

好きなこと、始めるなら今、手を動かさなくちゃ。

ISSN 2187-9664

USP MAGAZINE

for the sophisticated shell scripters

特集 シェルスクリプト・ハンズオン
春爛漫シェル芸人撩乱
勉強会体験記、ほか

TechLION 再録

小飼弾

人生を変えた
プログラム言語との出会い

新連載

すずきひろのぶの
“スズラボ通信”

シルネン・ブヤンジャルガルの
“自立への挑戦、情報整理技術の開拓!”

五十川員申

“メーカー達をつくってみた。”



From Editor

「こんな時間なのにまだ(もう)明るい!?」というほどに日も長くなり、気が付けば夏もすぐそこ。しかし「もう明るい」と言っている人は、早起き派ですか? 夜更かし派ですか? (私は夜更かし派です……)

さて今回の USP MAGAZINE は、summer 号だということにちょっと遅めの春の装い。

このページをめくるといきなり桜色のページが現れます。
TechLION レポートのページを見れば「出会い」という文字が飛び込んできます。
他にも 3D プリンターを初体験してみた人々の記事があったり、
モンゴルからやってきた新入社員のコラムがあったり……。
実は今回のテーマ、「始める」なのです。

今回、最もこだわったのはやはり特集記事「シェルスクリプトハンズオン」。
「なんかシェルスクリプトが最近スゴイらしい、という声は聞こえてくれど、
そのスゴイシェルスクリプトとやらを、どうやって始めればよいのかわからない」
このところよく聞かれるようになったこんな声に、何とか答えたいと思って制作しました。

新年度が始まったばかりの頃は、覚えることもたくさんあるし、何かと忙しいもの。
そんな忙しさも一段落した今だからこそ、これらの記事を是非読んでいただきたいと思います。

USP MAGAZINE 編集部

Contents

特集	こんなに身近なシェルスクリプトハンズオン	3
新連載	シルネン・ブヤンジャルガルの自立への挑戦、情報整理技術の開拓	
IT 美女図鑑	—— 私市智美	
新連載	スズラボ通信 すずきひろのぶ	
	メーカーたちをつくってみた。五十川員申	
	中小企業手作り IT 化奮戦記 第 7 回 菅雄一	
	漢の UNIX 第 6 回 後藤大地	
増井雄一郎氏に訊く、技術者哲学		36
	ユニケーエンジニアの作法 第 7 回	
	UNIX ネイティブの電子工作塾 第 2 講 大野浩之 高嶋健人	
	今私たちは何を学ぶべきか 第 9 回 大岩元	
	TechLION 再録 小飼弾が語る人生を変えたプログラム言語との出会い	
	うにつくすなやつら 第 3 回 長谷川猛	
	シェルスクリプト大喜利 第 9 回	74
Tech 数独		
天地概況	奈須瑠路 / 編集後記	

※ 薄字の記事はよりぬき版には収録されていません。是非正式版をお求めください。

特集

こんなに身近な シェルスクリプト ハンズオン

USP MAGAZINE 編集部

年度が変わって、まもなく3カ月が経過する。

新人は職場の空気に慣れると同時に現場の課題が気になる頃、上に立つベテランは自分に足りないスキルが見えてくる時期ではなかろうか。

「業務上のさまざまな課題を解決したい」「基礎からシステム開発について学び直したい」――。

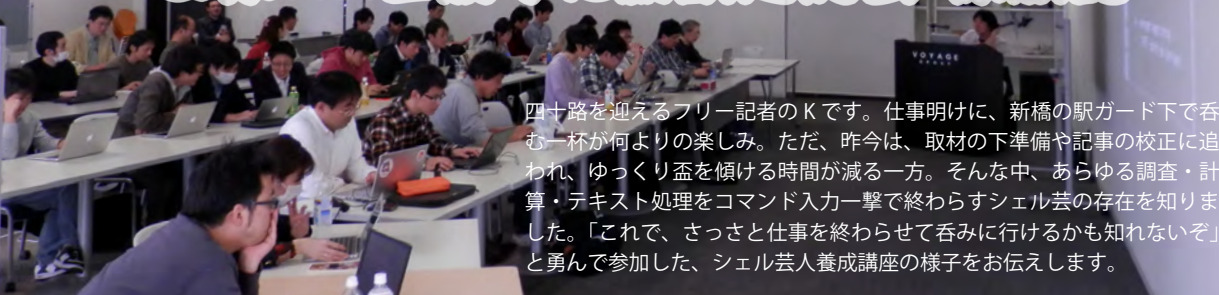
そうしたニーズに応えるべく本特集では、シェルスクリプトをこれから身に付けるビギナーから、

初心に立ち返ろうというベテラン勢までオススメの「学びの場」をご紹介します。

1つ目が、楽しみながらシェルプログラミングの真髄に触れられる USP 友の会主催「**シェル芸人養成講座**」、

2つ目が業務の改善や新事業の立ち上げを支援するユニケージ開発手法を体系的に学べる USP 研究所主催の「**ユニケージ開発手法教育講座**」である。どちらも実践的なカリキュラムだ。受講者の声を交えてナビゲートしよう。

第4回春爛漫シェル芸人撩乱勉強会 & 第22回勝手に新歓定例会 体験記



四十路を迎えるフリー記者のKです。仕事明けに、新橋の駅ガード下で呑む一杯が何よりの楽しみ。ただ、昨今は、取材の下準備や記事の校正に追われ、ゆっくり盃を傾ける時間が減る一方。そんな中、あらゆる調査・計算・テキスト処理をコマンド入力一撃で終わらすシェル芸の存在を知りました。「これで、さっさと仕事を終わらせて呑みに行けるかも知れないぞ」と勇んで参加した、シェル芸人養成講座の様子をお伝えします。



日頃の腕試しを兼ねて 実践的な問題に挑む

「素人、^{くろうと}玄人、どっちもバランスよく勉強できる——」「会社の新人さんにも、ぜひお声がけください！」という、こくちーずの文言に誘われて申し込んだ第4回シェル芸人養成講座。

ふだん、GUI画面を相手のマウス操作が中心で、UNIX系OSのコマンドを叩いたり、シェルスクリプトを書いたりする機会に久しい筆者。今更ながら場違いではなかったか、とひるむ心を押し鎮め、受講受付を完了します。「Linux環境のあるノートPC（Mac可）」を持参とあったので、Ubuntuを起動できるノートPCを携えて着座します。

シェル芸の目的は、「マウスも使わず、ソースコードも残さず、GUIツールを立ち上げる間もなく、あらゆる調査・計算・テキスト処理をコマンド入力一撃で終わらす」こと（シェル芸、という用語は最近、一般名詞化しているとか）。USP友の会が主催するこの勉強会は今回が4回目。回を重ねるたびに受講者数は増えてきました。

講師は、USP友の会会長 上田隆一氏。この日は、Web系をテーマにした全9問が出題され、1問あたりの回答時間に約15分が割り当てられます。

「ふだん、仕事でまったく文書の文字列整形しない、という初心者の方はいますか？」というので、恐る恐る手を挙げるとほかに、6～7名ほどの仲間がいます。私たちは、玄人、と呼ばれるベテランの方にサポートしてもらうことになりました。



複雑な制御構文の使用や 一般解の追求は逆効果!?

1問目は、与えられたHTMLのソースコードから、index.cgiというワードを取り除く、という問題。

問題1：肩ならし

次のファイルから「index.cgi」を除去して出力してください。
——ただし、p=index.cgiは残すこと。

```
$ cat 2.html
<a href="http://my.favorite.jk/index.cgi#jk"> 会 長
ブログ</a>
<a href="http://index.cgi.com/index.cgi?p=index.cgi"
">aho</a>
```

「p=index.cgiは残す」ということは、p=index.cgiの中に含まれるindex.cgiまで、消してはダメということか……。オーソドックスに、sedを使って、index.cgiを削除すると、一緒にp=index.cgiの中にあるindex.cgiまで消えてしまう……（ここまで5分近く経過）

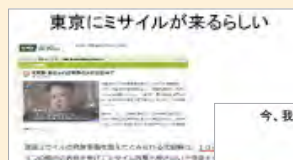
「できましたか？」

画面の前で硬直した姿を見かねて、隣の方（玄人さん）が優しく声をかけてくれます。「index.cgiではなく、『/』を付けて『/index.cgi』を、『/』に置き換える、と考えてみれば。そのあたりが、この問題の引っかけです」と私に小声でアドバイス。ついでに画面もチラッと見せてもらいました。

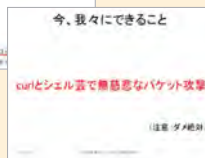
```
$ cat 2.html | sed 's/¥/index¥.cgi/¥//g'
<a href="http://my.favorite.jk/#jk"> 会長のブログ</a>
<a href="http://index.cgi.com/?p=index.cgi">aho</a>
```



今回が初参加、という受講者の方は「シェルはシステムの運用業務で使うことが多いが、今日の出題はどれも面白かった。Web上から店舗などのデータ一覧をダウンロードする面で使えるかも」とのこと。ちなみに今日の正解率を尋ねると「7～8割」。わが国とシェルの未来は明るい。



折しも、この勉強会の少し前ごろから、世界は不穏な空気に包まれていた。我々ができることは何か……。



さて、回答時間は終了。「これくらいだと、まだ玄人さんはうんちくを語れませんね」と講師。

正解を前方スクリーンに表示します。

```
$ cat 2.html | sed 's/index%.cgi//g'
<a href="http://my.favorite.jk/#jk">会長のブログ</a>
<a href="http://index.cgi.com/?p=index.cgi">aho</a>
```

別解として、awkを使った方法

```
$ cat 2.html | awk '{sub(/index%.cgi/, "", $0); print}'
<a href="http://my.favorite.jk/#jk">会長のブログ</a>
<a href="http://index.cgi.com/?p=index.cgi">aho</a>
```

