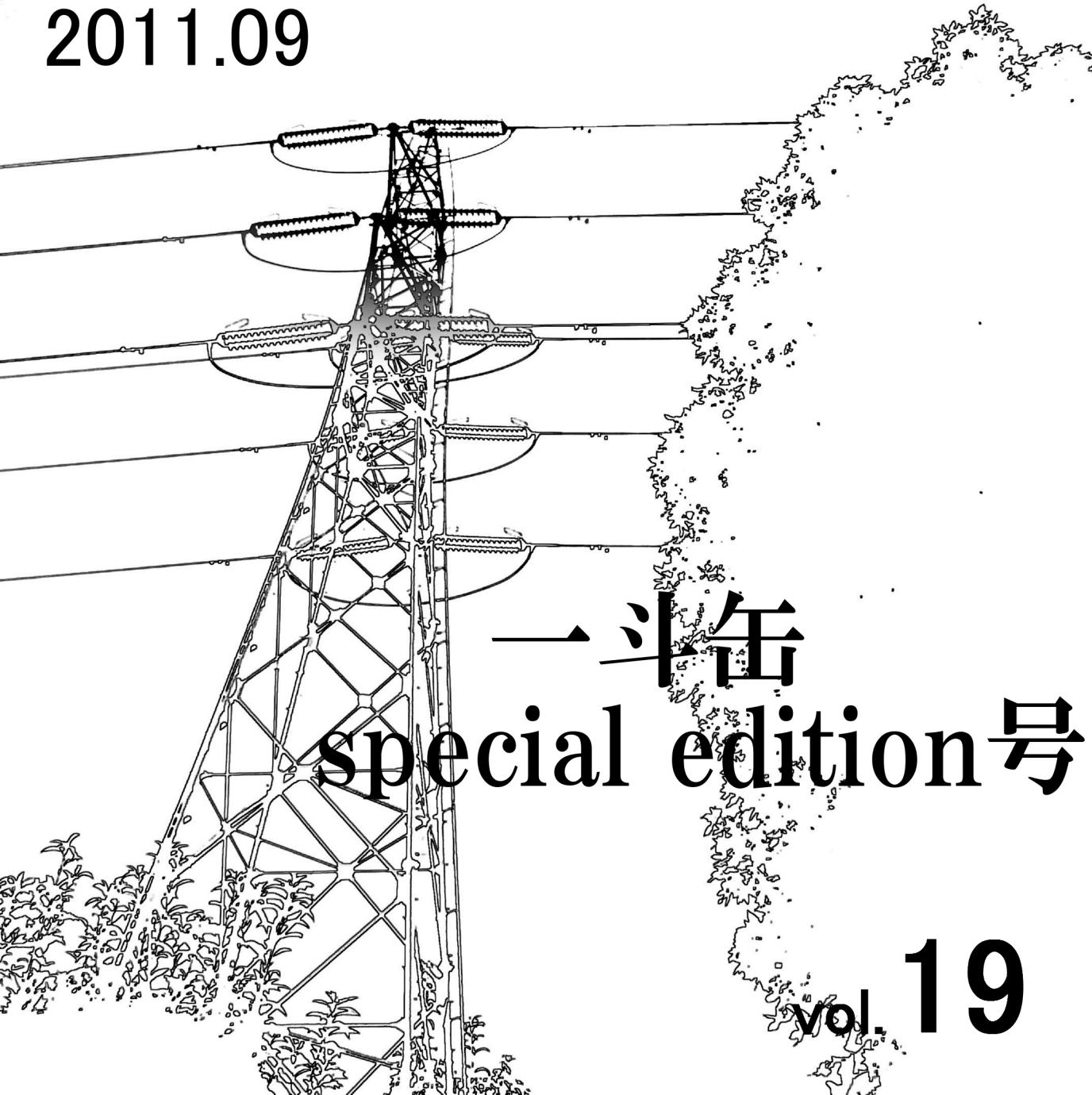


# WORD

From College of Information Science

2011.09



— 企画  
special edition 号

vol. 19

# 目次

p.3 真夏の夜の死闘 ~東日本縦断レース~  
(feat. GR な日々。VIII)

p.11 マニアック正規表現 序論

p.20 GNU screen から tmux へ

p.24 本格的♂イタリアン

p.30 全自動ふあぼ bot プロジェクト

p.35 情報科学類誌 WORD 読者アンケート

# 真夏の夜の死闘 ~ 東日本縦断レース

(feat. GRな日々。VIII)

文 編集部 Santarh 葡萄酒

## 始めに

「車 vs18 きっぷ」—— それは新宿から札幌という長大な行程を、自動車と鉄道のどちらが早く移動できるかを競うエクストリームスポーツである。

前夜から、この過酷な決闘の行く末を暗示するような雨が関東地方全域で降り続いていた。台風 12 号の影響である。出発を目前に控え、果たして両チームとも無事に北の大地を踏むことができるのであろうか、そんな不安を抱かずにはいられない夜の空模様であった。

## レギュレーション

### ▼共通

新宿駅を 2011 年 9 月 2 日の 23:10 に出発し、先に札幌駅に到着したチームを勝者とする。

### ▼ 18 きっぷチーム(以下「鉄道組」 / 担当 : 葡萄酒)

JR が発売する「青春 18 きっぷ(以下「18 きっぷ」)」を使い、JR 線の普通列車のみを用いて移動する。ただし 18 きっぷの特例に従い、蟹田-木古内間は特急列車の普通車自由席に乗車できる。

### ▼自動車チーム(以下「車組」 / 担当 : Santarh)

車組は有料高速道路を利用せず下道を走る。



移動経路概略

## #車 vs18 きっぷ

09/02 鉄道組

23:00 / 新宿駅

▼新宿駅の長い通路を抜けると、5番ホームにはお目当ての列車が既に到着していた。新宿発新潟行の臨時快速「ムーンライトえちご」である。この列車は夏休みなどの繁忙期にのみ運行される夜行の快速列車であり、指定席券を買えば18きっぷで利用することができるため、(一部の)旅行者には人気が高い。18きっぷで利用できる唯一の夜行列車であるムーンライトは、貧乏旅行には欠かせない存在である。

早速乗り込み座席に腰を下ろす。あとは何事も無ければ、寝ている間に新潟に着いている筈である——が、どうにも様子がおかしい。時は23時15分、既に出発時刻は過ぎているのだが、一向に列車は動き出す兆しを見せないので。不穏な沈黙の後、しばらくして車内放送が静寂を破り車内に響いた。

"山梨県内の大雨のため、中央線が遅延しております。

ご迷惑をおかけしますが、運行再開の予定が決まり次第お知らせいたします"

車内がざわつく。同じく新宿発白馬行きの「ムーンライト信州」は既に同日の運休が決定したと聞く。もしや、この列車も——そんな悪い予感が胸をよぎる。15分後、そんな我々の不安を拭い去るかのように、ムーンライトは力強い鳴動を上げて新宿駅を出発した。23時25分、予定よりも15分遅れで鉄道組は旅路の出発点に立った。

23:25 / 渋谷駅

▼ようやく一步を踏み出したかと思った次の瞬間、出鼻を挫かれることとなつた。次の駅、池袋でも出発が遅れている。京浜東北線で人身事故が発生したという。結局更に5分間の停車を余儀なくされ、車組に大きなリードを許すという最悪の滑り出しとなってしまった。

09/03

1:00 / 高崎駅

▼大宮を発ち、しばらくして日付が変わった。次の高崎からは18きっぷの適用が始まる。ここまで区間で、累計20分の遅延は7分程度に短縮されたようだ。どうやら車組も高崎付近にいる様子である。ようやく光明が見えてきた。隣のホームに停車していた急行「能登」を見送った後は、停車時間を短縮し定期に高崎を発車。完全に並走しているわけではないので正確な時間は判らないが、発車直後に車組を抜いたようである。

## 09/02 車組

23:00 / 新宿駅

▼ 23 時に新宿駅周辺を出発した我々は、まず新潟県を目指して国道 17 号線を走った。埼玉県に入るとさいたま市、熊谷市を通り、群馬県に入る。

## 01:00 / 群馬県

▼ 群馬県に入った途端、文明の光が明らかに少なくなったのでこれが噂のグンマーなのかと驚いた。とても暗い前橋市を通り、「\*\*\*D」に出てきそうな峠道をトラックに囲まれて走る。どうやら風が強いらしく、至るところで木の枝が道路の上を自走していた。

## 09/03

03:00 / 新潟県境

▼ 群馬の峠を攻めること 2 時間、ついに群馬県と新潟県の県境にある三国トンネルを通過。

「トンネルを抜けるとそこは雪……じゃなくて大雨なんだが？」

俺の人生は晴れ時々大荒れ……ッ！ イイ人生だよッ……！ 新潟に入り、流石雪国と言うべき広大な道路を走行していると、近くに越後湯沢駅があるのを発見した。そこで鉄道組の動向を確認すると、1 時間前に通過してしまったらしい。

## 05:00 / 新潟県中央

▼ そのまま新潟県を北上し、山道から日本海に向けて、夜明け前の空の下を走っていると綺麗な日の出を見る事ができた。土地と空を感じると、まさに日本海の空気であった。寝不足気味の体にはキツい朝ではあったが。



