明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

Perl

中村聡史

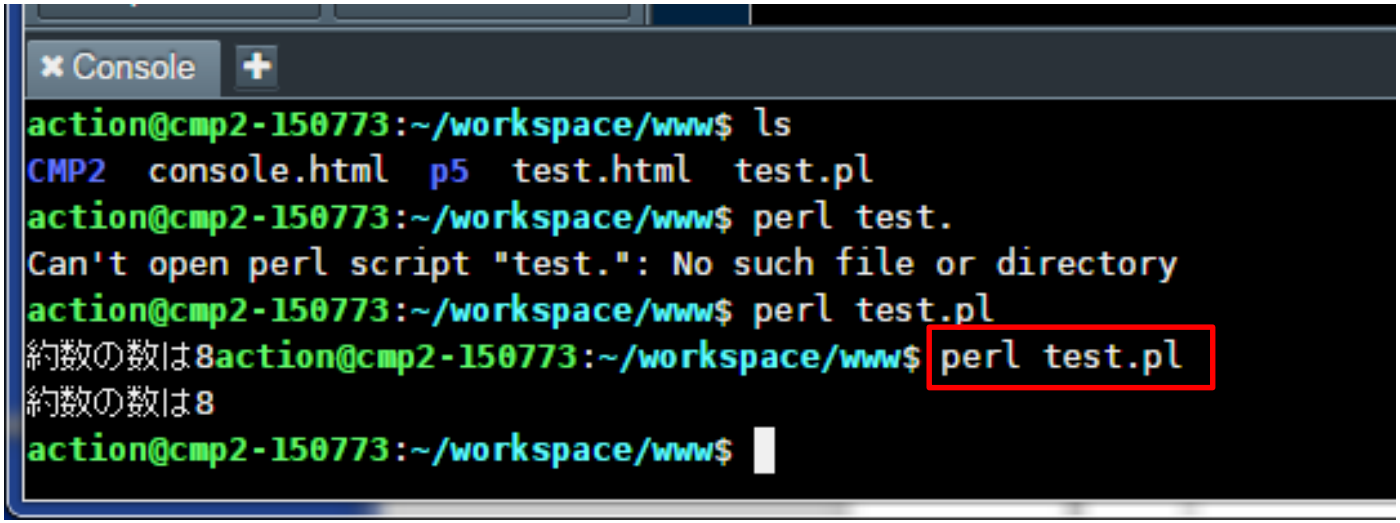
Perlとは

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

- プログラミング言語の1つ
- 文字列を処理するのに適している言語
- 長時間処理するようなものに適した言語
- 整形などをするのに適した言語

Perlの実行方法

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室



perlコマンドで実行！

```
% perl  test.pl
```

ProcessingとPerl

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

- 同じ所
  - プログラミング言語なので変数，計算，条件分岐，繰返し，メソッド，クラスなど同じ

12345の約数の数を数えるプログラム

```
int i = 1;
int count = 0;
while( i <= 12345 ){
  if( (12345 % i) == 0 ){
    // 12345をiで割った余りが
    // 0だったらcountを増やす
    count++;
  }
  i++;
}
println( "約数の数は"+count );
```

```
my $i = 1;
my $count = 0;
while( $i <= 12345 ){
  if( (12345 % $i) == 0 ){
    # 12345を$iで割った余りが
    # 0だったら$countを増やす
    $count++;
  }
  $i++;
}
print STDOUT "約数の数は".$count;
```

ProcessingとPerlの違い

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

- Perlでは変数名の最初には必ずドルが付く
  - 例： $value, $hoge, $ball などなど
  - 変数には型（int, Stringなど）が無い（勝手に解釈）
  - 変数を定義するときには my をつける
  - 変数への代入は「=」でつなぐだけ
- 配列名には「@」が最初に，連想配列（ハッシュ）には「%」が最初につく
- ユーザからの入力は <STDIN> で，ユーザに対する出力は STDOUT で行う

ProcessingとPerlの違い

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

- Perlではコメント行は # で始まる
  - Processingは // で始まる
- Perlで出力するには print を使う
  - print 出力先 出力内容; のようになる
    - 出力先は省略可能
    - 出力先にSTDOUTと書くと画面出力せよの意味
    - 出力先にファイルを指定するとファイル出力
- 文字列の連結には「.」を使う
  - Processingの場合は「+」を使う

ProcessingとPerlの違い

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

- if は同じだけれど else if が elsif になる
  - if(...){ ... } elsif(...){ ... } elsif(...){ ... } else { ... }
  - 数値が等しいという意味では ==/!=
    - if($id == 5){ 5の時の処理 }
    - if($id != 5){ 5じゃない時の処理 }

    こちらはProcessingと同じ.

    「かつ」「または」も「&&」や「||」で同じ
  - 文字列が等しいという意味では eq/ne
    - if($name eq '中村'){ 中村の時の処理 }
    - if($name ne '中村'){ 中村じゃない時の処理 }

ProcessingとPerlの違い

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

- メソッド（サブルーチン）は
  - 返り値型 メソッド名(引数){ ... } ではなく
  - sub サブルーチン名{ ... } となる
  - サブルーチンの呼出は &サブルーチン(引数);
  - 引数の処理は

    void show(int x, int y){ ... } というものを

```
sub show {
    my ( $x, $y ) = @_; # のように書く
}
```

ProcessingとPerlの違い

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

- サブルーチンを使っている例
  - あまり大きな差が無いことがわかりますか？

```
int getNumberofYakusu( int num ){
  int count = 0;
  int i = 1;
  while( i<=num ){
    if( (num % i) == 0 ){
      // numをiで割った余りが
      // 0だったらcountを増やす
      count++;
    }
    i++;
  }
  return count;
}
println( getNumberOfYakusu(12345) );
```

```
sub getNumberOfYakusu {
  my ( $num ) = @_;
  my $count = 0;
  while( $i <= $num ){
    if( ($num % $i) == 0 ){
      # $numを$iで割った余りが
      # 0だったら$countを増やす
      $count++;
    }
    $i++;
  }
  return $count;
}
print &getNumberOfYakusu(12345);
```

[演習]

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

- 約数を表示するサブルーチンを作り，1から1000までのそれぞれの数の約数の数を表示せよ！
- 引数として入力した数字が素数かどうかを判定するサブルーチンを作り，素数ならその値を表示するプログラムを作成せよ！
- 1から10000までのすべての素数を表示せよ

入力を取得する

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

- ユーザからの入力を取得して結果を変更したい！
  - 年齢を訊いて，お酒が飲めるかどうかを判断
  - リンゴ（120円）を何個購入するかを訊いて，合計の値段を表示
  - 苗字と名前を訊いて，確認したい

入力を取得する

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

- 取得する方法は単に <STDIN> と書くだけ！

```
print STDOUT "年齢は何歳ですか？";
my $age = <STDIN>;
if( $age >= 20 ){
    print STDOUT "お酒が飲めます";
} else {
    print STDOUT "お酒が飲めません";
}
```

```
print STDOUT "苗字は？";
my $myoji = <STDIN>;
print STDOUT "名前は？";
my $namae = <STDIN>;
print STDOUT "ようこそ" . $myoji . $namae . "さま";
```

[演習]

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

- リンゴ（120円）を何個購入するかユーザに質問し，その個数に応じて合計がいくらになるかを回答せよ
- リンゴ（120円）とみかん（30円）を何個ずつ購入するか質問し，ユーザの回答に応じて合計がいくらになるか回答せよ
- ユーザが1000円しか持っていないとき，リンゴをいくつ購入するか質問し，購入可能かどうか回答せよ

Perlで文字列処理

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

- ここからはPerlの得意な世界へ
  - 文字列の処理
  - ファイル入出力

ファイルから読み込む

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

- ファイルを開く
  - open(FILE_IN, "ファイル名");
- ファイルの内容を1行 $line に読み込む
  - $line = <FILE_IN>;
  - 1行読み込む際には「<>」をつける
  - $line の内容を表示するには...
    - print STDOUT $line;
- ファイルを閉じる
  - close(FILE_IN);