「仮に2行目のp=index.cgiの中の、index.cgiも消したいのであれば、sed 's/index%.cgi//g'。awkの場合は、subではなく、gsubを使います」。sedの場合、gを/の後ろに加えないと、最初にヒットした置換文字列のみに、変換対象が絞られます。

しかし——、もし、同一行中に「/index.cgi」が2つ以上含まれる問題が出たと仮定するならば、隣の玄人さんが教えてくれた方法のほうが、よりエレガントのような……。

「目の前に出された問題を淡々と片付けることがシェル芸のポイント。類似問題でも使いまわしの利きそうな一般解を探そうとすると、かえってドツボにハマることがありますよ」と講師が注意を促します。

シェル芸は、~~卑劣なミサイル攻撃を行う敵国のサイ~~トを煩雑な仕事をまたたく間に撃退する最終兵器瞬間芸。その解法は必ずしも一通りではなく、解き方に個人個人の好みも出ます。ただし、解法における汎用性

の過度な追求は、逆効果。頭の中でひねり過ぎて、簡単なパズルが解けなくなることもあるそうです。



処理結果を見ながら sed や grep で刈り取る

2問目は、与えられたhtmlファイルの中からurlを抜き出す問題。そして3問目は、別に与えられたhtmlファイルの中のタブを綺麗に整えて出力する問題です(p.7参照)。次第に、解答例のパイプの数も増え、コマンドの列がどんどん長くなってきます。筆者はもはや、講師が披露する解答手順をターミナル画面でナゾるので精いっぱい。写経ならぬ、sh経状態に、ひとり突入です。

「最終的な答えはワンライナーで出力するのが望ましいのですが、迷った場合は慌てず、各コマンドの処理結果を見ながら、一つひとつ進めてください」と講師。

すなわち、解答例は、パイプでつないだ一行のコマンド群ですが、問題を解く場合は、コマンドを一個一個打ち込み、結果を表示させながら、解いていくのがベター。たとえば、2問目ではおもむろに抜き出す文字列を含んだ各行に共通する適当なキーワード(例:「href=」)をgrepで抜き出します。これだけで、出力される全行数がググッと減ります。その処理結果を眺めて、不要な文字列をsedと正規表現を組み合わせ、空白や改行に置換していきます。そこからさらに、処理結果を見て、キーワードを選び、grepで絞り込み、

～お子様と一緒に勉強！会場の協力で実現！～



9カ月になる息子さんと一緒に今回、勉強会に初参加のあしかさん。
今回の勉強会は、Linux女子部のメールで知ったとか。結婚を機にプログラマーとして勤務していた会社を退職。いまは子育てのかたわら、Java、Objective-C、PHPを使った主にWeb構築の仕事在宅でこなしているとか。「今日は、子供を連れてきて大丈夫かな、と思ったけれど、意外に温かい雰囲気よかった」と、ほっと一安心。

女性エンジニアの多くは、出産を機に職場を辞めてしまう人が、やはり多いそうです。けれど、子は親の背中を見て育つもの。コミュニティは働くママと未来のエンジニアを応援しています！



お子さんを抱っこさせてもらい、満足げなスタッフ



問題7：データの収集

本日の最高気温をワンライナーで出力してください。(yahoo や tenki.jp などから)

解き方の例：

まずは、curl で、yahoo の天気情報のページをダウンロード。大量に表示された HTML のソースコード中から、grep を使って「最高」という文字を拾い出す。この段階で、最高気温に関係する数字を含む対象の行数が、一気に3行程度に絞り込まれる。

```
$ curl http://weather.yahoo.co.jp/weather/jp/13/4410.html 2> /dev/null | grep 最高
<meta name="description" content=" 今日・明日の天気と風と波、明日までの6時間ごとの降水確率と最高・最低気温を見られます。">
<td><small><font color="#ff3300">最高気温 (°C) [ 前日差 ]<br>16 [ -3 ]</font></small></td>
<td><small><font color="#ff3300">最高気温 (°C) [ 前日差 ]<br>16 [ 0 ]</font></small></td>
```

処理結果を眺めつつ、tail と head で、不要な文字列をサクサク刈り込んでいく。

```
$ curl http://weather.yahoo.co.jp/weather/jp/13/4410.html 2> /dev/null | grep 最高 | tail -n 2 | head -n 1
<td><small><font color="#ff3300">最高気温 (°C) [ 前日差 ]<br>16 [ -3 ]</font></small></td>
```

sed と正規表現を使って、文頭から、最高気温を表す数字（この場合、16）の手前の改行タグまでを、空白に置換。

```
$ curl http://weather.yahoo.co.jp/weather/jp/13/4410.html 2> /dev/null | grep 最高 | tail -n 2 | head -n 1 | sed 's/. *<br>/'
16 [ -3 ]</font></small></td>
```

さらに、16 の後ろにある『[』以下も、正規表現を使って、空白に置き換えると、必要な文字列だけが残る。そして、パイプで、つながったワンライナーのコマンド群ができあがる（コマンドの書き方は、ダウンロードする HTML や文字列整形の方法によって変わってくるので、あくまで参考とお考えください）。

```
$ curl http://weather.yahoo.co.jp/weather/jp/13/4410.html 2> /dev/null | grep 最高 | tail -n 2 | head -n 1 | sed 's/. *<br>/' | sed 's/[ ]*[ ]*/ /'
16 ← 最高気温を表す数字だけが残った
```

結果を見て、sed で刈る……。最終的にパイプでつながれた解答例のようなコマンドが並び、標準出力に抜き取りたい url や文言が抽出されています。

黙々と文字列を刈り続けるうちに、不思議と雑念も払われます。sh 経の効果でしょうか。

4 問目と7 問目（上のコラム）は、curl を使って実際にあるインターネット上の HTML ファイルをダウンロードし、そこから必要とする文字列を抜き出す問題でした。



正解は一つと限らず 講師を唸らせる解法も

第6 問は、curl や wget を使ってサーバーにデータを送る問題。xargs を使って、無慈悲な攻撃データのアップロードを行う様子を、ログを見ながら勉強します。

問題9：クローリング

どこかのサイトのトップページにあるリンク先のリストを作って、リストのページや画像等をダウンロードしてください。

- 相対パスにも対応しましょう。
- できたら再帰的にクローリング

プロードを行う様子を、ログを見ながら勉強します。

第8 問では、画像のダウンロード。

そして、9 問目はその応用を踏まえて、本日の集大成となる Web サイトのクローリングがお題です。

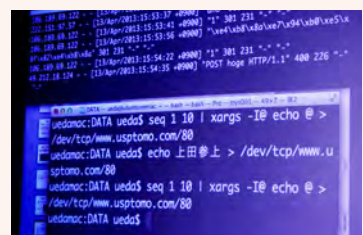
筆者は石になっていましたが、会場の皆さんは肅々と解いています。yahoo のページを対象にする場合、下記のような解答が提示されました。

```
$ curl http://www.yahoo.co.jp | sed 's/href="/>
$ curl http://www.yahoo.co.jp | sed 's/href="/>
```

一方、wget に再帰ダウンロードのオプション -r を

付けた wget -r を使えば、より短く同等の処理を実行可能という別解も披露されました。

それについて会場から



サーバーにデータを送る際、xargs で複数プロセスに分割処理。ログ（画面上）で挙動を観察（第6 問）

は「JPEG 画像だけを階層別に収集するのであれば、`wget -r -A JPG http:// ~`、と記述すればいいよ」と、指定した拡張子のデータだけを収集する方法や、「逆に、`wget -r -R [ 拡張子 ]`とすれば、特定の拡張子以外を取り放題」といった、オプションを駆使した、高度な解法が提示され、講師を唸らせる一コマもありました。

さて、勉強会の3時間は、あっという間でした。困った時は受講者同士、お互いのやり方を参考にしたり意

見を交わしたり、和気あいあいの雰囲気でした。

とはいえ、いろいろな正規表現やコマンドのオプションを知っていると、より手際よく問題を処理し、楽しめたはず。次回告知を見て、予習しておくといでしょう。

まずは目の前の課題に集中し、処理結果を見ながら、手を動かす。

この日は一般的なプログラム言語とちょっと違うシェル芸の真髄を、少し垣間見た気がしました。

第4回シェル芸人養成講座出題リスト

※本文で紹介した問題以外を掲載

問題2：urlのリスティング

次のファイルを作って、urlを抜き出してください。

【ヒント】 処理結果を見つつ、`sed`と`grep`で不要な文字列を切り取る。

```
$ cat 1.html
<a href="http://www.aho.com">
  アホどっと混む
</a>
<a href="#top">トップへ</a>
<a href="http://www.jk.jp">"JK"</a>
```

問題3：整形

インデントの汚いhtmlを整形してください。

——1行にtdを1個にすること

【ヒント】 `sed`を使って改行やインデントを要所に挿入。

```
$ cat 3.html
<table>
  <tr>
    <td>a</td><td>b</td>
    <td>c</td>
  </tr>
</table>
```

問題4：webページの切り取り

yahooのトップページからトピックスの見出しを抽出しましょう。

【ヒント】 ファイルのダウンロードは`curl`。htmlソースコードを見て、トピックス見出しを引っ張り出すキーワードをどれにするか。

解答例は、以下のURLに掲載しています。
<http://www.slideshare.net/ryuichiueda/20130406-18763665>

問題5：数値参照の変換

次のような数値参照、実体参照文字列を読めるように変換してください。

【ヒント】 Unicode文字を変換するコマンドは何？

```
$ cat numref
&#25105;&#12293;&#12399;&#12471
:&#12455;&#12523;&#33464;&#2015
4:&#12384;&#12290;
```

問題6：ポスト

`http://www.usptomo.com/TOMONOKAI_CMS/CGI/hoge.cgi`に、ポストでデータを送りつけてください。

こんなCGIスクリプト→

【ヒント】 使うコマンドは、`curl`や`wget`、`telnet`もアリ？

```
#!/bin/bash -xv

echo "Content-type: text/html"
echo ""

echo あなたのポストしたデータは
dd bs=$CONTENT_LENGTH
echo
echo です。
```