道の駅あつみ 夕陽のまち しゃりん

## #車 vs18 きつぶ

▼新潟に定着、太平洋側の悪天候が嘘のような朝焼けである。ここから白新線 に乗り換える、北へ。	04:51 / 新潟駅
▼村上、酒田と乗り換えて海沿いを走り抜け、秋田に到着。次の列車まで1時間以上あるので、昼食をとることに。この待ち時間で車組に抜かれてしまうが致し方ない、鉄道のダイヤは既に決まっているのだ。  秋田を出ると、関東とはまた違った風景が目に入る。沿線には広大な水田が広がり、如何にもコメ所といった趣である。また合間に植えられている鉄道林も独特で、やたらと細長い針葉樹が密集して生い茂っている。井川さくら駅という何とも人名のような不思議な響きの駅を過ぎて、次は大館である。	11:31 / 秋田駅
▼おおだて なみおか ▼大館で乗り換えた後、浪岡付近で車組を抜き、青森駅に到着。車組はしばらく青森でフェリーを待つようだが、鉄道組は青函トンネルを抜けて一足先に北海道へと踏み込む手はずである。例の如く列車待ちの時間に抜かれるようなので、折角だからと車組を待ち構えて撮影することに。	16:55 / 青森駅
▼青森を出て、多少遅延しながらも蟹田に到着。ここからはお待ちかねの特例区間である。青函トンネルを通過する普通列車は存在しないため、この旅行で唯一の特急列車に乗れる区間だ。  2分遅れで出発したものの、スーパー白鳥の快適なシートに身を委ねている間に、あっけなく青函トンネルを抜けてしまった。流石は特急列車である。何はともあれ北海道に上陸完了。	18:54 / 蟹田駅
▼快適な特急列車に別れを告げ、ここから先は試される大地のディーゼル車である。しかも冷房が付いていない。この時間になると流石に乗客も少なく、外の風景も暗闇に紛れて見えない。  余談ではあるが「知内は『しりうち』と読むのに木古内は『きこない』と読むのが納得いかない」という旨をTwitterにポストしたところ、現地の方から地名の由来に関する返信を頂いた。なんでも知内は「チリオチ(鳥の居る場所)」というアイヌ語が由来であり、木古内も同じように「リロナイ(潮の差し入る川)」という単語に当て字をして現在に至るらしい。北海道はアイヌ語が由来の地名・駅名が多く、日本語の感覚からすると風変わりなものがあって興味深い。	20:12 / 木古内駅

08:30 / 山形県

▼新潟からは日本海沿岸の国道 7 号線を北に進み、山形、秋田を経て青森へ向かう。

15:00 / 秋田県大館駅

▼ここまで鉄道組に先行されていた車組はついに、秋田の大館駅にて追いぬく。第一次デッドヒートが始まる。奥羽本線と並走する区間にて抜きつ抜かれつつ激闘したが、やはり先行されてしまう。

17:00 / 青森県青森駅

▼先に青森駅に到着した鉄道組が車組の通り道に待ち構えていたので、数秒ほど鉄道組の姿を確認することができた。謎のすれ違いに遭遇した後、本州北端の大間へ。運転に疲れた体を浅虫温泉で癒した後、フェリー乗り場まで到着。翌日朝のフェリーまで、睡眠をとることができた。

**09/04**

10:00 / 函館

▼フェリーに乗って函館へ。既に鉄道組は函館で一泊していたので、追いつかなければならない。道なりに国道 5 号線を北上することになったが、道が広いことに驚かされる。つくばの道も十分に広いが北海道の道は更に広く、走っていてとても快適である。一方、地名に目を向けると同じ名前の山が多かったり、大きい沼がある大沼など、安いなネーミングが多々見受けられる。一方、アイヌ語由来と思われる土地も多く、読めない名前も多い。「蘭越」など、とても面白い名前も多かった。道なりに進んでいると、大沼公園付近にて乗り換え待ちの鉄道組を追い抜き、大幅に先行することができた。この時点で勝ちも決まったものだと思っていたのだが、この慢心が後の悲劇を生むとは誰も思わなかつたのである。

11:30 / 長万部

▼内浦湾を北上し、長万部についたところでま○べくんを探すために我々は長い休憩を取った。鉄道組が 30 分後に長万部駅にて 2 時間待ちの乗り換えを行うので、それを見届けようとしたのだ。30 分後、定刻に鉄道組が乗っている車両が入ってきたわけだが、どうもおかしい。ホームに既に入っていた別の列車がすぐに出発してしまい、ホームには誰も残っていなかった。どうやら待ち時間無しで乗り換えてしまったらしい。我々のアドバンテージが、吹っ飛んだのだ。第二次デッドヒートの始まりである。すぐさま車に乗り込み、国道 5 号を走らせることとなつた。

## #車 vs18 きつぶ

▼函館に到着し、この日の全行程が終了、あとはホテルに帰って寝るだけである。何とこの日の移動距離、鉄道だけで 950km を越える。つくば市から新宿までの移動を考えると 1000km を越えたのではないだろうか。

到着直前に編集部のふあい氏より「ラッキーピエロというハンバーガーショップに立ち寄ると良い」という情報を頂いたので、寄り道してみることに。チャイニーズチキンバーガーを注文、揚げたての唐揚げに甘酢のようなソースがかかっており、とてもジューシーである。他にも独創的なメニューが多いので、函館にお立ち寄りの際は是非訪問することをお勧めする。

21:16 / 函館駅

09/04

08:17 / 函館駅

翌朝、函館を出て長万部へ。特急の通過待ちが多いため駅毎の停車時間が非常に長く、折角得た前日からのリードも車組に奪われてしまった。森駅でも長い停車時間があったので、かの有名な駅弁「いかめし」を購入することに。鳥賊、餅米、粳米、醤油、ザラメというシンプルな材料ながら実に旨い。柔らかく煮た鳥賊もさることながら、内包された米に煮汁が染み込んでおり、とても満足できる一品であった。

11:31 / 長万部駅

▼この日は朝から東の方で大雨が続いており、その影響で遅延する列車が多発していた。本来ならここで 40 分もの大きな待ち時間がある筈だったのだが、1 本前の 9 時 8 分発の列車が 2 時間以上遅れて長万部に停車していたため、存在しないはずの乗継ぎが成功してしまった。鉄道組はこれで到着時刻が相当早まったことになり、車組に対して大きなリードを得ることになった。同じく長万部駅にて油断していた車組を追い越し、小樽へと向かう。

14:26 / 小樽駅

▼小樽からは快速エアポートで札幌までひとつ飛びである。折角なので指定席券を購入して u シート（快速エアポートに設けられた指定席。普通の車両よりも快適で豪華なシートが魅力。指定席券 300 円）の快適な座席を確保した。決戦の地は近い。

15:06 / 札幌駅

▼ついに二日に渡る大移動を終え、終着点である札幌に到達した。札幌市街には車組とほぼ同時に突入りし、20 分ほどのごく僅差で鉄道組が先に到着した。

土壇場での勝負を分けたのは、鉄道組には市街地での渋滞など関係なかった点だろう。無論、大きな勝因は長万部での奇跡じみた乗り換えであったことは言うまでもない。

14:00 / 小樽市

▼列車と並走しつつ小樽市を抜け、ついに札幌へ。しかし札幌市内は道路が混雑しており、手稲駅付近で鉄道組に追い越されてしまう。

15:00 / 札幌市

▼札幌市内では神輿がトラクターに牽かれて走っていたり、ビルの狭間の草原に牛が飼われていたりする光景を横目に見つつ、Twitter にて鉄道組が札幌駅に到着したとの報告を受ける。

15:26 / 札幌駅

▼我々もその 20 分後に札幌駅へ到着。

この戦い、誰が 20 分差で決着すると予想ただろうか。全ては長万部の悲劇……  
我々がま○べくんの看板を見つけてしまったのが、勝敗を決してしまったのだ！



到着証拠写真 at 札幌駅

#車 vs18 きつぶ



快速「ムーンライトえちご」



青森駅で車組を激写



姫川で待ち合わせた特急「スーパー北斗」

# マニアック正規表現 序論

文 編集部 吉村 優

## まえがき

正規表現というと、情報科学類の方ならコンピュータリテラシなどで学習したと思います。多分講義で習うものは正規表現であつただろうと思いますが、正規表現はおかしい程パワーアップして、最近のものは正規表現とは呼べなくなってしまったのだそうです。

なので、これから出てくるものものは、もはや正規表現ではなくて、非正規表現であるものも混じっている可能性に注意してください<sup>1</sup>。

また、正規表現には色々方言があります。今回の内容であれば、おそらくどの方言であっても実現出来ると思います。しかし僕の知識がそこまでカバーしていないので、Perl 拡張正規表現と Vim 互換正規表現において、重要な概念であるゼロ幅マッチ(Zero Width Match)と量指定子(Quantifiers)について書きます。

## 表記

正規表現は<正規表現>という形で表記します。また解説では Perl 拡張正規表現の方を用い、脚注に Vim 互換正規表現を記述する形にします。

文章中には、正規表現の検索・置換対象を文字列と表記しますが、Vim の場合はファイルでのイメージしにくいかもしれません。この記事においては「abcdefg に対して……」と書く時はある行のみに対して検索を行なったと考えてください。

## Perl 拡張正規表現

プログラミング言語 Perl<sup>2</sup> の正規表現について書きます。Perl よりも多機能な正規表現をサポートしているのは Perl6 くらいだと思います。いずれは Perl の非正規表現にも触れてきますが、今回扱う範囲はここまで深くありません。

また、Perl の正規表現には「大文字・小文字を区別しない」など、正規表現全体に関する修飾子

修飾子	意味
/m	マルチライン、文字列を複数行として扱う。行頭や行末に対するメタ文字の挙動が変わる。
/s	シングルライン、列を一行として扱う。文字 が改行文字ともマッチするようになる。
/i	大文字・小文字を区別しない。

があります。それらの一部を軽く述べておきます。

これらはhogehoge/i<sup>3</sup> のように使えます。例えば`abcd`は `abcd` にマッチしますが、`ABCD` や `AbCd` にはマッチしません。

しかし先頭・末尾の扱いに関して、/m 修飾子がからみますが、それ以外はオマケです。

またこの内容は Perl5.10.1 の perlre<sup>4</sup> を元にしています。

---

\*1 どこまでが正規表現なのか僕にもよく分からないので、一応書いておきました。

\*2 今回扱う正規表現はバージョン 5.10 以上で動作します。

\*3 修飾子は複数同時に指定出来るので、`/hogehoge/msi` などと出来ます。また修飾子の重複も認められるので、`/py/iiiiiiii` なども OK です。

\*4 <http://perldoc.jp/docs/perl/5.10.1/perlre.pod>

## Regex I

### Vim 互換正規表現

これは有名なエディタ Vim<sup>\*5</sup> で用いられるだけあり、Perl ほどではありませんが充実した正規表現を備えています。Vim はメタ文字<sup>\*6</sup> の度合いを指定出来ます。これに関しては、デフォルトである *magic* の状態で話しを進めます。vimdocja<sup>\*7</sup> を参考にしました。

### 正規表現の実行

一番楽なのは Perl のワンライナー<sup>\*8</sup> を利用する事だと思います。Linux 系 OS や Mac であれば、ターミナルに、

```
$ perl -e 'qq|<文字列>|=~/<正規表現>/<修飾子> and print $&."¥n"'
```

