を進めていくぞ。

## 2-2. MoonScript とは

```

1 -- This is comment line.
2 str = "Hello, MoonScript"
3 print str --> Hello, MoonScript
4
5 ----- fibonacci numbers
6 fib = (n) -> n < 2 and n or fib(n - 1) + fib(n - 2)
7
8 print fib 10 --> 55
9
10 ----- MoonScript supports class
11 class Node
12   -- constructor
13   new: (val, left, right) =>
14     @val = val
15     @left = left
16     @right = right
17   -- class method
18   depth: () =>
19     if @left and @right
20       1 + math.max @left\depth!, @right\depth!
21     elseif ch = @left or @right
22       1 + ch\depth!
23     else
24       0
25
26 l = Node 1
27 r = Node 2

```

---

<sup>\*2</sup> `pdfTeX` から派生したもの。フォントの埋め込みが `TeX` よりも遥かに簡単にできる。コンパイル時間はものごと遅い。

```
28 root = Node 0, 1, r
29
30 print(root.val, root.left.val, root.right.val, root\depth!) --> 0 1 2 1
```

MoonScript とは Lua をより簡単に記述できる、いわば Alt Lua のような言語。文体は AltJS である CoffeeScript に似ている。

2015 年 5 月 15 日現在のバージョンは 0.3.1。これから分かるようにまだまだ開発中の未完成な言語である。MoonScript 自体が Lua 5.2 で書かれており、`require`<sup>\*3</sup> などで Lua 5.2 で動作するモジュールを使うことができる。もちろん moon ファイルも読み込める。

実は、`moonc`で上記のプログラムを Lua にコンパイルするともう少し大きくなる。

#### compiled lua file

```
1 local str = "Hello, MoonScript"
2 print(str)
3 local fib
4 fib = function(n)
5     return n < 2 and n or fib(n - 1) + fib(n - 2)
6 end
7 print(fib(10))
8 local Node
9 do
10    local _base_0 = {
11        depth = function(self)
12            if self.left and self.right then
13                return 1 + (math.max(self.left:depth(), self.right:depth()))
14            else
15                do
16                    local ch = (self.left or self.right)
17                    if ch then
18                        return 1 + ch:depth()
19                    else
20                        return 0
21                    end
22                end
23            end
24        end
25    end
26
```

<sup>\*3</sup> Lua の組み込み関数。簡単に説明すると、“渡した文字列 + .lua または .so”を LUA\_PATH という環境変数に示されたパスからカレントディレクトリから探して、該当する lua ファイルや so ファイルを実行するというもの。`require`される lua ファイルはグローバル変数を追加するものや値を返す関数というものが常である。後者ならば `var = require('file')` とすることで、`/PATH/T0/file.lua` が返す値を `var` に格納できる。

```

24     end
25   }
26   _base_0.__index = _base_0
27   local _class_0 = setmetatable({
28     __init = function(self, val, left, right)
29       self.val = val
30       self.left = left
31       self.right = right
32     end,
33     __base = _base_0,
34     __name = "Node"
35   }, {
36     __index = _base_0,
37     __call = function(cls, ...)
38       local _self_0 = setmetatable({}, _base_0)
39       cls.__init(_self_0, ...)
40       return _self_0
41     end
42   })
43   _base_0.__class = _class_0
44   Node = _class_0
45 end
46 local l = Node(1)
47 local r = Node(2)
48 local root = Node(0, l, r)
49 return print(root.val, root.left.val, root.right.val, root:depth())

```

処理系も結局のところ、上記のような lua ファイルに一旦コンパイルしてから、luaでそれを実行しているに過ぎないので、コンパイル時間分実行開始が遅い。

詳しい話は公式サイト<sup>\*4</sup> でチェック。公式サイトにはオンラインコンパイラもあるぞい。

### 2-3. イントロ

お手元のマシンで Lua と LuaRocks をご用意願います<sup>\*5</sup>。

```

$ su -c "pacman -S luarocks || apt-get install luarocks || emerge luarocks" || brew install
      luarocks
$ luarocks --local install moonscript

```

<sup>\*4</sup> <http://moonscript.org/>

<sup>\*5</sup> yum には無いらしいので、自力でビルドしよう。

LuaRocks は、Ruby でいうところの RubyGems みたいな、Lua のパッケージ管理システム。ユーザにより公式サイト<sup>\*6</sup>に登録されたパッケージを引っ張ってくれる。

luarocks コマンドで適当なパッケージをインストールすると、ホームディレクトリ以下に.luarocks というディレクトリが作られる。そこにある bin ディレクトリのパスを通して頂きたく。

ここですかさず moon -v とお手元のターミナルに入力して全筋肉を右手につぎ込んで力強く Enter を押すと……。

```
$ moon -v
MoonScript version 0.3.1
```

OK。それでは MoonScript をお楽しみください。

### 3. 機能

公式サイトの About にも書いてあるとおり、CoffeeScript に強く影響を受けている。そのために CoffeeScript と似たところが多い。また、公式サイトの Overview of Differences & Highlights を見ていただければ Lua との簡単な比較が分かる。

#### 3-1. クラスベースオブジェクト指向

Lua ではオブジェクト指向風なことができたが、MoonScript にはクラスがあるので、本格的にオブジェクト指向でプログラムを作ることができる。

Code 1: class.moon

```
1 class Animal
2   life: 100
3   roar: nil
4   cry: =>
5     print @roar
6
7 class Dog extends Animal
8   roar: "bow"
9
10 class Cat extends Animal
11   roar: "meaow"
12
13 class Man extends Animal
```

---

<sup>\*6</sup><https://luarocks.org>

```

14 roar: "Mamaaaahhh"
15
16 with Dog!
17 \cry! --> bow
18
19 with Cat!
20 \cry! --> meow
21
22
23 with Man!
24 \cry! --> Mamaaaahhh

```

### 3-2. Whitespace sensitive blocks

Luaにおいては、ブロック内ならインデントがぐちゃぐちゃだらうと関係なく動作するが、MoonScriptはPythonやCoffeeScriptのようにブロックの範囲がインデントによって決まる。

Code 2: indent.moon

```

1 f = ->
2   print "Hello!"
3   print "in f"
4
5 print "out of f"
6
7 f!

```

```

$ moon indent.moon
out of f
Hello!
in f

```

オフサイドルールなのでワンライナーが書きづらい（小学生並みの感想）。

### 3-3. 記号

記号が増えた。今更説明。

- +=, -=, \*=, /=, ^=

自己代入演算子。

● ->

関数宣言。以下に続くブロックが関数の中身となる。引数は `(x, y, z, ...)->` といった感じで渡す。

● =>

これも関数宣言。`->` の違いは関数の第 1 引数に `self` が暗黙的に付くということ。

● !

引数なしの関数を実行する時 `f()` とする代わりに `f!` と書ける。

● @

`self` の略記。また、`@var` は `self.var` と同じになる。

● \

Lua でいうところの `1:f()` のコロンの代わり。`1\f!` といった具合に使う。第 1 引数に `self` が渡される。

● \*

`table` の前につけてジェネリック `for` に渡すことで、`table` の要素を返すファクトリ<sup>\*7</sup>として働く。`table` に続けて `[m, n]` とすることで `m` 番目から `n` 番目までと範囲を指定できる。また、`[m, n, o]` とすることで `m` 番目から `n` 番目まで、`o` 個飛ばしといった具合に値が返される。

Code 3: ex\_factory.moon

```
1 for i in *{1, 1, 4, 5, 1, 4}
2   print i
3
4   -- 1
5   -- 1
6   -- 4
7   -- 5
8   -- 1
9   -- 4
10
11 tbl = [i for i = 1, 10]
12
13 for i in *tbl[0,6, 2]
14   print i
15
16   -- nil
17   -- 2
18   -- 4
19   -- 6
```

---

\*7 いわゆる“イテレータ関数”。Lua 界隈では“ファクトリ”と呼ぶのが一般的。

### 3-4. リスト内包表記

これまた CoffeeScript や Python のようなリスト内包表記ができる。

Code 4: list\_comprehension.moon

```

1 t1 = [i for i = 1, 10]           --> generate {1, 2, 3, 4, 5, 6, 7, 8, 9, 10}
2 t2 = [i * 2 for i = 1, 10]       --> generate {2, 4, 6, 8, 10, 12, 14, 16, 18, 20}
3 t3 = [i for i = 1, 10 when i % 2 < 1] --> generate {2, 4, 6, 8, 10}

```

table に入る要素のフィルタリングは `if` ではなく `when`。

### 3-5. 追加、削除された予約語

#### 追加された予約語

- `continue`

なんと Lua にはループ文における `continue` がなく、`if` などでうまく逃げるしかなかった。結局 Lua にコンパイルされるのだが、書く側としては嬉しい。

#### 削除された予約語

- `repeat, until`

`until` では内部ブロックで宣言した変数も評価の材料として扱えるので消さないで欲しかった。

- `goto`

実は Lua5.2 から `goto` 文が導入されているが、MoonScript にはご縁がなかったようだ。

## 4. v0.3.1 に残る問題点

メジャーバージョンが 1 にも満たない赤子なのであまり強くは言わないが、ガバガバである。

### 4-1. 処理系が命名する変数

2-2. あたりで moon ファイルからコンパイルされた lua ファイルをあげたが、MoonScript 処理系は lua ファイルの生成時、変数に規則的な命名をしている。さて、MoonScript 処理系が名付けるであろう変数名をこちらが先読みして使うとどうなるだろう。switch 文で `_exp_{0,1,2,...}` という変数を用いる。これについて見てみると……。

Code 5: trap\_compiledvar.moon

```

1 x = 3

```

```
2 switch x
3   when _exp_0
4     print x
5   else
6     print 1
```

```
$ moon trap_compiledvar.moon
3
```