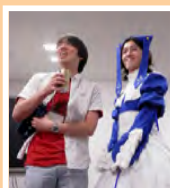
問題8：画像をダウンロード

yahooのニュースページ `http://headlines.yahoo.co.jp/hl`等々から、ウェブページに掲載されている画像を集めてください。

【ヒント】 まずは、htmlソースコードの中からイメージ（JPGやGIFなど）に紐づいているキーワードを、コマンドと正規表現と組み合わせで、じっくり引っ張り出す。

#!/beer/bash

懇親会 [#!/beer/bash] はLTで盛り上がる！



勉強会当日、別フロアのイベント（リカルハッカソン）に参加していた方がBeerBashに突如来訪し、動揺する講師。

懇親会は、乾杯の音頭の後に、参加者による交流会。さらに、有志による飛び入りLTも行われました。今回は、新規事業の立ち上げ支援などを展開するVOYAGE GROUPの渋谷オフィスをお借りしました。電源や無線LAN環境も含めて、たいへんお世話になりました。また、今回もUSP友の会有志により、Ustreamによる中継が行われました（スタッフの皆さま、ご協力有難うございました）。

次回勉強会のお知らせ

今回、参加に先だって、持参のノートPCにOpen usp Tukubaiをインストールしましたが、Tukubai コマンドを利用する問題は出題されず。未整形の文字列を処理する問題が中心だったためですが、今後のシェル芸人養成講座では、Tukubai コマンドが威力を発揮する整形文字列処理の問題が出るかも。Open usp Tukubai をぜひ、インストールして、会場に足を運んでみてください。

（インストール方法は、下記URLをご覧ください。または、USP MAGAZINE前号vol.8 P.15にも掲載しています）<https://uec.usp-lab.com/TUKUBAI/CGI/TUKUBAI.CGI?POMPA=INSTALL>

次回は、6月22日(土)開催予定です（変更の場合がありますので、詳しくはUSP友の会HPを検索の上、お確かめください）。

ユニケーj開発手法 教育講座レポート

はじめまして、今年の4月からUSP研究所で、お仕事見習い中のHです。いまは業務システム開発・運用について勉強中。やはり基礎が大事ですよ、ということで、ユニケーj開発手法教育講座を受講しています。この講座を一通り受ければ、業務で使われるバッチ処理システムやウェブアプリを開発・運用するために必要なスキルが身につきます。各コースを受講された皆さんの声を交えつつ、本講座の全体像、自分にあったコースを選択するポイントをお伝えします。



ユニケーj開発手法を 体系的に学べる

「ユニケーj開発手法教育講座」（以下、教育講座）とは、USP研究所が開催するエンジニア向けの講座のことです。ユニケーjエンジニアとしての知識量、そして問題解決能力を高めることができます。

そもそも、ユニケーj開発手法とはどういうものでしょうか。下段にまとめました（囲み記事参照）

シェルスクリプトは、UNIXとともに誕生したスクリプト言語です。そのシェルスクリプトをベースにしたユニケーj開発手法で作られたシステムは、高額なハードウェアや高度な専門知識を要するDBMSといったミドルウェアを基本的に使いません。また、現場の社員を主体としたシステムの開発・運用・保守が可能です。「システムを納品して終わり」ではなく、現場のユーザーがニーズに合わせて自らの手で拡張発展、メンテナンスしていくことを前提にしています。

これまで教育講座を受講されたのは、「業務を知っているけれども、現場の課題解決に資するシステム開

発の手法を身につけたい」という業務部門のユーザーさんや情報システム部門の担当者さん、または従来とは視点の異なるユニケーj開発手法をもとに顧客のシステム開発・運用を支援したい、というSI企業のエンジニアさんなどです。

教育講座では、コマンドの基礎やシェルスクリプトの書き方を学ぶほか、システム開発の起点となる実データの整理や配置の仕方を学びます。「WindowsやMacのOfficeツールしか使ったことがないのだけれど」という人でも、大丈夫です。S1、S2講座では、前提となるUNIXの基礎やエディター（Vim）の操作方法も一から身につけられます。

ユニケーj開発手法においてシェルスクリプトの作成に使うコマンドはUNIX標準のものもありますが、ユニケーj標準コマンド（usp Tukubai）を利用することで、開発生産性および日々の業務効率が高まります。ユニケーj開発手法では伝票などの実データをもとにアプリケーションを開発していきますが、データを取り扱う際に各種コマンドが威力を発揮します。これらのコマンドは、K1で学ぶことができます。

ユニケーj開発手法とは

本誌でもたびたび紹介してきたユニケーj開発手法とは、Linuxに代表されるUNIX系OSとテキスト、シェルスクリプトを組み合わせ、「やすく・はやく・やわらかく」情報システムを構築できるソフトウェア開発手法である。その主な特徴は次の通りだ。

- ・UNIX系OSのベーシックな機能を利用
- ・データベースにRDBではなく、テキストファイルを使う
- ・データの制御、操作はSQLの代わりに、UNIX系OSの性能を引き出すコマンドやシェルスクリプトを使う
- ・データ構造は、いわゆるキー・バリュー型データストアの考え方
- ・メンテナンス性の高い見通しのよい業務システムを開発できる
- ・柔軟なセキュリティポリシーに対応



演習を中心とする 濃密なカリキュラム

全講座は1日もしくは2日間のカリキュラムが組まれています。実際に、K1の講座を一例に、教室の様子をご紹介します（p.9上 囲み記事）。PCや受講用のテキストは、すべて講師側で用意しているので、受講者は基本的に「手ぶら」でOKです。大事なのは、実際に手を動かして、操作に慣れること。シェ

教育講座を受講した受講生Hの一日 ～「K1」コマンド学習編の場合

◆時間割（「K1」コマンド学習編）

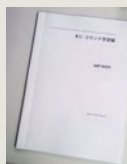
	1 日目	2 日目
10:00	はじめに ①	コマンド実習 2 sm2, sm4, sm5, ③ ysum, kasan, ratio, join0, join1, join2, loopj
11:30		
13:00	AWKの使い方 ②	コマンド実習 3 getfirst, getlast, up3, tarr, yarr, map, tateyoko
	sort と uniq	コマンド実習 4 marume, divsen, comma, keta, grep と正規表現, tr, sed
	コマンド実習 1 head, tail, ctail, gyo, retu, self, delf	コマンド実習 5 han, zen, mojihame, keycut, その他コマンド ④
17:30		

※ 紫色の文字はUSP研究所がリリースしたTukubaiコマンドです。

tateyoko コマンド
すごいねえ！



メモ① 講座テキストが配布される



講座の初めに、こんなテキストが配られます。目次から2日にわたる講座の内容がわかるのですが、ページ数は181。こんなにたくさん覚えられるのかなぁと最初不安でしたが、あとあと自習や、コマンドリファレンスとして使えるように考慮されているそうです。

技能検定に合格するために、必死でやっています！

メモ② 何でもできる AWK の実力を知る

講師がテキストの例を実際に入力し、実行結果を画面で見せてくれました。AWK一つで実に様々な処理ができることを理解しました。でも for や if 文が出てくると処理は一気に複雑化。講師曰く「最初に AWK で苦労しておくと、後で Tukubai コマンドの難みがわかります」とのこと。



```
$ cat hoge | awk '1<="20110303"[print $1,$2,$4]'
20110302 パナナ 4
20110302 リンゴ 5
20110303 パナナ 9
20110303 ナシ 3
```

メモ③ 列の合計値が簡単に出来る sm コマンドを学ぶ

2 日目は最初に sm2, sm4, sm5 を学習。表計算でよくある各列の合計を求めるコマンドでした。AWK でもできることは1日目の実習で想像できていましたが、sm シリーズだと簡単にできてしまいます。UNIX にはこの手のコマンドがありませんけど、COBOL や RPG にあった sum up みたいなものですね。

```
$ cat URE.IMO
001 さつまいも 20110405 32 1600
001 さつまいも 20110406 24 1200
001 さつまいも 20110407 49 2450
002 じゃがいも 20110405 102 5100
002 じゃがいも 20110406 98 4900
002 じゃがいも 20110407 121 6050
```

```
$ cat URE.IMO | sm2 1 2 4 5
001 さつまいも 105 5250
002 じゃがいも 321 16050
```

メモ④ mojihame コマンドで、<table> の中身などがとても簡単に作れることを知る

HTML で <tr> ~ </tr> の部分について一行だけ書いておいたテンプレートを用意しておき、この mojihame コマンドを使ってデータを与えれば、データの行数分の <tr> ~ </tr> が、値の入った状態で複製されます。これは HTML 画面作りへの応用でとても重宝する気がして、真っ先に覚えてしまいました。

```
リンゴ 40 円 3 個
キウイ 78 円 5 個
バナナ 98 円 2 房
トマト 50 円 1 個
レタス 220 円 1 玉
バセリ 100 円 2 袋
```

```
<table border="1">
<!-- MOJIHAME -->
<tr><td%1</td><td%2</td><td%3</td></tr>
<!-- MOJIHAME -->
</table>
```

mojihame → HTML

ルスクリプトの書き方に関する手順（お作法）は K2 で学びます。

ユニケーj開発手法は、従来型の開発手法と異なり、業務の中でやりとりされる各種伝票に記載される実データを起点に、集計や分析、出力などの処理を行うアプリを作ります。K3 では、K1、K2 で学んだ知識を基に Ajax を使った本格的な Web アプリケーションの作成を学びます。K3 は2日間に分かれ、初日は郵便番号から住所を検索するアプリを作成。2日目は、架空のスーパーマーケットを想定し、販売情報参照アプリを実際で作ってみるという内容です。皆さん、業務で類似の帳票類やシステムを利用しているだけに熱心に受講されています。K3 受講者の方からは、「CGI を本格的に使いつつ、ユニケーj開発手法で作ったシス