ファイルに書き込む

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

- ファイルを開く
  - open(FILE_OUT, ">ファイル名");
  - open(FILE_OUT, ">>ファイル名");
  - 「>」が1個の場合はファイルを一旦削除して新規作成，2個の場合はそのファイルに追記する
- ファイルに書き込む
  - print FILE_OUT "1行目" . "¥n";
  - print FILE_OUT "2行目" . "¥n";
  - 文字列を繋ぐ時は「.」で，改行は「¥n」
- ファイルを閉じる
  - close(FILE_OUT);

ファイルから読み込む

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

- ファイルの中身を読み込む！
  - 繰り返しでひたすら読み込む
    - while($line = <FILE_IN>){ ... }
    - whileはファイルの最終行まで来たら終了
    - おまじない chomp($line); という行を入れておくことによって，行の最後に入っている改行を削除することができる（¥nをカット）

[演習] 読み込んで表示

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

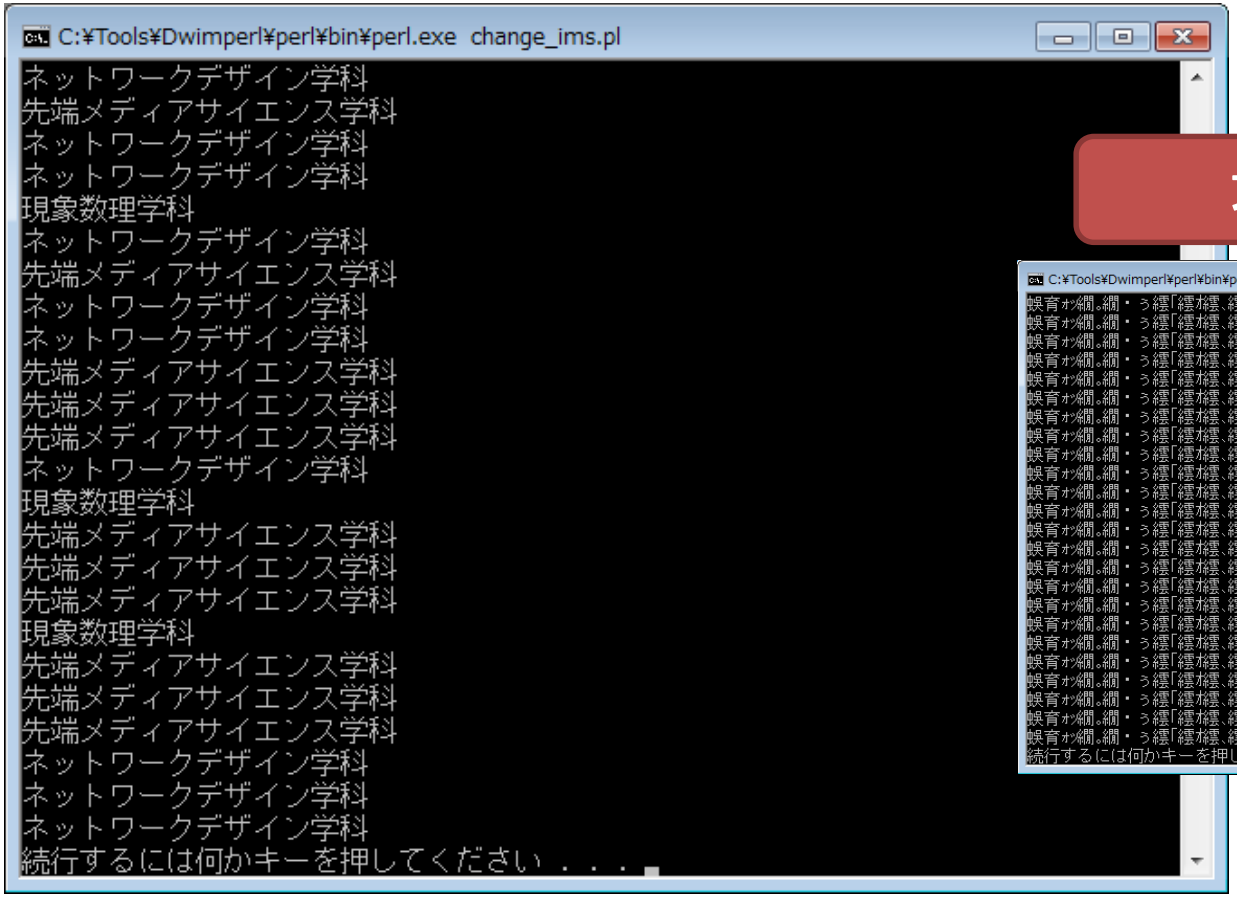
- 読み込んだ内容を表示してみよう
  - http://snakamura.org/files/ims50.csv
  - open で開き，print で表示する

- 読み込んだ内容に応じて画面に出力する内容を変更してみよう
  - 上記ファイルを読み込み，内容に応じて表示を変更する
  - MSなら現象数理学科，FMSなら先端メディアサイエンス学科，NDならネットワークデザイン学科と表示する

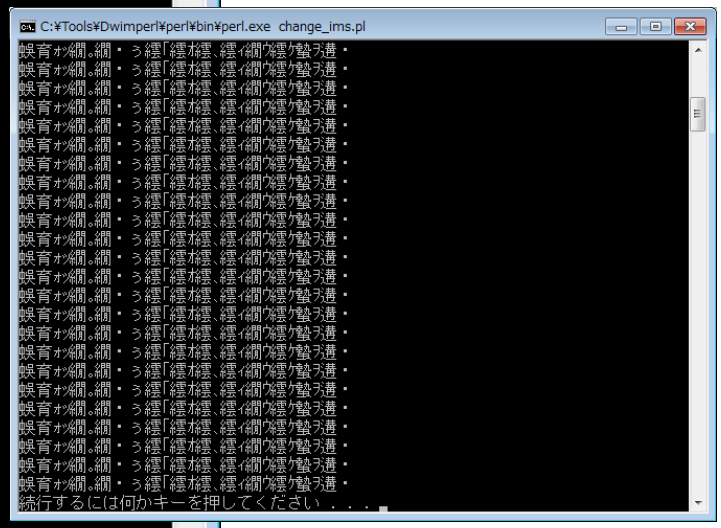
実行結果

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

- こんな感じになります．文字化けしている人は文字コード変換を！



文字化けた例



演習： 読み込んで表示

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

- 読み込んだ内容をそのままファイルへ書き込んでみよう（転写する）

- 読み込んだ内容に応じて出力を変更し，ファイルに出力してみよう
  - MSなら現象数理学科， FMSなら先端メディアサイエンス学科， NDならネットワークデザイン学科と表示する

演習：数える

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

- MS，FMS，NDの数を数えて表示しよう
  - 下記のファイルをダウンロードして，そのそれぞれの数をカウントしてみよう
  - http://snakamura.org/files/ims.csv
    - MS:　　個，FMS:　　個，ND:　　個
  - http://snakamura.org/files/ims2.csv
    - MS:　　個，FMS:　　個，ND:　　個

MS, FMS, ND数カウント

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室


```
open( FILE, "ims.csv" );
my $ms = 0;
my $fms = 0;
my $nd = 0;
while( my $line = <FILE> ){
  chomp($line);
  if( $line eq "MS" ){
    $ms ++;
  } elsif( $line eq "FMS" ){
    $fms ++;
  } elsif( $line eq "ND" ){
    $nd ++;
  }
}
close( FILE );
print "MS  = " . $ms . "¥n";
print "FMS = " . $fms . "¥n";
print "ND  = " . $nd . "¥n";
```

```
C:¥Tools¥Dwimperl¥perl¥bin¥perl.exe  ims_count.pl
MS  = 28572
FMS = 42877
ND  = 28551
続行するには何かキーを押してください . . .
```

```
C:¥Tools¥Dwimperl¥perl¥bin¥perl.exe  ims_count.pl
MS  = 3259
FMS = 3371
ND  = 3370
続行するには何かキーを押してください . . .
```

演習： 数える

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

- サイコロの目の偏りを調べる
  - http://snakamaura.org/files/saikoro10.csv
  - http://snakamaura.org/files/saikoro1000.csv
  - http://snakamaura.org/files/saikoro100000.csv
  - http://snakamaura.org/files/saikoro100000000.csv

