

WORD

From College of Information Science



- ・ 今日からマのつく制作者！
- ・ わかった気になるHTTP/2
- ・ Emacs Startup(基礎編)
- ・ WORD読者アンケート

目次

Emacs Startup(基礎編) 3
わかった気になるHTTP/2 7
今日からマのつく製作者！ 13
WORD読者アンケート 19
編集後記 29

Emacs Startup(基礎編)

文 編集部 rizaudo

1 はじめに















我らが Coins で行われている教育カリキュラムで最初に触るエディタは Emacs、メーラは Mew となっています。ですが、Emacs をカスタマイズする方法についてはほぼ触れられていません^{*1}。

私の考えでは、Emacs を快適に使うためにはカスタマイズは必須です。なぜなら、Emacs のデフォルトは時々良いと言えない挙動をするからです。

よって、この記事では Emacs のカスタマイズに必要と思われる情報を書きたいと思います^{*2}。この記事では、Windows は対象外となっています。対象とする Emacs は現在 (12/7) の最新版 24.4.1 にて確認しており、前提が無ければ 24.4.1 について書くものとします。

1.1 キー表記

この時期ではリテラシを情報科学類生は終えているはずですが、念のためにこの項で Emacs のキー表記について説明をします。

Emacs 文化での表記	実際のキー操作
	 を押しながら  を押す
	Meta キー、代替キーとして  を押しながら  を押す
	 (Escape キー) を押す
	 (Tab キー) を押す
	 (Return キー) を押す
	 (Delete キー) を押す

2 Emacs のカスタマイズ済みキットを使う

Emacs のカスタマイズが面倒で使えれば良いという方のために、この節ではカスタマイズ済みキットと分類出来るだろうソフトウェア^{*3}について書きます。

2.1 Emacs Prelude

Emacs Prelude^{*4} はデフォルトの Emacs 体験を拡張するために作られた Emacs ディストリビューションです^{*5}。ディストリビューションと名乗るだけあって、巨大なソフトウェアです。詳細は Github の README

^{*1} 少なくとも 13 年度入学では

^{*2} なお筆者に何か奢ってくれるか、十分に好感度を上げると、Emacs のカスタマイズ相談にのります

^{*3} それぞれのソフトウェアで別の名称を使っていたので、総称が存在しませんでした

^{*4} <https://github.com/bbatsov/prelude>

^{*5} <https://github.com/bbatsov/prelude/blob/master/README.md> より

を読むとよいのですが、様々な独自キーバインドを追加していて独特な物となっています。使って気に入らない場合でも、カスタマイズの参考にはなるかと思います。導入方法は以下のとおりです。

cURL を使った方法

```
curl -L https://github.com/bbatsov/prelude/raw/master/utils/installer.sh | sh
```

Wget を使った方法

```
wget --no-check-certificate  
https://github.com/bbatsov/prelude/raw/master/utils/installer.sh -O - | sh
```

- Git を使った方法^{*6} ::

```
git clone git://github.com/bbatsov/prelude.git path/to/local/repo  
ln -s path/to/local/repo ~/.emacs.d  
cd ~/.emacs.d
```

2.2 Spacemacs

Spacemacs^{*7} は、Evil という Emacs 上で Vim ライクなエディタを実現する拡張を主軸とした、発展キットです^{*8}^{*9}。Spacemacs は公式にあるスクリーンショットなどを見るとわかるように、操作は Evil がベースとなっていることから、Vim ライクになっているため Vim から移ってきた人だとすぐに使えると思われます。導入方法は以下の通りです^{*10}

```
git clone --recursive http://github.com/syl20bnr/spacemacs ~/.emacs.d
```

3 推奨パッケージ

本節では Prelude や Spacemacs などのカスタマイズ済みキットに頼らず、手動でパッケージを入れて改造していく際に有用であるであろう情報について書きます。

^{*6} 念のため書きますが path/to/local/repo は環境に応じて変更してください

^{*7} <https://github.com/syl20bnr/spacemacs>

^{*8} <https://github.com/syl20bnr/spacemacs/blob/master/README.md> より

^{*9} ん？ Emacs もエディタだって？ HAHHAHA ご冗談を。環境ですよ？

^{*10} README より、GitHub は <https> が好きしいですが、これは原文ママとなっています

3.1 準備

現在 Emacs には標準で `Package.el` というパッケージマネージャが存在しており、Emacs のパッケージを入れる時にはこれを使うと非常に楽になります。`init.el` 等の設定ファイルに書く好ましい設定としては以下の通りとなります。

```
(require 'package)
(addlist 'package-archives
  '("marmalade" .
    "http://marmalade-repo.org/packages/"))
```

ここでは Marmalade という、ユーザがパッケージを自由にアップロード出来る Emacs Lisp リポジトリを読み込みリポジトリのリストに追加しています。パッケージをインストールする際は、

```
M-x package-install [package-name] RET
```

と指定してインストールをするか、または

```
M-x package-list-packages RET
```

で表示されるリストの好きなパッケージにカーソルを合わせ、☐ **i** でインストールをするマークをつけ、☐ **x** で実行をするとインストール出来ます。

3.2 Better Defaults

一部の人にはおなじみの `technomancy`^{*11} という方が作っている、Emacs のデフォルトを少し改善するパッケージです。内容としては `ido`, `uniquify`, `saveplace` に対してのちょっとした設定、括弧の対応の有効化、Menu-bar などの無効化が入っています。導入方法は以下の通りです

```
M-x package-install better-defaults
```

3.3 Starter-kit

Better Defaults と同じ作者によるパッケージ、及びガイドです。推奨しておきながら何ですが、パッケージ版がかなり古いため、試しに使ってみて、設定を参考にするのに良いかと思って紹介しました。ガイドのほうは中々良いパッケージをオススメして、一見の価値がありますので、是非参考にどうぞ。導入方法は以下の通りです。

```
M-x package-install starter-kit
```

^{*11} <https://github.com/technomancy>

4 カスタマイズに有用な情報もしくは知っておくと良い事

Cheatsheet Cheatsheet とは要するにカンペです。Emacs の基本的なキーバインドのチートシートをどこか目に見える所に貼っておくと覚えがよくなるでしょう。また、パッケージによってはチートシートを用意してくれている場合もあるため、使っているパッケージのチートシートが無いか探してみるのもよいでしょう。