と入れると、正規表現にマッチした文字列全体が表示されます。

### 前提知識

まずはコンピュータリテラシで扱う程度の正規表現<sup>\*9</sup> を表に並べます。これらは egrep などを対象にしたものなので、Perl や Vim になると若干意味が変わります。違いについては後々書いていきますので、とりあえずこれでいいです

意味	Perl表記	Vim 表記
改行を除く任意の一文字	.	.
直前の正規表現を0回以上繰り返し	*	*
直前の正規表現を1回以上繰り返し	+	¥+
直前の正規表現が1回または0回	?	¥=
グループ化	(?: )	¥( ¥)
文字クラス	[ ]	[ ]
否定文字クラス	[^ ]	[^ ]
選択		¥
単語の先頭	無し	¥<
単語の末尾	無し	¥>
行頭	^	^
行末	\$	\$

\*5 今回は MacVim Custom Version 7.3 KaoriYa 20110801 で動作を確認しました。

\*6 very *magic*, *magic*, *nomagic*, *very nomagic* の四段階によるメタ文字の指定があります。

\*7 <http://sites.google.com/site/vimdocja/pattern-html>

\*8 たった1行だけのプログラムです。Perl のワンライナーについて詳しく知りたい場合は 404 Blog Not Found の記事 (<http://blog.livedoor.jp/dankogai/archives/51026593.html>) を参考にするとよいです。

\*9 <http://www.coins.tsukuba.ac.jp/~yas/coins/literacy-2011/2011-06-24/index.html>

### ゼロ幅マッチ (Zero Width Match)

正規表現では一般的に検索対象の文字にマッチするのですが、ゼロ幅マッチは位置にマッチします。若干厳密にマッチの位置を示します。

マッチ位置	Perl表記	Vim 表記
単語 <sup>*10</sup> の先頭	無し	¥<
単語の末尾	無し	¥>
単語 <sup>*11</sup> の終端または末尾	¥b <sup>*12</sup>	無し
文字列（ファイル <sup>*13</sup> ）の先頭	¥A	¥%^
文字列（ファイル）の末尾	¥z	¥%\$
文字列（ファイル）の先頭か改行直後	^	^*14
文字列（ファイル）の終端か改行直前	\$	\$ <sup>*15</sup>

Vimについてはこの表を全ての場合に当てはめて良いですが、Perl拡張の場合は注意が必要です。  
`/m`修飾子をつけない場合、`^`、`$`は改行の直前・直後にマッチせず、`¥A`、`¥Z`と同じ扱いを受けます。ですが正規表現に慣れていない人の事を考えると、改行の直前・直後にもマッチするかしないかを明瞭にするためにも、文字列の先頭・末尾を扱う時は`¥A`や`¥Z`を使うのがよいと思います。

ここで`¥A`と`^`は同じ意味です。しかし`$`と`¥Z`とは違う意味になります。何が違うのかというと、`¥Z`は完全に文字列終端の位置にマッチしますが、`$`は文字列終端が改行文字だった場合、改行文字の直前の位置にマッチします。例えば、`abcdef¥n`に`/$`をあてると、`.`がマッチするのは`f`です。ところが`¥Z`とすると、`.`は改行文字にマッチしないのでこの正規表現は失敗します。

なのでPerl拡張正規表現の`/m`修飾子を指定した場合、文字列の終端または終端の改行文字直前にマッチさせたい場合は`¥Z`を用いるとよいです。

---

\*10 Vimにおける単語とは *iskeyword* オプションで指定された文字による文字列を指します。

\*11 Perlにおける単語の定義は正規表現の修飾子によって変化します。このあたりはPerl拡張正規表現の仕様 (<http://perldoc.perl.org/perlrecharclass.html#Backslash-sequences>) に書かれている`¥w`の *Word characters* が単語の定義ですのでそれを確認してください。

\*12 `¥b`は文字クラス内で使うとバックスペースを表すので注意が必要です。基本的にゼロ幅マッチは文字クラス内では使えません。

\*13 これはVimで編集中のファイル全体や、選択部分に対しても適応出来ます。Perlがファイルに対して正規表現を実行する事はないので、このファイルという表記はPerlには関係ないです。

\*14 `^`は正規表現の先頭で使った場合のみメタ文字として扱われます。つまり、正規表現の途中で用いるとただの`^`を表わしてしまいます。正規表現のどこでも使える`¥_^`というものがありますが、後述する、絶対にマッチしない正規表現を生み出す可能性が上がるのをお勧めしません。正規表現の先頭でただの`^`にマッチさせたい場合は`¥^`とすればOKです。

\*15 `$`に関しても正規表現の末尾、選択の終端（`¥`の直前）、グループの終端（`¥`の直後）、改行文字の直前（`¥n`の直前）でのみ末尾を示すメタ文字として扱われ、それ以外の場所ではただの`$`として扱われます。`¥_¥`という位置に制限されない方もありますが、絶対にマッチしない正規表現を生み出す可能性が上がるのを理由からお勧めしません。ただの`$`は`¥$`とすればOKです。

## Regex I

ゼロ幅マッチは非常に便利ですが、絶対にマッチしない正規表現を生産する可能性を孕みます。例えば、`/hogehoge$foofoo/`<sup>16</sup>に対し、`hogehoge\nfoofoo`がマッチするかどうか考えます。

結論はマッチしません。`hogehoge`が改行位置の直前にあるのはいいですが、`foofoo`が改行位置の直後にあるわけではないです。改行位置の直後にあるものは、改行文字です。よってこの正規表現はどんな文字列にもマッチしません。

### 先後読み (Look Around)

ゼロ幅マッチの一つ、つまり位置にマッチする正規表現です。とりあえず一覧を載せます。

意味	Perl表記	Vim表記
肯定先読み	(?= hoge)	\( hoge \)\@=
否定先読み	(?! hoge)	\( hoge \)\@!=
肯定後読み	(?<= hoge)	\( hoge \)\@<=
否定後読み	(?<! hoge)	\( hoge \)\@<!

これらは先（後）に任意の正規表現がある（ない）位置を表します。例えば`/foo (?=bar)/`<sup>17</sup>は、直後に `bar` がある `foo` という文字列にマッチします。

ただし、この先後読みはゼロ幅マッチである事に注意が必要です。ゼロ幅マッチでも言いましたが、割りとひつかかるのでもう一度書きます。

ここで`/foo (?=bar)foo/`<sup>18</sup>に対して、`foobarfoo`がマッチするか考えてみます。

結論から言うと、マッチしません。これは`(?=bar)/`がマッチする位置を考えると、`foo|barfoo`です。`|`の直後に `foo` が存在しなくてはならないので、この正規表現はマッチしません。

### 正規表現の論理積 (AND)

これはかなり強引な手法ですが、一応先読みで実現出来るので紹介します。論理積とはかつですね。つまり、複数の正規表現に同時にあてはめたい時にも、先読みを使うことで出来ます。

ちなみに論理和は選択 (|) で出来ます。

例えば、次の要件を満す正規表現を考えます。

- どこかに `foo` を含む
- どこかに `bar` を含む
- どこかに `bazz` を含む

なんとなく`/foo.*bar.*bazz/`<sup>19</sup>などとやればいいそうですが、これだと `foo`、`bar`、`bazz` の順番が固定されてしまうので駄目です。選択を使って、次のように順番を全部カバーする方法もあります。

`/foo.*(?:bar|bazz|bazz|bar)|bar.*(?:foo|bazz|bazz|foo)|bazz.*(?:foo|bazz|bazz|foo)/`<sup>20</sup>

見て分かるように色々な意味で最悪です。

---

\*16 `/hogehoge$\$foofoo/`

\*17 `/foo$(bar$)\@=/`

\*18 `/foo$(bar$)\@=foo/`

\*19 `/foo.*bar.*bazz/`

\*20 `/foo.*$(bar.*bazz$bazz.*bar$)$|bar.*$(foo.*bazz$bazz.*foo$)$|bazz.*$(foo.*bazz$bazz.*foo$)/`

先読みを用いる、`/(?=.foo)(?=.bar)(?=.bazz)/21` で OK です。

`.`は改行を除く任意の文字列を表すので、これら三つの先読みは全て対象文字列の先頭にマッチすることになります。

例えば`(?=.foo)(?=.bar)(?=.bazz)`を `hoge foo hoge bar hoge bazz`に対して用いた場合、`(?=.foo)`は `hoge foo` の先頭の位置にマッチし、`(?=.bar)`は `hoge foo hoge bar` の先頭の位置、`(?=.bazz)`は `hoge foo hoge bar hoge bazz` の先頭の位置

今回は肯定先読みを使いましたが、否定先読みを使って「全ての条件を満さない」や「これは満すけどこれは満さない」を作れます。

例えば次の条件を満す正規表現を考えます。

- どこかに `foo` を含む
- どこにも `bar` を含まない

この場合`(?=.foo)(?!.*bar)22`という感じになります。

### 量指定子 (Quantifiers)

量指定子とは最初の表にもあった、`*`や`+`、`?`などのことです。文字通り、直前の正規表現の量を指定するメタ文字です。これら以外にも、次のような量指定子があります。

意味	Perl 拡張	Vim 互換
ちょうどn回	{n}	¥{n}
n回以上	{n,}	¥{n,}
n回以上、m回以下 <sup>23</sup>	{n,m}	¥{n,m}

### 貪欲な量指定子 (Greedy Quantifiers)

ここで例えば、直前の正規表現が 1 回または 0 回 (?) を使って次の場合を考えます。`abcde` に `/abcde?/24` を行うと、マッチするのは `abcde` か `abcd` かどちらになるでしょうか。

この場合、`?`の直前にある `e` が 1 回（ある）または 0 回（ない）なのでどちらでもよさそうです。しかし Perl 拡張、Vim 互換とともに、`abcde` にマッチします。その理由は正規表現における一般的な量指定が**最長一致**だからです。

最長一致は出来るだけ長くなるようにマッチします。例えば`abc*`に `abcccccc` を与えると、`*`は貪欲になるべく長くマッチしようとし、結果 `abcccccc` にマッチします。

ちなみに上の表で紹介した量指定子は全て最長一致です。

\*21 ¥(.foo¥)¥@=¥(.bar¥)¥@=¥(.bazz¥)¥@=/

\*22 ¥(\*foo¥)¥@=¥(\*bar¥)¥@!/

\*23 理由は謎ですが、Perl では `n`、`m` に 32767 以上を入れるとエラーで終了しました。ならば `n` 回以上 (`{n,}`) は、実質は `n` 回以上 32766 回以下なのかと思いましたが、どうやらそうではないようです。また Vimにおいて同様の実験したところ、`n` や `m` に 50000 などを入れても普通に動作しました。