数えるプログラムは面倒

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

- 数える対象分の変数を用意するのは大変
- 配列を使おう！

配列

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

- 配列の最初には「@」をつける
  - @count
    - countという名前の配列をつくる
  - 要素数を指定する必要はない
  - 配列の各要素は「$」ではじめ，要素番号は0から，要素は [] で指定する
    - $count[0], $count[1], $count[2], …
    - print $count[0];
    - $count[0] = 3; $count[2]++; など
- 配列に一度に値を入れるには = () を使う
  - @count = (0,1,2,3,4,5);

配列（文字列も一緒）

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

- 文字列の配列も同じように利用
  - $name[0] = 'nakamura';
  - @name = ('nakamura', 'komatsu', 'kikuchi');

- 「@_」というメソッドの最初に登場する不思議なおまじないは，引数の配列という意味
  - my ($x, $y) = @_; は，@_ が要素が2つの配列であり，$x = $_[0]; $y = $_[1]; としているという意味

[演習]

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

- 下記のファイルをダウンロードし，1～100までの数字が何個ずつあるか表示せよ
  - http://snakamura.org/files/num100.csv
  - なお，数を数える時は配列を利用せよ

パターンがあると大変

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

- 数値なら配列でいいが，文字列の場合はどうやって数えたら良いの？
- その文字列分，変数を用意する？
  - それはあまりに面倒すぎる…

連想配列（ハッシュ）を使おう！

連想配列とは...

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

- 配列の場合は 0, 1, 2, 3, 4, ... がインデックス（目次）となってそれぞれの値を呼び出したりするが，FMS, MS, NDなどをインデックスとしてしまう

0	55
1	48
2	63
3	55
4	59
5	47

FMS	191
MS	140
ND	120
SFC	300
IAMAS	420
FUN	380

連想配列

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

- 連想配列の最初には「%」をつける
  - %count_ims
    - count_ims という名前の連想配列をつくる
- 配列の場合は 0, 1, 2, ... で値を取得するが, もっとわかりやすい語で配列を作るもの
  - $count_ims{ 'FMS' } = 190;
  - $count_ims{ 'MS' } = 140;
  - $count_ims{ 'ND' } = 120;
  - print STDOUT $count_ims{'FMS'};
  - print STDOUT $count_ims{'MS'};
  - print STDOUT $count_ims{'ND'};

連想配列

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

- 連想配列の最初には「%」をつける
  - %price
    - price という名前の連想配列をつくる
- 配列の場合は 0, 1, 2, ... で値を取得するが, もっとわかりやすい語で配列を作るもの
  - $price{ 'apple' } = 100;
  - $price{ 'orange' } = 80;
  - $price{ 'melon' } = 950;
  - print $price{'apple'}*5 + $price{'orange'}*3;

連想配列

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

- 連想配列の最初には「%」をつける
  - %name
    - name という名前の連想配列をつくる
    - $name{ 'nakamura' } = 'satoshi';
    - $name{ 'komatsu' } = 'takanori';
- 何かの数を数えてみる
  - %ims_count;
    - $ims_count{ 'FMS' }++;
    - $ims_count{ 'MS' }++;
    - $ims_count{ 'ND' }++;

連想配列の内容を表示

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

下記どちらでも値は取れます

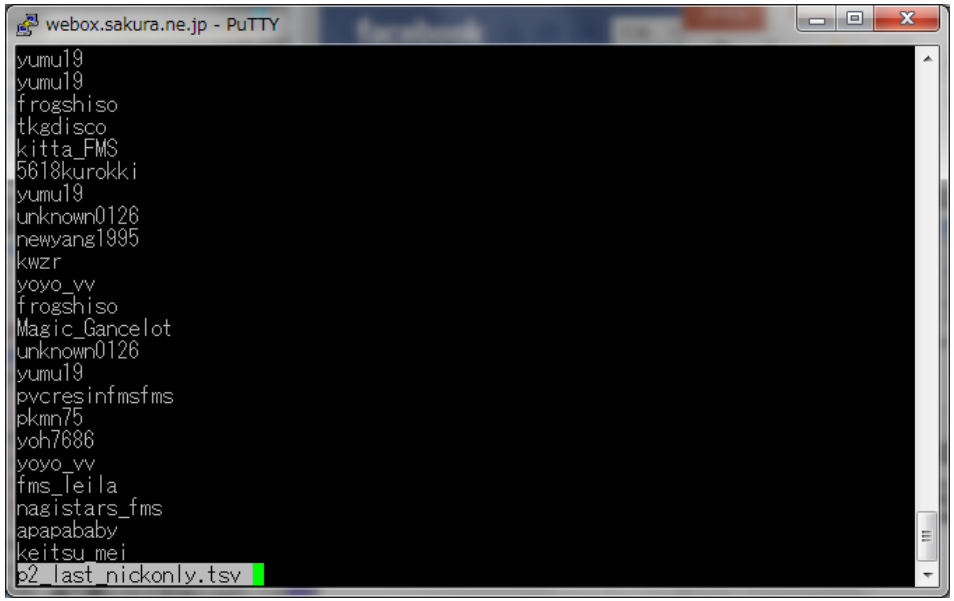
```
foreach my $key(keys(%count)){
  print "$key: ".$count{$key}."¥n";
}

while (my ($key, $value) = each(%count)){
  print "$key: $value¥n";
}
```

key	value
FMS	191
MS	140
ND	120
SFC	300
IAMAS	420
FUN	380

[演習] 数えて表示する

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

- ツイートした人のニックネームを一覧化してみる
  - http://snakamura.org/tsv/p2_all_nickonly.tsv（プログラミング演習2発表会の全ツイートのニックネームを抽出したもの）をダウンロードしてニックネームごとにその数を数え，出力してみよう



答え

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

```
my %nick_count;

open( FILE, "p2_all_nickonly.tsv" );
while( my $line = <FILE> ){
  chomp( $line );
  $nick_count{ $line }++;
}
close( FILE );

foreach my $key( keys( %nick_count ) ){
  print "$key: ".$nick_count{$key}."¥n";
}
```

ニックネーム順に並べたい

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

- ソートする
  - Perlではsortと書くだけで簡単にソートできる

```
my %nick_count;

open( FILE, "p2_all_nickonly.tsv" );
while( my $line = <FILE> ){
  chomp( $line );
  $nick_count{ $line }++;
}
close( FILE );

foreach my $key( sort keys( %nick_count ) ){
  print "$key: ".$nick_count{$key}."¥n";
}
```

多い順に並べたい

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

- ソートする
  - 下記のプログラムを書くとソートすることが出来る
  - 意味としては，$nick_count{$a}という何かと，$nick_count{$b}という何かを比較して並び替える

こんなのは覚える必要ない，必要に応じて検索！

```
for my $key ( sort {$nick_count{$b} <=> $nick_count{$a} || $a cmp $b} keys %nick_count) {
	print $key . "=" . $nick_count{$key} . "¥n";
}
```

トップ10を表示

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室


```
my %nick_count;

open( FILE, "p2_all_nickonly.tsv" );
while( my $line = <FILE> ){
  chomp( $line );
  $nick_count{ $line }++;
}
close( FILE );

my $num = 1;
for my $key (sort {$nick_count{$b}<=>$nick_count{$a} || $a cmp $b} keys %nick_count) {
   print $num . ": " . $key . "=" . $nick_count{$key} . "¥n";
      $num++;
   if( $num > 10 ){
      last;
   }
}
```

[演習]