M-x info このコマンドで出てくる、Emacs に標準で付属のリファレンスが、大体の場合に於いて有用です。

C-h デフォルトでこのコマンドには **Help** が割り当てられており、この **Help** はこの関数は何をするのか、このキーバインドはどの関数に割り当てられているのかなどをチェックする時に有用です。入れたパッケージの事について知る時にはほぼ必須だと思われます。

4.1 Emacs

Coins 計算機利用の手引き 2014 年版^{*12} Coins では計算機利用のための手引きを毎年発行しています。その中に Emacs の利用の基礎について書いている章が存在するため、Emacs の基礎がわからないという場合は参照すると良いでしょう。

EmacsWiki Emacs Wiki には大抵の Emacs に関する情報が集まっています。以上。というのは少し酷いので追記すると、情報が古いこともあります、大抵のパッケージの説明が載っているため、困ったら Emacs に聞く次ほどに参照すると良いでしょう。

4.2 Emacs のカスタマイズ

Emacs JP Emacs JP は日本の Emacs ユーザーのためのハブサイトで、Emacs 全般についての情報を掲載しています。ここの中の Emacs for Beginners^{*13} という記事は少し参考になると思われます。残念ながらコンテンツは少ないのですが、質問を **GitHub Issue** で受け付けているため、分からない事があったら聞くと良いのではないのでしょうか。

5 おわりに

時間がなくて、一、二時間で書いた記事ですが、何かの役に立てば幸いです。この記事の上位互換として、私が作っている Emacs customize 101 ^{*14} というサイトがあるので、もっと情報が欲しいという人は見てください。おわり。

^{*12}<http://www.coins.tsukuba.ac.jp/tebiki/2014/tebiki2014.pdf>

^{*13}<http://emacs-jp.github.io/beginner.html>

^{*14}<http://rizaudio.github.io/emacs-customize-101-jp/> URL は変わる可能性があります

わかった気になる HTTP/2

文 編集部 polamjag

1 はじめに

こんにちは、polamjag と申します。この記事では、現在 IETF の httpbis WG^{*1} において Working Group Last Call が終了した^{*2}HTTP/2^{*3}について、HTTP/1.1 をだいたい知っていることを前提として、プロトコルの概要を噛み砕いて紹介していきます。厳密な説明を目的としているものではないことを予めご了承ください。

2 HTTP/2 の概要

HTTP/1.1 が最初に策定された RFC2616 は、なんと 1999 年 6 月に発行されています^{*4}。それから本稿執筆時までおよそ 15 年強が経過しています。その間にインターネットを取り巻く環境は大きな変化を遂げ、結果として HTTP/1.1 はある意味時代遅れなプロトコルになってしまいました。

そこで、Google は HTTP/1.1 をベースに高速・効率化した SPDY (スピーディー)^{*5} というプロトコルを独自に提唱し、導入しました。SPDY 2 以降は仕様も公開され、多くのブラウザやサーバソフトウェアが対応したこともあり、Google や Facebook などのようなサービスでは実際に導入されています。

HTTP/2 は、この SPDY をベースに策定されています^{*6}。

HTTP/2 は、現在の最新バージョンである HTTP/1.1 と比べて、GET や POST といったメソッド、メッセージなどの変更はありません。これは、HTTP/2 を導入するのに、Web アプリケーションなどのコードを変更する必要はほとんどの場合無いということを意味します。

HTTP/2 の大きな特徴としては、おおまかに以下のような点が挙げられます：

1. HTTP/1.1 までのようにテキストベースではなく、バイナリベースである
2. 一つの HTTP/2 接続の中で複数の接続を擬似的に実現する“ストリーム”の導入
3. サーバからのプッシュが可能
4. 高度なヘッダの圧縮
5. ALPN または HTTP Upgrade による HTTP/1.1 以前との共存

^{*1} Hypertext Transfer Protocol ワーキンググループ

^{*2} だいたい仕様が固まったということです

^{*3} ちなみに、HTTP/2 の正式名称は HTTP/2.0 ではなく HTTP2 または HTTP/2 であるとされています

^{*4} つい最近 RFC7230～7239 でアップデートされています

^{*5} SPDY も IETF で標準化が進められています

^{*6} HTTP/2 のドラフトの一番最初のものは SPDY の仕様をそのまま写したのになっています

2.1 フレーム

HTTP/2 において、プロトコルのもっとも基本的な単位であるのがフレームです。すべての情報はフレームとしてやりとりされ、それらははすべてバイナリベースのフォーマットで表されます。

フレームには、表 1 で示す種類が含まれます*7。

フレームタイプ	説明
DATA	HTTP リクエストやレスポンスのペイロードを送信するのに使用
HEADERS	圧縮されたヘッダ
PRIORITY	ストリームの優先度を設定
RST_STREAM	ストリームの即時終了。キャンセルやエラー発生時に使用
SETTINGS	ストリームの最大同時アクティブ数のような、接続に関する設定に使用
PUSH_PROMISE	サーバ側がストリームのプッシュの開始を通知する
PING	アイドル中の接続の死活監視、遅延測定
GOAWAY	ストリームの作成の終了をリモートピアに通知
WINDOW_UPDATE	ストリームのフロー制御に使用
CONTINUATION	連続する HEADERS、PUSH_PROMISE を別のフレームで送る際の続きに使用

表 1: フレームタイプと説明

2.2 ストリーム

ストリームは、クライアント・サーバのどちらからでも確立できる、フレームの双方向シーケンスです (図 1)。要するに、フレームのやり取りの集合みたいなものです。一つの HTTP/2 接続の中には複数のストリームも含むことが可能です*8。ストリームの詳しいライフサイクルに関してはここでは記述しませんが、鍵となる機能について紹介します。

優先度の設定

ストリームには優先度が設定できます。優先度の設定には、新規ストリームでは HEADERS フレームを、既存のストリームに対しては PRIORITY フレームを使用します。これにより、画像ファイルより CSS ファイルを優先的にダウンロードさせたり、といったことが可能になります。

*7 RFC のドラフトに説明が登場する順に並べています

*8 HTTP/1.1 までは、サーバとクライアントが複数通信する際には毎度新規に HTTP 接続を確立する必要があり、そのオーバーヘッドが問題視されていました。また、同時接続数にも制限があるため、多くのファイルを同一ホストから一斉にダウンロードしようとするとう時間がかかってしまうという問題もあります