\*24 /abcde¥=/

## Regex I

### 怠惰な量指定子 (Lazy Quantifiers)

最短一致はその逆で、なるべく短くなるようにマッチします。最短一致するメタ文字の一部を次の表に紹介します。

意味	Perl 拡張	Vim 互換
直前の正規表現を 0 回以上繰り返し	*?	¥{-}
直前の正規表現を 1 回以上繰り返し	+?	¥{-1}
直前の正規表現が 1 回または 0 回	??	¥{-0,1}
n 回以上	{n}?	¥{-n}
n 回以上、m 回以下	{n,m}?	¥{-n,m}

こうして見ると Perl 拡張は?をつける、Vim 互換は{}と-を使うという感じに統一されています。よって、*abcccc* に/*abc\**?/<sup>25</sup> を使った場合、マッチする部分は *ab* になります。

### 強欲な量指定子 (Possessive Quantifiers)

最短一致の量指定は「出来るだけ短く」と、貪欲さを抑圧していました。今回は貪欲さをパワーアップして、**強欲**にしてみましょう。

強欲な量指定子は**絶対最大量指定子**などといわれ、次の表で示します。

意味	Perl 拡張
直前の正規表現を 0 回以上繰り返し	*+
直前の正規表現を 1 回以上繰り返し	++
直前の正規表現が 1 回または 0 回	?+
n 回以上	{n}+
n 回以上、m 回以下	{n,m}+

Vim 互換に関しては、正規表現のグループ直後に¥@>を置きます。例えば/\*+/を Vim 互換で書くと、¥(\*¥)¥@>となります<sup>26</sup>。

これが何の役に立つかというと、正規表現の高速化に若干貢献します。

前述した否定先読みを使って、確実に失敗する/a+b? (?!)/<sup>27\*28</sup> を考えてみます。

これは否定先読みがその後に続くあらゆるものを否定するので、確実に失敗します。この正規表現を *aaab* に適応した場合、どういう経緯で失敗するのか見てみます。基本的に全部(?!))で失敗しますが、+と?がどのように変化するか見てみます。

---

\*25 /abc¥{-}/

\*26 Vim 互換の¥@>に相当する Perl 拡張に(?>)というものがあります。なので先程の正規表現は、/(?>.\*)/と同じです

\*27 /a+b?¥(¥)¥@!/

\*28 Perl の場合は(?!))の代わりに、(\*FAIL)を使う方が読みやすいです。実質的に同じ意味ですし、(?!))は内部的に(\*FAIL)に最適化されます。

試行順	マッチした部分	$n - m^{*29}$	備考
1	aaab	1 - 4	+ や?が先頭から末尾までマッチした
2	aaa	1 - 3	?がbを捨ててみた
3	aa	1 - 2	+ が一個aを捨てた
4	a	1	さらにaを捨てた
5	aab	2 - 4	開始地点を二文字目に変更
6	aa	2 - 3	bを捨てた
7	a	2	aを一つ捨てた
8	ab	3 - 4	開始地点を三文字目に変更
9	a	3	bを捨てた

通常、量指定子はこの様に終端をずらす事でなんとかマッチしないかと頑張ってしまいます。ちなみに始点をずらしているのは正規表現のエンジンです。

とりあえず、+を絶対最大量指定子の++にしてみます。

試行順	マッチした部分	$n - m$	備考
1	aaab	1 - 4	全部
2	aaa	1 - 3	?がbを捨てた
3	aab	2 - 4	開始地点を二文字目に変更
4	aa	2 - 3	?がbを諦めた
5	ab	3 - 4	開始地点を三文字目に変更
6	a	3	bを諦めた

さらに、?も強欲な?+にしてみます。

試行順	マッチした部分	$n - m$	備考
1	aaab	1 - 4	全部
2	aab	2 - 4	開始地点を二文字目に変更
3	ab	3 - 4	開始地点を三文字目に変更

\*29 nからm字目がマッチしたということです。ただし、先頭の文字を1文字目として扱います。

## Regex I

絶対最大量指定子は一度確保した部分を諦めないので、このようになります<sup>\*30</sup>。

絶対最大量指定子の動きを紹介したところで、もう少し実用的なものを考えます。例えば、HTML の font タグの中身を探す/*font[^>]\*>.+<\\$font>*<sup>\*31</sup>を考えます。

これに*font abcdefghidk ..... 「>」以外の文字が一万文字くらい .....*などと、確実に失敗する文字列を与えます。

確実に失敗するのですが、ここで[>]\*はまず文字列の最後まで飲み込みます。*>*はないので最後の一文字を諦めて、また*>*を探し失敗と、「失敗→一文字諦める」をひたすら一万回くらい繰り返します<sup>\*32</sup>。

この一万回くらいの繰り返しが無駄です。いくら繰り返したところで成功するはずがないからです。ここで絶対最大量指定子に変更してみます。正規表現は/*font[^>]\*>.+<\\$font>*<sup>\*33</sup>です。こうすると、[>]\*は最後まで飲み込み、当然最後には無いので失敗します。

失敗しても絶対最大量指定子は確保した部分を諦めないので、そのままマッチに失敗します。このように、上手く使えば無駄な処理を省けます。これは特に選択と組み合わせると効果抜群です。

### あとがき

今回は序論ということでそこそこの内容しか触れませんでした。次の機会には正規表現の最適化やバックトラックについて触れるかもしれません。

何か扱って欲しいテーマなどがあれば、アンケートなどで教えてください。

---

\*30 正規表現の適応開始位置が変更されると、今までのバックスラッシュが破棄されるので、このような結果になります。先頭や末尾などのゼロ幅マッチを活用して、マッチ開始位置を指定することは正規表現の高速化にも貢献します。

\*31 /*font[^>]\*>.+<\\$font>*/

\*32 開始位置を変更して再度試行を開始しますが、今回の正規表現は正規表現の開始がくという不变の文字なので、失敗の判断にそれほど時間はかかりません。

\*33 /*font\\$([>]\*\\$)@\>.+<\\$font>*/

## tmux

# GNU screenからtmuxへ

文 編集部 Flast

### GNU screen

みなさんは GNU screen(以下 screen)を使っているでしょうか。多くのディストロで既に入っているか、パッケージが用意されているので導入コストは非常に低いと言えるでしょう。サーバ管理者などは ssh セッションが切れても大丈夫なように使っていると思います。

しかしデフォルトでは常用するには耐えられません。なぜなら今ウィンドウをいくつ開いているのかが見えないからです。もちろん.screenrc を書けば良いのですがシェルのプロンプトの設定並にわかりづらいです。

### tmux

そんな screen ですが、最近対抗馬となる tmux(ティーマックス)というソフトウェアが注目を集めています。

tmux は sourceforge<sup>\*1</sup> 上で BSD ライセンスのもと公開されている OSS です。screen で用意されている基本的な機能は網羅しており、もちろんデタッチ/アタッチも出来ます。特にステータスラインがデフォルトでそれなりに使える<sup>\*2</sup>と思うので便利です。

また、screen では CVS の trunk のみで使用可能な左右分割<sup>\*3</sup>がリリースバージョンで利用出来ます。画面分割系は screen のそれよりも強力なので、man 等をしっかり読むと面白いと思います。

### 基本

tmux の rc ファイルは ~/.tmux.conf です。このファイルに各種設定を書くことで実行時に適用されます。tmux の操作はデフォルトで Emacs バインドですが vi バインドを希望される場合は ~/.tmux.conf に

```
setw -g mode-keys vi
```

と加えてください。

tmux を実行すると、最下段にステータスラインが表示された状態で、シェルが待機していると思います。この状態では通常のシェルと同じですが、<prefix>キー(デフォルトでは<C-b>)を叩くと tmux が入力を受け取るようになります。まずは<prefix>?を押してみてください。現在 tmux がバインドしているキーとそのコマンドの一覧が表示されます。

以下の様な一覧が出力されると思います。

---

\*1 <http://tmux.sourceforge.net/>

\*2 vim を起動したらちゃんと vim と表示してくれます。

\*3 もっとも、多くのディストロで提供されてるパッケージは trunk のパッチが当たってたりしますが。

C-b: send-prefix  
 C-o: rotate-window  
 C-z: suspend-client  
 ...

[44/44]

コロンの左側が<prefix>に続くキーで、右側がそのキーを押すことで実行されるコマンドです。コマンドは <prefix>:でステータスラインがコマンドラインになるので、そこで実行することも出来ます。

また、各バインドの移動方法でカーソルを動かすことが出来ます。右上に [44/44] のように出ているかと思いますが、これは 【現在の行/バックログの行数】 です。現在どの辺を見ているのかがわかるので便利です。

Emacs バインドでは<C-w>を、vi バインドでは<Enter>を叩くことで通常のシェルへもどることが出来ます。これはコピー mode と呼ばれるモードでも使用されます。

コピー mode とは過去の出力を閲覧、コピーすることの出来るモードです。<prefix><C-[>でコピー mode に入ることが出来ます。コピー mode 中では各バインドの移動方法でカーソルを移動させ、任意の範囲をコピーすることが出来ます。

Emacs バインドでは<C-Space>で先頭をマークし、<C-w>でコピー mode を抜けると同時に取り込みます。vi バインドでは<Space>でマーク、<Enter>で取り込みます。どちらのバインドでも<prefix><C-]>で取り込んだものをペーストすることが出来ます。

通常では最後に取り込んだものがペーストされますが、過去に取り込んだものもペーストする choose-buffer コマンドなどが用意されており、様々な操作を行うことが出来ます。ペーストバッファの操作周りについては man で tmux(1) の BUFFERS を参照してください。

### アタッチ/デタッチ

screen や tmux はアタッチ/デタッチと呼ばれる機能を持っています。これはターミナルを終了しても、デタッチしたセッションは終了せずに裏で動いていて、別のターミナルからセッションに接続するとそれまでの出力が復元されるという機能です。

このアタッチ/デタッチがもっとも活躍する場所は、回線の細い環境や移動している状態で ssh 等でサーバに接続をしていたときです。細い回線が切れたりトンネルなどに入ることで通信できなくなると ssh のセッションが切れ、たとえ実行途中のものでも強制的に終了されてしまいます。