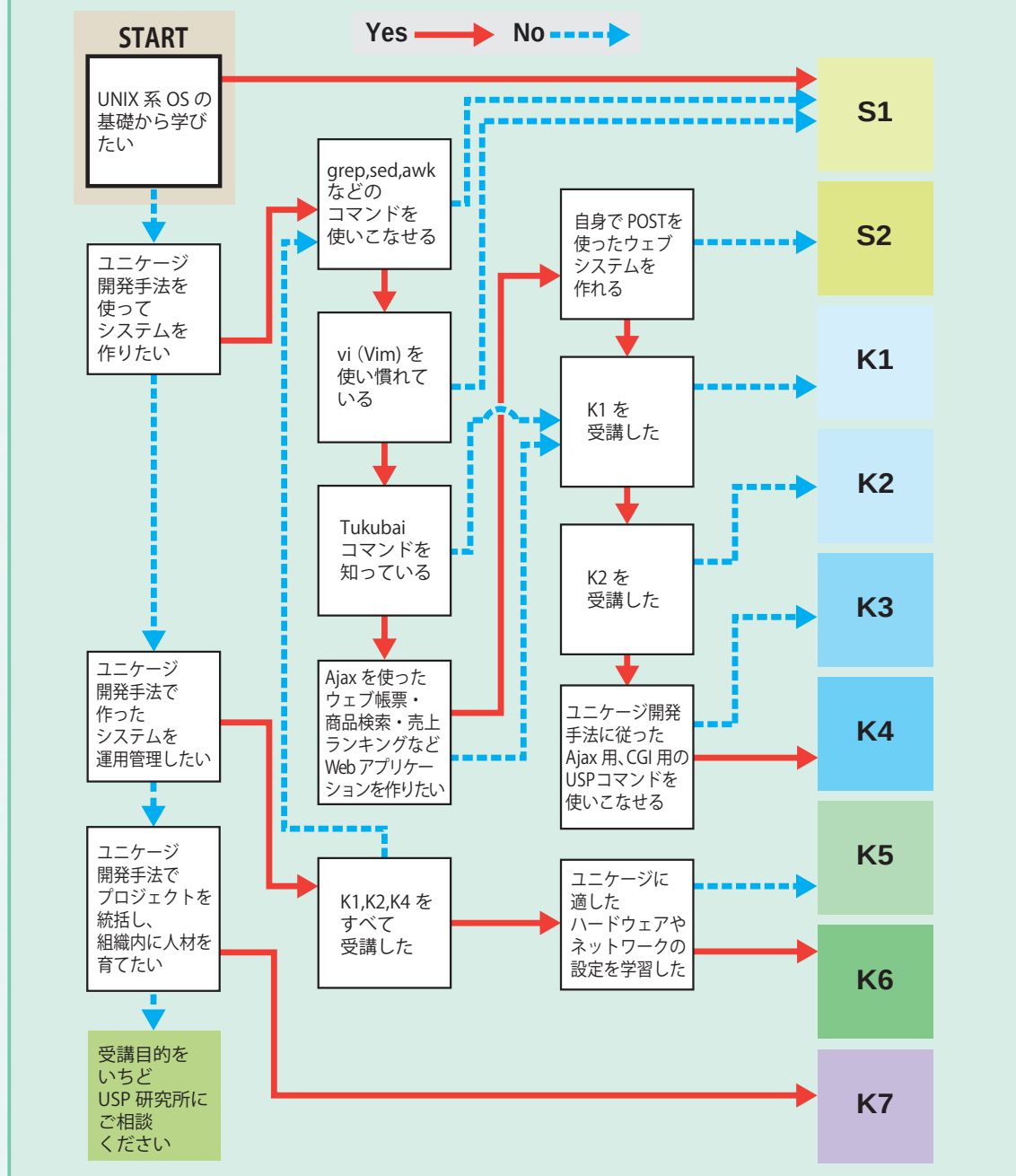
テムの処理結果を Web 上に表示するまで体験できた。わかりやすかった」という感想をよく耳にします。



業務で使う実データを起点に アジャイルなアプリ開発

ユニケーj開発手法で取り扱う実データは、日々業務でやりとりされる帳票データなどです。それをもとにマスタデータを作ります。ただし、原本となる生データをいきなり使うのではなく、テキストファイル上でコマンドによって処理しやすいように文字列を整形します。このような下準備をして、コマンドでいつでも処理できる状態になったデータ群が、L4 または POMPA データといわれます（コマンドとオプション、ファイル名などを入力し、ポン、と実行キーを押すと

あなたが最初に受ける講座はこれ！ 教育講座フローチャート



パツと処理結果が出る、という意味)。K4では、このようなユニケー開発手法におけるデータ構造やデータモデリングを学ぶことができます。さらに、K5では、ユニケー開発手法に適したシステム環境（ハード、ソフト、ネットワークなど）を学習します。K6では運用管理を行う上でのポイント、具体的にはログデータの活用したエラー分析や運用監視アプリケーションの作成方法などを勉強できます。今後開講予定のK7

では、プロジェクトマネジメントにおける人材育成やプロジェクト推進のコツを学べます。

教育講座の開催場所は、主に、USP 研究所本社のある東京・西新橋にある会場です。過去には大阪と沖縄で、出張形式の教育講座も開催されました。大阪では、業務システム開発の内製化に取り組む大手百貨店の社員の方を中心に、講座が開催されました。

さて、私（筆者 H）もこの数カ月間で、いろいろなコー

教育講座ラインナップ

	名称	内容・習得事項	日数
 Win/Mac ユーザーも 一からわかる	S1 ユニケージのための UNIX 入門編	ユニケージでデータを加工するために必要な前提知識を学びます。 内容：UNIX 系 OS の生い立ちや考え方、ファイルやディレクトリ、コマンドの操作、エディタ（Vim）の実習。	1
 Web アプリの基礎を 総ざらい	S2 ユニケージのための Web 入門編	ユニケージ開発手法で Web アプリケーションを構築するために必要な前提知識を学びます。 内容（解説と簡単な実習）：HTML、JavaScript、CGI	2
 データを操る コマンドの強い手	K1 コマンド学習編	ユニケージ開発手法において使用頻度の高いコマンド（特に grep, sed, awk）、USP コマンドを学習します。	2
 大量データも サクサクさばく	K2 シェルスクリプト学習編 （バッチ処理）	ユニケージ開発手法で用いる bash の文法、ユニケージ開発手法にしたがったバッチシェルスクリプトの書き方を学習します。	2
 会社の帳票システムを Web 化できる	K3 シェルスクリプト学習編 （ウェブアプリケーション処理）	ユニケージ開発手法にしたがった web アプリケーションの開発方法、Ajax 用、CGI 用 USP コマンドを学習します。	2
 実データのマネジメント におけるコツ	K4 ユニケージアーキテクチャ編	ユニケージ開発手法の標準的なシステム構成を学びます。ハードウェア配置、データの整理方法、ソフトウェアの起動スケジューリングと連携、エラー処理や障害対応、セキュリティについて、ユニケージ開発手法の考え方・作法と共に具体的に学習します。	1
 最適な動作環境を つくる	K5 ユニケージ開発環境 セットアップ編	ユニケージ開発手法でソフトウェア開発するためのコンピュータの導入、OS のインストール、サーバ・ネットワーク環境の整備方法を具体的に学習します。	1
 セキュリティや 保守も万全	K6 システム運用・管理編	ユニケージ開発手法のシステム監視方法、運用監視、障害復旧、セキュリティ、アクセスコントロール、バージョン管理、ソフトウェアの保守を学びます。	1
 継続的な組織作りの ポイント	K7 プロジェクトマネジメント ・人材育成編	ユニケージ開発手法のシステム開発の進め方を学習します。ユニケージ開発手法に適した人員配置、要件定義方法、アジャイル的な開発方法、テスト方法、「床屋方式」や「一休方式」などの独特の方法を学びます。	1

スを受講してきました。ただ、教育講座は皆さんのニーズや、お持ちのスキルによって、必要なコースだけをピンポイントに受講することもできます。

「でも、どの教育講座を選べばよいのか、わからない」というご質問があるのも確か。そこで皆さん一人ひとりにピッタリの講座がわかるフローチャートを作成しました（上図）。参考にしてみてくださいね！



講師

「ユニケージ開発手法の目的は、業務上の課題を解決すること。たとえば、業務効率を上げる、コンピュータ画面に向かう時間は最小限にしてお客様と接する時間を増やす、新たな業務を立ち上げるなどがあります。そのために業務知識のある現場の方自らがシステム開発・運用・保守を行うことが可能になっています」

教育講座のご案内

教育講座の最新スケジュールや、申込みなど、より詳しい情報は Web サイトをご覧ください。

USP 教育

検索



増井雄一郎氏
に訊く、

技術者哲学

面白いという気持ち、作りたいという衝動には逆らわない。
迷った時、Noと言われなければ、
Yesにフラグを立てて、信じたほうに動いてみる。



増井雄一郎 1976年生まれ。大学時代に北海道でWebシステム開発会社を起業。フリーランスを経て、Ruby on Railsを活用したWebアプリの開発や書籍の執筆などを手がける。2008年に渡米し、Windows95/98のチーフアーキテクトだった中島聡氏とともにアプリ開発会社を起業。2010年に帰国後、アプリSDK「Titanium Mobile」のエバンジェリストとして活動を開始。2012年に退職し現在、FrogApps取締役CTO。自身のブログは「@masuidrive blog」。

ここへ辿りつくまでにいくつものトライ&エラーがあった。RDBMS 不要のCMS「PukiWiki」、アプリSDK「Titanium Mobile」、iOS/AndroidアプリをRubyで作成できるフレームワーク「MobiRuby」など数多のツールの開発者として知られる増井雄一郎氏は、学生時代に起業後、海外でビジネスを立ち上げた。ただ、思い通りに事が運ばない局面に幾度も遭遇した。そういう時に必ず確かめる。自分が何をやりたいのか。そして、思いやイメージを周囲に伝える。ツールで。対話で。ブログで。その時たとえ苦手な英語を使うことになっても。むしろ、自らを表舞台にさらけ出すことで人とつながり、道が拓ける。(まとめ：柏崎吉一)