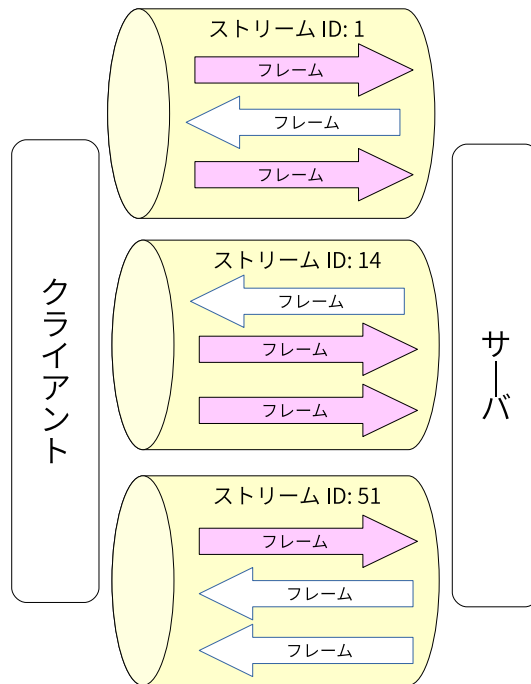


図 1: フレームとストリーム

依存関係

あるストリームに他のストリームを依存関係として追加することで、リソースの優先的な割り当てを明示することが出来ます。依存関係として追加されたストリームは、それぞれ依存関係の順番に応じて、優先度に比例したリソースが割り当てられ、処理されるべき (SHOULD) とされています。

フロー制御

ストリームは WINDOW_UPDATE フレームによりフロー制御を実現しています。フロー制御は DATA フレームのみが対象とされ、また無効化することは出来ません。

ただし、HTTP/2 は、WINDOW_UPDATE フレームのフォーマットと意味だけを定義し、実際のアルゴリズムや実装については何ら取り決めていません。

2.3 サーバからのプッシュ

HTTP/2 では、サーバからレスポンスをプッシュすることが出来ます。これにより、例えば HTML ファイルの前に CSS ファイルや画像ファイルを前もって送信する、というようなことが可能になります。プッシュされたレスポンスは、基本的にクライアントによりキャッシュされます^{*9}。

^{*9} キャッシュ可能かどうかは、サーバ側がヘッダのフラグを使用して通知します

ブッシュでは、まずサーバから PUSH_PROMISE フレームが送信され、その後にブッシュレスポンスが DATA フレームとして送信されます。

サーバブッシュのレスポンスは、通常のリクエストに対するレスポンスと変わらないものとされており、PUSH_PROMISE フレームは HTTP リクエストヘッダの代わりの役割を果たします*10。

2.4 ヘッダの圧縮

HTTP/2 では、HPACK という仕組みを使ってヘッダの圧縮を実現しています。

HPACK

HPACK は、SPDY の EXI 圧縮という仕組みを元にしています。EXI 圧縮は、行ベースである HTTP のヘッダを XML 化し、それをバイナリで表現してから最後に deflate*11 で圧縮するというものです。

それに加えて、HPACK では Indexing Tables とハフマン符号化を用いることで、SPDY よりも効率的なヘッダの圧縮を実現しています。

Indexing Tables はヘッダのキーを整数の符号と対応させる仕組みのことで、よく使うヘッダについて予め定められた Static Table と、動的に決定される Header Table との 2 つから構成されています。実際の通信時には、ヘッダのキーとして整数値を用いることで通信量の削減を図ります。

2.5 HTTP/2 と HTTP/1.1 以前の共存

HTTP/2 とそれ以前の HTTP が共存している環境の場合、クライアントとサーバのやり取りによってどのバージョンの HTTP を使用するか決定する必要があります。これには、以下の 2 つの手法が用いられます。

ALPN (Alternate Protocol Negotiation)

ALPN は、TLS 上でプロトコルネゴシエーションをするための拡張です。TLS のハンドシェイク時に HTTP/2 が利用出来ることを通知することで、HTTP/2 での接続を開始できます。HTTPS 接続を使用する場合はこちらが利用されるでしょう。

HTTP Upgrade

HTTP/1.1 では、Upgrade ヘッダフィールドが定義されています。これを使い、クライアントがリクエスト時に HTTP/2 が利用可能であると通知することで、接続を HTTP/2 に切り替えることが出来ます*12。

3 HTTP/2 を手軽に試す

3.1 クライアント

Google Chrome (または Chromium) の現行安定版 (39.0 系) は、HTTP/2 の draft-14*13 を実装しています。chrome://flags/から有効にすることで、HTTP/2 が有効になります。

*10 ちょっとややこしいですが、クライアントからリクエストを投げた結果のレスポンスと、サーバブッシュによるレスポンスとは全く変わらないものとして扱われるということです

*11 いわゆる ZIP ファイルや gzip、PNG ファイルなどで使用されている圧縮アルゴリズム

*12 Websocket でも同じ仕組みを使って HTTP でのリクエストから Websocket での通信に切り替えています

*13 ゼロオリジンで 14 番目に策定された仕様のドラフト (草稿) という意味

Mozilla Firefox の現行安定版 (34.0 系) も draft-14 に対応しています。ロケーションバーに `about:config` と打ち込むと開くページで `network.http.spdy.enabled.http2draft` を `true` にすることで有効にすることができます。

Internet Explorer では HTTP/2 は Preview Release となっています。開発状況は <https://status.modern.ie/http2> から確認することが出来ます。

また、`curl` コマンドも 7.36.0 以降で draft-14 に対応しています*14。

3.2 Web サービス

現時点では、殆どの Web サイトや Web アプリケーションは HTTP/2 に対応していません。

有名どころでは Twitter (twitter.com) が HTTP/2 に対応しています。日頃よく使っているサービスなら HTTP/2 を有効にした際のパフォーマンスの違いを体感するのもそう難しくないと思います*15。

接続で HTTP/2 が有効になっているかを簡単に確かめるためのサイトとしては、[iij-http2](https://http2.iijplus.jp/) のもの (<https://http2.iijplus.jp/>) や、[golang.org](https://http2.golang.org/) のもの (<https://http2.golang.org/>) などが挙げられます。

3.3 Web サーバ

httpbis WG の GitHub Wiki 上に HTTP/2 に対応したサーバの一覧がまとめられています。その中からサーバソフトウェアを選び、適当に立ててみるのもよいでしょう。

HTTP/1.1 サーバのリバースプロキシとして HTTP/2 サーバを配置することで、既存の Web サービスなどに手を加えることなく HTTP/2 を試すことも出来ます。