そんなとき、screen や tmux を実行していると不意な切断でも再度サーバに接続することで途中から作業を再開することが出来ます。

デフォルトでは<prefix>d でデタッチすることが出来ます。もちろん screen にあったオートデタッチも搭載されています。アタッチするには tmux の実行時に

```
$ tmux attach
```

で直前のセッションにアタッチ出来ます。

全てのセッションをリストアップするには

## tmux

```
$ tmux list-sessions  
0: 1 windows (created Sat Sep 10 19:01:09 2011) [80x23]  
1: 1 windows (created Sat Sep 10 19:01:11 2011) [80x23]
```

とするとセッションの一覧が見られます。一番左の数字がセッション番号です。

複数セッションがある場合、

```
$ tmux attach -t n
```

とすると n 番のセッションにアタッチ出来ます。

### 画面分割

screen では画面のレイアウトがあり、そこに任意のものを表示してました。当然左右で同じものも表示出来ました。

一方 tmux は screen と違い、ウィンドウとペインという概念があります。ステータスラインにはウィンドウの一覧がリストアップされていますが、各ウィンドウの中にペインがあります。通常ペインはウィンドウ全体を占有しますが、ウィンドウを分割し複数のペインでウィンドウを共有することが出来ます。

ただし、あるウィンドウに属しているペインは別のウィンドウへは移動出来ません。screen とは若干違うので、screen に慣れてる人は最初戸惑うかもしれません。このウィンドウとペインという概念ですが、文章で書かれても多分難しいので、実際に手を動かして見ると良いと思います。

また、screen では画面のレイアウトはすべて共通でしたが、tmux では各ウィンドウ毎にレイアウトを持っているので、ウィンドウ毎にペインを好きな配置で表示することが出来ます。

ウィンドウを左右に分割するには<prefix>%で、上下に分割するには<prefix>"でそれぞれ可能です。新たにウィンドウを作るには<prefix>c で可能です。

ウィンドウ間の移動は<prefix>n か<prefix>p で出来、ペイン間の移動は<prefix>とカーソルキーで出来ます。

### キーバインド

tmux には screen にない unbind<sup>4</sup> があります。これは非常に便利です。例えばデタッチするには<prefix>d と説明しましたが、これは非常に押し間違えそうで危ないので筆者は unbind して<prefix><C-d> に変更して使っています。

デタッチを<prefix>d から<prefix><C-d>に変更してみましょう。以下の設定を~/.tmux.conf に書き加えてみてください。

```
unbind d  
bind C-d detach-client
```

\*4 登録されているキーバインドを削除するコマンド

1 行目で既存の<prefix>d を unbind しています。これで<prefix>d を入力してもデタッチされることはできません。私の知る限り、screen では別のコマンドで上書きするしかありませんでした。

2 行目で<prefix><C-d>にデタッチを bind します。これで<prefix><C-d>を入力することでデタッチすることが出来ます。

あとは好みのキーバインドを書いていけば良いです。どのようなコマンドがあるのかは man で tmux(1) の COMMANDS 以下の項目を読んでください。

ところで今まで<prefix>は変更していないので<C-b>のままだと思いますが、unbind-bind ではこれを変更することは出来ません。私はいつも<C-z>を<prefix>として常用しているのでこれに変更しようと思います。と、同時に<C-z><C-z>で<C-z>を送るように変更しましょう。

```
set-option -g prefix C-z
unbind C-b
bind C-z send-prefix
```

1 行目で<prefix>を<C-z>に設定しています。それまでの設定を上書きするのでこの時点で<C-b>を入力しても<prefix>として食われることはありません。

2,3 行目で<C-z>を送る設定をしています。もちろん別のキーで<prefix>を送るように設定することも出来ます。

## 面白い機能

tmux は面白い機能も搭載しています。

### ・ペインの移動

レイアウトを維持したままペインの表示する位置を変更することができます。

試しにいくつか縦横にペインを開いて<prefix><C-o>か<prefix><M-o>を入力してみてください。ペイン全体がローテーションする様子がわかると思います。

<prefix>{か<prefix>}はペイン全体を動かすのではなく、現在選択しているペインを他のペインと入れ替えます。

### ・時計

<prefix>t で現在のペインに時計を表示することができます。しかし、時計を表示している間は現在のペインは隠れてしまいます。誰得でしょうか。

### ・choose-session コマンド

これはデフォルトでどのキーにも割り当てられていませんが、デタッチすることなく別のセッションに接続することができます。

<prefix>:choose-session<CR>と入力すると現在存在するセッションのリストが表示されます。各キーバインドの上下で選択し、<Enter>で選択したセッションへ接続することができます。

## まとめ

tmux 使いましょう。

# 本格的♂イタリアン

文 編集部 IX

**What kind of foods is "Italian"?**  
**Is it spaghetti? or yakisoba?**  
**"Italian" is neither.**  
**"Italian" is "Italian" !!**

～「みかづき@WEB」トップページより抜粋～

## ■ 前書き

皆さん！イタリアンはお好きですか？  
好きですよね！  
年に一回はイタリアン食わないとだめな感じですよね！  
可愛い！イタリアンちゃんジャンクフード可愛い！  
イタリアンちゃんジュルジュルペロペロペロれ……  
え？ご存じない！？  
それ以上……じゃなかった、それはいけない！  
速く通達せねば！イタリアンを！  
イタリアン(新潟)を！迅速に！この常陸の国の民達に！

■ イタリアンとは

イタリアンとは、ソース焼きそばにトマトソースをかけた新潟名物のことです。新潟市周辺に店舗を展開する「みかづき」で食べることができます。

公式サイトでは、以下の様な説明が述べられています。

- ・麺の見た目や歯ごたえは焼きうどんのよう……でも、中華麺なので焼きそばなのです。
- ・なぜかトマトソースが合うんですよ。
- ・フォークで食べるのもおもしろいでしょ？
- ・他にも定番ではカレーソース・ホワイトソースなどがあり、季節によって別バージョンが出ます。
- ・もやしとキャベツがたくさん入ってるのでビタミン・ミネラル・食物繊維などが多く含まれていて、特に女性のお客様に喜ばれているようです。
- ・毎日食べても飽きない味……という点でも、イタリアンかもしれませんね(イタリアの日常食、つまりイタリア料理って、毎日食べても飽きないそうですから)。

なんか風雲拳<sup>1</sup>みたいな食べ物だな……

■ イタリアンを食べに行こう！

本場のイタリアンを食べずに、  
私は大学生活を終えられるだろうか、  
いや終えられない。

そう思った私は、去る 2011 年 8 月 20 日、とあるポケモンチャットのオフ会に参加するという名目で新潟に向かいました。オフ会は和気藹々と進行し、お互いに対戦したり、漫画の原稿を見せ合ったり、カラオケを行ったりと、実に充実した内容でした。

そして、14 時過ぎにお昼を食べようと言うことになりました。どうやら行き先ははっきりと決まってない模様。ここで言わねばどうにもならんよなあ！

なあ……イタリアン食いにいこうや……

---

\*1 風雲拳：格闘ゲーム風雲默示録の主人公、ショー・疾風が用いる格闘スタイルのこと。「風雲拳……それは、実戦空手道とブーメランをくみあわせた、まったく新しい格闘技である。」という台詞は一部で局地的にあまりにも有名。

## イタリアン(新潟)



さあ、やって参りました、みかづき万代店。夜行バスで 7 時間、オフ会前半戦で 7 時間、合計 14 時間を経て、愛しのイタリアンちゃんにようやく会うことができます。

中に入ってみると日本に上陸したばかりの時のド○ド○バーガーのような雰囲気でした。「あれ?ここだけ時が止まってるんじゃね?」的な印象を受けました。

カウンターで注文してから約 10 分、イタリアンちゃんとの感動のご対面です!



図2 ノーマルイタリアン大盛り

ホントにソース焼きそばの上に  
トマトソースがかかつてやがる……



図3 フライドポテト

なぜかポテトまでついて来やがる……



図4 メロンソーダ

## わざとらしい……メロン味……

実に美味しそうですね！早速食べてみましょう！

まずは、イタリアンの脇に添えてある沢庵らしき物体を一口……**ショウガだこれえ！**

味はまさに牛丼にのっている赤いアレ。公式サイトによると紅ショウガではなく白ショウガなのがミソだそうで。

いきなり出鼻をくじかれたので、いったんポテトに逃げよう。ケンタッキー系統の太めのポテトを一口……**ポテトだこれえ！**

味はまさにジャガイモ。揚げたてサクサク、実に食べ応えのあるフライドポテトでした。

塩気にやられてのども渴いてきたので飲み物を飲もう。明らかに自然界に存在しないタイプの緑色液体を一口……**メロンソーダだこれえ！**

味はまさに典型的かつ伝統的なメロンソーダ。メロン味ではなくメロンソーダ味でした。

さて、いよいよイタリアンちゃんとのファーストコンタクト。  
トマトソースとソース焼きそばを分けて食べるなんてまどろっこしいことなんてせずに豪快に一口……**イタリアンだこれえ！**

味はまさにイタリアン。極太麺にソースとトマトソースが絡み、独特の味を醸し出しています。味に関して、いい意味で裏切られました。粉チーズで味付けされているとのことですが、チーズの風味は前面には出ておらず、全体の味をまとめてくれているような印象でした。

ぶつとい麺にしやきしやきのモヤシ、濃いめの味付け……どこかで食べたことがあるような…  
**二郎<sup>2</sup>だこれえ！**

なるほど、どうりで私の口に合ったわけだ。ジャンクフード万歳！

## **■まとめ**

**イタリアン = 脂分を減らし洋風要素を追加した二郎系焼きそば**

私が実際に食べてみて、導き出した結論このようになります。正直、思っていたより美味しかったです。しかも大盛りにしても 390 円<sup>3</sup> と実にリーズナブル。いい意味で正統派ジャンクフードでした。

今回紹介した以外にも、イタリアンが入っているプラスチック容器は自社製、誕生から半世紀経っている、家庭版イタリアンがついに発売されたなど、イタリアンに関するエピソードはまだ絶えません。

新潟を訪れる際は、是非イタリアンを食べに「みかづき」を訪れてみてはいかがでしょうか？

---

\*2 二郎：ラーメン二郎にて提供されるラーメンのこと。豚骨醤油ベースのスープに極太麺、大量の野菜、背脂、ニンニク、(チャーシューではなく)豚によって構成される。今流行りの G(がつり)系ラーメンの開祖的存在。「二郎はラーメンではなく二郎という食べ物だ。」という言葉は割と有名。