——2012年11月のTechLION vol.10では「どこでも生きていけるエンジニアを目指した後に見えるもの」というお題で登壇いただきました。その後、取り組まれていることを教えてください。

増井 当時、僕はAppcelerator Inc. から株式会社FrogAppsに軸足を移した頃でした。現在はFrogAppsで主に、お料理写真共有アプリ「ミイル (mill)」というサービスの開発・運用を行っています。CTOという立場で、システム開発と、運用まで見据えた開発を任せられる人材の育成を担当しています。2011年にリリースしたミイルは今年5月に累計30万ダウンロードを突破しました。

——増井さんはどこでコーディングを身に付けたのですか。

増井 ほとんど独学です。コンピュータに興味を持ったのは中学で、高校時代にMSXを買ってもらい、毎日プログラム漬けでした。僕は北海道札幌市にある工業高校に通っていて、当時、レジ打ちのアルバイトを募集していた近くのDPEショップに面接に行っただです。履歴書の特技欄にプログラミングと書いたら、『ちょうどPC担当者が辞

めて人手が足りない』というので、僕がそのお店の販売管理などの業務系システムを Lotus1-2-3 とか dBase で作ることになりました。出来たシステムを見た取引先の会計士の人に「うちのお客さんのところもやってよ」と頼まれ、冬休みに道内にある企業の販売や棚卸の管理、送り状発行に使うシステムを作って、モデム通信などで納品しました。アルバイト代で、8 万円ほどもらった記憶があります。自宅でコーディングして、プリントアウトしたものを高校の授業の合間にデバッグして、帰宅してからまたコーディング、という生活をほぼ 3 年続けました。C やデータベース、OS など、いろいろな本を買って勉強しました。みんな僕が出入りさせてもらっていた会社の経費で落としてもらったんですけど(笑)。

札幌大学では経営学部でお世話になった八鍬先生の手伝いで、ある企業とイントラネットキットを構築する案件に関わりました。インストール用のドキュメントを作成し、Web 上で公開したところ、当時、オン・ザ・エッチの社長だった堀江さんの目に留まった。それが縁で、北海道にいて同社の Web システムなどをリモートで開発するようになりました。ちなみに当時、同社の CTO は小飼弾さん。まさか小飼さんと将来、メディアの記事で対談したりするようになるとは、その頃、夢にも思いませんでした(笑)。

学生時代に起業

増井 僕は、いままで 2 度会社を潰しているんです。一度目が、携帯電話向けアプリの開発を主事業とする株式会社ハイセックです。二度目が、2008 年、(Windows95/98 のチーフアーキテクトだった)中島聡さんと起業した Big Canvas Inc. という iOS/iPhone 向けアプリの開発会社です。

ハイセックでは仲間のエンジニアと一緒に、大手通信キャリアの公式 Web サイトなどを受託開発していたのですが、営業や経理なども僕の担当でした。会社勤務の経

験がなく、学生でいきなり起業ですから、慣れない社長業に右往左往しました。商談における根回しや交渉術に疎かったのです。会社はそこそこ食べて行けたんですけど、一方で、僕はやっぱりコードを書きたかった。僕は 26、7 歳でしたが、社長業は 40 歳からでもできるけれど、エンジニアは 40 歳からやるのは厳しいような気がした。本心を仲間に打ち明けたら、いいよ、と。彼らは、いったんフリーランスになり、ハイセックのお客さんを引き継ぐ、ということで会社を畳みました。2003 年です。いまそのときの仲間の一人が、株式会社インフィニットループという会社を札幌で立ち上げて活躍しています。

米国でのビジネス立ち上げ

——増井さんも解散後は、フリーランスになったのですね。

増井 そうです。その間、Linux ディストリビューションのエヴァンジェリストやシステム開発手法などを社会人向けに教える講師の職を得て、そのかわり、Ruby を使ってアプリやフレームワークを開発するようになりました。Ruby を知ったのは 98 年ごろですが、2004 年に Ruby on Rails が出て、これは仕事に使えるぞと思い、本格的に勉強しました。2006 年にサンノゼで Ruby on Rails のカンファレンスがある、というので行きたい、と妻に言ったところ、行って来たら、と背中を押された。

中学以来、英語をろくに勉強していないのに初めての海外一人旅ですから、イミグレをしどろもどろで通ったりと、散々でした。でも、すごく良い刺激を受けました。帰国後、妻に話をしたところ「それならアメリカで働いたら英語も覚えられていいんじゃない」と勧められました。ただし、米国で日本人がエンジニアとして長期にわたって働くには、米国法人に勤めるか、現地法人のある日本法人に勤めるか、の大きく二つの選択肢しかなかった。それで、当時海外進出を検討していた、ある日本企業に勤めたのです。

その頃ですね。読んでいた中島聡さんのブログ記事に、

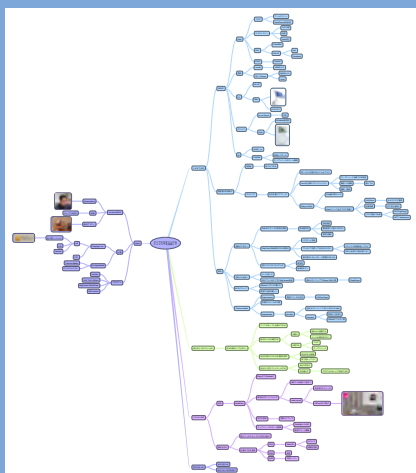
お料理写真共有アプリ「ミイル」

「ミイル」の最大の特徴は「食に最適化した写真フィルター機能」。投稿写真の 40% が「Facebook」または「Twitter」にも投稿されている。初期バージョンのコアとなる画像処理エンジンを増井氏が開発した。



リラックマの iPhone ケースなど。「誕生日の Wish リストに書いたら、たくさんの方から頂いてしまいました」。

マインドマップ



「これまでの10年」「現在」「これからの10年」について、記した増井氏のマインドマップ。中心にあるテーマは、どこでも生きていけるエンジニアを目指して。

米国でのエピソードを見つけて、現地のことを伺おうと連絡したのは。直接の面識はありません。ご本人のメールアドレスも知らなかったの、勤め先の会社に連絡したら、「多忙なので」と担当者に断られた。広報としては正しい対応ですが、僕も諦めずに『そのメールを、中島さんに転送してください』と食い下がったところ、後日、ご本人から直接連絡が来て、ランチミーティングで30分だけならば、と時間を作ってくれたのです。それでお会いし、話しているうちに意気投合し、米国に会社を作ろう、となった。「奥さんも一緒に行くのだから、確認とらなくて大丈夫?」と尋ねられたのですが、その場ですぐ問題ないです、と言い切っちゃいました。彼女は、僕の事後報告にあまり驚きもせず、予想通りOKしてくれました。それで、僕も勤めていた会社を退職し、米国に渡りました。

リーマンショックが直撃

——中島さんとはどんな仕事をするつもりだったのですか。

増井 実は、何をやるかまったく決めてなかったんです(笑)。僕に英語やビジネスのいろは、が身についてないので半年間、現地の英会話学校やビジネススクールに通いながら考えよう、ということにしました。その矢先の2008年3月にiPhoneが米国でリリースされて、手に取った僕たちは、これだ!とiPhoneアプリを作り始めたんです。というのも、同年7月までにアプリを完成させれば、AppleStoreのローンチにアプリを登録できる、というキャンペーンがあったので。2人でほぼ3カ月間、大急ぎで作りました。英会話学校とビジネススクールの話

は後回しです。それで、何とかリリースできたと思ったら、その年の9月にリーマンショックが起きた。広告出稿がガタ落ちです。ビジネスモデルの見直しを余儀なくされました。セールスやマーケティングの強化も検討しましたが、僕らだけでは回しきれない。僕の英語力やビジネススキルが足りず、組織を引っ張り切れなかったことが悔やまれました。結局、Big Canvasは中島さんの別会社に譲渡する形で閉じたのです。

——そこからAppceleratorのテクニカルエヴァンジェリストとして活動するようになったんですね。

増井 Big Canvasを解散したころ、Appceleratorから連絡がありました。Titanium Mobileの開発をお前がやれ、と。僕は2010年の初頭から、一ユーザーとしてTitaniumを実際に使い、ここが不便だと批評を書いたり、コードを送りつけたりしていたので、目をつけられたのでしょう。ただ、Appceleratorはその頃、まだ売上も少なく、とても僕にWork Permitのあるビザを提供できる状況になかった。しかし、僕の持っている会社付きのビザに期限が迫り、日本へ帰国しようとした前日に、Appceleratorへ900万ドルの増資という発表を偶然目にしたのです。これは、もしかしたら、と直感しました。僕は、同社のトップに、ビザを発行してくれないか、と直談判しました。その間、日本行きの僕と妻の荷物は船会社に連絡し、倉庫に留め置くように手配して。

交渉の末、僕はビザを申請してもらう約束を取りつけ、晴れてAppceleratorの社員になりました。ところが、日本に戻り、ビザが取得出来るまでということでエヴァンジェリストとして活動を開始したところ、ある月にTitanium Mobileの全ダウンロード数の1/3が日本からという状態になってしまいました。これで彼らは日本市場に着目。僕に日本に残って活動を続けてほしい、と遠回しに言ってきたのです。僕は、米国で働きたい気持ちもあったのだけれど、体制が整ったら戻ってきてもらうからというので日本に残りエヴァンジェリストとして活動を続けることにしました。給料はドルベースだったため、当時の円高で2割くらい実質目減りしたのは結構きつかったです。後で、Appceleratorを辞めた後は反対に一気に円安です。何の呪いかと思いました。