3.4 その他

パケットキャプチャ・解析ソフトとして有名な Wireshark も HTTP/2 の解析・表示に対応しているので、実際にパケットを眺めてみるのもいいと思います。

4 HTTP/2 をもっと知る

以上の説明は大変ざっくりとしたものだったので、プロトコルや実装の詳細を知りたい方にとっては物足りないものだったと思います。以下では、より詳しく HTTP/2 を知るための情報について記します。

HTTP/2 JP (<http://http2.info/>) というコミュニティが、GitHub 上での仕様の和訳、勉強会・ハッカソンなどを活発*16に行っています。また、ウェブサイトには“HTTP/2 最速実装”と題して、HTTP/2 に対応したソフトウェアを実際に実装するための資料も掲載されています。

また、HTTP/2 の正式な仕様 (RFC) は、早ければ来年の 2 月頃には正式にリリースされるのではないかと言われています。httpbis WG のメーリングリストやドキュメント、GitHub の [http2/http2-spec](https://github.com/http2/http2-spec) レポジトリ*17 の Issue などを見てみることで、議論の過程を追っていくのもいいでしょう。

*14 OS のパッケージマネージャなどによっては新しいバージョンのものが降ってきていないかもしれません。筆者の環境 (Arch Linux) ではこれよりも新しいものでした

*15 Chrome Dev Tools や Firefox 開発者コンソールの Network パネルから、CSS や JavaScript、画像などがどのようにダウンロードされているか観察すると面白いです

*16 本場に活発です

*17 <https://github.com/http2/http2-spec/>

4.1 参考

<https://tools.ietf.org/html/draft-ietf-httpbis-http2-16>

HTTP/2 draft-16 (記事執筆時の最新版)

<https://tools.ietf.org/html/draft-ietf-httpbis-header-compression-10>

HTTP/2 ヘッダ圧縮仕様 draft-10 (記事執筆時の最新版)

<https://github.com/http2jp/http2-spec-ja>

HTTP/2 JP による上記ドラフトの和訳

<https://github.com/http2/http2-spec/wiki/Implementations>

HTTP/2 実装の一覧

<http://datatracker.ietf.org/wg/httpbis/documents/>

IETF の httpbis WG のページ

<http://qiita.com/advent-calendar/2014/http2>

HTTP2 Advent Calendar 2014

<http://mozaic.fm/post/83421293098/2-http2>

mozaic.fm というポッドキャストの、HTTP/2 がテーマになっているエピソード (2014 年 4 月配信)。

実際にサーバソフトウェアを実装したエンジニアの方による対談が収録されています

今日からマのつく製作者！

文 編集部 ジオン

こっちから読んでください

コミPo!登場



今日も一日



1 コミ Po!とは

コミ Po!とは株式会社ウェブテクノロジー・コムが販売している 3D で誰でも簡単にコミックがポツと作れてしまうソフトなのである。2010 年 12 月 15 日にリリースされ、拡張パックが 2014 年 8 月に発売されるなど今でも更新が続いている。

コミ Po!の良いところは製作物を個人・商用問わず許可無く自由に使うことができることだろう。そのためコミ Po!があれば自分のホームページ上の挿絵、サークルの冊子などでマンガやイラストをふんだんに使うことができる。

とはいえ恩恵が大きいだけあってフリーソフトではない。Amazon ではダウンロード版が 6340 円、パッケージ版が 7530 円(2014 年 11 月現在)と、即座に購入するには些か高価である。

そこで今回はコミ Po!の製作過程、利点およびコミ Po!の仕様などを紹介する。この記事が読者の方のコミ Po!を購入するきっかけになれば幸いである。

2 マンガをつくろう

コミ Po!でマンガを作る各ステップを紹介していく。

2.1 ページの設定

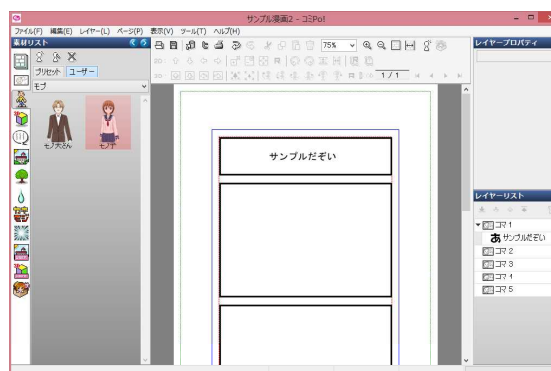
マンガを新規作成するときはまずページの設定をする。用紙は縦横各 2 タイプあり、それぞれ 4 コマ、ページマンガ用がある。4 コマはもちろんのことページマンガにも予めコマ割がしてあるテンプレートが用意されている。もちろん白紙から好きなようにコマを配置することもできる。

2.2 マンガの制作

ページ設定をすると編集画面中央に原稿用紙が表示され、マンガがつくれるようになる。

左側にあるのが素材リストで 3D キャラクター、吹き出し、背景などの各オブジェクトがまとめられている。ここから使いたいオブジェクトを選択して、コマの中にドロップをすると配置することができる。

右側にあるのがレイヤー関連のバーでレイヤーの管理・編集をここで行う。レイヤーとは層のことで主にオブジェクトの奥・手前の関係を調整するときに関わってくる。レイヤーの概念は他で触れる機会が多いと思うので割愛する。



2.2.1 キャラメイク

マンガの中心はなんといってもキャラクターだ。コミ Po!のキャラクターは髪・顔・その他胴体(服装含む)にパーツが分かれており、これらを組み合わせることで自分だけのキャラクターを作ることができる。

作ったキャラクターは保存して素材リストの 3D キャラクターからいつでも使うことができる。

2.2.2 表情&ポーズ変更

最初に言っておくとコミ Po!は用意された表情・ポーズしか使うことができない。つまり、ユーザが自由に目の開く具合を変えたり、関節を調整したりすることはできない。

しかし、自由が効かない分収録されている表情とポーズは多く、表情は約 350 種、ポーズは約 200 種用意されている。また、ポーズとは別に手の形がパーツとして変更可能で、さらに首と手首の角度が自由に調整できるため同じポーズでも細部を変更することで表現の幅は意外と大きい。