\*3 390 円：ノーマルイタリアン(320 円)+大盛り(70 円)。今回はポテトセット(180 円)もつけたので合計で 570 円でした。

## 全自動ふあぼ bot プロジェクト

# 全自動ふあぼbotプロジェクト

文 編集部 zer0day

## 序文

こんにちは。zer0dayです。

情報科学類のみなさんには、Twitterはかなりなじみ深いものと思われますが、そんなTwitterにこのたび新たなbotが出現しました！

その名も、「湯猫bot(@B\_Otsuki)」。

かつてふあぼ魔・RT魔として猛威をふるったあの「湯猫<sup>\*1</sup>」が、botになって帰ってきました！

さて、この湯猫botですが、いったいどういう仕組みで、何をするんでしょうか。

## 機能

主な機能として、湯猫botはまず、ふあぼります<sup>\*2</sup>。とことんふあぼります。でもただふあぼるだけではありません。先述のとおりこのbotには元ネタとなる人物が存在するのですが、彼の振る舞いをそっくり再現することを目標としています。このbotがふあぼる対象のツイートは本人が過去にふあぼったツイートを解析して、いかにも彼がふあぼりそうなツイートをヒューリスティックにふあぼります。それ以外にも、リプライチェーン<sup>\*3</sup>を丸ごとふあぼる、観測したリツイートをまわす、しゃべるなどといった本人の行動に基づいた機能も搭載しております(一部予定)。

これはいかにして実現されているのでしょうか。

## しくみ

湯猫botはPerlで書かれており、モジュールにNet::Twitter, AnyEvent::Twitter::Stream, MeCab, DBIを採用しております。それぞれの役目は以下の通りです：

- Net::Twitter
  - PerlでTwitterAPIをペロペロする<sup>\*4</sup>ためのライブラリ
- AnyEvent::Twitter::Stream
  - Userstreamを使うためのAnyEvent<sup>\*5</sup>ベースのライブラリ。このbotはUserstreamに対応しており、その反応速度は半端ではありません
- MeCab
  - 日本語のツイートの解釈に不可欠である自然言語処理プログラム「MeCab」のラッパ
- DBI
  - SQLを操作するためのライブラリ

これらは大部分が標準でインストールされていないので、以下を自分で実践する場合は各自CPANやaptなどを使ってライブラリをインストールしましょう。

---

\*1 あえてご本人様のIDは出しません。

\*2 「お気に入りに追加」の俗語。なぜか虜になる人が多い。

\*3 会話の履歴のこと。

\*4 APIを「叩く」という暴力的な表現は嫌いなので、こう言い換えました。

\*5 Event-drivenなプログラムを組むときに役立つライブラリ。

## 実践

ではまず、Userstream でタイムラインを取得してみましょう。

```
1#!/usr/bin/perl
2
3use strict;
4use warnings;
5use Encode;
6use utf8;
7use AnyEvent::Twitter::Stream;
8
9my $done = AnyEvent->condvar;
10
11my %keys = { #OAuth keys を下の各フィールドに代入してください
12    consumer_key      => "",
13    consumer_secret => "",
14    token             => "",
15    token_secret     => "",
16};
17
18my $stream = AnyEvent::Twitter::Stream->new(
19    %keys,
20    method   => "userstream",
21    on_tweet => sub {
22        my $tweet = shift;
23        my $id   = $tweet->{id};
24        my $user = $tweet->{user}{screen_name};
25        my $text = decode_utf8( $tweet->{text} );
26        print '@' . $user . ":" . $text . "\n" if ($id);
27    },
28    on_error => sub {
29        my $error = shift;
30        warn "Error: $error\n";
31        $done->send;
32    },
33    on_eof => sub {
34        $done->send;
35    },
36);
37
38$done->recv;
```

## 全自動ふあぼ bot プロジェクト

%keys の中身は各自自分の OAuth キーを取得し代入してください。

とくに各部の説明もせずにさらっと書きましたが、これで、Userstream による TL 取得結果が標準出力に流れできます。

でもそれだけでは受動的すぎるので、ツイートを取得し次第一瞬でふあぼる馬鹿極まりないスクリプトを書いてみましょう。

以下に先述のソースコードに追加する差分<sup>\*1</sup> を示しました。

```
7  use Net::Twitter;  
  
17 my $actions = Net::Twitter->new(  
18     %keys,  
19     access_token      => $keys{token},  
20     access_token_secret => $keys{token_secret},  
21     traits            => [qw/API::REST OAuth WrapError/],  
22 );  
  
33     $actions->create_favorite($id);
```

Net::Twitter でふあぼの API をペロペロします。このクラスを\$actions に呼び出したので、\$actions を通じて Twitter の API のひとつである create\_favorite メソッドにアクセスできるようにしました。

なぜ %keys を引数にとったのに 19-20 行目が必要かというと、Net::Twitter と AnyEvent::Twitter::Stream とでは引数の名前が違うからです。ちょっと面倒。

無差別ふあぼ爆撃にぴったりですね。ではこのふあぼ魔スクリプトに少し知能を持たせてみましょう。

以下は特定キーワードを探し当ててふあぼるスクリプトです。

```
29 my $stream = AnyEvent::Twitter::Stream->new(  
30     %keys,  
31     method   => "userstream",  
32     on_tweet => sub {  
33         my $tweet = shift;  
34         my $id    = $tweet->{id};  
35         my $user  = $tweet->{user}{screen_name};  
36         my $text  = decode_utf8( $tweet->{text} );  
37         print '@' . $user . ":" . $text . "\n" if ($id);  
38  
39         my @keywords = qw/湯猫 fav_cat 大槻 ふあぼ 加古川/;  
40         foreach my $keyword (@keywords) {  
41             if (    $text =~ decode_utf8($keyword)  
42                 and $tweet->{favorited} == 0 )
```

---

\*1一番最後に示すソースコードに全体の完成図が載っておりまます。

```

43      {
44          $actions->create_favorite( $tweet->{id} );
45          last;
46      }
47  }
48 },

```

39 行目にある配列@keywords 内の語句に反応してふあぼります。

ついでにリツイートをまわす機能も実装してみましょう (39-40 行目)。

以下はこれまでに説明したことすべて盛り込んだ、コメント付きの完成図になります。

```

1  #!/usr/bin/perl
2
3  use strict;
4  use warnings;
5  use Encode;
6  use utf8;
7  use Net::Twitter;                      # Handles API actions
8  use AnyEvent::Twitter::Stream;        # Handles Userstream API
9
10 my $done = AnyEvent->condvar;        # Handles condition
11
12 # Put OAuth keys
13 my %keys = (
14     consumer_key    => "",
15     consumer_secret => "",
16     token           => "",
17     token_secret    => "",
18 );
19
20 # Initiate Twitter-API methods
21 my $actions = Net::Twitter->(
22     %keys,
23     access_token      => $keys{token},
24     access_token_secret => $keys{token_secret},
25     traits            => [qw/API::REST OAuth WrapError/],
26 );
27
28 # Initiate Userstream
29 my $stream = AnyEvent::Twitter::Stream->new(
30     %keys,
31     method   => "userstream",

```

## 全自動ふあぼ bot プロジェクト

```
32      on_tweet => sub {
33          my $tweet = shift;
34          my $id    = $tweet->{id};
35          my $user   = $tweet->{user}{screen_name};
36          my $text   = decode_utf8( $tweet->{text} );
37          print '@' . $user . ":" . $text . "\n" if ($id);
38
39          $actions->retweet({id => $tweet->{id}})
40              if ($tweet->{retweeted_status});
41
42          my @keywords = qw/湯猫 fav_cat 大槻 ふあぼ 加古川/;
43          foreach my $keyword (@keywords) {
44              if (    $text =~ decode_utf8($keyword)
45                  and $tweet->{favorited} == 0 )
46              {
47                  $actions->create_favorite( $tweet->{id} );
48                  last;
49              }
50          }
51      },
52      on_error => sub {
53          my $error = shift;
54          warn "Error: $error\n";
55          $done->send;
56      },
57      on_eof => sub {
58          $done->send;
59      },
60  );
61
62  $done->recv;
```

もっと知能を与えたいたいですね。次回は、ふあぼった発言の名詞と動詞の記録を使ったヒューリスティックふあぼり機能を実装し、また、某しゅ●まいやえり●ん Bot のようにしゃべらせてみましょう。

それでは。

# 情報科学類誌 WORD 読者アンケート

文・題字 編集部 ふあい

。+°'°+☆+'°'+。+°'°+☆+'°'+。+°'°+☆+'°'+。+°'°+☆+'°'+。+°'°+☆+'°'+。+°'°+☆+'°'+。

## 情報科学類誌 WORD 読者アンケート プレゼント当選者

°+,。,+☆+,。,+☆+,。,+°+,。,+☆+,。,+☆+,。,+°+,。,+☆+,。,+☆+,。,+°+,。,+☆+,。,+☆+,。,+°+

### 怪しいクリアファイル トクナガさん

### 怪しいクリアファイル トクナガさん

### 怪しいテープのり トクナガさん



WORD マスコットキャラクター・WORD 娘

## WORD 読者アンケート

### ■あいさつ

みなさん、おはこんばんちは<sup>\*1</sup>。第3回目の読者アンケート集計です。今回は**13人**の方から回答をいただきました。**そのうち1名は何も書いてありませんでした。**今号もアンケートを実施するので、面白い回答を待ってます。例によってアンケートに回答してくれた方のうち、希望する人に粗品をさしあげます。希望される方はアンケート提出の際にWORD編集部室(3C212、情報科学類生ラウンジ横の怪しい部屋)まで直接お越し下さい。特に、徳永先生<sup>\*2</sup>から送られてきた**例のグッズ<sup>\*3</sup>**がいまだに余っているので、ドンドン持っていってください。マジで。