話は前後するのですが、私は前から自分の作りたい物をリストとして公開しており、それを読んだ友人が「その

中の一つと同じ物を作ってる人が日本にいる」と教えてくれました。それで帰国直後に中村仁を紹介してもらいました。実は、中村は、豚組というレストランを経営しており、お客さんの Twitter 情報を利用して、商品開発や顧客サービスの向上につなげるアイデアを形にしようと動いていました。それがきっかけで、料理の写真をおいしく撮影・アップできるミールの開発にシフトし、初期の画像処理のエンジンは僕が、開発しました。Appcelerator から、FrogApps に移ったのが、ちょうど、TechLION vol.10 に出させてもらった 2012 年 11 月頃です。

コードを書く上で気を付けること

——増井さんは、短時間でコードを書くことで有名ですね。

増井 飛行機などでの移動時間を使って作ることもありますね。時間がはっきり決まっているので、集中しやすいです。MobiRuby も米国から日本へ帰国する飛行機で、基礎となる部分を作りました。ヒューストンから 12 時間以上のフライトですが、その飛行機が、座席に AC 電源のあるタイプで、ネットワークはないけど、かえって邪魔が入らず作業がはかどりました。

開発で気を遣うのはユーザーに簡単に使ってもらえるようにすることです。PukiWiki で DB を使わないのもそういう観点からでした。当時、DB を使える ISP が少なかったし、DB の設定自体が面倒くさいので、避けたかった。MobiRuby のインストーラーをシェルスクリプトでごりごり書いたのも、インストールに必要なものを減らすため。初めの一步で躓くとユーザーの心が折れちゃうんです。シェルはどんな OS にも入っていてポータビリティが高いのでしばしば使うことがあります。

意識的に知名度を上げる

——好きなことをやる、といっても、食べていかなければなりません。増井さんにとって、お金は仕事のモチベーションですか。

増井 仕事を受けるかどうかの判断基準に報酬はもちろんありますが、それ以上に、面白いかどうかはかなり大きいウェイトを占めます。趣味で作っているものは本当にモノになるかどうかわかりませんが、面白いからやっています。ただ、仕事は仕事できちんと区別しています。基本的



風呂グラマー

寝るとき以外は、ほぼコーディングをしているという増井氏。「昨日も風呂場で作っていた」とのこと。「結構な頻度でやっても PC が壊れたことはないのにおすすめです」

には、依頼された仕事は、講演でも何でも断らないようにしています。

最初の会社を畳んでフリーランスになってから講師の仕事を引き受けたのは、仕事の時間がある程度、決まっていたからです。最低限の生活費を稼ぎ、あとは自分の好きな開発に専念できる。米国で起業したときも、売上が当初すぐ出ないことはわかっていたので、少なくとも半年分食べられるだけの、米国からリモートでできる日本での仕事を持っていきました。

ちなみに、仕事の単価を意識して高めに取っています。サラリーの意味もあるし、プロジェクト単位での報酬もそうです。一般に同じ作業を受ける時は、知名度の高い人の方が、料金を高く請求できますよね。その意味もあって、日頃、僕は自身の知名度を上げるよう意識的に活動しています。GitHub や SNS のつながり、講演の仕事などが、まさにそうです。

ただ、自分のアプリに値段をつけるのはいまでも悩みます。僕には自分が目標とする年収があり、アプリの開発に 1 カ月かかるとすれば、年収の 1/12 を価格として考えるようにしています。値決めの基準は、人それぞれだとは思いますが。

——どこでも食べていけるようになるには、評判をないがしろにできないと。

増井 そう思います。国内外問わず、フリーランスを続けていく場合は、対外的な信用が必須です。米国でビザを取得するときにも、あなどれません。米国の H-1B という



クラウド型メモ帳サービス「wri.pe」

「このメモ帳を作るのも、実は2回目。5年前に作ったけど、思ったように面白く行かなくていちど中断。僕は飽きっぽいので」（増井氏）。今回、あらためて、いくつか思いついたことを加え、Rubyで作り上げた。

専門的な職業に携わる外国人向けのビザは基本的に、大学での学位と、日本での職種、米国での職種が一致しないといけない。僕は、男子校卒だったこともあって大学ではあえて文系に進んだのですが、そのためにH-1Bビザを簡単に取得できなくなるとは想像しませんでした（笑）。例外規定を駆使しないと取れないんです。過去に何か大きな賞をもらったとか。有名な大学の先生のレターを書いてもらうとか。職位何年で、学位何年分、と換算するとか、いろいろ面倒です。ただし、米国のルールは都市によっても、州によっても経済状況によっても変わります。米国の失業率が上がると、ビザが取りにくくなったりするので、仕事をするには最新情報を得るようにしてほしいと思います。不法就労が見つかったと厳しく罰せられるので。



勉強会はネットワークングのため

増井 ところで僕は、勉強会でも、単に勉強のために行く、ということはありません。勉強するのであれば、コーディングもそうですが、自分の好きなときにWebや本を見たり、YouTubeを見たりしたほうがずっと効率的です。勉強会は自分が講師をする場合を除いて、ほぼネットワークングのためです。いまのFrogAppsでも、iOSに詳しいエンジニアがほしいな、と思ったら、そろそろセミナーや勉強会に行って、参加者を一本釣りしなきゃ、といった話をしています。

——「masuidriveの作りたい物 & 試作品リスト」にも、英語関係が多いですが、これも勉強のためですか。

増井 英語は必要に駆られてやっています。これまで独学でやってきたのですが、そろそろ感覚だけでは限界に来了。それで、どういう風に勉強しようかと思って、まずはGitHubのissuesで使われている英単語を調べてみたんです。動詞などを原形に直して形態素解析をかけた。その分布を取ったところ、300語あれば、GitHubのissuesの95%をカバーできることがわかりました。しかも、そのうちの2百数十語は、中学生で覚える単語なんです。だから高校・大学と英語をちゃんと勉強してこなかった僕のような人でも、100語未満の単語を覚えるだけで、issuesに書かれる英語の9割以上は、書けるはずですよ。

英語を勉強するもう一つの理由は、ツールを英語圏でリリースするからです。MobiRubyでアプリを書きたい人が日本だと数百人くらいしかいない。それを英語圏まで対象を広げると、その10倍から20倍はパイが大きくなります。その代り、MobiRubyも、Ruby同様、英語のドキュメンテーションやプレゼン資料を用意しなければならないので、英語をやらざるを得ません。

悩ましいのは、Phrasal Verbsです。『Get into』とか『Put on』とか、それぞれの単語は中学生レベルだけど、組み合わせると、文脈の中で意味が変わる。それがどういう風に使われて、どんな返事がなされているのか、合成語の使いこなしが英語の肝だと思います。

米国で辛かったのは、昼飯を食べているときの会話です。テレビで見たニュースの話題から、家の話、彼女の話、犬の話、めちゃくちゃ飛ぶんで話が追えない。

でも、話題をコンピュータなどのドメインに絞れば、なんとかなる。そうした経験をベースに、通訳や翻訳をやっている人と組んで、日本人向けに本を書ければいいなと思って、出版社にアプローチしているところです。本を書くと自分も勉強するし、一石二鳥です。

リスクをコントロールできる自由

——行動力とスピードが出会いを呼び込んでいるように見えます。

増井 コーディングも英語も、僕より上を見たら切りがない。でも、旺盛な好奇心や衝動に素直に従うスタンスは、僕の数少ない取り柄です。基本的にNoと言われなければ、Yesのほうにフラグを立てる。怒られたらその時、考え直せばいい。

まつもとさんと、mrubyについてやりとりするときには主にメールですが、どうしても直接ネゴシエートしたいときがあって、まつもとさんをお願いして、羽田から成田に移動するタクシーに同乗させてもらったことがあります。車中で僕が開発したツールのコードを見せ、道中でHerokuのオフィスに立ち寄ってデバッグしていただいた後に、成田から旅立ってもらいました。まつもとさんには感謝ですが、僕も機会を逃さず、身軽に動くには、時間が自由に使えないといけない。普通の会社員の立場じゃ、ちょっと難しいかもしれません。例えば、毎朝決まった時刻に必ず出社しなさい、という会社とか。

それから、一社にすーっと勤め続ける、というのも、僕にとっては、リスクに感じることです。たしかに、あれこれ手を出すということはリソースが分散されてしまい、一つの事業集中型の人と比べるとパフォーマンスで劣るかもしれない。そこは一長一短です。いろいろやることで生まれる刺激や、あるプロジェクトが頓挫しても、他のプロジェクトでリスクをヘッジできる選択肢は必要です。実際、米国の会社では、いきなり『この製品ラインを止めます』と、プロジェクトがキャンセルされて転職活動を余儀なくされることがよくあります。失業率などの数字以上に人が頻繁に動いています。

——逆に、増井さんが人を雇う場合、「ここだけは見ている」というポイントはどこですか。

増井 自分で何か作っている、とか、コーディングを好き



でやっているかどうか、をじっくり見ますね。ちゃんとしたプロダクトになっていなくても、GitHubに上げていなくても構わない。個人用の便利ツールでいいんです。大事なのは、手を動かしていること。Titanium Mobileと接続するMySQLに関連するライブラリを作っても、最終的にユーザーに届く訳ではない。けれども、ライブラリやアルゴリズムを作るのに面白さを感じている人も少なくない。僕には到底書けないような凄いコードを書く人も、いっぱいいます。MobiRubyは、iPhoneアプリを作るためのツール。そういうのって、年単位で地下に潜ることが多くてしんどいけれど、それはそれで面白さがある。ユーザー向けのプロダクトを作る以外にも、モチベーションは人それぞれです。僕は、最終ユーザーの目に見える、見えないを問わず、作っているものを両方評価したいですね。

また、仕事の全体像がわかって技術を深掘りすると、わからずに深掘りするのではだいぶ違ってきます。デザインをわからないとデザイナーとは話が噛み合わない。プロの色々な仕事を全般的に知っておいて損はないです。

僕は、人より、技術のほうに興味があるほうですが、それでも、人生の節目で探しているのは、この場面を動かしているのはこの人だ、というようなキーパーソンです。意識的にそうした人の近くにいたい心がけています。

——そろそろ社長業に復帰のご予定は？

増井 いや、まだないですね。哲学を語るのも、もっと歳を取ってからでもいいかなと。いまのところ、起きている間は、ずーっとコードを書きたいんです。

——増井さんの元気の源がちょっぴりわかったような気がします。本日は有難うございました。

シェルスクリプト大喜利

第九回

司会：『もっと吹く』編集長・みかん

皆様こんにちは。今月もシェルスクリプト大喜利(略してsh大喜利)のコーナーがやってまいりました。

六月です。カタツムリやアジサイの花をよく見かける季節ですが、カタツムリをアジサイに置くと逃げてしまうとか……。実はアジサイには「ヒドラシアノシドA」(シアン化合物)という毒があるんですって。へえ～。

この話の**一体どこがシェルスクリプトと関係あるのか?**
……**ありませんっ!**トリビアってヤツですよ。だってほら、この大喜利だってトリビアみたいなもんじゃないですか。全く実用性を求めてないわけですから。ではどういうコーナーか?というわけで、まずは本コーナーのご説明!