2.2.3 3D オブジェクトの配置

次にコミ Po!のセールスポイントでもある 3D オブジェクトについてみていく。

3D オブジェクトに対する操作は 5 種類ある。3D オブジェクトを選択した時の図をみてほしい。

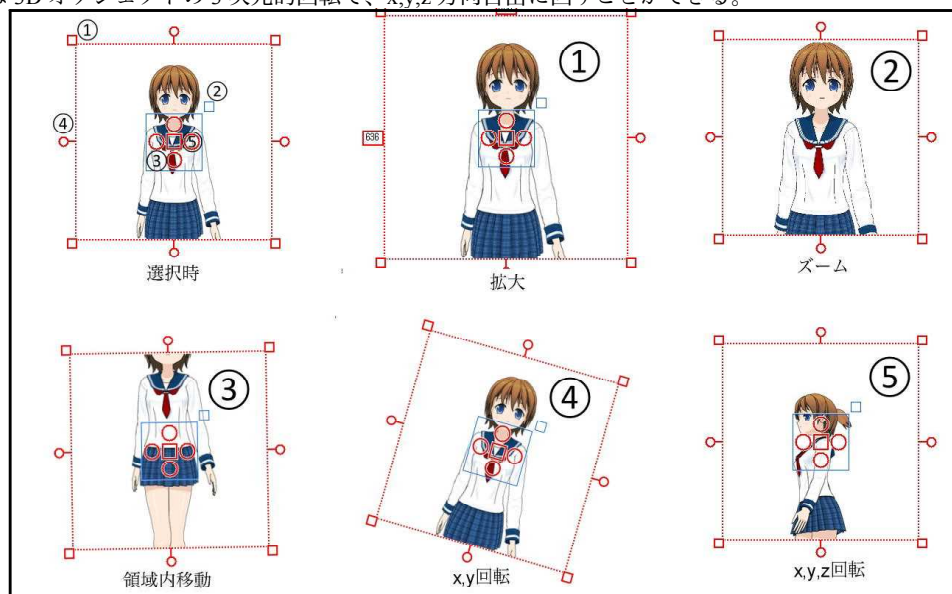
①の四角形をつまむと縮小・拡大をすることができる。ただし単純な縮小・拡大なので、拡大し過ぎると 3D オブジェクトの輪郭がぼやけてしまう。

②は 3D オブジェクトのズームである。こちらはオブジェクト自体のサイズを変えていくので拡大してもぼやけたり荒くなったりしない。①の操作と合わせてモデルの大きさを調節しよう。

③は描画領域内の移動である。画像の一番外側がオブジェクトの描画領域でこの領域内を移動することで描写する範囲を変えることができる。

④は描画領域の回転である。

⑤は 3D オブジェクトの 3 次元的回転で、x,y,z 方向自由に回すことができる。



2.2.4 その他素材の配置

吹き出し・集中線などの素材は拡大・縮小と回転のみすることができる。吹き出しや背景の素材は元から豊富なうえ、ユーザが選んだ jpg や png 画像を取り込むこともできる。

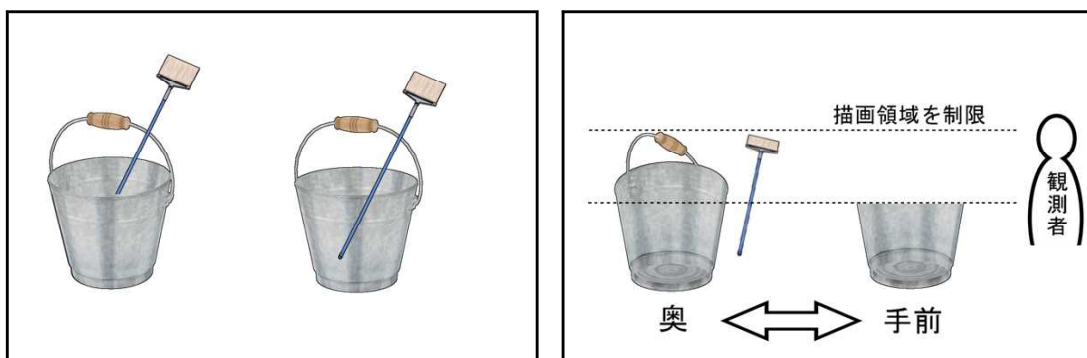
2.3 出力

素材の配置が終わったら画像として出力する。出力できる画像形式は jpeg/png で、またコミ Po!から直接 Twitter に投稿する機能もついている。

3 コミ Po!でできること・できないこと

前章にてコミ Po!でのマンガ制作を解説したが、コミ Po!のウリ文句通り簡単にできるため実際に使えばすぐに理解できるだろう。しかし、その簡単さ故に複雑な描写が仕様上できない事もある。ここではコミ Po!を購入する前に知っておきたいコミ Po!でできること・できないことを紹介していく。

3.1 3 次元的物体の配置



コミ Po!のセールスポイントは 3D でマンガが描けること……だが、オブジェクトを 3 次元的に配置できるわけではない。

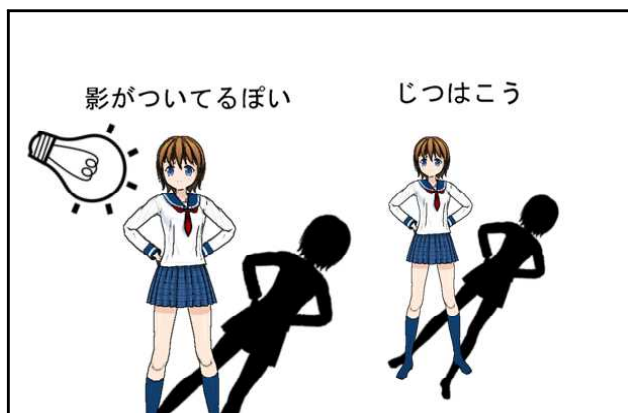
例えば上の図の左側はバケツの中にタモが入っているが、コミ Po!で何も考えずにバケツの中にタモを入れようとする右側のようになる。

実は左側のバケツは 3 枚のレイヤーからなっている。奥と手前にバケツのレイヤを作り、タモのレイヤを挟む。そして手前のバケツレイヤをトリミングし、タモの上部分が見えるように工夫してあるのだ。

配置によってはこのような工夫を用いてオブジェクトを組み合わせる必要がある。

3.2 影をつける

立体感を出すためにキャラクターに影を付けたいときもあるが、コミ Po!に光源を指定して影をつけるなんていう気の利いた機能はない。影はユーザの涙ぐましい努力によって再現される。例えば図のように影が伸びているように見えるのは影をつくるキャラクターのレイヤをコピーし、明度を下げた影っぽく見せている。実はこの影をつけるテクニックも 3.1 で行った立体的にものを配置するテクニックも全てコミ Po!の公式ページで紹介されている。