### ■粗品一覧

目新しいものが無いので、詳しい説明は省きます。詳しい説明は前号をご覧下さい。

#### ◆例のグッズ

COJTロゴ入りのクリアファイル5枚とテープのり1つ。まだまだ余ってるよ～。

#### ◆メモリ型定規

おおよそ13cmのメモリそっくりな定規。実際にPCにさして使える(かもしれない)本格仕様。

### ■集計の前に、言い訳と訂正

前号のアンケート用紙で「6.Mac便利機能への誘い」と書くべきところを「2.Mac便利機能への誘い」と表記してしまいました。これは前行の先頭「1.表紙」に合わせて、一太郎が勝手に番号を2に変えやがったからだと推測されます。Microsoft Wordとかでもそうだけど、こういったお節介機能は本当に要らないですね。こういったお節介機能がデフォルトでオンになってるせいで、プログラミング入門のレポートを出すときに、ソースの行頭が勝手に大文字になったり、カーナビの地図が勝手にクルクルまわって「なんで大学の南西に霞ヶ浦があるんだろう？？」ってなったりするんだ！！！！！

---

\*1 おはこんばんちは：おはよう+こんにちは+こんばんは

\*2 徳永先生：私が聞いた初めての徳永先生の言葉。それは「俺は億万長者です。」でした。その声は低くてダンディーで、こんな素敵な言葉を貰える私は特別な存在だと思いました。一緒にお金も貰えたら良かったのですが。そんな徳永先生の知られざる(知らなくても良い)実態を知りたい方は、Twitterにて@cojt\_onlineをフォローするべし(しなくても良い)。そのTwitter情報によれば、地震が起きると脱ぐ習性があるらしい。某誌と違って上品さをモットーとする本誌ではとても書けないようなアレな発言をしている。専門的な事はともかく、色々あってこの先生から例のグッズが送ってきた。「とくちゃん」「COJT先生」「C某JT先生」「C棒JT先生」「C♂JT先生」「棒先生」「棒」「魔王」「魔棒」「巧い棒」「オスなが先生」「メスなが先生」「とくながおじさん」など、色々な呼び方があるので、好きな呼び方で呼ぼう。

\*3 例のグッズ：情報科学類・情報メディア創成学類3年生向けの授業「組み込み技術キャンパスOJT」の教室デザインアンケート用に、偉大なる徳永先生が道楽で作ったグッズ。COJTグッズとも言う。よほど大量に作ったのか、WORDや情報メディア創成学類誌MASTにも流れ込んできた。これをCOJTグッズ大海嘯事件と言う。COJTシャツとは何だったのか。

**■アンケート集計**

**◆ Q1:所属を教えて下さい**

情報科学類：4人

応用理工学類：1人

社会工学類：1人

生物学類：1人

新人文学類：1人

(第四学群) 二次元創成学類：1人

惑星ベジータ：1人

芝：1人

びぶんとかぎりしやもじとかつかうむずかしいがくるい：1人

色々つっこみ所はあるけど、とにかく惑星ベジータの方に読まれているというのは特筆すべき点でしょう。地球外生命体に読まれている学類誌は WORD だけです！！ びぶんとかぎりしやもじとかはじょうほうかがくるいでもつかうよ？ **芝とは一体……**

**◆ Q2:性別を教えて下さい**

- ・男：4人
- ・牡：1人
- ・非ノンケ：1人
- ・その他(男)：1人
- ・芝：1人
- ・無性生物：1人
- ・非実在少年：1人

とうとう女性読者が居なくなってしまった……。と思ったけど、ド・モルガンの法則より **ノンケの補集合は同性愛者**であることが言えるので、女性が好きな女性であると信じています。あと、その他(男)って書いた奴！！男っていう欄があつただろ！！そして**芝とは一体……**

**◆ Q3:WORD の公式サイト「WORD Press」(<http://www.word-ac.net>)はご存知でしたか？**

- ・知ってる ZE:3人
- ・知らぬぞ:5人
- ・見たことない:1人
- ・見てるけど更新間隔が長い：1人
- ・404 Not Found : 1人
- ・芝:1人

更新間隔長くてごめんね～。~~教育上よろしくない出来事しかない~~書くような出来事が無いんです。たまによく結構稀にサーバが落ちるので、404 になることがたまによく結構稀に有ります。

**そして芝とは一体……**

## WORD 読者アンケート

### ◆ Q4:良かったと思う記事があれば教えて下さい

前述の通り、一太郎君のお節介機能によって 2 という回答が目次なのか「Mac 便利機能への誘い」なのか判別が付かなくなっていますが、どうせ目次には票は入らないだろうという独断と偏見のもと集計しています。ご理解とご協力を強制いたします。

- ・ 9.アメリカンパワーフード：7人
- ・ 10.WORD 読者アンケート：5人
- ・ 8.Good Bye,Windows ! : 3人
- ・ 3.GR な日々 : 2人
- ・ 6.Mac 便利機能への誘い : 2人
- ・ 11.編集後記 : 2人
- ・ 1.表紙 : 1人
- ・ 4.Android でコーディング : 1人
- ・ 7.Perl6 入門-オブジェクトを躊躇する！ : 1人
- ・ 12.裏表紙 : 1人

芝とは…… あれっ、書いてないじゃん！ 最後まで芝を貫き通せよ！！！以下、回答欄をそのまま載せます。

10：次もイカちゃんをおして下さい。というかやれ。

11：ナマコの存在がすごかった。

(新人文学類 吉永才人さん)

まあ私は小麦ちゃん<sup>4</sup>派だけね。

9 僕も好きです、お菓子作り

10 みなさん個性的でおもしろいですね

(情報科学類 変☆態(★★★)さん)

お菓子すきかい？<sup>5</sup>

1 2 10 11 12

ゆるい記事を希望。アンケートの集計おもしろい。

(情報科学類 IMAGINE THE FUTURE.さん)

今後もこの記事は浦安の地盤くらいゆるい記事にしようと思います。続けばだけどね……

\*4 小麦ちゃん：かわいい

\*5 お菓子すきかい？：うん、大好きさ！わからない人は「頭の中にダイナマイト」で検索すべし

⑨アメリカンパワーフード  
—わたしを RanRan に連れてってー (お願いします)

(情報科学類 C.Ma н д e л さん)

(^,^)  
( ° ω ° )  
((c ))  
( \_ \ ^ )  
ε ≡ Ξ ∫ ∫ J

8 私も Windows には嫌悪感を抱いております。そろそろ Ubuntu でも始めようかと考え中。  
10 自分のコメントが.....

(社会工学類 黒谷駆さん)

今週も全部さらします。

4 : こんなこともできるのか！と感心  
7 : Perl6 手を出そうとしていたので丁度よかったです  
8,9 : 普通に面白かった

(情報科学類 tak8n.さん)

はい。

号名。  
理由 : 激しい

(惑星ベジータ とりすーぷさん)

他にもサブタイ候補はありましたが、なぜかあの長いサブタイが採用されました。

Mac いいよ

(芝学類 芝学類長さん)

**芝学類とは一体.....**

## WORD 読者アンケート

3 GR 良いよ GR。でも富士のスーパー CCD ハニカム EXR には劣る。富士フィルム最強！  
8 Ubuntu 便利。4 年くらい前から使ってます。  
9 馬鹿っぽいから（良い意味で）

（びぶんとかぎりしゃもじとかつかうむずかしいがくるい ふんぬらばあ！さん）  
そういうえば「GR な日々」ってカメラ記事でしたね。

2 : GR な日々…秘境、いいですよね。  
9 : アメリカンパワーフード…生地の崩れっぷりに吹いた。

（応用理工学類 Yellow 13 さん）

「2 : GR な日々」だと……？ この人もお節介機能にやられたに違いない！

9、トルコのお菓子の「バラクヴァ」また「甘口イチゴスパ」を知っている私にとっては  
まだまだ"甘い"です。  
10、次のお題を（こんな ~~未~~ かけた面白いアンケートを終わらせてたまるかよ！）

（（第四学群）二次元創成学類 多加枝 鉢見さん）

「バラクヴァ」だと……？ バクラヴァじゃないの……？ それはともかく、スニッカーズパイの  
生地の如く、この記事もだいぶ崩れかかっているので近々消えるかも……。

次のお題については Q7 にて発表させていただきます。

### ◆ Q5: 良くなかったと思う記事があれば教えて下さい

- ・ 5.LTE レビューその後 : 1 人
- ・ 9.アメリカンパワーフード : 2 人
- ・ 11.編集後記 : 1 人

以下、回答欄をそのまま載せます。

9 : 生地が崩れる様子が趣深いのに……。

（新人文学類 吉永才人さん）

申し訳ありません。代わりに記事が崩れる様子をご覧ください。

プログラミング系がつまらん。数学・物理系がいいな。  
AKBみたいに WORD メンバーの選挙やって TKB ○○♂をつくってほしい。

(情報科学類 IMAGINE THE FUTURE.さん)

「面白くなればテレビじゃない」というキャッチフレーズをご存知でしょうか。われわれ情報科学類誌 WORD は「プログラミング記事がなければ WORD じゃない」というキャッチフレーズを宣言いたしますのでご理解とご協力を強制いたします。あと、TKB ○○♂と書いてありますが女性編集部員も居ますよ～。たくさん居ますよ～。

#### 5 LTE レビューのその後

つづきなのはわかるけど、せめて前回を読んでいない人のために  
簡単に LTE を説明する文(又は形容詞節)がほしかったですね。

(情報科学類 C.Мандељさん)

アンケート史上、最も参考になる意見でした。ありがとうございます。

#### 9 もちろん全部食べきたんですよね☆

(情報科学類 tak8n.さん)

関係者に尋ねたところ「食べましたよ、当然。」というコメントをいただきました。

編集後記

ITF

(惑星ベジータ とりすーぷさん)

どうやら IMAGINE THE FUTURE.のロゴを太字にする作業が中途半端なところで終わっていることに対して怒っているらしい。

4.情ペ最強、サムチョン嫌い。iPhone ?ハゲ社長が電波を何とかしない限りゴミだよ、ゴミ。  
それとアリア社長可愛い。ぶいにゅ。  
そしてこのアンケート用紙。タイトル一覧の中に2.が2つ有る。わけがわからないよ。

(びぶんとかぎりしやもじとかつかうむずかしいがくるい ふんぬらばあ！さん)

親に「家族そろって iPhone にしよう」といわれましたが、電波が弱いことを理由に断固反対しました。定期的に鉄道で僻地にいく私にとって、電波が弱いというのは由々しき問題なのです。  
記事の番号がおかしいのは前述の通りです。俺は悪くない！

## WORD 読者アンケート

### ◆ Q6:過去の記事に関する感想を教えて下さい

教室に落ちてたのを拾ったので、過去もクソもありません。  
最初はサツマイモを焼くのに使えそうと思ったのですが。

(新人文学類 吉永才人さん)