シェルスクリプト大喜利とは

シェルスクリプト大喜利特有のルール

一、sh大喜利はクイズやテストではありません。なので決まった答えというものはないのです。あえて言うなら面白いスクリプトが正解!

二、面白いスクリプトとは、例えばこんなもの。

イ、人が考えつかない意外性がある

ロ、美しい、芸術的、記述がシンプル、高速、など

ハ、アイデア・こだわりが光る

ニ、ネタになるバカバカしさ、くだらなさがある

など。でも最後のは段位強制返還の恐れありよ。:-)

三、スクリプト動作環境はLinuxとし、基本的にLinux JM(<http://linuxjm.sourceforge.jp/>)に記載されているコマンド及び機能のみ使用可能とします。これは多くの人を楽しめるようにするためなのです。(JMにあるので、Cシェル系での解答もOK!**あとksh、zshもOKにひまひた**)

四、sh大喜利はシェルスクリプトを披露する場なので、**PerlやRuby、Pythonなどは使っちゃダメ**です。そもそもJMにも載っていません。逆にシェ

ルスクリプトにとって不可欠なawkやsed等はOKです。JMにもありますし。でも、よっぽど面白ければ、Perlとかもなきにしもあらず??

五、Open usp Tukubai(<http://uec.usp-lab.com/>)も使用OK!但し、使う意義がそれなりに感じられないと採用はキビしいですよ～。

ルールもおさらいしたところで、さあ始めましょう!

本番開始

<第一問>

1～n (nは引数で指定) の自然数の中に存在する『**セクシー素数**』を全て求めるシェルスクリプトを書いてください。

整数数列のお題の4回目。お題作りがちょっと安易すぎませんか?とツッコまれそうですが、だってこのお題にすると投稿集まるんですもん!

それでセクシー素数というのは互いの差が6である素数ペアなわけですが、早速投稿見ていきましょう。

◎**ahsokai**さんの解答

```
1 #!/bin/sh
2 awk -v n=$1 '
3 BEGIN {
4     split("", p)
5     for (i=2; i<=n; i++)
6         if (is_p(i) && is_p(i-6)) {print i-6, i}
7 }
8 function is_p(n) {
9     if (n<2) {return 0}
10    for (j=0; j<length(p); j++) {
11        if (n==p[j]) {return 1}
12        else if (n%p[j]==0) {return 0}
13    }
14    p[length(p)]=n
15    return 1
16 }
```

一人目の解答は、AWK一撃解法ですな。素数かどうかを判定する関数を作り、差が6の二数を同時に判定して両方とも素数だったら表示するようですね。

教科書通りの解答といったところかな、**初段授与**。

◎熱々!お茶!さんの解答

```
1 #!/bin/sh
2 for n in $(seq 1 $1);do factor $n;done | awk 'NF=
  =2{print $1}' | sed 's/:// ' | awk '{print $1,$1+6
  }' | sort -k1,1 > q1. $. $. tmp
3 cat q1. $. $. tmp | sort -k2,2 | join1 key=2 q1. $. tm
  p | sort -k1n,1 | tail +2 | delf 3
4 rm q1. $. $. tmp
```

投稿者コメント「primes コマンドは JM に収録されておらず factor コマンドで素数作りしました。そして2列目に +6 した値を追記して、それをキーにして自分自身と join1 コマンドで結合して…(略)」ということで Tukubai を使ってくれました。ありがとう!

なるほどね、ウマイ join1 の使い方だ! 更に primes コマンドが無い事にも配慮してくれて、よし! これは二段……といきたかったんだけど、残念! **初段獲手**。

いやあ〜これねえ、ホント残念。何が起こったかという、と、熱々!お茶!さんが投稿してくれた時点では確かに primes コマンドは JM にはなかったんだけど、2013 年 4 月に JM に追加されたのですよ! **sh 大喜利史上まさかの大波乱**だ。というわけで primes を使って短く書いた方がむしろ模範解答ということになってしまいました。しかし今回 primes を使って投稿してくれた方々は、果たして知っていたのだろうか!?

◎Kさんの解答

```
1 #!/bin/sh
2
3 primes 1 $((($1 - 5)) | ¥
4 awk '{printf "%d ", $1; system("primes " $1+6 " "
  $1+7); printf "%n"}' | ¥
5 awk 'NF==2{print}'
```

コメントによると「Linux JM に bsd-games の primes が入っていたので、ありがたく使わせていただきました」とのことで、しっかり確認とってます。

うむ、よくぞ確認してくれた! エライ! さすが常連さんだ。よし **二段獲手**。これで六段ですな。

◎yyamaokaさんの解答

```
1 #!/bin/sh
2 primes 2 ${1:-2} | awk '{if($1-6 in p){print $1-6
  , $1}p[$1]=1}'
```

これは今回一番シンプルな解答でした。そして最速。

うーん、これも primes 使ってるね。まあ後ろの AWK は、配列にどんどん素数が溜まっていくので大きい数を求める時ほど遅くはなるだろうけど、記述のシンプルさは気に入った! よし **二段獲手**。



さて、第二問いってみよう。

<第二問>

HTML テキストを与えると、タグのツリーを返すシェルスクリプトを書いてください。

[出力例]	<html><body!>	html
	笑い声ベスト3	body
	- 笑 - w	ol
	わはは </body> →	li
	</html>	li

このコーナーはトリビアみたいなものと申しましたが、このお題に関してはちょっと実用性あるかもしれないですね。まあ、閉じタグ無しのもの(
 など)無しだとか、属性も無しだとか、ということでお題を簡単にしていますが……。

これも結構解答にバリエーションがありましたよ。そして、意表を突く解答が一つ……(苦笑)

◎tnazukaさんの解答

```
1 #!/bin/sh
2 cat target.html |
3 sed 's/</¥
4 </g' |
5 sed 's/>/>¥
6 /g' |
7 awk '
8 BEGIN {indent="";}
9 $0~/<¥// {indent=substr(indent,1,length(indent)
  -2);next;}
10 $0~/</ {print indent substr($0,2,length($0)-2);
  indent=indent " ";next;}'
```

「タグ1個で1つの行にしまえばあとはawkで処理できることに気が付きました」とコメントが付いていました。sed と AWK の合わせ技ですね。AWK で処理するのに都合がいいように予め sed でテキストを置換しておくという……。実は今回、投稿された解答はこの sed&AWK のパターンと sed だけのパターンに分かれました。一つを除いてね。

うん、sed&AWK な解答は正攻法だね、分かり易いし。ただ大喜利的には普通かな。というわけで **一段獲手**。これで三段だ。

◎Kさんの解答

```
1 #!/bin/sh
2
3 cat $1 |
4 sed -e '/[<]*$/d; s/[<]*¥([>]
  [^>]*¥)>[<]*¥1¥
5 /g' |
6 sed -e '/[ ]*$/d' |
7 awk 'BEGIN{l=0}{if ($0 ~ /¥//) {l++} else {print
  l, $1; l++}}' |
8 awk '{for(i=0; i<1*2; i++) {printf " " print
  $2}}'
```

はい、1 問目に続き投稿ありがとうございます。こ

ちらも sed&AWK 解法ですね。ただ、投稿者によって使い方が微妙に異なるところが興味深いですね。

というわけで、こちらも **一段撥与**。これで七段だ！さて今度は、もう一つのパターン、「sed だけ解法」の投稿者を紹介していきましょう。

◎松澤二郎さんの解答

```
1 #!/bin/sh
2 sed -n 'A
3 ${
4 s/[[:space:]]*/g;s/⟨[ ]⟩#{1, *}⟩/⟨
5 ¶1/g;p
6 }
7 N;s/¶n/ /g;b A' |
8 sed -n ' /'⟨[ ]⟩#{1, *}⟩/{
9 g;s/ //;h;d
10 }
11 /'⟨[ ]⟩#{1, *}⟩/{
12 G;s/⟨[ ]⟩#{1, *}⟩.¶n¶(.*)/¶2¶1/p;g;s/ /;h
13 }
```

一人目の解答ですが、sed2 段構えとのこと。前半の sed は一行一タグ化を行うもの、後半の sed は階層に応じたインデントをつけてお題が求める形にするものです。

いやー、sed って中のコマンドが全て 1 文字だから全体的に眺めると呪文のようだよな。よし、**二段撥与**。これで五段。

◎sed 内 ker さんの解答

```
1 #!/bin/sh
2 n=' '
3 '
4 sed 's/⟨/"$n"/⟩g'
5 sed 's/⟩/⟩/"$n"/g'
6 sed -n ' /'⟨p'
7 sed 's/⟨¶/¶(.*)⟩/-/'
8 sed 's/⟨¶(.*)⟩/¶1/'
9 sed -n 'x;G;s/¶n//;s/+//;p;s/[[:alnum:]]*///;x'
10 sed 's/+//
11 sed -n ' /'⟨[[:alnum:]]⟩p'
12 sed 's/+//g'
```