3.3 3D データの読み込み

コミ Po!は obj 形式、mqo 形式、pmd 形式に対応しており、読み込んで素材として使用することができる。

obj ファイルは多くの CG 系ソフトウェアが対応している頂点座標や頂点法線、頂点テクスチャの座標を記述し、ポリゴンを構成したファイルの事である。情報科学類なら CG 基礎の講義を履修した際にこの形式で三角柱などを作った人があるかもしれない。

mqo 形式は 3D モデリングソフトウェア「Metasequoia」のファイル形式である。詳しくは各自で調べて欲しい。

pmd 形式は MikuMikuDance(以下 MMD)のファイル形式である。MMD と言えばニコニコ動画や最近では MMD で作られたアニメが放映されているので、ご存知の方も多いかもしれない。となると早速 MMD の素材置場から艦〇れや東〇 Project のモデルをダウンロードして使ってみたいところだが、コミ Po!が対応している pmd 形式は最新の MMD ファイル形式ではないので、モデルをダウンロードしても使えない可能性がある。実際に素材をダウンロードしてもらえばわかるが、最近の MMD モデルは pmx というフォーマットが使われている。pmx を pmd に変換する方法はあるものの、pmx は pmd を改善・拡張した形式であるため pmd に変換すると様々な情報が欠落する可能性がある。そのため、流行りの MMD モデルをコミ Po!で使えるかと言えば微妙な話である。MMD が pmx に対応したのが 2011 年の春なので、それ以前の MMD モデルならば大抵読み込めるはずだ。ちなみにコミ Po!が発売されたのは 2010 年 12 月だが、pmd に対応したのは 2013 年の 12 月である。うーん？



pmd 形式モデルの一例

また、moq、pmd ファイルをコミ Po!に取り込んだとしてもコミ Po!上でモデルを編集することはできない。ポーズなどを変えたいときは外部の専用ソフトでポーズを変更し、保存して再びコミ Po!に取り込む必要があり、とても面倒である。

これらの点から実際に取り込んで使えるのはポーズなどが関係ない小物ぐらいだと考えても良いかもしれない。

以上が私が購入前に知っておくべきだと思ったコミ Po!の仕様だ。コミ Po!には体験版があり、一部の機能は制限されているが3.1、3.2の配置などは実際に体験することができる。購入を検討する場合は是非1回は体験版を使うことをおすすめする。公式ホームページは「<http://www.comipo.com/>」

4 おわりに



情報科学類誌 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

りざうど

1. あいさつ

ツインテールになるには属性力が不足していました、ひだるまです。今回はアンケート記事担当の次代引継ぎのため、後輩がところどころ書いています。よりまともな文章を書いている方がひだるまです。はっきりわかんだね。

↑○○は嘘つき -後輩

2. 今回の粗品

アンケートを直接編集部に持ってきていただいた方に差し上げている珍品もとい素敵グッズ。当編集部は3C棟213です。インターホンを鳴らすと編集部員が顔を出します。

今回は皆さん待望の新グッズが入荷しました！例によって先着優先とさせていただきます。

2.1. メモリ型定規

皆さん待望のメモリ型定規！最近ではDDR2系が逆にレア！でもどれに当たるかは運次第！たくさんアンケートを送って欲しい定規をゲットしような！！！！！！

2.2. WORD ステッカー



本命はこちら。WORDロゴ（新）のステッカーが深夜の謎テンションによって発注！！数日の内に発送！！！！！！到着！！！！！！！！サイズは20mm×50mm。さりげないアクセントとしてどうぞ。当ステッカーに関して何らかの不都合が発生した場合当編集部では責任を負いかねますのでご了承ください^{*1}。

*1 三回見ても死にはしないはずです - 後輩

3. アンケート集計

Q1：所属を教えてください。

工学システム学類：1人

IMAGINE THE FUTURE.^{*2}：2人

浪人生（迫真）：1人

残留思念：1人

頭狂理k大学基礎工学部材料工学科 ~~学類~~

TKB以外は学科だ
わかったな？
↙

浪人生という単語は高校ではともかく大学では大した意味を持たないんじゃない（先制ネタ潰し）。

こっちは志望のところの「学科」を毎回「学類」に書き換えてきたんだぞ！これくらいいいだろ！！

残留思念！ 何時までも筑波に居たいと思わせる何かが大学にあったのでしょうか。

Q2：性別を教えてください。

男：1人

美少女：1人

浪人：1人

デュエリスト：1人

女：1人

IMAGINE THE FUTURE.^{*3}：1人

性別を書こうな。最近はゴタゴタもありましたが、自身に対して正直な性であれば構わないとひだるまは思っています。自分の胸に手を当てて、はいそこ揉まないで、今一度自分のチェックした性別を見返してくれよな。特にないか。そうか。

Q3：年齢を教えてください。

24歳、浪人です：1人

14さい（食べごろ）：1人

0x13歳：1人

17歳美少女：1人

1956年生まれ：1人

114515歳：1人

平均は19094.5₍₁₀₎歳でした。

食べごろとは何を指しているのでしょうか。

WORD読者にも安定して美少女がいるようで安心です。

*2 WORDでは表現上不適切なもの、記入のない事項に関して IMAGINE THE FUTURE. と置き換えて表記しています

*3 WORDでは表現上不適切なもの、記入のない事項に関して IMAGINE THE FUTURE. と置き換えて表記していますつつてんの

Q4：良かったと思う記事があれば教えてください。

- | | | |
|---------------------------|--------------------------|-------------|
| 1. 表紙 | 7. スピコちゃん | 13. 裏表紙 |
| 2. 号名 | 8. 今日から始める賞金稼ぎ | 14. アンケート用紙 |
| 3. 目次 | 9. Gentooカレー | 15. 配布場所 |
| 4. KindleでWORDを読む | 10. 実録！ベクシンスキー絵画探求記（実践編） | 16. 配布時期 |
| 5. 1452円で鳥取の老舗ラーメン屋に行けるのか | 11. WORD 読者アンケート | 17. 配布媒体 |
| 6. Clojure ライブラリレポート | 12. 編集後記 | 18. 冊子の厚さ |

集計結果：

5.：1 票	9.：1 票
7.：1 票	10.：2 票
8.：1 票	14.：1票

おっアンケート用紙に票が入っている！やっぱり細やかな気遣いや数代に渡って洗練されたレイアウトが評価されたんですね、いやー参っちゃうなあ、でもやっぱり分かる人にはわかるもんなんですねこういうの、それとも毎号の一発ネタとか実は楽しみだったんですね？もっと早く投稿してくれれば良かった