サツマイモを焼くのであれば、ちょっと過去に刷りすぎた号が大量に余っているので、それを使ってください。

MAX プリン。コスプレ。

(情報科学類 IMAGINE THE FUTURE.さん)

今年の新歓では MAX キャラメルが登場しました。新たな名物となるか！？

ミクかわいい！最高！

(生物学類 Zimbabwe さん)

まあ私は小麦ちゃん<sup>\*6</sup>派だけどね。

読んだことないです

(情報科学類 変☆態(★★★)さん)

時間のある時にバックナンバー見よ～と思います。

(情報科学類 tak8n.さん)

記憶が……

(情報科学類 C.М а н д е л さん)

過去は振り返らない主義！

(びぶんとかぎりしゃもじとかつかうむずかしいがくるい ふんぬらばあ！さん)

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

みなさんまとめて読みましょう～。

---

\*6 小麦ちゃん：かわいい！！！！

総集編ないんですかあ！？ フルカラーで。

(惑星ベジータ とりすーふさん)

フルカラーではありませんが、実はすでに4回発行しています。1~3は私が入学する前の事なのでよくわかりません。4は今年の2月に「つくばSF・コミックフェスタ2011(通称:つっこみ)」にて1部100円で売りました。この様子を記事にする予定でしたが、#氏がスニッカーズパイを焼き上げていたころにmitty氏の私生活が焼き上がり、#氏がスニッカーズパイの生地を崩したころにmitty氏がつっこみの記事を崩してしまいました。そういうことです。

#### ◆ Q7:試される回答欄～自由記述欄～

第三エリアの食堂にそおしあ～るが落ちているのを見たことがあるので、配布エリア外に落ちているのは学類誌の宿命かもしれません。そおしあ～る配布場所の真横で配布して「そおしあ～るより面白い！情報科学類誌WORD」という看板でも立ててやろうかというネタも考えましたが、第一エリアでの配布許可をとるのがめんどくさくなつてやめました。

「経済」の授業で教室に落ちたのを見て、いちおう読んでみました。まあ次号に期待ですね。  
どうしてこれが一学の教室に落ちていたのかは永遠に謎です。都市伝説にしたいです。というかしろ。  
一学でも読書をうやせということ？  
最近うなぎが食べたいところときがあるんで  
すけどどうしてでしょうか。もうすぐ土用ですけどういう時期でもないときに食べたくなるんです。

あとSFCのソフト、全部ぼくに下さい。  
SO2は神ゲー。異論は認めない。

\*歴代総理大臣を身に付けてものに例えるシリーズ  
←その1  
○伊藤博文 → 塩こんぶ  
○原敬 → ~~カキ~~ つけづくね  
○村山富市 → 手ゆ毛

(新人文学類 吉永才人さん)

時間を創り出すライフハックですか。私が知っているのは「東京→大阪の移動で新幹線を使うと通常よりも6時間短縮できる」「ドンキーコング2はなるべく1時間以内にクリアする」くらいでしょうか。マジレスすると、時間を創り出すライフハックを知っている編集部員は一人も居ません。編集部員になること自体が時間を無駄にしているからです。

どうも、三編 from 高専 & COJT@SW 氏です。  
一学期に講義取り過ぎ?で課題過多を70%...  
じゃなくて\*にそろ...  
ネギま！最新刊限定版が欲しいが金もかけねば時間もねー！  
その他諸々漫画読みたいアニメを見たいと思ってます...  
\*重篤なアニメ成分不足、アリア見たいな...  
た"が"。

時 間 行 無 し

そこで時間を作り出すライハックの特集を希望します。  
私はWORDを応援します！

P.S.

誰か～！麻雀やうす～カラオケ行こう～Tセシ行k(ry

(情報科学類 tak8n.さん)

## WORD 読者アンケート

初音ミクがいいません どうしたことでいいのか

(生物学類 Zimbabwe さん)

どうした事でしょう。まあ私は 小麦ちゃん<sup>\*7</sup> 派 だけどね。

今期はカイジ 星空へ架け橋 トトコ スーピースー がいいな。  
OJTグッズほしいです。

(情報科学類 IMAGINE THE FUTURE. さん)

カイジの後に始まるユルアニ？ こそ今期ナンバー 1 でしょう！！！ そんなことより OJT グッズ取りに来てください。どんどん持って行ってください。早く！！！ 1 1 1 !!

引用符がバラバラじゃないですか、校正お願ひしますよ。表紙の右下にたぬき油性マジックをつかったことについてここに明記しておきます。

(情報科学類 C. Манде л さん)

そんなことよりなんか色々書いて消した跡あったけど、扱いに困るじゃないですか。それと性別の回答なかったけど回答お願いしますよ。

私は統計学を愛しています。  
是非皆様も統計学の世界へ  
因子分析の美しい異常（異常ではない）

(社会工学類 黒谷駆さん)

是非あなたも朝鮮音楽<sup>\*8</sup> の世界へ。朝鮮音楽の美しさは偉大。「攻撃戦だ」「統一の虹」「一つの大家族」「女性は花だね」「祝杯を持とう」「不平等歌」「フルラリ」「我らの将軍様を仰ごう」「若さは急行列車」「祝福します」「曜日解釈の歌」「祝願の花吹雪」あたりがノリが良いのでオススメです。

\*7 小麦ちゃん：かわいいいいいいいいいいいいいい～～～～！！！！

\*8 朝鮮音楽：北の共和国のほうです。

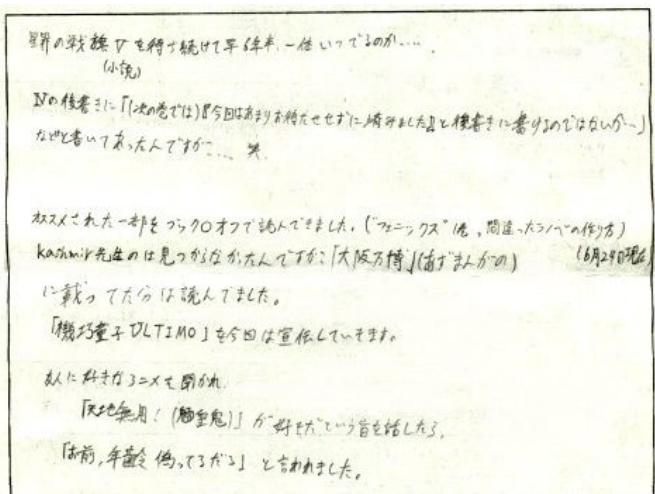
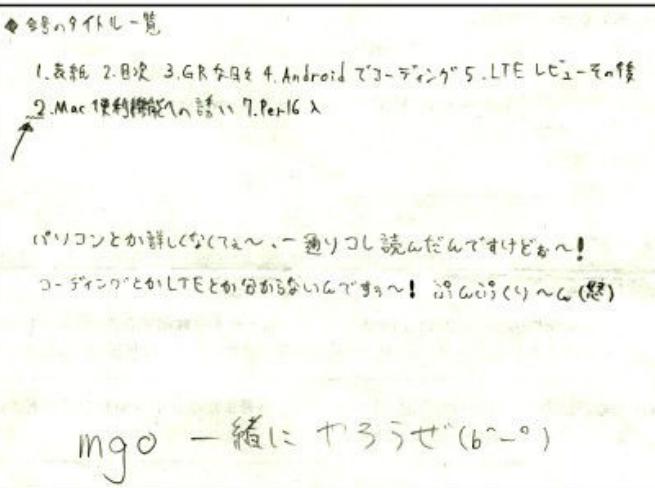
mgo とかあ～、スーパーファミばかりやってる私には全然わからないんですう～。ぶんぶくり～ん(怒)

パネルでポン、みんなもやろうぜ(b^ー。)

(情報科学類  
変☆態(★★★)さん)

ひー、文字小さすぎるー。ここまで小さいと印刷時につぶれて読めなくなってるかも。個人的に重要な情報だけ抜き出すと「天地無用！魑皇鬼」が好きという旨の文が最後の方にかかれています。私は何故かあずまんがから天地無用！にのめり込んでしまいました。ちなみに今年の私はPCを3台自然故障させてしまう美星さん状態です。助けてください

(応用理工学類 Yellow 13さん)



リトバスは非常に絵柄がかわいらしくて好みなのですが、兎にも角にも私が

**小麦ちゃん<sup>\*9</sup>派**

である事にはかわりありません。

(びぶんとかぎりしゃもじとかつかうむずかしいがくるい ふんぬらばあ！さん)



\*9 小麦ちゃん：ばんざーい！ばんざーい！ばんざーい！

## WORD 読者アンケート

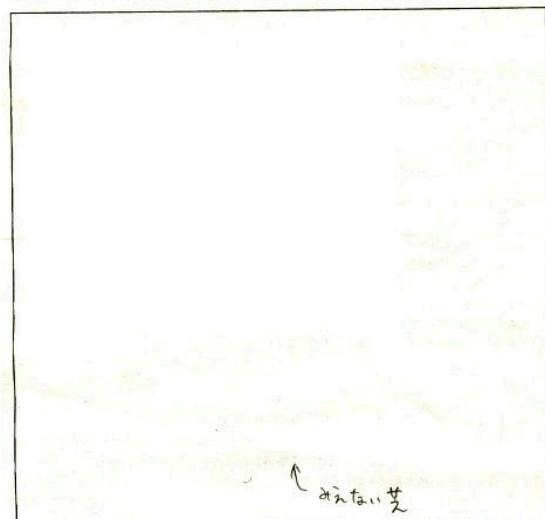
もう秋だけどね。それより、IMAGINE THE FUTURE.の最後のピリオドが欠けてるけど大丈夫ですか。IMAGINE THE FUTURE.信者(過激派)におそわれても知りませんよ~。



アンケートが始まって以来、すべてのアンケートに回答している方はあなただけです。えらいっ！

ダグドリオが3人の女の子(ロングとショートとツインテール)になっているのは良いのですが、私はシニヨンが一番好きなのでよろしくお願ひします。

シニヨンが可愛いという事を念頭に置いた上で、ウベシティーに生息する幻のポケモン「エコハちゃん」を描いて下さい、お願ひします。



### ■はい、おしまい。

また次号でお会いしましょう。

---

## 編集後記

---

情報科学類誌

WORD

From College of Information Science

一斗缶  
special edition号

発行者

情報科学類長

編集長

石川 陽一

制作・編集

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)

2011年09月30日 初版第一刷発行

(512部)