普通にアクセスできるやないかーい（完）。もう少し説明すると、switch 文は Lua にコンパイルされると if 文になる。

Code 6: trap\_compiledvar.lua

```
1 local x = 3
2 local _exp_0 = x
3 if _exp_0 == _exp_0 then
4   return print(x)
5 else
6   return print(1)
7 end
```

変数名を付ける時は処理系の気持ちを考えて付けよう。処理系が作る変数の名前は概ね '\_' から始まるので、それ以外なら基本的にコンフリクトしない。

## 4-2. クラス変数

実は問題ではなく仕様なのだが<sup>\*8</sup>、クラス変数の宣言時にも注意が必要な箇所がある。

Code 7: trap\_classvar.moon

```
1 y = 3
2
3 class Cls
4   :x
5   :y
6
7 with Cls!
8   print .x, .y
```

<sup>\*8</sup> 私はこれをバグだと思っていたのだが、<https://github.com/leafo/moonscript> にて Issue を投げたところ、leafo 氏本人から「これは仕様だ」との回答を頂いた。実際ドキュメントを見てみるとなるほど確かに。うーん、しかし……。

さて、メインブロックにて `y` を宣言し、`Cls` クラスにてクラス変数の `y` が、値は無いが宣言されている。これを実行すれば標準出力に `nil` が 2 つ並ぶと誰もが想像するのだが……。

```
$ moonc trap_classvar.moon
nil 3
```

何故なのか。`moonc` にて Lua にコンパイルしてみると……。

Code 8: trap\_classvar.lua

```
1 local y = 3
2 local Cls
3 do
4     local _base_0 = {
5         x = x,
6         y = y
7     }
8     _base_0.__index = _base_0
9     local _class_0 = setmetatable({
10         __init = function() end,
11         __base = _base_0,
12         __name = "Cls"
13     }, {
14         __index = _base_0,
15     .....
```

`trap_classvar.lua` の 4~6 行目に注目すると、どうやらクラス変数は `_base_0` という `table` に格納されるようだ。値無しでクラス変数を宣言すると、`y = y` となる。この右辺の `y` は 1 つ前のブロックの `y`、つまり 1 行目で宣言されている `y = 3` を示している。

ところで、MoonScript にはこんな機能が追加されている。

Code 9: add\_to\_table.moon

```
1 x = 1
2 y = 2
3 z = 3
4
5 for k, v in pairs { :x, :y, :z }
6     print k, v
7
8 -- x    1
9 -- y    2
```

table 同じブロックにある変数を、変数名を key、変数の中身を value として table に追加することができるというのだ。trap\_classvar.lua を見ればわかる通り、class も内部的には table である。つまりそういうことである。この table の機能も使いどころによっては便利であるが、Code 8 のような問題をうっかり起こす可能性が十分にありうる。使うときは細心の注意を払ってほしい。尚、初期化されていないクラス変数を定義したいときは x: nil とするのが良いらしい。MoonScript には protect のような機能がないので、そんなことはしなくともインスタンス生成後に値を追加するのも可能ではある。

これらの問題もバージョンアップで解消されていくか、あるいは MoonScript から Lua にコンパイルせずにそのまま実行されるような別の処理系が出れば解消されるだろう、うん。

## 5. おわりに

ロゴが美少女バトラーのセーラーなムーンを想起させるんですが。それにこんなマイナー言語じゃあ飯は食えないよハッハッハ～～それではアディオスアミーゴス

## 6. 付録

MoonScript を書くにあたり便利なものを紹介する。これで MoonScript ライフを楽しんでいただきい。

- **ZeroBraneStudio** <sup>\*9</sup>

Lua で書かれた Lua のための IDE。MoonScript 用のプラグインがある。

- **moonscript-vim** <sup>\*10</sup>

このプラグインを入れることで Vim 上で MoonScript のシンタックスハイライトとかオートインデントがされる。MoonScript の作者もやはりVimmer じゃないか。

- **mooni** <sup>\*11</sup>

MoonScript の REPL。luarocks コマンドで簡単導入。

- **moor** <sup>\*12</sup>

私が上記の mooni に手を加えたもの。先方よりもう少し使いやすくなっている……はず。pullreq 待ってるぜ。

<sup>\*9</sup> <https://github.com/pkulchenko/ZeroBraneStudio>

<sup>\*10</sup> <https://github.com/leafo/moonscript-vim>

<sup>\*11</sup> <https://rocks.moonscript.org/modules/steved/mooni>

<sup>\*12</sup> <https://github.com/Nymphium/moor>

- **mooncheck** \*<sup>13</sup>

またしても拙作。MoonScript のシンタックスチェック。出来がひどいのでこんなもの信用しないか、あるいは pullreq を送ってくれると嬉しいです☆

- **Vim** \*<sup>14</sup>

ビム。MoonScript も書ける。この記事も書ける。

- **Neovim** \*<sup>15</sup>

ネオビム。

- **百合漫画**

開発途中のバグバグな言語を書くのがつらい時に読むと心がスーッとする。ほぼ必須といつても間違いない。

他にも、公式サイトの Extras の Editor Support の項を見ていただけると色々紹介されている。

---

\*<sup>13</sup> <https://github.com/Nymphia/mooncheck>

\*<sup>14</sup> <http://www.vim.org/>

\*<sup>15</sup> <https://github.com/neovim/neovim>

# カツジニズムを感じよ

文 編集部 Moneto

## 1. はじめに

筆者の「カツジニズム」との出会いは、5年ほど前の冬のことだっただろうか。

筆者が参加した「全日本学生ゲーム開発者連合<sup>\*1</sup>」という学生の同人ゲーム開発者の集いにおいて、真紅の箱に収められた「活蔴 28」という恐ろしげな栄養剤が、とある参加者の机に置かれていた。その参加者の方曰く、「ユ●ケルの低価格帯の製品<sup>\*2</sup>よりも明らかに効く。ゲーム開発には持って来いである」とのことであった。

筆者は、何よりゲーム制作への熱意故、その集いへの参加を重ねたが、参加を重ねる毎に、明らかに周辺の参加者に「活蔴 28」の飲用者が増えているのを感じていた。そしていつしか、私の作業机にもその真紅の箱が、真紅の栄養剤「活蔴 28」が鎮座するようになっていたのである。

ここに筆者は「活蔴主義」、名付けて「カツジニズム」を感じた。本記事は、その「カツジニズム」の中核的存在たる「活蔴 28」について、筆者の独断と偏見に基づいて紹介するものである。

## 2. 「活蔴 28」

「活蔴 28」は、右図のようなパッケージに収められている。本誌は白黒印刷なので、その溢れ出るエネルギーを象徴せんとするばかりの「紅」色を表現することはできないのであるが、それでも一般的な「栄養ドリンク」と呼ばれる医薬部外品、清涼飲料水の類などからは圧倒的にかけ離れた風格を感じができるであろう。

さて、このデザインは健康的な生活を送っていらっしゃる読者の皆様方にはそうそうお目にかかるないものではなかろうか。それも当然、この製品は薬事法に規定されている「第三類医薬品」であるため、薬剤師ないし登録販売者が監督するドラッグストアの類にしか陳列を許されていないのである。



「活蔴 28」とそのパッケージ

また、「マツモ●キヨシにはよく行っているけれど、こんなものは見たことも聞いたこともないぞ」とお

\*1 学生のゲーム開発者のための交流会。公式 Web サイト：<http://d.hatena.ne.jp/zengeren/>

\*2 1本 400 円程度で手に入る類のもの。ユ●ケルの製品群は幅広く、「箱入り」のものでも 400 円～3,000 円と様々な価格帯の製品が存在する。

思いの方もいらっしゃることだろう。それもまた当然、「活蓼 28」は一般的に名前が知られている「ユ●ケル」等の他社製品群とは異なり、つくばセンターの Q't 内店舗を始め全国区で展開されているドラッグストア「コクミンドラッグ」及びその系列のドラッグストア等の限られた店舗<sup>\*3</sup>でしか販売されていないのである。如何に「活蓼 28」という存在が希少なものであるか、その理由の一部だけでもお分かり頂けただろうか。

## 2. 1 成分について

ここで「活蓼 28」の成分について、パッケージの記載を示そう。私には東洋薬学の心得がなく、成分に関して細かく批評を加えることは当然できないのであるが、栄養剤などの医薬品の成分に詳しい「その手の方」には有益な情報となるだろうか。

・コウジンエキス（高麗紅蓼六年根）	600.0mg（原生薬として 1,200mg）
・オウギエキス	35.7mg（原生薬として 100mg）
・ショウキョウエキス	41.0mg（原生薬として 1,000mg）
・ゴミシエキス	105.3mg（原生薬として 200mg）
・クコシエキス	111.1mg（原生薬として 200mg）
・タイソウエキス	357.1mg（原生薬として 750mg）
・ロクジョウチンキ	0.05mL（原生薬として 14.3mg）
・ビタミン B2 リン酸エステル	5.0mg（リボフラビンとして 3.934mg）
・ビタミン B6	10.0mg
・ニコチン酸アミド	25.0mg
・無水カフェイン	50.0mg
(いずれも 1 日量 (1 本) 当たり)	

「活蓼 28」最大の特徴は「1 本売り 800 円未満という中程度の価格帯でありながら、この価格帯ではまず見られない『高麗紅蓼六年根』を使用している」という点である。

実はこの成分、おなじみ「ユ●ケル」シリーズでは最上級品の「ユ●ケルスター」にしか用いられていない成分なのである。「ユ●ケルスター」は 1 本売りで平均 3,000 円程度、どんなに安くなると 2,000 円を下

\*3 「コクミンドラッグ」系列のドラッグストア以外では、東京都区内に展開しているドラッグストア「一本堂」に取り扱いがある。余談だが筆者の通っていた高校の目の前に「一本堂」があったため、実は「活蓼 28」は大変身近な存在のように感じられた。また、最近では一部ドラッグストアのネット販売でも取り扱いがあるようだ。