14. 消去法

【浪人生（迫真） 黒塗りの浪人 さん（24歳、浪人です）】

絶対に許さない。

消去法という事は、良い記事が無かったのですか。良い記事を寄稿してくだされれば良い記事が出来ますよ！^{*4}

^{*4} WORD 編集部では何時でも良い記事の寄稿を待ち受けています 注：最近投稿に関する規定ができたらしい <- 知らなかった - 後輩

10 男の裸体に興味をもっている

【IMAGINE THE FUTURE. IMAGINE THE FUTURE. さん (1956 年生まれ)】

美少女であるところのひだるまには関係ない話ですね。

↑おっ、上で自分が書いた文章に思いを馳せた後、胸に手を当てて考えてから発言してくださいね！-後輩

↑気持ちよかった。

5、10、8

3回見たら死ぬ絵を3回見たら死んでしまった。

【工学システム学類 ちんとしてぼ さん (0x13 歳)】

いいですか、

A. ソクラテスは人間である。

B. 人間は死ぬ。

C. 朝陽きゅん^{*5}マジ完璧メイド。

D. A、B よりソクラテスは死ぬ。

E. だが、海兵隊は永遠である。

F. よって、貴様らも永遠である^{*6}！

永遠なので頑張ってください。

7、これは偉大

【残留思念 IMAGINE THE FUTURE. さん (14 さい食べごろ)】

スピコちゃん良いですよ、自分コンビニ行くと何時もスピコちゃん買っちゃうんですよ～

9. レッドブル感があってよかった。それと STAR ☆ ANIS.

【頭狂理 k 大学基礎工学部材料工学科 kmin さん (17 歳美少女)】

レッドブル感。第七感か何かだろうか。

*5 大金持ちの次男坊で超絶美少女の父親、お優しいルナ様を母親に持つ天使の化身

*6 「口からク○をたれる前と後にサーをつけろ！！」

Q5：良くなかったと思う記事があれば教えてください。

10.：2 票

流石のベクシンスキーニ、良い記事も良くなかった記事も1位入賞です。

10. この絵を見たせいで浪人生になって社会的に死にました

【浪人生（迫真） 黒塗りの浪人 さん（24 歳、浪人です）】

浪人生と社会的死に関連性はない（名推理）。

「死んだら教えて」という文言を見て、

昔、「欠席してるやついるか？」と朝のHRで聞いてくる教師のことを思い出した。

【工学システム学類 ちんとしてぼ さん（0x13 歳）】

言わんとすることはわかりますが、この言い方だと欠席した子の友人が「〇〇君が休んでいます」と答える流れっぽいなどと思いました。友達……あつ（察し）。

ちなみに編集部でこんなことを言い出すと、「いかに死亡を検知して報告を行うシステムを実現するか」の議論が始まります。

1 1, Q.5 が抜けていました
(指摘)

【残留思念 IMAGINE THE FUTURE. さん（14 さい食べごろ）】



なるほど、前任者をシベリア送りだ

Q6 : 過去の記事に関する感想を教えてください。

Vol.4(2004 年)の登さんの AC の記事は AC で良かった。

もっと AC して (お願い)

【頭狂理 k 大学基礎工学部材料工学科 kmin さん (17 歳美少女)】

あーあのきじですか、あれはとてもそうでしたねー。はい、えー……。うん、われわれもこんごね！も
っと、こう、ああして……。

振られた記事内容を思い出せないそんなあなたに！ WORD の過去記事は

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

から見るができますよ！！

……載ってるの新ナンバリング(2007-)からでした orz。真偽を確かめたい君は広大なネットの海か辛く悲
しい現実を探し歩いてください……。

アラブから来た革命戦士がもう一回見たい。

【浪人生 (迫真) 黒塗りの浪人 さん (24 歳、浪人です)】

31 号「WORD は 2D プリンタで印刷されています号」より「WORD ☆ STYLE」の記事のことでしょうか*。
大学内で度々目撃情報が挙がるがあるので、運が良ければ見つかるかもしれません。許可のない撮影その他
行為についてはご遠慮ください。

つり革にレモンをはめ込む選手権がおもしろかった

【残留思念 IMAGINE THE FUTURE. さん (14 さい食べごろ)】

ああ～～ 偉大なる P 同志が J ○ のつり革にレモンをはめこんで行く姿という、涙と青春などが詰まった
あの回ですね～～～*8

*7 「何だよ、そんなの知らないぞ！」とガタガタ震える皆さん、過去の WORD 記事を読むのであれば、
<http://www.word-ac.net/> へどうぞ！

*8 「そんなのあったか？」となった皆さん、過去の WORD 記事を読んで記憶を確認しましょう！

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

Q7：手から石油を出す方法を教えてください。

なぜかこの質問には比較的まとも⁹に答えてくれるアンケ投稿者ズ。

泥にうまって長い時間たてば手が石油になってるんじゃないか？

【浪人生（迫真） 黒塗りの浪人 さん（24歳、浪人です）】

それは「手が石油になる」方法なんじゃないかなあ。とりあえずまだ仮説なので、是非実証レポートをお願いします！ 石油が出てきたらまた投稿してください！！あと、**某WORD娘とイ○娘に関連はありません**¹⁰。

Googleに「手から石油を出す方法」できいてみましょう

【IMAGINE THE FUTURE. IMAGINE THE FUTURE. さん（1956年生まれ）】



Yah O o 知恵袋エ……。というわけで、残念ながら有用と思われる情報を得ることは叶いませんでした。とある先生も「Googleで出てこなかったら論文になる」とおっしゃっていましたが、方法論を確立された方はチャンスですよ！！

⁹ まともとはいっていない

¹⁰ 3年ほど前に表紙に何度か出現していたWORDのマスコットキャラクター。某漫画のキャラクターに非常によく似ていると言われる。編集部にはこの話題に敏感な人間がいるとかいないとか。えね子ちゃんでもないよ。過去の表紙を見るには <http://www.word-ac.net/> にアクセス！

異能バトルモノとかに、手から石油を出せるやつ登上してほしいですね。

【工学システム学類 ちんとしてぼ さん (0x13 歳)】



異能バトルモノ (確信) ^{*11}。「登上」は原文ママ。直接戦闘中は安易に火をつけて戦うと自分の手が燃え始めるから、むしろ密閉空間を油で満たして窒息とか姑息な手段を使って、失敗しそうならその場で発火、みたいな。