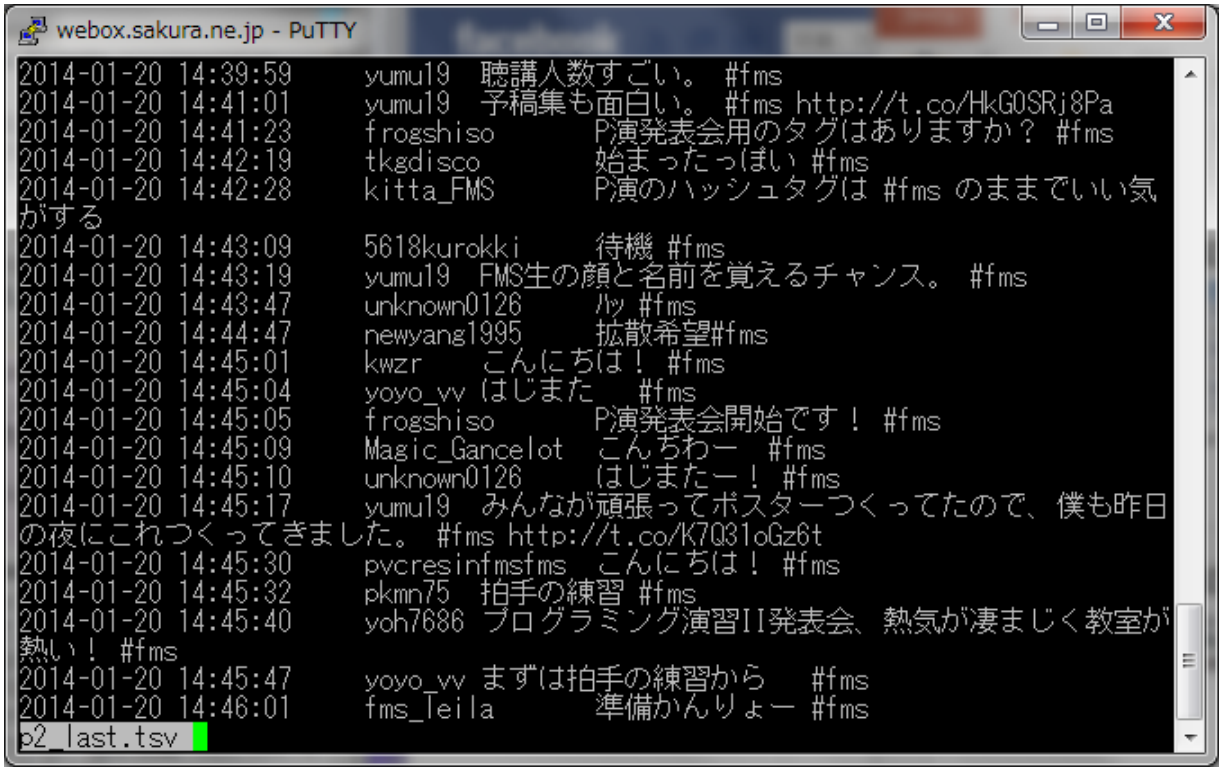
明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

- 先述のファイル（p2_all_nickonly.tsv1）について，書き込みが多い順に10件の結果を表示せよ
- 同様の処理を下記ファイルについても実行してみよ
  - http://snakamura.org/tsv/p1_all_nickonly.tsv

内容を抽出しよう

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

- 抽出前のデータはタブ区切りで日時，ニックネーム，書き込みが並んでいる
- ここから書き込み時分だけ，ニックネームや書き込みの内容だけを抽出するには？



その前に

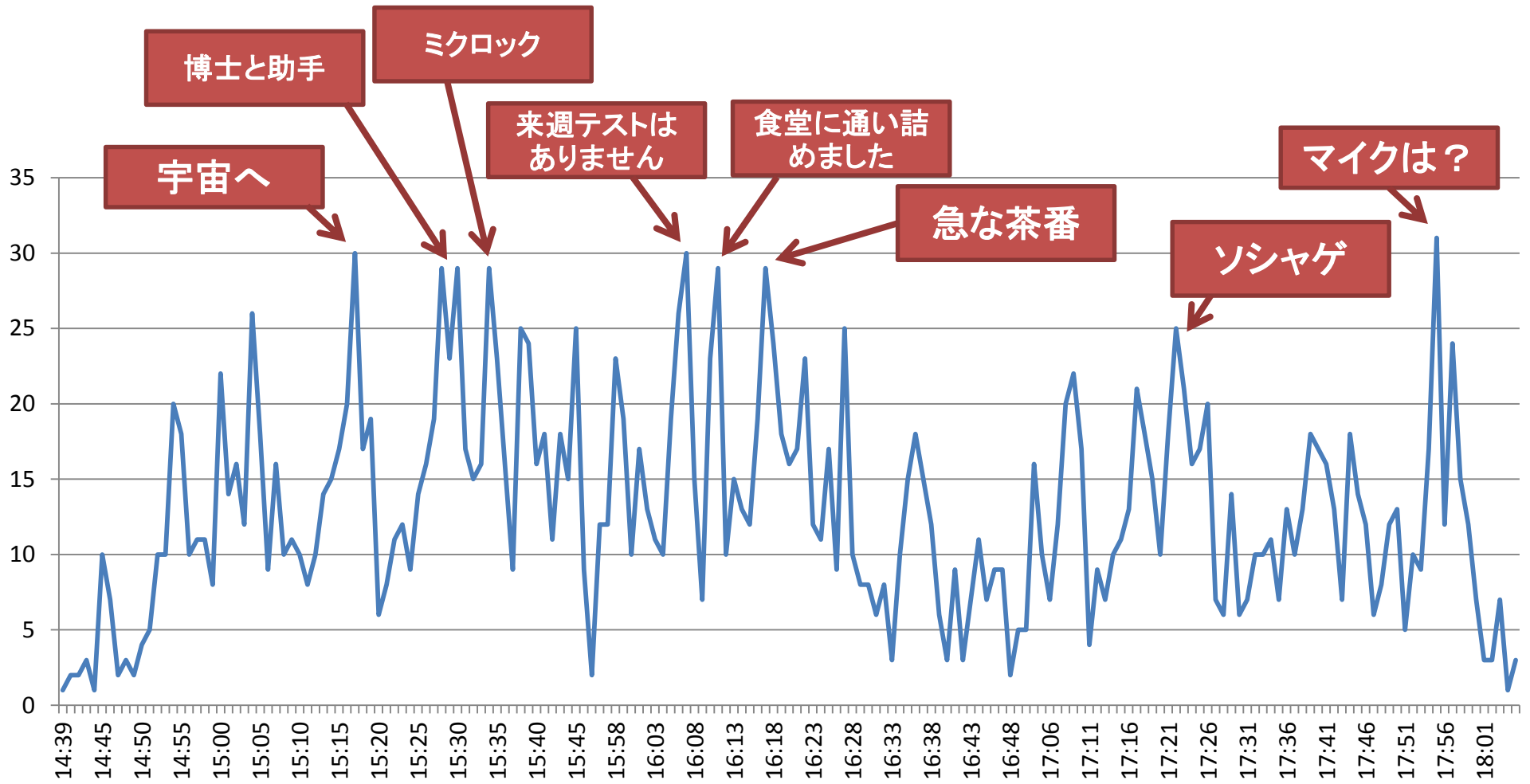
明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

- 書き込み時間だけを抽出したものについて，書き込み時分のみを取り出してカウントしたい

webox.sakura.ne.jp - PuTTY

```
2014-01-20 14:39:59
2014-01-20 14:41:01
2014-01-20 14:41:23
2014-01-20 14:42:19
2014-01-20 14:42:28
2014-01-20 14:43:09
2014-01-20 14:43:19
2014-01-20 14:43:47
2014-01-20 14:44:47
2014-01-20 14:45:01
2014-01-20 14:45:04
2014-01-20 14:45:05
2014-01-20 14:45:09
2014-01-20 14:45:10
2014-01-20 14:45:17
2014-01-20 14:45:30
2014-01-20 14:45:32
2014-01-20 14:45:40
2014-01-20 14:45:47
2014-01-20 14:46:01
2014-01-20 14:46:09
2014-01-20 14:46:13
2014-01-20 14:46:25
p2_all_timeonly.tsv
```

こんなグラフを作りたい！



正規表現（抽出）

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

- 基本的には下記のフォーマットにのっとる

変数 =~ /正規表現パターン/;

- 正規表現でまず覚えておくこと
  - 普通に書いた文字はそのまま適用される
  - () で囲まれた部分を抽出する
  - [] で囲まれた部分に含まれる文字列とマッチ
    - [1234] だと1234のいずれかの文字にマッチする語
  - [] の中で ^ を書くとそれ以外という意味

[演習]

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

- MS，FMS，NDとマッチする語が何回登場するかをそれぞれカウントしてみよ？
  - http://snakamura.org/files/ims.csv

– ヒント
  - if($line =~ /MS/){ ... } でMSというキーワードとマッチング
  - if($line =~ /FMS/){ ... } でMSというキーワードとマッチング
  - if($line =~ /ND/){ ... } でMSというキーワードとマッチング