回る事例は私の見る範囲では観測されていない。原生薬の使用量こそ違うものの、「大変本格的な栄養剤」を求めて一部の読者の皆様方においては実にピッタリの栄養剤となるかも知れない。なお、「活蔴 28」には上位品として「活蔴 28 ROYAL」なる製品があり、こちらは無印の「活蔴 28」と比較すると 1 本 1,300 円超と高価なものであるが、その分含有成分はより豊富になっている。

## 2. 2 味わい

飲み口は、「ユ●ケル」の独特の痺れる辛みを少なくし、甘いともほろ苦いとも言える深みのある味わいを感じられるようなものとなっている。勿論、この種の栄養剤を初めて服用する方は独特の「喉が焼ける」ような感覚を味わうことだろうが、良薬口に苦しとはよく言ったもの。服用して少し経てば、コンビニエンスストア等で簡単に手に入る「栄養ドリンク、エナジードリンク」とは全く違う感覚を味わえるに違いないであろう。

## 2. 3 価格

ところで肝心の価格であるが、先述の「コクミンドラッグ」では 1 本 700 円強、3 本 1,700 円強、10 本 4,600 円強<sup>\*4</sup> といったところだろうか。店員さんに聞けば出てくる「裏メニュー」として 50 本 20,000 円強という「ケース」での販売もあるが、余程の事情がない限りまずケース単位で購入することはないだろう。

なお、筆者は先日、「活蔴 28 を 50 本ケース<sup>\*5</sup> でご購入の方にはご自宅まで無料配達します！」というキャンペーンの文字列に心を動かされかけたものである。



「活蔴 28」10 本入り箱

---

\*4 10 本単位で買うと、1 本 460 円程度となりかなり割安。

\*5 そもそも 50 本ケースは常時の在庫がないらしく、「お持ち帰りの場合も数日前までにご連絡をお願いします」とのことだった。つくば広しといえども、圧倒的カツジニズムによりドラッグストアに 50 本ケースの在庫を置かせるような熱狂的カツジニストは居ない模様である。50 本ケースで買うと 1 本 410 円程度と更に値段が下がるが、「活蔴 28」を 50 本も飲むような生活は想像したくないものである。

### 3. カツジンカツドウ！

さて、この記事をここまで読み進めて下さった奇特な読者の皆様方の中には、「ユ●ケル」と「レッ●ブル」を混合した通称「ユンブル」なる魔の混合液に手を出したことのある方もいらっしゃることだろう。筆者も勿論その魔の混合液を服用し、疾風怒濤の如く睡眠時間と己の健康を犠牲に様々なプログラムを生み出したことのある人間のうちの1人である。

さて、賢明な読者の皆様方であれば本章の展開は非常に容易に「 $1+1=2$ 」を計算するよりも容易に予測頂けることであろう。

——飲んだ者全てに翼を授け、幾人もの窮地を救ってきた真紅の猛牛「レッ●ブル」。そして、王者の風格すら感じる真紅の箱が、真紅のラベルが全てを物語る「活蔴28」。2つの「真紅」が禁断の融合を果たす時、一体何が生まれるのであろうか——

時は平成27年5月3日、午後6時。筆者は、某サークルにおいて「48時間ゲームジャム」という、5月2日の正午から5月4日の正午までの48時間で1作のゲームを仕上げる文字通りの耐久戦に参加していた。〆切まであと18時間、時間も体力も着実に無くなりつつある筆者の手元には勿論「活蔴28」があり、情報科学類ラウンジの自動販売機には狙いましたかのように「レッ●ブル」が鎮座している。これは、「やるしかない」。

というわけで早速「レッ●ブル」を購入し、若干量を飲んだ「レッ●ブル」の缶に「活蔴28」を投入する。折角なので写真を撮っておけばよかったものの、透明なコップの用意等を怠ってしまったのは不覚極まりないことであったが、こうなっては視覚以外の「五感」に訴える表現をするしかない。



禁断の融合

まず、「聴覚」。ただただ筆者のThinkPad T440pのファンの音が響くばかりである。

次に、「触覚」。先刻まで自動販売機に格納されていた「レッ●ブル」をベースにしているため、大変よく冷えている。

そして、「嗅覚」。これも、「活蔴28」の深い芳香をよく感じられるものとなっている。普段の「レッ●ブル」から感じられる甘ったるい芳香は、きっと「活蔴28」の堂々たる風格に押されてしまったのだろう。

更に、「味覚」。「活蔴28」の味が「レッ●ブル」と比較してかなり強烈なためか、「レッ●ブル」全体に「活蔴28」の味がしているような感覚を味わうことが出来た。普段は「レッ●ブル」など一気に飲んでし

もう筆者であっても、こればかりはとても一気に飲めない。

最後に、「第六感」。筆者はおよそ「ニュータイプ」や「強化人間」、「ウィスパード」、「超能力者」等の特殊な属性など持たぬ只の凡人である。だが、これだけは感じる。否、目の前に鎮座する「レッ●ブル」の缶に入った液体の禍々しいオーラが、私に感じさせてくるのだ。

# これアカンやつや。

ちなみに、この液体を服用した筆者の気力は、翌朝5時頃まで持続したことを書き添えておこう<sup>\*6</sup>。

## 4. おわりに

健康的な生活で、健全な生産活動を。日々の少しずつの積み重ねが、あなたの健康を左右します。

この記事に記載されている内容は特に注記されている事実を除き、筆者の独断と偏見によるものです。  
この記事を読んだことに起因する如何なるトラブル、及び心身の健康に発生した如何なる不調にあっても、筆者及びWORD編集部は一切の責任を負いかねます。

---

<sup>\*6</sup> 無事に成果物も完成した。以下のURLよりダウンロード可能。URL：<http://dev.flat-soft.com/Uploads/Clavis.zip>

# GeoGuessr 現代～トップランカーを目指す若者たち～

文 編集部 みみずのひもの

## 1 旅は唐突に



Let's play!

読者の皆様は「電波少年」というTV番組をご存知だろうか？放映していたのはもう10年以上昔になる。ヒッチハイクでユーラシア大陸を横断したり、スワンボートで国境を横断したりと、現在では到底実現できないようなひどい企画ばかりの番組だったが、冒頭の定番として「目隠しをしたまま何処かへ連行される」というお約束があった。国内に限らず、海外に飛ばされることも少なくなかったそうである。一体どうやって目隠ししながら税関をくぐり抜けたのだろうか。

その番組についてはさておき、目隠しを外した瞬間見知らぬ光景が広がっていたら、人間は一体どのような気分になるのだろうか？言語も理解できないような僻地に飛ばされた絶望感、日が暮れるまでに衣食住を確保しなければならない焦燥感……あまりポジティブな感情だとは考えにくいだろう。だがそんな状況に対して、少し憧れを抱いてしまった経験はないだろうか？何も恥ずかしいことではない。無味乾燥な日常に別れを告げたいという非日常への憧れ、そして誰に指図されるでもなく何処へ行くにしても自由な開放感への憧れ、それはすなわち「独り旅」の欲求に相違ない。

本当は誰しもが、心の何処かで独り旅を欲しているはずなのだ。青春18きっぷを駆使して聞いたこともないような無人駅に降り立ち、ゴローちゃん<sup>\*1</sup>のようにふらっと入った定食屋で美味しい飯を食い、絶景の写真を撮ってFacebookに投稿するような旅を羨ましく思ってしまうのだ。

ほぼつくば在住で時折東京に出るか実家に戻る程度という閉塞的環境に置かれた筑波大生の皆様も、その例外から漏れることはないだろう。しかし我々学生にとって現実は厳しい。まず時間もかかるし、費用もか

<sup>\*1</sup> 漫画「孤独のグルメ」に登場するエンゲル係数がどちら高そうな主人公。ドラマ版も既にSeason4と絶好調。

かるし、何より海外では強盗や遭難などといった生命の危機に瀕するリスクが高まるからだ。危険な旅となると当然ご両親からの反対は免れないだろうし、ついうっかり御不幸が……なんてことになると、自身の卒業文集が昼夜のワイドショーに延々垂れ流される辱めを四十九日まで堪能することになるだろう。

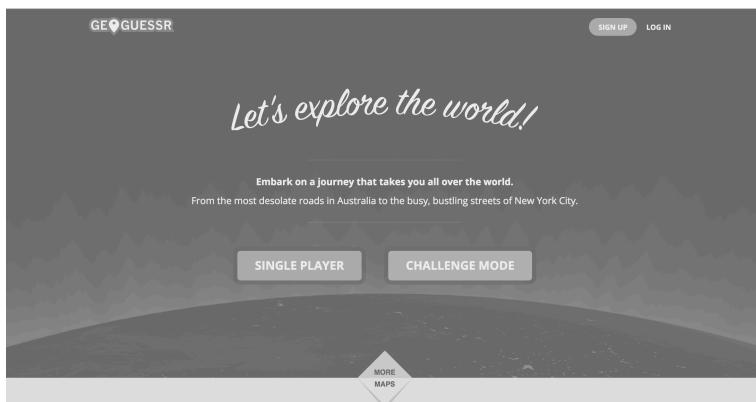
「ああ、オイラも縮地法<sup>\*2</sup>が使えたらしいのになあ(ホホ)」などと思いを馳せる方へぜひお聞きしたい。では旅行にどのくらいの時間なら割くことができるだろうか？1週間？1日？それでも多忙な現代人は首を縊に振らないかもしれない。だがもし5分と言われたらどうだろうか？「てーきゅう<sup>\*3</sup>」や「あいまいみー<sup>\*4</sup>」を視聴するぐらいの時間で、旅行欲求を満たすことができる画期的な方法があると聞いたら、どうだろうか？

よろしい、ならばGeoGuessrだ。

## 2 GeoGuessr とは？

それでは皆様お待ちかね、GeoGuessr（ジオ・ゲッサーと読む）のご紹介に移ろう。GeoGuessrはGoogleストリートビューが用いられた地理探検型のゲームで、2013年9月にリリースされた。その目的は単純明快、「自分が今何処にいるか」を世界地図の中から答えることだ。以下のURLからすぐに遊ぶ事ができる。