二人目。おお来ましたね、この方！第七回の大勝利では sed を 62 個もパイプで繋ぐという変態的回答を送ってきたんですよ。あれは参りましたが、今回もまた sed づくしですな。

工程が細かく分かれていますので、sed を一個目まで、二個目まで……と実行していくと変化の様子が見えて面白いね。これも **二段撥与**。これで五段だ。

◎gori.sh さんの解答

```
1 yarr | tr -d ¶ | sed 's/⟨([a-zA-Z]*)⟩[ ]*/¶1
¶n/g' | awk '{if (/¶/) {s=substr(s, 1, l-2)} else {p
rint s$0;s=s" ";l+=2}}' | sed 's/body1/body/'
```

これが、意表を突く解答です。**実はおこの題には出**

題ミスがありまして、まんまとツッコミを入れられてしまいましたよ。

一行野郎な gori.sh さんの解答は、一番最後に s/body1/body/ という置換が入っています。前ページのこのお題の「出力例」をよーく見てください。ほら「</body1>」と typo してるでしょ。まさか解答をもってしてツッコミ喰らうとはねえ。一本とられましたよ。

うーん、参った！でも tr -d '\n' とやれば yarr コマンド要らないしー、それにこのコードだと出力結果の最後に余計な空行が付くしー、などとツッコミ返して、**一段撥与**しかないもんね！でも八段、ぐぬぬ……。

◇ ◇ ◇

お待ちかねの第三問。また変り種のお題ですよ。

＜第三問＞

へんな名前のファイルが出来て消すのに苦労したことはありませんか？そこで「そんな名前のファイル作んなよ〜」と思わせるファイル名のファイルを作るシェルスクリプトを書いてください。そのファイル名がどうしてへんのか理由も添えて……。

古参の UNIX ユーザーならきっとありますよね、消せないファイルで困ったという思い出が。脳内に格納されている .history ファイルの中からその忌むべきコマンド行を grep して投稿するだけです。簡単でしょ？……ってそれが難しいか。でも三つ解答来ましたよ。

◎イタローさんの解答

```
1 #shまたはbashにしたらワンライナーで実行
2 a='deep';b=$(awk 'BEGIN{ORS=" ";b=" $a";print b;for(i=1;i<=199;i++){print "/" b}}');i=0;while [ $i -lt 1000 ];do mkdir -p $b;j=$?;cd $b;j=$((j+$?));i=$((i+1));echo $i;[ $j -ne 0 ] && break;done;i=0;while [ $i -lt 200 ];do mkdir $a;j=$?;cd $a;j=$((j+$?));i=$((i+1));echo $i;[ $j -ne 0 ] && break;done
```

投稿者コメント「ディレクトリの深さの限界に挑戦！途中でエラー出ますがとにかく最後まで実行させましょう。パスが長すぎて特に bash では消すのに一苦労すると思いますぜ」と、ということで**実行する場合は自己責任で！**

おお、これは……。消す時、変なエラーが出て苦労するし、おまけに bash だとプロンプト表示がいちいち重くなるぞ！うん、確かに「そんなファイル作んなよ〜」だな、こりゃ。**一段撥与**。これで二段ね。

◎C₆H₈O₇ (クエン) さんの解答

```
1 touch `printf '¥377' `clear`
```

コメント「これは私がイタズラ目的で書いたコードで、実質的に FreeBSD 専用です。9.0 以降の /bin/sh

でタブキーを押してコマンド補完した場合や、ports から jls コマンドをインストールして使うと、clear コマンドのエスケープシーケンスが効いて画面が消去されます。最初の printf は、ファイルがキャラクターコード順でなるべく最後に表示されるようにするための工夫で、イタズラに念を入れます」とのことです。

イタズラされたんじゃなくてアナタがしたのか!……と最初にツッコミを入れておきたいが、確かにこれはイジが悪いねえ。FreeBSD9 の /bin/sh でこのファイルの置かれているディレクトリに移動して、[v][i][] [タブキー] とかタイプすると画面消えるし。

「そんなファイル作んなよ!」と思うけど、消すのは難しくないね。touch と書いてあるところを、rm に変えるだけで消せるじゃない。まあ勿論、こんなヘンなファイル名になっていることを知らなきゃ消せないけどね。よし、これも一段授与するぞ。現在二段。

そして最後。……やれやれ、またヘンなのが来ましたよ。いや、ヘンな解答はこの大喜利ではむしろ歓迎なんですけどね。

◎gori.sh さんの解答

```
1 curl http://www.kanjijiten.net/jiten/yomi06.html
  | iconv -f Shift_JIS -t UTF-8 | sed -n '/id="y649
  9"/,/k8157/p' | sed 's/[ -~]/%/g' | sed '/$/d'
  | tail -n +2 | xargs touch
```

この解答には curl とか iconv とか Linux JM 外のコマンド含まれてますが、まあネタとして面白いので掲載しました。でも段位進呈はナシですからね。ましてや gori.sh さん、八段だし。

さて、投稿者コメントですが「オンライン漢和辞典サイトから、『へん』と読む字を探し出し、それをファイル名にします。これは間違いなくヘンなファイルです。異論は認めません。シェルスクリプト自体は普通。curl してゴミ掃除して touch するだけです」とのこと。

おめでとう!それは「ヘンなファイル」の意味が違うぞー!と、一応リアクションするけど、うーむそう来たのね……。



これにて本日的大喜利はお開き! 読者の皆さん、投稿してくれた皆さん、今回もありがとうございました。

投稿大募集

次回のお題

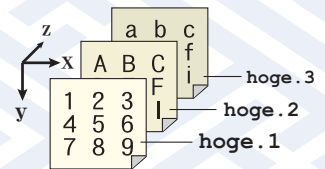
一、1 ~ n (n は引数で指定) までの自然数の中に存在

する「スミス数」を列挙するシェルスクリプトを書いてください。

二、3D プリンタの記事がありましたね。UNIX の各種フィルタコマンドも 3D 対応にしなくっちゃ!

図のような 3 行 3

列のファイル(横の文字、例えば 1,2,3 の間には半角スペースが 1 つ入っ



ている) が 3 つで構成されたデータがあります。これに x-z 平面方向や y-z 平面方向でコマンドを作用させたいとなったら、予め x-z 平面や y-z 平面で 90° 単位の回転ができればよさそうです。

そこで、今回は x-z 平面で 90°、180°、270° 回転させられるシェルスクリプトを書いてください。3 つはバラバラのシェルスクリプトで構いません。(90° 回転時、回転後の hoge.1 は 1 行目から順に "a A 1", "d D 4", "g G 7" となれば正解とします)

三、rev コマンドって知ってますか? 与えられたテキストの各行内で、文字の左右の位置を入れ替えるコマンドです。例えば 12345 という行があったら、54321 が出てきます。でもコレ、何につかえばいいのでしょうか? 是非ウまい使い道を提案してください、シェルスクリプトで。

投稿のしかた

お題への解答は、お名前(ペンネーム)、解答したお題番号、解答スクリプト、簡単な補足の四点セットで下記の宛先へ。一人何問でも何個でも解答可! 尚、次回締め切りは**8月26日(月) 午前0時**とします。しかもその間は何度でも解答の修正を受け付けます。

お題もどしどし送ってくださーい

お題の投稿も大募集。こっちは締め切りなしでずーっと募集中。そして、考えてくれた方にも段位を授与します。自分で出題して解答するのも、OK!

投稿先

どちらも投稿先は、mag@usp-lab.com です。面白ければ(ワリと)何でもあり! じゃんじゃん投稿待ってます!

TALK LIVE TUESDAY 23 JULY TechLION vol.13 たなか くにひろ



MC : 法林浩之、馮富久

TalkGuests : 田中邦弘、紫竹佑騎、田中和紀、ほか

2013.7.23 Tuesday

Place : 六本木 SuperDeluxe

Start 19:30 ~ Close 22:00 (予定) Ticket : 前売り・予約 2700 円、当日 3200 円 (1ドリンク込)

主催 : ユニバーサル・シェル・プログラミング研究所 協賛 : 技術評論社、オライリー、ハートビーツ、クラウドワークス、他

後援 : 日本 UNIX ユーザ会、EMZERO、キャリアデザインセンター「エンジニア Type」

おもしろエンジニアークライブ 「TechLION (てっくらいおん)」 詳細は WEB で!

TechLION

検索

USP 友の会 会員募集中

シェルでつながる技術の輪! 基礎から、プロのライブコーディング・ライブまで、いろいろやっています。おもいきり真面目に、時に面白く、心から楽しめます。WEB 会費は無料です。勉強会や定例会の参加費は営利目的ではない為、良心的です。



UNIX/Linux/ シェルスクリプトの可能性を
極限まで追求します!

詳しくは

<http://www.usptomo.com> にアクセス!

約 400 名の会員が定例会や勉強会で活発に交流をしています。



@usptomo

検索



USP 友の会

検索

USP MAGAZINE バックナンバー 好評発売中!

vol.0 2011 spring~vol.4 2012 spring



vol.5 2012 summer



vol.6 2012 autumn



vol.7 2013 winter



vol.8 2013 spring



USP MAGAZINE は、世界初のシェルスクリプト技術情報誌。でもご存知のとおり、シェルスクリプトはグルー言語。OS 深層から様々な言語・アプリの話まで、さらには技術の先にいるエンジニア達にもスポットを当てます。

目指すは、シェルスクリプト力とエンジニア達の地位向上!

自炊不要 定期購読又はバックナンバーをお申込みいただくと、PDF 版が無料で手に入ります。

ご購入、お申込みはこちらから⇒

USP 出版

検索