MS, FMS, NDを数える

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室


```
open( FILE, "ims.csv" );
my $ms = 0;
my $fms = 0;
my $nd = 0;
while( my $line = <FILE> ){
  chomp($line);
  if( $line =~ /MS/ ){
    $ms ++;
  } elsif( $line =~ /FMS/ ){
    $fms ++;
  } elsif( $line =~ /ND/ ){
    $nd ++;
  }
}
close( FILE );
print "MS  = " . $ms . "¥n";
print "FMS = " . $fms . "¥n";
print "ND  = " . $nd . "¥n";
```

結果がおかしいのは何故？

[演習]

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

- 時間と分だけを抜き出して出力するには？
  - p2_all_timeonly.tsv をチェックすると以下の様なフォーマットになっている

2014-01-20 14:39:59	14:39
2014-01-20 14:41:01	14:41
2014-01-20 14:41:23	14:41
2014-01-20 14:42:19	14:42
2014-01-20 14:42:28	14:42
2014-01-20 14:43:09	14:43
2014-01-20 14:43:19	14:43
2014-01-20 14:43:47	14:43
2014-01-20 14:44:47	14:44
2014-01-20 14:45:01	14:45

見ていくと…

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

2014-01-20 14:39:59
2014-01-20 14:41:01
2014-01-20 14:41:23
2014-01-20 14:42:19
2014-01-20 14:42:28
2014-01-20 14:43:09
2014-01-20 14:43:19
2014-01-20 14:43:47
2014-01-20 14:44:47
2014-01-20 14:45:01

1. 「数字4つ」
2. 「マイナス」
3. 「数字2つ」
4. 「マイナス」
5. 「数字2つ」
6. 「スペース」
7. 「数字2つ」
8. 「コロン」
9. 「数字2つ」
10. 「コロン」
11. 「数字2つ」

正規表現を書いてみる

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

- 「数字2つ」「コロン」「数字2つ」の部分を抽出したらよい

```
$line =~ /([0123456789][0123456789]:[0123456789][0123456789])/;
```

- [] の中を何回繰り返すか？は続けて
  - {数字} で数次回繰り返す
  - + で1回以上繰り返す
  - * で0回以上繰り返す

  となるので…

```
$line =~ /([0123456789]{2}:[0123456789]{2})/;
```

マッチした情報をどう出力？

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

- （）でマッチした結果はそのカッコの順番に応じて $1, $2, $3, $4, … と取得できる

```
$line =~ /([0123456789]{2}:[0123456789]{2})/;
```

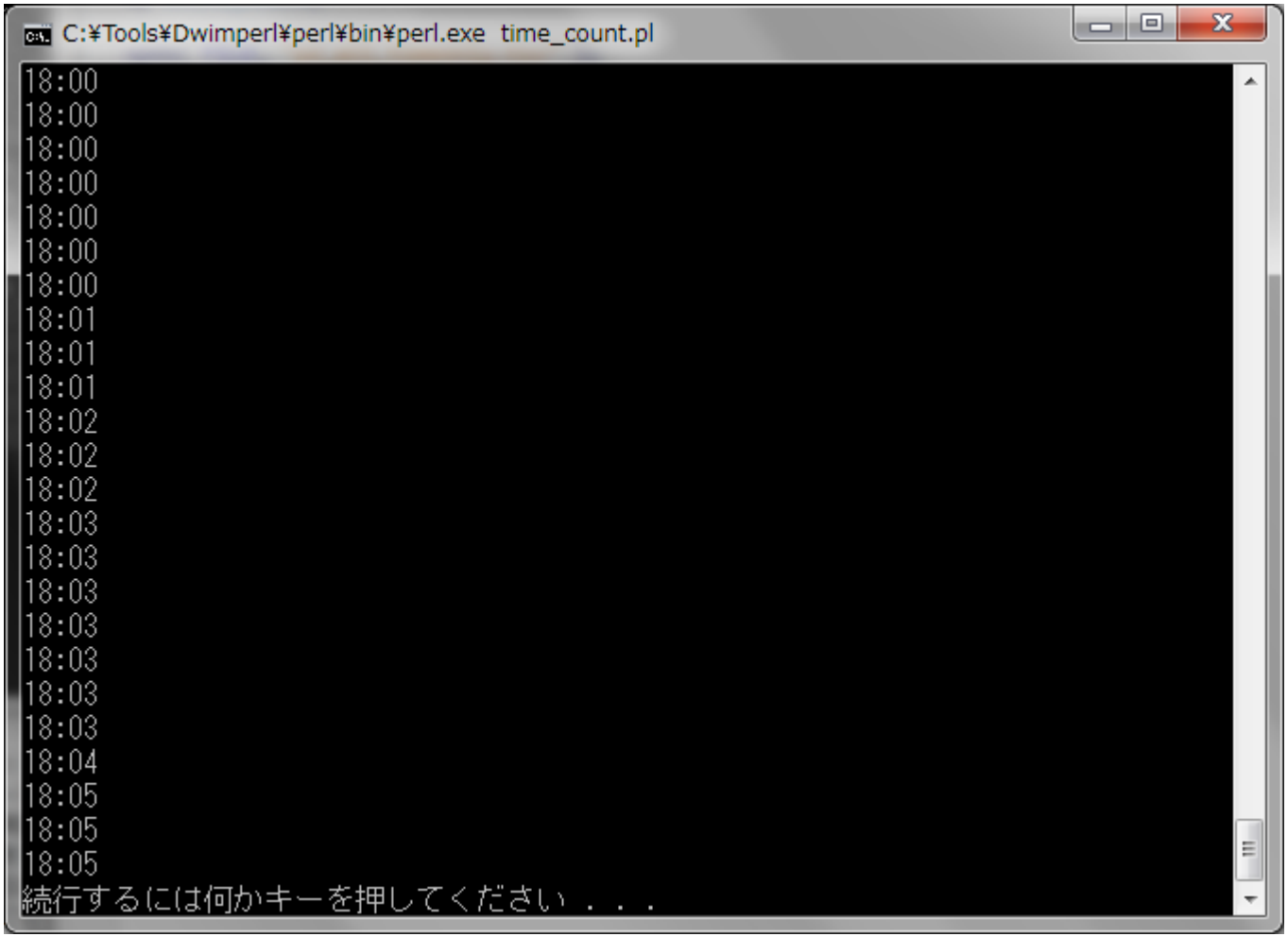
- というパターンの場合，1つ目の（）のものがそれに該当するので，

```
print   STDOUT  $1 . "¥n";
```

- というコードで結果を表示できる．¥nは改行

出力結果

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室



ちなみに

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

- 「0～9までの数字2つ:0～9までの数字2つ」というパターンだと，時分だけじゃなくて分秒もひっかかってしまいそうだが…
- 正規表現は最初にマッチしたところを優先するため，今回は問題なし
  - きっちりやるなら，最初にスペースを入れるとか，最後にコロンを入れるとかしたらよい

```
$line =~ / ([0123456789]{2}:[0123456789]{2}):/;
```

0123456789と打つのは面倒

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

- 省略できます！
  - 「0-9」と書くと，0から9までの全ての数字という意味

```
$line =~ /([0-9]{2}:[0-9]{2}):/;
```

  - 同様に[a-z]はaからzまでの小文字の英字，[A-Z]はAからZまでの大文字の英字となる
  - [0-9a-zA-z]とかくと，英数字とマッチする

```
$line =~ /(¥d+:¥d+):/;
```

  - 「¥d」で数字とマッチングすることが可能

時間と分を分割して取りたい

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

- カッコで分割して数字だけそれぞれ取得
- その後，$1と$2で結果を表示する

```
$line =~ / ([0-9]{2}):([0-9]{2}):/;
print  STDOUT  $1 . "時" . $2 . "分¥n";
```

[演習]

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

- 上記ファイルについて，「2014-01-20 14:39:59」を「2014年01月20日 14時39分59秒」と表示するようにプログラムを作成せよ

[演習]

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

- 下記のファイルから英字のみの行を抽出して表示せよ
  - http://snakamura.org/tsv/mix100.csv

- 下記のファイル（上記と一緒）から数字のみの行を抽出して表示せよ
  - http://snakamura.org/tsv/mix100.csv

[演習]