<https://geoguessr.com>



お好きなモードから選べるドン！

ゲームの流れは以下の通り。まずゲーム開始直後、プレイヤーは「世界のGoogleストリートビュー上で移動可能な任意の点」に飛ばされる（一部地域は除外）。国土面積の都合上米国や露国に行き着くことが多いが、一応同様に確からしい確率で地球上の何処かに流刑されるようだ。

---

<sup>\*2</sup>中国発祥のテレポーテーション仙術。青獣や北の將軍様や庭球部の中学生が使えたりする。

<sup>\*3</sup>本編2分広告2分で構成されているテニスアニメ（テニスをするとは言っていない）。

<sup>\*4</sup>あびやあ～～～。

それからプレイヤーは Google ストリートビューの操作 UI に従って移動し、地名のヒントになりそうな情報を収集する。万国共通で参考になるヒントとしては「標識」「看板」「地形」などがある。ちなみに最も目につきやすいであろう「車のナンバープレート」には、当然自動的にモザイクがかかっているため、情報としての意味を成さない。このゲームのなかなかよく出来ているところだ。

最後に画面右下の Google マップ上から、スタート地点と思わしき場所へピンを立てて「MAKE GUESS」をクリックする。すると画面が遷移し、答えの地点と自分がピンを立てた地点との距離、及びそれに応じた得点が表示される。得点は 5000 点満点で換算される。例えば答えの地点に対して地球の裏側に位置する国を選択してしまうと 0 点、同じ国内なら 3000 ~ 4000 点、10m 近くまで接近できたら 5000 点という具合だ。これで 1 ラウンド終了である。

GeoGuessr 全体は 5 ラウンドで構成されており、全ラウンド終了後の合計点数がスコアとなる。ちなみに GeoGuessr ではユーザー登録をすることで自分の戦績を記録することも可能だ。自分のスコアの向上を調べたり、自分がプレイしたマップを他人と共有することもできる。

と、ここまでが世間における一般的な GeoGuessr の紹介となる。ストリートビュー中で見かけた文字列を翻訳にかけたり、店の名前を検索にかけたり、時間をたっぷりかけて調べたり、飽きたら保存して次の日から再開したり、のほほんと高得点を目指す競技、これが「軟式 GeoGuessr」である。



なんともどかなプレイ中の画面

では早速軟式 GeoGuessr で遊んでみて、感想はいかがだろうか。確かに、Google ストリートビューで未踏の地をぐるぐる探索しているだけでも、独り旅の気分は十分に味わえるかもしれない。だが少し待って欲しい。本当にそれでよいのだろうか？ 安易にインターネットのチカラを乱用し、焦躁感に責め立てられることもなく終える旅でよいのだろうか？ 否、本当の独り旅とは真に孤独であり、故に自分の知識のみで戦わなければならない。そして旅は実時間を消費するものであって、閉じていたタブを開きなおして再開するもので

はない。表に対する裏、光に対する闇、まんがライフに対するまんがライフ WIN、軟式 GeoGuessr に対する存在として提唱された競技、それこそが「硬式 GeoGuessr」である。

### 3 硬式 GeoGuessr とは？

競技レベルにまで洗練された硬式 GeoGuessr では、以下の様な制約が追加されている。

- 1 ラウンドの制限時間を 5 分（もしくは 7 分）とする。
- No Google（ググり禁止）。他同様に一切のカンニング行為を禁ずる。
- プレイヤーへの一切の口出しを禁ずる。

まず何と言っても時間制限の存在が大きい。時間無制限の場合、最悪国さえ判明すれば総当たり攻撃を仕掛けることで、無理矢理探索することが可能となってしまう。しかし時間制限が設けられたことによって、プレイヤーは探索時間と回答時間を切り分けるという、まるでセンター試験のようなテクニックが要求されるのだ。

ググり禁止もゲーム性を大きく高めることになる。植生、地形、標識の色、言語などといった情報を、自身の知識として取り込む必要性が生じるようになったのだ。すなわち今までどれだけ硬式 GeoGuessr で鍛錬してきたか、修行の積み重ねがプレイヤースキルへと直結するようになったのである。

こうして硬式 GeoGuessr は WORD 編集部内にて産声を挙げ、すくすくと成長していった。特に ioriveur 氏、わさお氏のこれに対する執着ぶりには、筆を取らずにはいられないほどの力強さを感じ取ったほどだ。ある時は看板も見当たらないような砂漠の一本道を永遠に走り続け、またある時は知床半島の山道に降り立ち、またある時はタイの麻畑<sup>\*5</sup>のど真ん中にスポーンしたりと、彼らはそれこそ朝から晩まで世界中を飛び回ったのだ。彼らの目標は硬式 GeoGuessr 甲子園への出場と公言している。

### 4 名言集

ここでは硬式 GeoGuessr 競技中にさらっと飛び出した名言達を紹介する。

- わさお「日本はいいですね、やっぱり凱旋って感じ」

GeoGuessr はその特性上海外出張の機会が多くなる。米国、ブラジル、オーストラリアの僻地へと何度も飛ばされ、だがそれでも高得点を叩き出し、日本へ帰ってきた時の一言がこれである。無論帰国してきた GeoGuessr プレイヤー達を叱咤激励する観客など画面上には存在しない。だがそれでも、看板に表記された日本語や丁寧に舗装された日本特有のアスファルトから、GeoGuessr プレイヤー達は帰国できた喜びを感じ取るのだ。

---

<sup>\*5</sup> 麻は衣服の繊維や油としても使われる万能植物だ！ 決して楽しい気分になるためだけのものじゃないぞ、いいね？

- ioriveur 「この国は高得点出しづらいんですね」

開幕数秒での名言。まず何処の国であるかということまで把握しきった上で、この国のどの地域かまでを当てるのは難しいと述べている。あくまでも GeoGuessr を観光旅行ではなく競技として捉えていることが分かる一言だ。一体彼はこのゲームを何処まで極めたというのだろうか。

- ioriveur 「ThinkPad プレイヤーは腕立てから始めるべきかもしれない」

ほぼ一本道で構成されるステージは、高速に移動して速やかに情報を収集することが得点への鍵となる。そのためクリック連打速度も競技の勝敗を分ける重要なファクターだ。本来 E-sports に区分される GeoGuessr だが、この名言で彼は筋力の必要性も説いている。

- わさお「僕はこれで食べていこうと思うんで」

さらりと飛び出す人生方針。一体誰がスポンサーに付いて、どのような経緯で金銭が発生するのか詳しくお聞きしたい。

- こぼる「東？ 西？」わさお「熊本県です」

とある海外ステージでのやりとり。とある国の中で東と西どちら側なのだろうかというこぼる氏の誘導質問に対し、わさお氏は答えを返さず自身の出身地でピシャリと戒めている。相手に有利な情報を与えず、かつ地元をリスペクトするフェアプレイだ。

- わさお「大会になるとアンチの妨害なども考えられるので、こうやって BGM を流しながら」

既に大会の将来性まで見据えた一言。一体誰のアンチがどのような目的を持ち、どういった手段で妨害してくるのか詳しくお聞きしたい。

- ioriveur 「お腹空いた、飯食わなきゃ、でも読み込み始まっちゃった。あ、オーストラリアだ」

午後 3 時頃に飛び出した名言。寝食を忘れて GeoGuessr に没頭する ioriveur 氏の様子が窺える。彼にとって画像の読み込み開始は GeoGuessr の開幕を意味するので、途中で止めることは許されないらしい。なお開幕即オーストラリアを断定しているところからも、プロフェッショナルとしての本能には逆らえないようだ。

## 5 解説と共に学ぶ GeoGuessr

こうして来る 5 月 8 日の深夜頃、世界初ともなる公開 GeoGuessr 霊言解説がしめやかに執り行われた。お話を伺ったのは今年 GeoGuessr 業界に鮮烈なデビューを飾り、「バラボラアンテナの向きによる国の判別」という新たな風を吹き込んだ、WORD 硬式 GeoGuessr 部部長 ioriveur 氏だ。なおこれは本来「第 1 回硬式 GeoGuessr 世界大会」となる予定だったが、参加者が 1 名しか集わなかつたため急遽こののような形となったことをご了承頂きたい。

果たしてプロプレイヤーはどのような思考過程で MAKE GUESS にまで至るのか。その真意に迫る。

—「では始めましょうか、よろしくお願ひします」

iori (ioriveur 氏)「はい、こちらこそ」

—「まずは開会式ということで、開幕のテーマを何にしましょうか？」

はろべり氏『祖国は我らのために』(ソビエト連邦国歌)で』

iori 「いいね」

—国家齊唱中—

iori 「じゃあ GeoGuessr に絡めた話でも。ソビエト、じゃなくてロシアの看板って道がクソ長くて荒れてる上に詳細が少ないんですよ。大きな街だったらしいんですけど、大体ゴミみたいな町であることが多い。そんなん Google マップでも拡大してもわからない上、国土がめちゃくちゃでかい。ということでロシアが出てくると高得点が狙いづらいんですね」

—「なるほど、プロプレイヤーの名に恥じない、含蓄のある言葉ですね。では早速競技に移りましょうか」

iori 「では始めますか。まず GeoGuessr を WORD に持ち込んだのは僕なんですよ。存在自体は 1 年以上前から知っていたんですが、WORD で流行らないかと思ったら自家中毒的にのめり込んでしまいました」

—「なるほど」

iori 「で、アップデートが行われたのも大きいですね。今までそのゲーム固有の ID が URL として発行されるのはゲーム終了後だったので、多人数で競技を行う際には第三者がゲームメイクする必要があったわけですよ」