アゼルバイジャン人になる。

【残留思念 IMAGINE THE FUTURE. さん (14 さい食べごろ)】

なるほど、つまり我々のような人間は来世ガチャ^{*12} にかけるしかないのでしょうか。要修正

皮フに藻を飼え。

【頭狂理 k 大学基礎工学部材料工学科 kmin さん (17 歳美少女)】

むしろ光合成とかできるようになって食物摂取しなくて良くなりそうですね。

神 と 和 解 せ よ 。

【IMAGINE THE FUTURE. むむむむむ さん (114515 歳)】

「来世は美少女にしてもらって可愛い女の子とイチャイチャできるようにする」という条件の下合意へ向けた話し合いが行われる予定です。

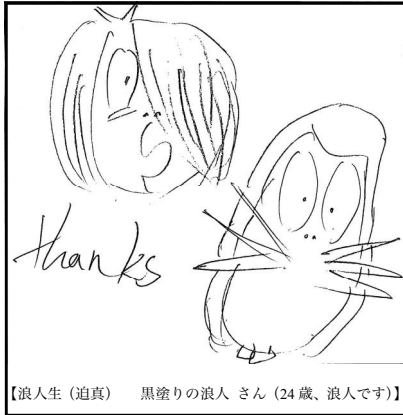
*11 画像は Amazon.com から。

書籍情報：カッコカワイイ宣言！ 3(ジャンプコミックス),地獄のミサワ著,集英社,発売日:2012/2/3

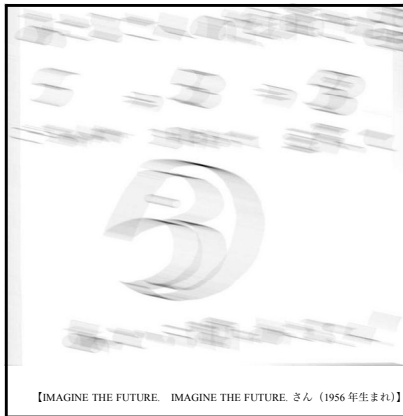
ISBN-13: 978-4088703817

*12 来世が何になるかはガチャガチャのようなもの。石油王の娘もワンチャンありますよ

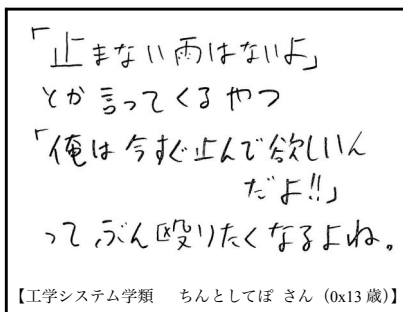
Q8：自由記述欄。



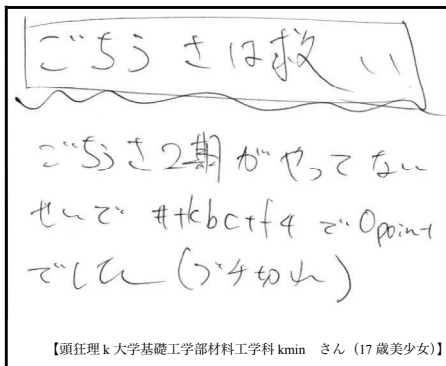
突然の鬼○郎^{感謝}。何に対する thanks なのでしょう。ご両親ですか、ご家族ですか、先生ですか、友達ですか、この前ズボンから発掘された 100 円ですか、毎週提供されるアニメですか、ツイエンテールですか、100 万本のエクスカリバーですか、羊飼いですか、八十八カ所のゲストヒロインですか、何を言っているかわからない旦那ですか、幼馴染みは何言っているのかいつもよくわからない薄幸そうな女の子ですか、勇者ですか、（仮）ですか。



スキャン失敗しました。

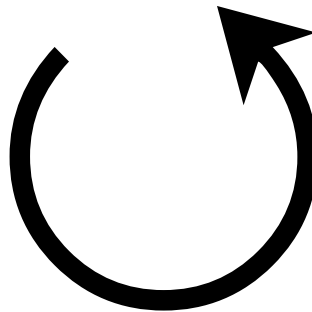
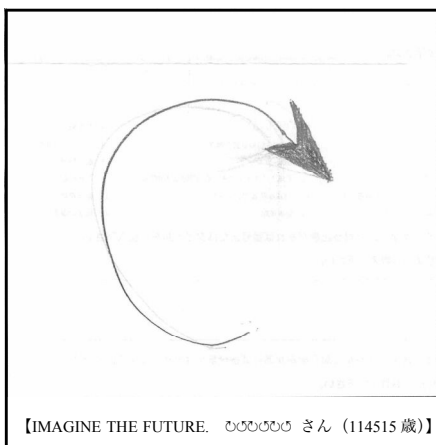


「じゃあ、ちよつと待ってて」
頬を思い切り殴られたというのに、圭太はにこやかに笑う。
「……は？」
殴られ返されるのを期待していた敏は毒気を抜かれ、喉から間の抜けた音を出す。圭太は敏に向かいもう一度微笑むと、踵を返し駆けだした。止める間もなく、どこかへ圭太が走り去った。残されたのはそんな事は意にも介さず降り続く雨と、握りしめたままの拳を持てあます敏のみであった。
だが、降り続くと思われた雨は、形容しがたい感情をそのままに敏が家に帰るまでに、その影すらなくしていた。顔を空へと向け続け、少し首が悲鳴を上げていた敏の耳にふと、
「これでどうだい？」
と、消えた圭太の声が聞こえた気がした。一瞬の目眩。その刹那に敏に視えた景色の中で、圭太は笑っていた。どこか神々しさとおどろかしさを感じさせる肉塊と、そこから止めどなく流れ飛沫く血の雨の中で。



2014年の筑波大学学祭「雙峰祭」の企画として行われたtkbctfの参加者の方のようですね。

救いとは欲して与えられるものではなくただただ施されるものなのではないでしょうか。



4. 終わりに

そういえばこの度、WORD アンケートの Web フォームを試験的に導入することになりました。天下のGoogle様のツールを用いますので、セキュリティ対策を自前でガチガチに組まなくても良いので楽。Webフォームのアドレスはこちら

<http://goo.gl/forms/wikjAX28U4>



試験的導入ですので、今後どうするかは未定です。

情報科学類誌

WORD

From College of Information Science

どうして発行するんですか号

発行者

情報科学類長

編集長

根本 晃輔

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)

2014年12月17日 初版第一刷発行
(512部)