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

- 連想配列などを使い，下記のファイルから，時分毎のツイート件数を求めるプログラムを作成せよ
  - http://snakamura.org/tsv/p2_all_timeonly.tsv
- また，上記のファイルについて，時分ごとのツイート件数が多い順に20件出力せよ

[演習]

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

- 正規表現にチャレンジ
  - 郵便番号を抽出するにはどうするか？
  - 電話番号を抽出するにはどうするか？
  - 携帯電話かどうかを判定するにはどうするか？
  - メールアドレスを判定するにはどうするか？
    - これはかなり難易度が高い
  - URLかどうかを判定するにはどうするか？
    - 「://」があるかどうかで判定

正規表現の注意点

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

- 幾つかのそのままでは表現できない文字が存在する．その場合は「¥」エスケープシーケンスを利用する
  - ¥n　　改行
  - ¥t　　水平タブ
  - ¥¥　　文字としての「¥」
  - ¥$　　文字としての「$」
  - ¥/　　文字としての「/」
  - ¥?　　文字としての「?」
  - ¥+　　文字としての「+」
  - ¥*　　文字としての「*」
  - ¥-　　[]内にマイナスを書く時は...
  - ¥'　　シングルクォーテーション(')
  - ¥"　　ダブルクォーテーション(")
  - ¥.　　文字としての「.」（単に「.」を使うと任意の文字）
  - などなど

正規表現の用法

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

- 「.」は任意の文字にマッチ
- 「a?」はaまたは空文字列にマッチ
- 「a*」はaの0回以上の繰り返しにマッチ
- 「a+」はaの一回以上の繰り返しにマッチ
- 「a{m,n}」はm回以上n回以下の繰り返しにマッチ
- 「a{n,}」はn回以上の繰り返しにマッチ
- 「a{n}」はn回の繰り返しにマッチ.

色々あるが基本的なことだけ記憶して,
後は検索したらOK！

[演習]

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

- timeonly や nickonly ではない下記のファイルをダウンロードし，ニックネーム毎の投稿数ランキング，時分毎の投稿数ランキングを作成せよ
  - http://snakamura.org/tsv/p2_all.tsv
  - http://snakamura.org/tsv/p1_all.tsv
  - ただし，タブは「¥t」となる

- 上記ファイルについて「宮下」「菊池」「小松」という語が何回出てくるかカウントしてみよう

Webから直接情報をとる

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

- CPANを利用してLWP::UserAgent というモジュールをインストールして利用（後述）
- 下記の getweb.pl というファイルを作成しよう！

```
# ウェブページの情報を取得するときに使う
use LWP::UserAgent;

# UserAgent（ウェブアクセスするもの）を用意する
my $ua = LWP::UserAgent->new;
# ウェブページから情報を取得する
my $response = $ua->get('http://snakamura.org/teach/fms/');
# 取得した内容を表示する
print $response->content;
```

CPANで色々インストール

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

- CPANとは色々なライブラリ（モジュール）をお手軽にインストールする仕組み
  - Consoleで下記のように実行しよう

```
$ cpan
 (全部Enterでとりあえずは OK)
cpan[1]> o conf init
 (全部Enterでとりあえずは OK)
cpan[2]> install LWP::UserAgent
cpan[3]> exit
$ perl getweb.pl
```

cpan を起動する

cpan を初期化する

LWP::UserAgent をインストールする

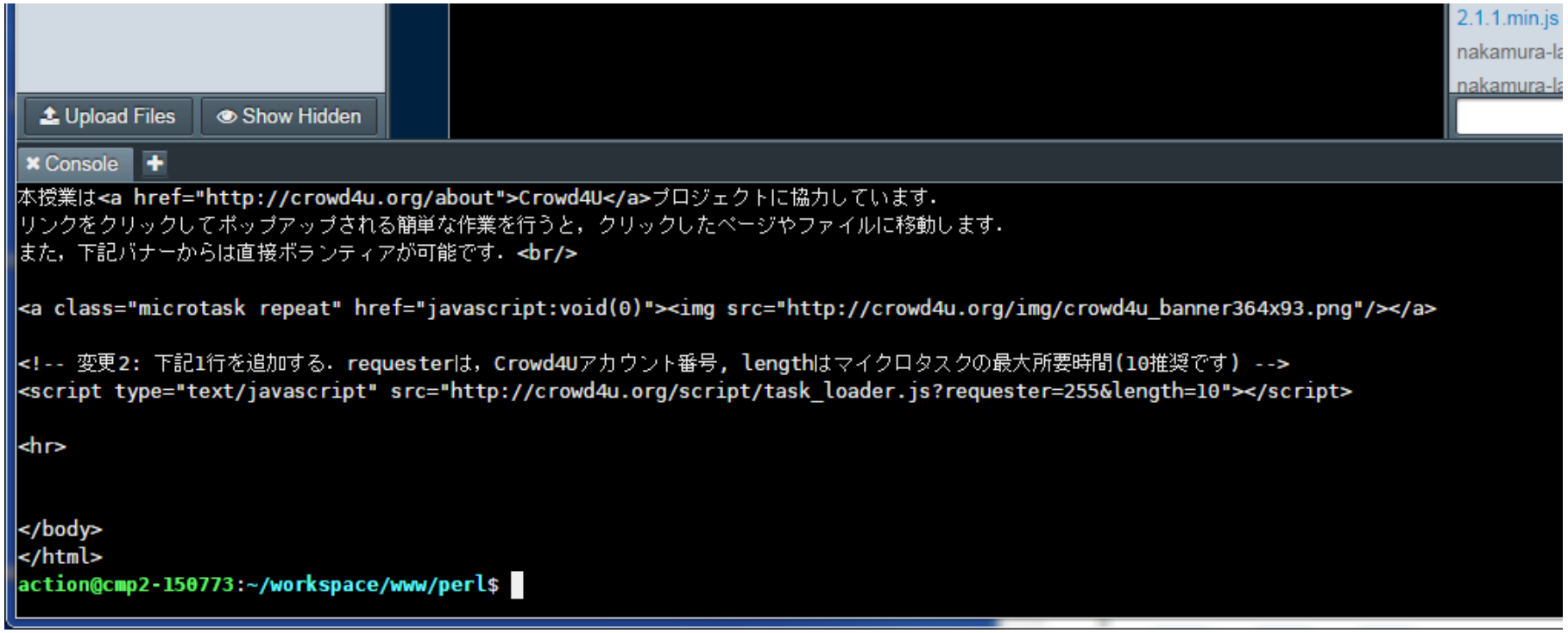
cpan を終了する

作ったファイルを実行する

ウェブページを取得

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

- perl getweb.pl
  - を実行してみよう！



[演習]

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

- 上記のウェブページからURLだけを抽出するにはどうするか？
- http://wiki.fms-meiji.jp/ の演習発表会のページから発表者のリストを取得するには？
- 出力した結果をファイルに格納するには？

文字列を分析しよう

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

- ツイートでどんなことが書かれていたか気になるが分析するのは大変である
- 自然言語処理をしてみよう！
- Yahoo! APIのテキスト解析API
  - http://developer.yahoo.co.jp/webapi/jlp/ma/v1/parse.html
  - XML::Simple というライブラリをインストール

```
action@cmp2-150773:~/workspace/www/perl$ perl getweb.pl
名詞: 庭
助詞: に
助詞: は
名詞: 二羽
名詞: にわとり
助詞: が
動詞: いる
```

CPANで色々インストール

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

- まず，色々とローカルにインストールする準備をしておこう！
  - nitrous.io にはルート権限（全てに対する書き込み権限）が無いので，ローカル（自分のディレクトリ内にインストールする）
  - そのインストール先にパスを通しておく

```
$ cd
$ emacs .bashrc
PERL5LIB="/home/action/perl5/lib/perl5"; export PERL5LIB;
という行を中に追加
$ bash
```

XML::Simpleのインストール

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

- XML::Simpleのインストールはそのままではできない（XML::Simpleという名前のものがいくつかあるため）
- i /XML::Simple/ で探す