—「そうでしたね」

iori 「ところが 4 月の末にたまたまアップデートされまして」

—「そうそう、タイムリーに」

iori 「『SINGLE PLAYER』と『CHALLENGE MODE』が実装されたんですよ。CHALLENGE MODE は開始時に固有の URL が表示されるので、その URL を配布するだけで競技できるようになった。ついでにとある脆弱性も修正……いや、これで皆安心してプレイできるようになったというわけで。そんな感じでやって

いきたいと思います。今回は『WORLD』のマップで、今回制限時間無しですが、だいたい1問ずつ10分から20分ぐらいということで。まぁ運にもよるんですよね。で、画面をローディングしたら……もう画質が悪いんですけど」

—「ああ、今日は厳しい……」

iori「これは森の中ですね。太陽の方向が何となく南東の方向に見えるんですけど、中央分離帯も黄色でよくある感じになっているんで、これも分からないんですけど。まぁ、カナダかアメリカかなと思うんですけど……何しろ画質が悪い。Googleストリートビューって田舎の方だと画質悪いことがあって、特にオーストラリアの方は多いんですけど。これは違いますね」

—「オーストラリアではない？」

iori「ええ、まず道路標識の向きが違いますよね。これは明らかに右側通行の国だし、太陽が南にも登っているんで、南アフリカやオーストラリアではないことが明らかです。道や車の感じからして……あっ、この『2』って書いてある標識なんんですけど、これはアメリカの国道表記のマークなので、アメリカで確定です」

—「おお」

iori「ではもう少し情報を集めてみましょう。ここに『SPEED LIMIT 55』って書いてありますね。カナダと北米どっちかで迷った時には、こういう速度標識を見るといいと思います」

—「というのは？」

iori「カナダの場合には『MAXIMUM』って書いてあって、アメリカだと『SPEED LIMIT』って書いてあるんですよ。今のところ例外は見つけてないんで多分。他にも『MAXIMUM』って看板は独特なので、一緒にフランス語があったりするとカナダのケベック州であることがほぼ確定するとか、まぁそういう感じで速度標識も馬鹿にならないと、おっ」

—「これは？」

iori「『Wilson Lake Inn』とありますね。いわゆるモーテルんですけど、こういうのどこまで信用するかですね。実際に Wilson Lake の近くにあるのか、ひょっとしたら日本で言うカラオケの『コートダジュール』みたいなものかもしれない」

—「なるほど」

この後も ioriveur 氏の推理は着々と進むものの、具体的な地名を当てるには至らず。今回は 2000 点を切ってしまう結果となった。

—「なかなか厳しいですね」

iori「そうですね、情報証拠は揃っていたんですけど。これもまた GeoGuessr です。では次行きましょう。えーっと、まぁ今回は暖かそうなところですね。北から太陽も差してるので」

—「この木はバナナでしょうか？」

iori 「いや、ココナツですかね。おや、これは『STOP AHEAD』……STOP という単語だけ輸入しているところはあるんですけど、STOP AHEAD となると英語圏の国であることは確定しますね。もしかしてまたアメリカですかね、太陽の方角からすると考えづらいですが。マルボロの広告とかも出ている。だいぶ北から差してるんで南半球かな……？」

なかなか苦難する ioriveur 氏。しかしここで強力な証拠が。

iori 「KFC の広告……こういう広告もヒントになることがあってですね。ドルとか、ループルとか」  
—「あー、通貨単位ですか」

iori 「そうそう。おや、Here now on Guam……ってあるんで、これはグアムですね。もうグアムをポチー (MAKE GUESS) で高得点取れますよ。これも SPEED LIMIT って書いてあるんでアメリカなんですよね、でも北から太陽が差していたんで、グアム・サイパン・ハワイかなみたいな、勘が当たった感じですね」

後はグアムもそう広くない島、直ちに場所の目処がついてくる。ここからはいかにして 5000 点満点まで近づけられるかという、プロの仕事だ。

iori 「とりあえずハイウェイに出てどちらへんかとか、あるいは海まで出てどっち海岸かとか。お、ヒントが見つかりましたよ。ここ『Dededo Central Park』って(マップ上に)あるじゃないですか」

—「おお、引き当ててきましたね」

iori 「で、さっき見えた『International Contry Club』ってこれですか。で、ここで一度スタート地点に戻って、スタート地点が『Aga』とかいう小道だったっぽいんですけど、あ、これですね。で、スタート地点から進んでいって 1 個目の交差点なので、もうここで MAKE GUESS します。あ、5000 点取れましたね。距離は 18m」

—「うーん、お見事」

iori 「はい、では次行ってみましょうか。目の前には白樺が広がってますね。今回は北国のようにです。それから……あ、デンマークかスウェーデンみたいですね」

—「ほう！ というのは？」

iori 「この看板ですね。あ、フィンランドかもしれません。A ウムラウト(Ä)があるので、ノルウェーではないけれどスカンジナビア半島



見事 5000 点満点を獲得する ioriveur 氏

の方かな。名前、地名がそんな感じ。あー、デ

ンマークですかね。標識が赤丸に白抜きっていう日本にありがちな標識。これはスカンジナビア半島付近あるいはデンマークだと分かります。で、特に真ん中が黄色くなっているものはデンマークですね」

—「おお、A ウムラウトだけでそこまで断定できるんですか？」

iori 「いや、ノルウェーじゃないってことしか分からないですね。本当はできれば語尾がベルク (Berg) とボルク (Borg) ぐらいの違いが分かるといいんですけども」

—「こういったは知識は経験則ですか？」

iori 「ええ、経験則ですね。国ごとにいくつか特徴量ってあって、それだけ注意してればそんなに覚えることはないんですよ。繰り返しになりますが、ロシアアメリカみたいな大国は高得点を狙いにくいんですよ。でも国や気候が分かっても全然点にならない。その辺はその国のニッチな予備知識が必要になってくると」

その後しばらく情報集めに奔走するものの、決定的な場所特定には至らない。その後 ioriveur 氏は「ノルウェーじゃないかも」という疑念を徐々に浮かべ始める。残念ながら編集者みみずのひもの眼氣にも限界が近づいてきたので、早々に MAKE GUESS。

iori 「あら、スウェーデンでしたね！」

—「うーん、残念」

iori 「あ、でもこっちの方がさっき（第 1 ラウンド）より点数いいですね」

—「なるほど、アメリカだと同じ国内でも点数が低いってことが起こりうる」

iori 「これはちょっと敗因の分析が必要ですよ。標識の黄色が確かにデンマーク以外でも 1 回あってピッタ記憶があるんだけど……（自分自身で）イレギュラーだと思ってたんですかねえ」

—「言語の違いについては？」

iori 「確かに看板見てデンマークじゃないかもって言い始めたのは、ウムラウトとオングストローム (Å のようなリング付き文字) が両方同じ単語にあって、やっぱりちょっとデンマークっぽくないなあと」

点数向上のための努力を欠かさない姿勢、これこそトップランカーを目指す者のたゆまぬ努力だろう。

その後も ioriveur 氏は部長の名に恥じぬ丹念なプレイで残り 2 問を消化し、序盤のミスを挽回する形で 20000 点超えを達成した。競技終了後 ioriveur 氏から GeoGuessr をプレイする上で参考となる指南を享受して頂いたが、紙面を割くにはあまりにも文量が多いので、今話題のパラボラアンテナに関する件の部分のみ抜粋しよう。

iori 「……えーっと、特に太陽ってのは南北に上下するんですけど、人工衛星は常に静止軌道上にあるんで

上下しないんですね。ずっと赤道上にあるんです。ということで BS や CS を受信するパラボラアンテナがどっちを向いているかで国的位置が分かるんですよ」

—「おっと、パラボラアンテナ判別法ですね？」

iori 「はい。例えばアンテナがメッシュ柄になっているやつは南米に多いんですね。そういう型式でも分かると思うんですけど、やっぱり向きで決定的になります。さらに言えば多分アメリカ大陸ってこの辺（エクアドル周辺）の上空（赤道上）に有名な人工衛星が浮かんでるんですよ。例えばコロンビアとかだと屋根の上に仰向けになったパラボラアンテナが置いてあったりするんですけど、ブラジルのような赤道付近のところだと少し西を向くように起こしてあるんですね。で、その角度が今度南側に結構深いと、ブラジルでもさらに南側にあたる可能性が出てくる。この周辺は台地になっていて降水量が少ないし、土も独特の色をしているんで分かりやすいんですよ。例えば（現在地が）ブラジルだということしか分からない時、アマゾンかなとか思って MAKE GUESS して答えがこの台地の周辺だったりすると、同じ国内でも 1000 点とかになっちゃうんですけど。でもこの辺の台地かなって推測できれば、探しやすいですし点数も上がってくる。まあそうすると今度アルゼンチンの可能性もあるんで、そういったことも考慮しないといけないんですが」

—「はええ……（既に意識を失いかけている）」

こぼる「ママー」

Santarh 「ちゅばちゅば」

iori 「（中略）まぁ GeoGuessr で今のところ唯一の救いは、中国とインドにほとんど Google ストリートビューが入ってないんで、特に中国には一切入ってないんですよ（香港・台湾を除く）。インドにはぼつぼつ、ドイツも法的な問題で観光地以外はほとんど入ってないですね。これらが参入したら阿鼻叫喚になると思いますよ。アメリカ・ロシア・ブラジルの攻略がこれからは課題ですね。これから頑張って行きたいと思います」

zion 「GeoGuessr でグローバルな活動してきました、って面接とかで言えばいいんじゃない？」

iori 「お、いいねそれ」

—「なるほど。ひとまずインタビューは終了です。ありがとうございました」

iori 「お疲れ様でした」

## 6 終わりに

ということで本記事では GeoGuessr の世界を垣間見た訳だが、いかがだっただろうか？ 今号ではプレイヤー視点からの GeoGuessr として、わさお氏による記事も同時掲載しているのでそちらもお楽しみ頂きたい。

最後になるが、硬式 GeoGuessr 部では共に甲子園出場を目指す部員を募集している。甲子園では 3 人 1 組のチーム戦となるので、それぞれの分野に特化したエキスパートを揃えたいところだ。米国特有である画質の悪い平坦な道から何州か判別できる猛者、スラウ語派やウラル語族における各言語の区別がつけられる猛者、国道・県道・州道命名の規則性に詳しい猛者などを随時募集している。

## 地理推定部 入部記

文 編集部 wasao

平成 27 年の春。快晴に恵まれた今日も、新入生の私は地理推定部の活動に参加しています！

地理推定部の活動は至って簡単で、ウェブサイト「ジオ・ゲッサア」での高得点獲得を目指し日々練習を積み重ねることです。

有りもしない「ジオ・ゲッサア」の大会を妄想し、そこでの活躍を目標にすることが部の慣わしとなっていて、今のところ「甲子園優勝」が部の標語です。

新入生の自分もそれは単なる戯言であると理解できています。実際、そのような慣習はどうでも良いようで、部員各人が、「ジオ・ゲッサア」に対して別の目的を持っているみたいです。

共通しているのが知識欲。都市名・公用語・名所・気候・植生・土色といった地理の知識と、国特有の道路標示・標識に関する知識を用いて、地図考察を行う先輩達の姿。

それは既知の知識と新たな発見が結びつく快感に溺れていく姿でもありました。

「なるほど、面白い」

その姿に惹かれ、私は入部を決意したんです。

思い出してみれば、入部初日。

私が持っていた「ジオ・ゲッサア」への印象はガラリと変わることになりました。



無作為に表示される景色を元に、その場所を特定する遊びである「ジオ・ゲッサア」。

2・3 回程しか遊んだことがなく、実のところそれ程まで面白いとは感じていなかった。

実際、部活動が必須であることのほうが大きかった。先輩の姿に惹かれたのは嘘ではないが、楽そうであるのが入部の一番の要因であった。

放課後、部室に向かうと入口付近で部長が談笑していた。そして私を見つけるやいなや、その柔らかな表情から一転して、まるで敵を見据えるような面付きで話しかけてきた。

「やあやあ、新入生ちゃん、地理推定部へようこそ。今から、1 ラウンド 5 分、検索なしの制約を付けてプレイしてもらう。ステージは日本だ」

それは、つまり日本の景色が出題されるということ。

「ジオ・ゲッサア」は1ゲーム5ラウンドの勝負になっており、1ラウンドにつき1つの場所が出題される。各ラウンドに付き5,000点満点、合計25,000点の得点を競うものだが、部活動見学時点とは異なり、この部がただの点数競いをやっているとは思えなかった。

殺氣。そう、殺気が感じられた。

この部に漂うこの殺気は私がこれまでプレイしてきた「ジオ・ゲッサア」に依るものではないと、薄々ながら理解した。その実相は捉えきれず、生半可な態度ではやられる、それは真であると察し、戦慄きを抑えるので必死になった。

ここで特筆すべきことは、時間制限の有無。私がこれまで遊んだ2・3回共、時間制限はなかったが、単純にそれだけの違いから、この雰囲気が創りだされているのか。疑問は私を焦燥へと駆り立てた。地理推定部特有の「ジオ・ゲッサア」、それは一体どのような形を成しているのか。

「では、始めてもらうよ」

ラップトップとマウスを渡され、指定の席に着く。

「ジオ・ゲッサア」が、始まった。

---

画面にまず映されたのは、一本の山道。木々に囲まれた場所だ。この様子から夏だと分かる。太陽の位置からして、お昼時だろう。無機質な虚空と言うべきか。空は只青かった。何の風も匂いも無い。

不慣れのためか、中々前に進めない。ヒントに成りそうな標識も得られない。一台の車とすれ違ったが、ナンバープレートはぼやかされている。

ここで「ビー」とビープ音が鳴り、一分経過のアナウンスがされた。

やっと少し拓けた場所に出た。相変わらず両脇には木しか見えない。雲の白さに影響され、アスファルトの色は徐々に薄くなって澄んでいくようだった。木々が風で揺動する。アスファルトからの放熱が、風と絡む。抱擁に近いその熱気は心地よく、眠気がそれに答える。

それでも私は前に進まなければならない。

山を感じる。その山気を吸い込む様に空が生き生きとした。

ここで、二分経過のアナウンス。

私は無心に山路を進んでいた。

ここまで 10 キロ程進んだろうか。 前方に青色の案内標識を発見し、胸を躍らせた。

その瞬間、私はその地に降り立った感覚を持った。

次第に連続性を帯びてきた視覚は、他の四感と統合され、その地で駆ける旅人へと私を変貌させた。

初めは虚無に感じた世界も、この時点では実在を持っていた。

ここで理解する。「ジオ・ゲッサア」での時間制限は、旅における目的と対応していることを。

目的のない旅、無意味に見るストリートビューに纏わり付く哀感など其処にはなかった。

通常では哀感が邪魔をし、集中を遮るだろう場面を、時間制限によって回避していたのだ。

集中は部室で座っているはずの私の五感を、その場所へ移した。

「県道 353 号線、十勝岳温泉迄 20 キロ」

これで都道府県は北海道、そしてその中央付近が出発点であると分かった。

ここで、四分経過のアナウンス。

右下のマップから、353 号線を探す。焦りは高揚に似た。

十勝岳周辺を拡大し、353 号線を探す。無い、見当たらない。

残り三十秒のアナウンス。

ここで偶然にも 353 号線のマークを発見した！

制限時間ギリギリであったため、急いでマークの上にピンを置く。

そして、終了のアナウンスと同時に「MAKE GUESS」のボタンをクリックした。

---

同様に、残りの 4 ラウンドも座りながらにして、私は日本を駆けた。その殺気に近い気勢が、先輩方のものも含め、部室には溢れていたのだった。

「部長、これは一体 ……！」

「ジオ・ゲッサアだよ」

至極明白な答えに感銘を受け、「ジオ・ゲッサア」そして地理推定部の存在が、質量を持って大きく感じ取れた。そして地理推定部での活動の決心は確固たるものになった。

—  
—

あの入部から二週間が経つ。活動に慣れ、現在では先輩同様世界で旅をしている。

帰宅前、部長から話しかけられた。

「やっていけそうかね」

「はい。なんというか、自分を見つけられそうです」

嗜み合わない会話も、ここでは、通じる。

# そうだ 畑 行こう

文 WORD TOKIO 部 くりす

## 1 あはん

- ・ワ・「やつたー」
- ・ワ・「ごーさいんでた」
- ・ワ・「よさんがついたー」
- ・ワ・「ぶろじえくとほつそくー」
- ・ワ・「けんこく、けんこく」
- ・ワ・「はたけつくろー」←今ココ

## 2 やあ

みなさん、おはこんばんちわ。くりすです。

WORD 32 号掲載の拙稿「世捨ておじさんの週末」を覚えていらっしゃいますでしょうか\*1。去年の5月にわたしが滋賀県の祖母宅で畠仕事をして悟りを開いたというものです。ぜひ本稿とあわせてご一読くださいませ。そういうればあれの続報をWORDには掲載していなかったのですが、短く表すならじゃがいもが170kg 収穫されて最高だったというところでしょう。

あれからもときどき野良仕事に首を突っ込んでおりましたが、なんと、ここつくば市内に自分の農場を手に入れた次第でござります。本稿ではつくばで農業を始めて約3週間となる現在までの進捗について述べます。

## 3 すべてはここから始まった

今回借りた農場は、イースつくばの道向かいにある「ファミリー農園」\*2の10坪区画です。いつ頃だったかはもはや覚えておりませんが、だいぶ前のある時「つくば市内に激安の貸し農園がある」という情報を小耳にはさみました。なんでも、5坪なら年間5,000円、10坪なら年間9,600円で借りられると



図1: 筆者

\*1 読んでないよ！ という方はどうぞこちらからバックナンバーをダウンロードしてくださいませ！ - <http://www.word-ac.net/>

\*2 株式会社谷口企画が運営する貸し農場。 <http://kasinouen.web.fc2.com/>



(a) 我々の使用区画



(b) ファミリー農園・南側

図 2: ファミリー農園・4月 19 日

いうのです。じやがいも畑での素晴らしい体験以来農業に並ならぬ思い入れがあったわたしは、寒さも和らいできたことすこしある農業を始めようかと思い、貸し農園に連絡をとりました。電話口の向こうは農園オーナーの谷口さん。「じゃあ今週末にでも見に来てください」とのお話を受け、我々 3 名<sup>\*3</sup>はファミリー農園に赴くのでした。

## 4 Hello, Field!

4月 18 日土曜日、農園に到着した我々はただただ圧倒されました。広大というにはちょっと大きさかもしませんが、イースつくばを西に望む住宅街の間に、合計 1,200 坪強の、立派な農園が広がっていました。が、注目すべきは広さというよりは、農園全体の活気の良さでした。あちこちに立派な畝やビニールトンネルが並び、元気の良い葉が青々と育ち、そしていつも誰かしら利用者がせっせと作物のお世話をしているのです。

—そこは豊穣の神の宿る地でした。

我々は駐車場からの便の良い 10 坪の区画を借りることに決定し、翌日に再び現地を訪れて使用契約を結ぶ旨の約束をし、ひとまず現地をあとにしました。

さて、念願の農場を手に入れたので、次にすることは種の調達です。我々は車を走らせ、石岡市の JA やさと柿岡直売所へと赴きました。あとから知ったのですが、実はイースつくばに JA 土浦が店舗を構えているそうで、こんなに遠出する必要は実のところありませんでした。まあ、軽いドライブはわたしの人生を豊かにするのに不可欠なので、必要だったということにしておきましょう。