パスの設定がおかしくなることがあるようなのでとりあえず毎回初期化

```
$ cpan
cpan[1]> o conf init
 （全部Enterでとりあえずは OK）
cpan[2]> i /XML::Simple/
 （出てきた中からXML-Simple-X.XX.tar.gzを探す）
GRANTM/XML-Simple-2.20.tar.gzが見つかった場合
 （次のページ参照）
cpan[3]> install GRANTM/XML-Simple-2.20.tar.gz
cpan[4]> exit
```

こんな感じで探す

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室


```
action@cmp2-150773:~$ cpan
Terminal does not support AddHistory.

cpan shell -- CPAN exploration and modules installation (v1.960001)
Enter 'h' for help.

cpan[1]> i /XML::Simple/
Going to read '/home/action/.cpan/Metadata'
  Database was generated on Thu, 16 Oct 2014 20:17:02 GMT
Module  < Catalyst::Action::Deserialize::XML::Simple (FREW/Catalyst-Action-REST-1.16.tar.gz)
Module  < Catalyst::Action::Serialize::XML::Simple (FREW/Catalyst-Action-REST-1.16.tar.gz)
Module  < Catalyst::View::XML::Simple (LORN/Catalyst-View-XML-Simple-0.022.tar.gz)
Module  < Data::Serializer::XML::Simple (NEELY/Data-Serializer-0.60.tar.gz)
Module  < Gtk2::GladeXML::Simple (AMNESIAC/Gtk2-GladeXML-Simple-0.32.tar.gz)
Module  < MP3::PodcastFetch::XML::SimpleParser (LDS/MP3-PodcastFetch-1.05.tar.gz)
Module  < MooseX::Storage::Format::XML::Simple (BRUNORC/MooseX-Storage-Format-XML-Simple-0.04.tar.gz)
Module  < RPC::XML::simple_type  (RJRAY/RPC-XML-0.78.tar.gz)
Module  < Template::Plugin::XML::Simple (ABW/Template-XML-2.17.tar.gz)
Module  < Test::XML::Simple      (MCMAHON/Test-XML-Simple-1.04.tar.gz)
Module  < WebService::Simple::Parser::XML::Simple (YUSUKEBE/WebService-Simple-0.21.tar.gz)
Module  < XML::LibXML::Simple    (MARKOV/XML-LibXML-Simple-0.94.tar.gz)
Module  = XML::Simple            (GRANTM/XML-Simple-2.20.tar.gz)
Module  < XML::Simple::DTDReader (ALEXMV/XML-Simple-DTDReader-0.04.tar.gz)
Module  < XML::Simple::Sugar     (CAMSPI/XML-Simple-Sugar-v1.0.2.tar.gz)
Module  < XML::Simple::Tree      (AAKD/XML-Simple-Tree-0.03.tar.gz)
Module  < XML::SimpleObject      (DBRIAN/XML-SimpleObject-0.53.tar.gz)
Module  < XML::SimpleObject::Enhanced (DBRIAN/XML-SimpleObject-0.53.tar.gz)
Module  < XML::SimpleObject::LibXML (DBRIAN/XML-SimpleObject-LibXML-0.60.tar.gz)
19 items found

cpan[2]> install GRANTM/XML-Simple-2.20.tar.gz
```

APIで文字列を分析する

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

- 先述の方法でXML::Simpleをインストールしておく（Yahoo! APIのキーも必要！）

```
use LWP::UserAgent;
use XML::Simple;
my $ua = LWP::UserAgent->new;
my $xml = XML::Simple->new();
my $response = $ua->get(
'http://jlp.yahooapis.jp/MAService/V1/parse?appid=XXXXXX&results=ma&sentence=庭には二羽にわとりがいる');
my $data = $xml->XMLin( $response->content );
my $count = $data->{ma_result}->{total_count};

for( my $i=0; $i<$count; $i++ ){
   print $data->{ma_result}->{word_list}->{word}[$i]->{pos}.": ";
   print $data->{ma_result}->{word_list}->{word}[$i]->{surface}."¥n";
}
```

Yahoo! APIのキー

[演習]

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

- 文字列処理
  - まずはテキスト部分を抽出する準備
  - どういう正規表現を書けば，語の部分だけを取り出すことが可能だろうか？
    - 正規表現の文頭は「^」文末は「$」で表現する
    - ～でないは[]の中に「^」を書く
      - [^¥t] タブ以外の文字であれば～
    - 1回以上繰り返すは「+」を使う
  - 次に，順次APIに処理を投げて，結果を連想配列などに格納していく！

さて

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

- Twitterの書き込みをテキスト解析APIで分析し，なんという語がよく登場しているかを出力！
- 手順
  - 各ユーザのツイートを順に取得する
  - ツイートをAPIに入力
  - APIからの出力を取得する
  - 取得した結果を分析する
  - 連想配列でカウントする
  - ソートして表示する

[演習]

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

- 適当なウェブページからテキスト情報を抽出し，そのページにあらわれる語の頻度を求め，出力するとともに，ファイルに保存せよ

Perlは色々省略できる言語

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室


```
%nick_count;
open( FILE, "p2_all_nickonly.tsv" );
while( <FILE> ){
  chomp;
  $nick_count{ $_ }++;
}
close( FILE );
```

my は省略してもよい
$_ は <> からの入力
=~ など色々省略可能

```
%time_count;
open( FILE, "p2_all.tsv" );
while( <FILE> ){
  /([¥d]+:[¥d]+)/;
  $time_count{$1}++;
}
close( FILE );
```

[課題]

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

- Tweetの書き込み頻度の高い名詞トップ20を作成し，その頻度ともに示せ
  - その際，明らかに名詞で無さそうなものは省け
- Tweetの書き込み分析結果で，名詞が連続するものは1つの名詞として扱い頻度を求めよ
  - 「総合数理学部」が「総合」「数」「理学部」となってしまう問題を解決する
- Tweetの書き込み頻度の高い形容詞トップ20を作成し，その頻度ともに示せ

[課題]

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

- 適当なスポーツニュースサイトを選定し，そこからニュースタイトルを列挙せよ
  - 列挙結果をファイルに保存せよ
- 適当なウェブページから画像のリンクを取得し（画像リンクはIMGタグ），その画像URLリストを作成せよ
  - また，画像をダウンロードせよ
  - LWP::UserAgent を使えば下記でできるよ！

```
open(OUT, ">ファイル名");
binmode OUT;
print OUT $response->content;
close(OUT);
```

Net::Twitter::Liteを使う！

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

- Net::Twitter::Lite というTwitterを利用するためのモジュールをインストールしよう！
  - Twitterの検索結果などを取得することが可能！

```
$ cpan
cpan[1]> o conf init
 (全部Enterでとりあえずは OK)
cpan[2]> install Net::Twitter::Lite
 (途中で何か質問されるのでy/nで回答)
cpan[3]> exit
```

#fmsの検索結果を取得

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室


```
binmode(STDOUT, ":utf8");
use strict;
use warnings;
use Net::Twitter::Lite::WithAPIv1_1;
my %CONSUMER_TOKENS = (
    consumer_key    => 'XXXXXXXXXXXXX',
    consumer_secret => 'XXXXXXXXXXXXX',
    access_token    => 'XXXXXXXXXXXXX',
    access_token_secret => 'XXXXXXXXXXXXX',
    ssl => 1,
);
my $twi = Net::Twitter::Lite::WithAPIv1_1->new(%CONSUMER_TOKENS);
my $search_str = '#fms';
my $res = $twi->search({ q => $search_str });

my $results = $res->{statuses};
foreach my $line (@$results){
  print $line->{user}->{name}." (".$line->{user}->{screen_name}."¥n";
  print $line->{text}."¥n";
}
```

データベースとの連携

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

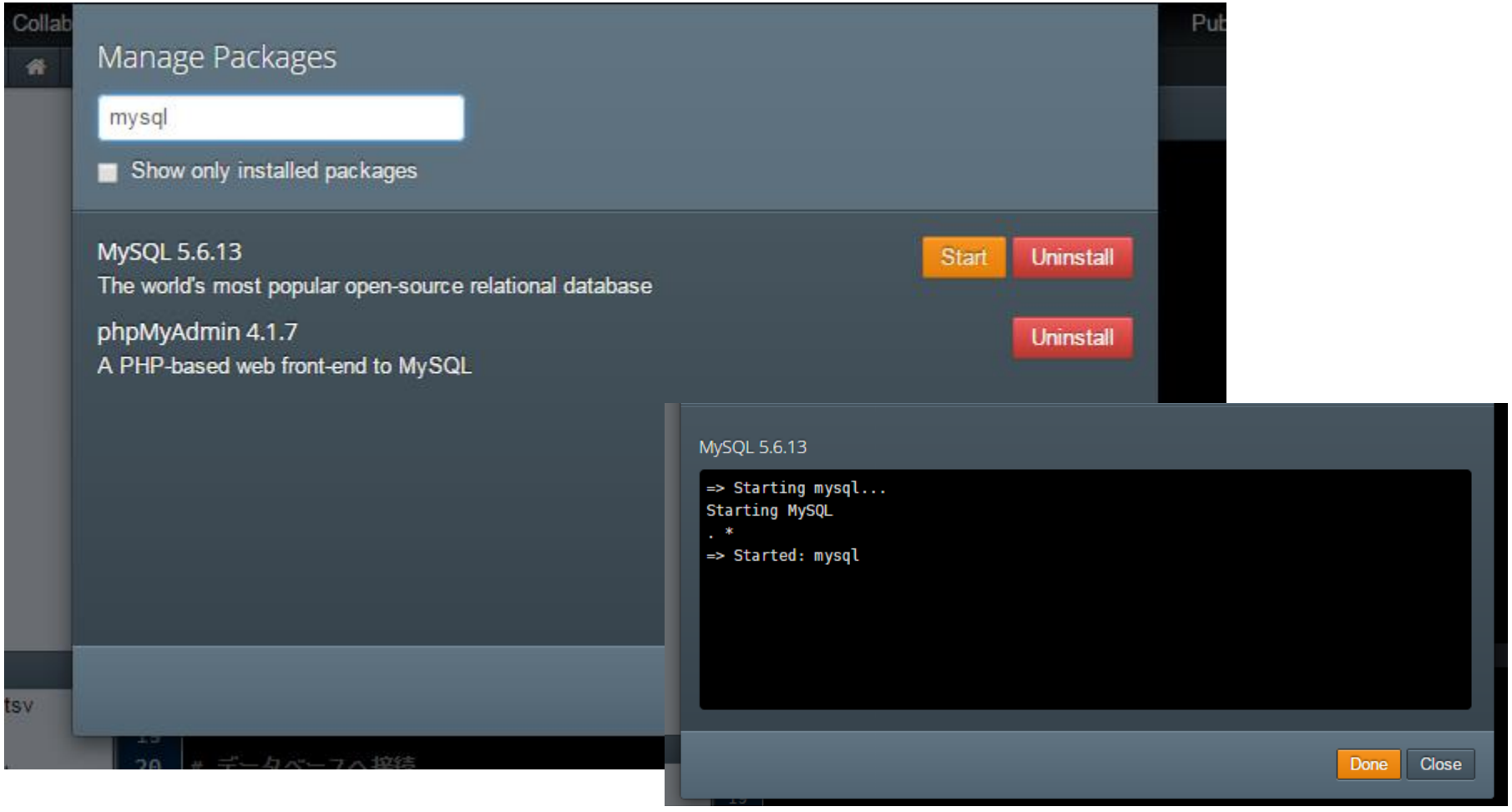
- DBIというデータベースを利用するためのモジュールをインストールしよう！
  - Twitterの検索結果などを取得することが可能！

```
$ cpan
cpan[1]> o conf init
 (全部Enterでとりあえずは OK)
cpan[2]> install DBI
cpan[3]> exit
```

先にデータベースを起動

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

- AutoParts から mysql をインストール
  - –んでもって「Start」で起動！



コンソールでmysqlに接続

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

- mysql に接続

```
$ mysql -u root -p
 (rootというアカウントでパスワード指定して接続)
Enter password:
 (パスワードは最初はないのでそのままEnter!)
mysql>
 (このmysqlのコンソールでDBの操作が可能)
```

- mysql のコンソールで
  - create database でデータベースを
  - create table でテーブルを作成しよう！
  - （データベースの操作は過去資料参照）

DBとtableを作成

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

- tweet_db というデータベース
- tweet_table というテーブル
- tweet_table は下記にしてみましょう

```
create table tweet_table
    (id int PRIMARY KEY AUTO_INCREMENT,
     tweet_time datetime,
     screen_name text, message text);
```

- tweet_table に値を放り込んでいこう！

```
insert into tweet_table
    (tweet_time, screen_name, message)
    values ('2014-10-17 18:12:50', 'nakamura', 'hello!' );
```

PerlでDBを構築！

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

- p2_all.tsv からデータを読み込んでひたすら挿入していこう！
  - 正規表現例 /^([¥d¥-]+ [¥d:]+)¥s+(¥w+)¥s+(.+)$/

- データベースの接続・操作方法
  - use DBI; でDBIモジュールを使うという宣言
  - $db = DBI->connect(各種設定); で接続
  - $db->do(SQL文); でSQL実行
  - $db-disconnect; で終了

```
binmode(STDOUT, ":utf8");
use strict;
use warnings;
use DBI; # DBIモジュール

my $user = 'root'; # MySQLのユーザ名
my $pass = ''; # MySQLのパスワード
my $database = 'tweet_db'; # 使用するデータベース名
my $hostname = 'localhost'; # データベースサーバのアドレス
my $port = '3306'; # データベースサーバに接続する時のポート番号

# データベースへ接続
my $db = DBI->connect(
  "DBI:mysql:$database:$hostname:$port", $user, $pass
) or die "cannot connect to MySWL: $DBI::errstr";

# ファイルを開いて1行ずつ取得してInsertしていく！
open( FILE, "p2_all.tsv" );
while( my $line = <FILE> ){
  chop( $line ); # 改行をカット
  $line =~ /^([¥d¥-]+ [¥d:]+)¥s+(¥w+)¥s+(.+)$/;
  my $sql_insert  = "insert into tweet_table (tweet_time, screen_name, message)";
  $sql_insert .= " values (¥'".$1."¥', ¥'".$2."¥',¥'".$3."¥')";
  print STDOUT $sql_insert . "¥n"; # sqlを念のため表示してみる
  $db->do($sql_insert) || die $db->errstr;
}
close( FILE );

# データベースから切断
$db->disconnect;
```

SQLでランキングに挑戦

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

- SQLでTweet数が多い順にニックネームと投稿数を1位～10位まで表示しよう
- SQLでつぶやかれている数が多い時分ランキングを作ってみよう
- 何らかのキーワードで呟いているTweetリストを作成してみよう
  - like を使えばOK！

こんな感じでDBから
値を取ることが可能！

```
binmode(STDOUT, ":utf8");
use strict;
use warnings;
use DBI; # DBIモジュール

my $user = 'root'; # MySQLのユーザ名
my $pass = ''; # MySQLのパスワード
my $database = 'tweet_db'; # 使用するデータベース名
my $hostname = 'localhost'; # データベースサーバのアドレス
my $port = '3306'; # データベースサーバに接続する時のポート番号

# データベースへ接続
my $db = DBI->connect(
  "DBI:mysql:$database:$hostname:$port", $user, $pass
) or die "cannot connect to MySWL: $DBI::errstr";

# $sqlの実行準備
my $sql  = "select * from tweet_table";
my $sth = $db->prepare($sql);
$sth->execute; # SQL実行
# fetchrow_arrayを使って行データを項目の配列として取り出す
while (my @data = $sth->fetchrow_array) {
   print STDOUT "[".$data[1] . "] @" . $data[2] . " " . $data[3] . "¥n";
}
# SQL文を解放
$sth->finish;

# データベースから切断
$db->disconnect;
```

[課題]

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

- 2人で1組の課題
  - Perl を利用してクロール（収集）を行い，DBを構築して，その収集したデータを元にPHPやJavaScriptなどを用いて検索閲覧可能とするシステムを構築せよ.
  - システムの仕様を説明するA4 1ページの報告資料も作成せよ（提出期限は11月7日の13時）

- 11月7日に発表会
  - 投票で最も評価が高かった発表グループはその日の懇親会無料（おごります）