<sup>\*3</sup>筆者くりす、および編集部員のらいりゅう、Mt.Shadow 両氏。

JA やさと柿岡直売所に到着したのは正午すぎでした。ここでらいりゅう氏が昼食を所望したのですが、そこは JA、お弁当やお惣菜、おにぎりも（ほぼ）産地直送で販売されています。種を買うついでに昼食でもどうでしょうということで、おにぎりを各々 2 個ずつと、白身魚のフライ、それからメンチカツを 3 人で分けあいました。味は申し分なく、それが「農協で買った食品である」という事実がもたらすプラシーボ効果であるかは我々には確かめる術はなかったのですが、まあおいしかったのでよしということで。

最終的に、入手した種は次のようになりました。

- |          |        |        |         |
|----------|--------|--------|---------|
| • ラディッシュ | • 青首大根 | • オクラ  | • モロヘイヤ |
| • サニーレタス | • 水ナス  | • えだまめ |         |

とりあえず楽しそう・おいしそうな野菜を選んでいたのですが、このときはつくばの気候を中間地だと仮定していました。ところが、あとで谷口さんに尋ねたところ、この時期のつくばは比較的冷涼で、我々が用意した種はまだ発芽に適さないという事実が明らかになりました。これだから茨城はよく南東北と言われるのでです。

結局、実際に種をまいたのは、日中の気温が安定して 20°C を超える 5 月に入ってからとなりました。

## 5 種まき・第一弾

5 月 2 日土曜日。ゴールデンウィークが盛り上がりをみせるこの日、我々<sup>\*4</sup>はラディッシュ、サニーレタス、大根、そしてナスの種をまきに畑を訪れました。この 3 日前に谷口さんに苦土石灰をまいて耕耘機で耕してもらったので、土の柔らかさと pH 調整はバッチリでした。土はまるで布団のように柔らかく、新しい芽生えを擁するにぴったりでした。

水はポンプ井戸から汲みます。もちろん手動。ここで地下水をじょうろやペットボトルに汲み取り、畑まで運んで使います。井戸なんて小学校の宿泊学習でも使った記憶がありませんでした。

いよいよ農業らしくなってきました。畑仕事といえば鋤です。鋤で畠を盛り、てっぺんに指で溝を掘って種をまき、種の上に軽く土をかぶせたら水をまきます。3 人で順繰りに取り掛かると、自然と流れのよいパイプライン工程が出来上がっていました。プロジェクトとは本来こうあるべきですね。

こうして畑での初日はあっけなく、それでいてゆったりと過ぎていきました。

ところが、その翌日、らいりゅう氏より

「アリガレタスの種を巢の中に運び込んでいる」

という情報が入りました。早くもハプニングです。アリってレタスの種を食べるんでしょうか？ それとも好奇心から持ち去っただけなのでしょうか？ いずれにせよ困るのでアリさんたちには死滅してもらいます。ごめんなさい、わたしたちは人間なのでこういう発想をしてしまうのです。

---

<sup>\*4</sup>筆者くりす、編集部員のらいりゅう、polamjag 両氏。



(a) 井戸



(b) はたらく農夫たち

図3: ファミリー農園・5月2日

## 6 萌芽、種まき・第二弾

5月6日水曜日。ゴールデンウィーク最終日となるこの日の午後には、オクラの種をまきに畑を訪れました。この日も、土を耕して畝を作つて種をまくという、何日か前と同じ工程を実行することが主たる目的だったのですが、大きな変化がひとつ、ありました。

らいりゅう氏が満面の笑みで

「くりすさん、ほら、あれ！」

とラディッシュを植えたあたりを指さします。見てみると、さっそく芽が出ていたではありませんか！ 新たなる芽吹きに無類の感動を覚え、思わず写真をパチリ。

でもって、畝の向こう側ではアリさんたちが築城しておりました。申し訳ないのですがそこにアリエンパイアを建国されてしまうと、我々の食糧供給が脅かされてしまうのです。内心楽しみつつ心を鬼にして、彼らの地下帝国を水攻めにしました。水は対アリ兵器としてどれほど効くのかは存じ上げませんが、化学兵器の使用はマイルールで禁じられておりますゆえ……。

種まき・第二弾はこれにて完了です。現時点では、発芽のためにもうちょっと高い気温を要するモロヘイヤとえだまめはまだ植えていません。こちらの種まきは6月になるでしょう。



図4: ラディッシュの芽

## 7 名言集

- polamjag 「畠って上から見るとメモリスロットみたい。  
葉っぱも緑色だしどう見ても RAM でしょ。」  
らいりゅう 「(承前) サーバラックっぽい気もする。」  
らいりゅう 「畠を作つて種をまいて水をまくの、3人でやると完全にパイプラインだ。」  
polamjag 「Redmine でガントチャートとかチケットとか駆使してプロジェクト管理しよう。  
もちろん Slack 連携で。」  
らいりゅう 「最高のプロジェクトだ！」  
polamjag 「うんうん、これもまたアイカツだね！」

せっかくテクノロジーを捨て去り大自然に裸一貫で飛び込んだというのにこの有り様です。しかしながら、テクノロジーもひとつの地球生命体たる人類が産み出したものですから、自然の摂理と共通点があるということは何ら不思議でないのかもしれません。人類のあるところにテクノロジーあり、プロジェクトあり。

## 8 おわりに

いかがでしたか。農業の奥深さや、農作業がもたらす充足感はもはや言語化できるものではありません。早くも農業の醍醐味を味わったといえる気もするのですが、本番はまだこれからです。このプロジェクトはまだまだ始まったばかりです。

次号ではめきめきと成長していく作物たちの様子や、うまくいけば収穫の様子をお伝えします！……だつたらいいな。



# 情報科学雑誌 WORD 読者アンケート

題字 元編集部 ふあい

文 編集部 ひだるま

## 1. あいさつ

先号では久々にアンケート記事無しでの発行となりました。さびしかった？さびしかったならさびしかったって言え！ひだるまです。今号では 34 号アンケート、35 号アンケートの結果を 2 記事に分けてお送りします。「先々号の内容なんて覚えてないよお（><）ふええ」という方は

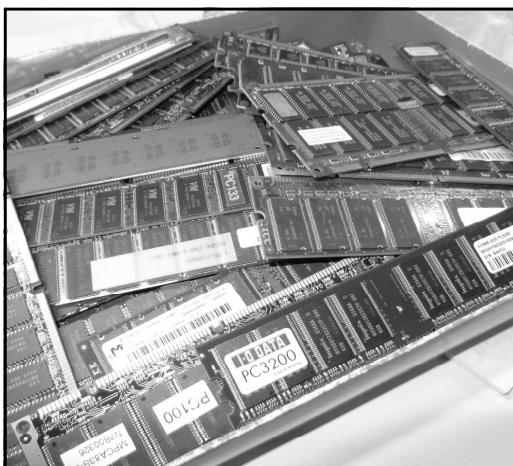
<http://word-ac.net/>

で過去記事を読むことができます！

ノルマ消化したので次にいきましょう。

## 2. 今回の粗品

アンケートを提出する際、各置き場の回収箱ではなく 3C ラウンジ横の WORD 編集部（3C212）まで直接持ってきてくださった方には先着で粗品を差し上げています。最近扉の見た目が変わったとか違いがよく分からないとか。近々インタヒョンを実装したいなあ（願望）。



### 2.1.メモリ型定規

WORD 名物メモリ型定規。捌けてきたと思ったら誰かしら追加するので減らない。



## 2.2.WORD ステッカー

全然アンケート回答者が取っていかないのに何故か減っていく WORD ステッカー。なおこのシールの使用者の方について当編集部では一切の責任を負うことができません。

### 3. アンケート集計

#### Q 1 : 所属を教えてください。

情報科学類	: 1 人	カワグチ	: 1 人
システム情報工学専攻 : 1 人			筑波大学勇者部 : 1 人
弟が筑波大生	: 1 人	筑波大学 OB (白熱)	: 1 人

今回は基本に忠実に情報系学生中心のようです。正直毎回コメントする必要があるのかは分からぬんですが、変わった所属の人間についてだけコメントしているとそんなのばっかりになっていくので。そのままの貴方たちでいいんです。

期間が開きすぎたためにひょっとしたら結城友奈が勇者だったことを忘却している方もいるでしょう。編集部にも別口で地球を守る組織所属の人間がいます<sup>\*1</sup>。4.1といいつつ大分システムが改善されているようで、ruby の 1.8 から 1.9 への変更みたいな感じでしょうか。

#### Q 2 : 性別を教えてください。

男 : 5 人	秀吉 : 1 人
女 : 0 人	

あつバカテス最終巻まだ買ってないや。今回は「女」の項目も載せてみましたが、より残念感が出ていて個人的には気に入っています。

#### Q 3 : 年齢を教えてください。

24 歳 : 2 人	++0x15 歳 : 1 人
24 歳 (厄) : 1 人	じゅうなな歳 : 1 人

---

\*1 EDF! EDF!

37 歳：1 人

圧倒的 24 歳。厄年を経験した人間が口々に「厄年はある」と発言していくので恐怖ですよね。僕は前厄の段階で既に自転車が 4 度壊れるという事態に遭遇しました。

じゅうなな歳。ひらがなで書くと「じゅうな」を含んでいるんですね。気づきを与えてくれた貴方に「ぜひいざさんくすおんりー」の言葉を贈りましょう。

#### Q4：良かったと思う記事があれば教えてください。

先々号の記事一覧はこちら。

- |                   |                      |           |
|-------------------|----------------------|-----------|
| 1. 表紙             | 6. EmacsStartUp(基礎編) | 11. 配布場所  |
| 2. 号名             | 7. WORD 読者アンケート      | 12. 配布時期  |
| 3. 目次             | 8. 編集後記              | 13. 配布媒体  |
| 4. 今日からマのつく創作者！   | 9. 裏表紙               | 14. 冊子の厚さ |
| 5. わかった気になるHTTP/2 | 10. アンケート用紙          |           |

#### 集計結果：

- |          |           |
|----------|-----------|
| 1. : 2 票 | 9. : 1 票  |
| 2. : 1 票 | 10. : 1 票 |
| 3. : 1 票 | 11. : 1 票 |
| 4. : 1 票 | 12. : 1 票 |
| 5. : 1 票 | 13. : 1 票 |
| 6. : 2 票 | 14. : 3 票 |

**7. : 2 票**

8. : 1 票

記事の割合だけみると技術記事と娯楽系記事の割合が半分くらいでした。娯楽記事が増えるとアンケート回収数が上がる傾向が確認されています。だらしねえな？

7.

**6. 私は ThinkPad を愛用してます**

14.

【システム情報工学専攻 ムラサキのバラの人 さん (24 歳)】

僕も新機は ThinkPad になりました。以前は Let's note SX だったんですが、あのシリーズというか Let's note 系は総じて解像度が上がってくれないので泣く泣く切り替えました。X250 の不満点はやはり最大メモリ 8GiB の制限ですねえ。

### 逆にあったと思いますか（威圧）

【弟が筑波大生（弟が筑波大生）トヨ(C)さん（24歳（厄））】

良い記事しかなかったよなあ（すっとぼけ）。

6. 勉強になった（小学生並の感想）

1. 時事ネタ（why-hukko）

【筑波大学勇者部 勇者（派遣）さん（じゅうなな歳）】

小学生でもガリガリ書いている人間は書いていて辛かつたりするこの業界。

タイトル自体は正確には表紙ではなく号名なんですが、まあいいや。

## 14. 手に取りやすい

【筑波大学OB（白熱） 勇者（派遣）さん（じゅうなな歳）】

となると今号とかはちょっと重いかもしれませんね。

### Q5：良くなかったと思う記事があれば教えてください。

1. : 1票

6. : 2票

15. : 1票

6. ムズカシイ

【カワグチ カワグチさん（24歳）】

入門（初心者向けとは言っていない）。「初めての～」とかついている本も全然初学者向けじゃない気がしますよねまず厚さ的に。とはいえる原書では全然別のタイトルがついていることもあったりして今回もそういうことなのかもしれないですね（適当）。

## 6. 宗教はNo

【情報科学類 清き一票さん（++0x15歳）】

良かった記事に6だけ含めない辺り徹底しています。とはいえる「郷に入りては Go is God」と古事記にあ

るようすに、大学の授業で使うと言っているのだから授業中くらいは頑張ってみましょう<sup>\*2</sup>。

## 1. コンテンツの並びが目次、本文と違う

【筑波大学OB（白熱）匿名希望さん（24歳（厄））】

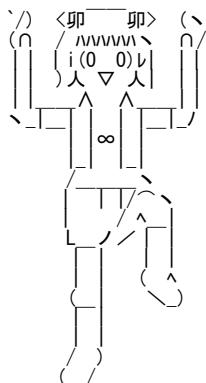
表紙はデザイン優先だからね。仕方ないね。

14. 年末だからあわてて出しました感が.....

4. ホメ春香は出してはいけない！いいね？

【弟が筑波大生（弟が筑波大生）トヨ（C）さん（24歳（厄））】

わっぽい わっぽい



編集部員には一年間での努力目標があるのは部内では有名な話。それとアンケを溜めすぎると大変になることがあるとかないとか<sup>\*3</sup>。まあ、アンケート記事は全体で作ってる訳じゃないので本誌発行のタイミングによっては息をしてないことがあるんじや。

### Q 6：過去の記事に関する感想を教えてください。

過去記事は <http://word-ac.net/> から読むことができます！

2013.4 別冊WORD内の宿舎でできる自炊

味霸と本つゆを出せるとは... わかっているな...

【弟が筑波大生（弟が筑波大生）トヨ（C）さん（24歳（厄））】

味霸は一度に使う量が割と少なくて余らせやすいというのが難点ですね。というか会社分裂したんでしあつけあそこ。

\*2 Apple アレルギー起こしかけているひだるまが言うのもアレですが。

\*3 今みたいに！

進捗がないと記憶が消える呪いにかけられてしまってな。

【筑波大学勇者部 勇者（派遣）さん（じゅうなな歳）】

か、格好いい……！ そういえば「記憶を消してもう一度見たい／やりたい作品」なんてのをたまに聞きますが、上手く進捗を調整すれば実現できるんですかね。

進捗がないと学籍が消える皆さんも頑張ってください(◐◑)♪”

読んでたのは 15 年以上前なので忘れた  
└ (旧情報学類のWORD時代)

【筑波大学 OB（白熱）匿名希望 さん（37 歳）】

熱い古参アピール。そう、WORD は歴史ある団体なんですよ。「そしあのパクリですか」とか言う心ないコメントを頂いたこともありましたが MS-Word よりも古いんですよ一応。流石にその年代となると

# http://www.word-ac.net/

には置けてないかもしれませんねえ……。

## Q7：使っているシャンプーを教えてください。

使っている（意味深）。先日部内で突発的にアンケートをとったところ、髪に気を使っている人間があまりいないという事が発覚したのを受けてこのような質問を行いました。「覚えてない」はともかく、「ボディソープ」は流石にね？

# V i m

【情報科学類 清き一票 さん（++0x15 歳）】

シャンプーに Vim 使うとか言い出す人間が「宗教はNo」とか言い出すのなんなんだ（驚愕）。

## アニキからの泡立たない白いシャンプー

【弟が筑波大生（弟が筑波大生）トヨ(C) さん（24 歳（厄））】

「**アニキから**」「**白い**」頂きました。あれでもあれ泡立つて話だしよっぽど粘度高いのを使って（検閲削除）

飲まんでもいい

【筑波大学勇者部 勇者（派遣）さん（17歳）】

別に「シャンプーを飲む」というワケではなくて、「声優の構成物質を摂取することによって自身の構成を声優に近づけ、関係性の接近を図る」一手段という見方もあるのではないか。

**Q 8：自由記述欄。**

(先行さんが不安にならぬ様子)  
 来年も一年  
 カンばるぞい！

【筑波大学勇者部 勇者（派遣）さん（17歳）】

NEWGAME! 2巻出ましたね。僕も精神安定用に購入しました。

:(  
 :)

【情報科学類 清き一票 さん（++0x15歳）】

発行のスパンが短かすぎやしそせんかね  
 届けたアンケートが無駄になってるんですが  
 誠意を見せて頂さたい！  
 本当に、どうに発行したんですか……  
 そりゃあ、のんのんびりの2期タイトルが  
 リピートに決定しましたね。て、ヨリ叶なみけの様  
 におかわりとか、おかえりみたいな物だと思つたら  
 まさかの横文字、ん？おかわり…？フユ………  
 、  
 おいいいですよな！カブトムシ!!

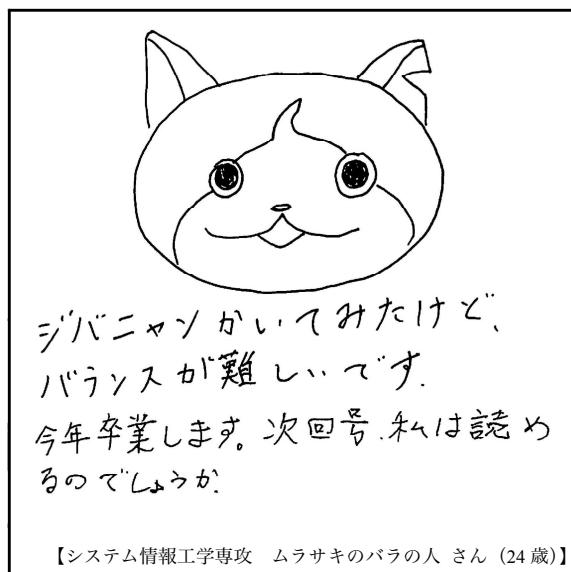
【弟が筑波大生（弟が筑波大生）トヨ(C)さん（24歳（厄））】

誠意（ぼろん）。まあ口先ではなんとでも言えますからね。人生に無駄なものなんて何一つないんです。言い換えれば無駄じゃ無いものなんて何一つないんですよきっと。そんな感じ。

田舎モノで「リピート」とか繰り返しつづくワードを聞くとどことなく不気味な感じしません？しませんかそうですか。



ほんげ～



自由投稿欄なので掲載が明らかにNGな爆弾を持ち込まなければOKなんですが、イラストの感想ってときどきすごく難しいなって最近思います。

皆巣立っていくんですね。感慨深いものが……え？ 次回号のWORDが読みたい？ そんな貴方に

**<http://www.word-ac.net/>**  
っていうWebサイトがありましてね！？ 過去記事がアップロードされるんですよ～。紙媒体に比べ遅れますが、ゆっくり待っていてね！！

現役の学生の皆さん、

おはこんばんちわ (死語)

あ、僕浪人じやん。ちなみに「おはこんばんちは」ですが、今号で編集部員に使用者がいることが判明しました。探してみてくれよな！

【筑波大学 OB (白熱) 匿名希望 さん (37歳)】

#### 4. 終わらない

まだもうちょっとだけ続くんじやよ。落ちてなければ<sup>\*4</sup>。あ、そうそう Web 版アンケートですが反響が芳しくなかったので暫く塩漬けですねこれは。

\*4 落ちたってよ。

## WORD 編集部への誘い

文 編集部 カブ

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。

**小林秀和** @KOBA789

**リザウド** Lisp 大好きクラブ会員

**びしょ～じょ** 百合仙人

**もねと** 生きるジャーゴン辞書

**ioriveur** つくばの革命家

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

学年ビリのWORD部員が  
1年で精神年齢を20下げて  
幼稚園に入園した話号

発行者  
編集長  
制作・編集

情報科学類長  
根本 晃輔  
筑波大学情報学群  
情報科学類WORD編集部  
(第三エリアC棟212号室)

2015年5月26日 初版第一刷発行  
(512部)