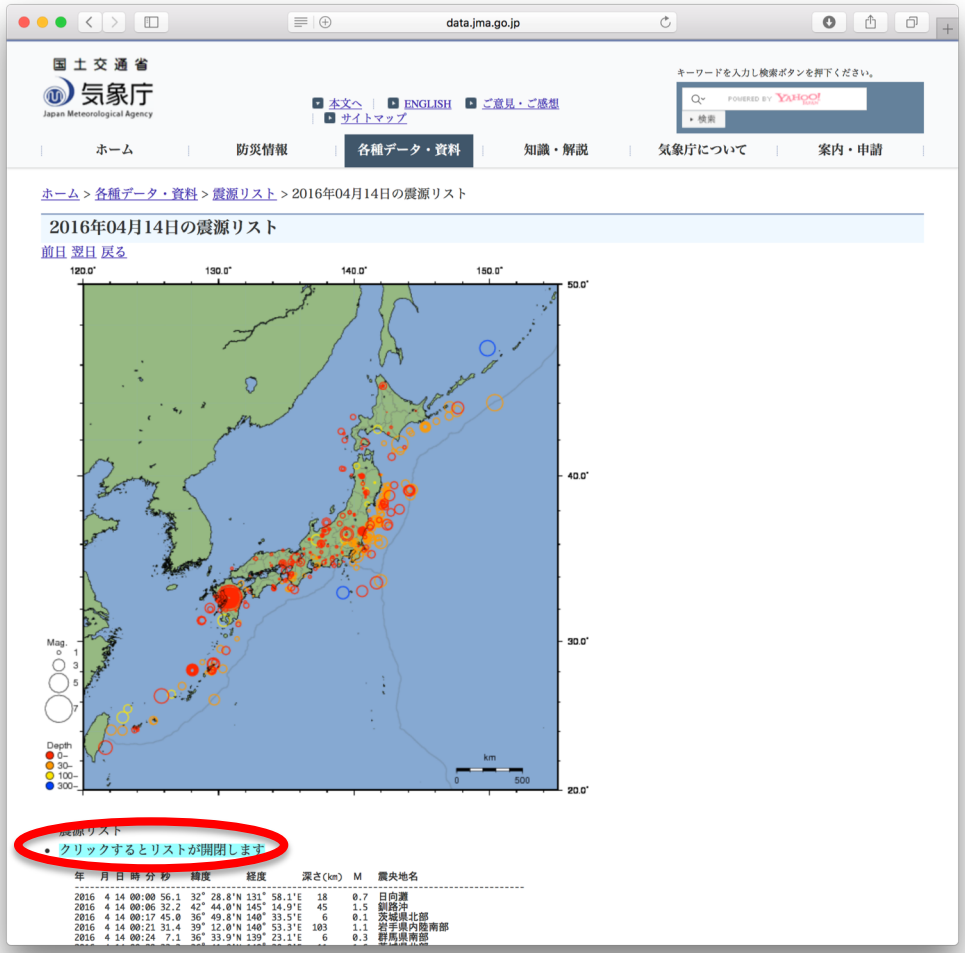
CLI: コマンドラインインターフェイス
CLI: Command Line Interface

トレーニング
Training

- 地震の情報をダウンロードしてExcelやCSVを作る
  Download earthquake info, make Excel or CSV
  http://www.data.jma.go.jp/svd/eqev/data/daily_map/20160414.html



シェルとは
What's Shell?

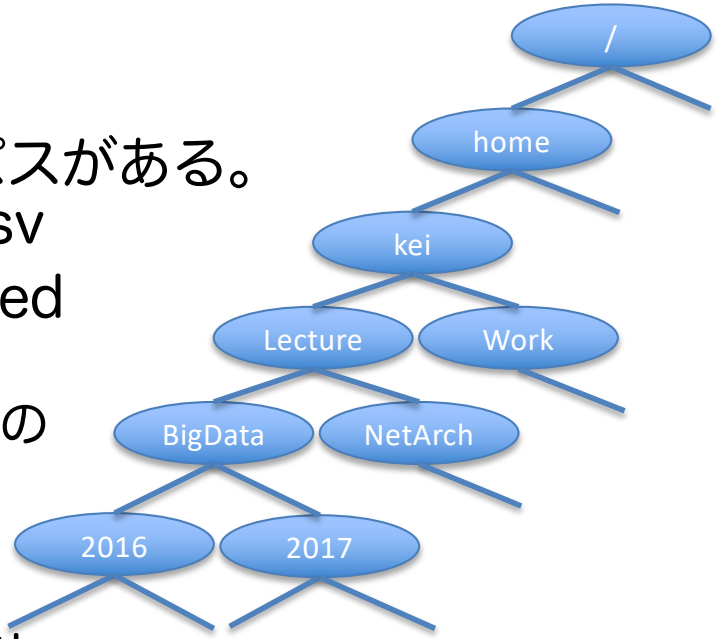
- 元はUNIX用語
- It was terminology used in UNIX.

- lS等のコマンドを実行するためのCLI (Command Line Interface)
- CLI (Command Line Interface) to execute a command such as "ls".

- 大きく、文法の違う2種類のシェルがある
- There are two types of shells, these has different grammar.
  - sh系 / sh type (sh , bash…)
  - csh系 / csh type (csh, tcsh…)

ディレクトリ操作
Directory operation

- Windows/MacOS/Linux等のOSのファイルシステムは、木構造の（階層化された）ディレクトリ構造を採用している。いわゆる「フォルダ」のこと。
- Windows/MacOS/Linux has tree structured file system. It is known as "Folder".



- CLIではファイルの指定には絶対パスと相対パスがある。
  例: /home/kei/Lecture/BigData/2017/5.csv
- Absolute path and relative path can be used to specify a file.
  - 相対パスの場合は、ワーキングディレクトリからのパスとなる。
    例: BigData/2017/5.csv
    (WDが/home/kei/Lectureの場合)
  - Using relative path, you have to specify a path from working directory
    ex: BigData/2017/5.csv (in case of working directory is /home/kei/Lecture)

ディレクトリ操作
Directory operation

- 操作 / Operation
  - ディレクトリの作成 / Make a directory　　mkdir
  - ディレクトリの削除 / remove a directory　　rmdir
  - WDの変更 / change WD　　cd
  - 現WDの表示 / print WD　　pwd

- データを扱う場合は、関連のあるデータごとにディレクトリを整理すると良い
  - 例:　Lecture/BigData/Radiation
    Lecture/BigData/Twitter
- When you store data to a file system, it might be good idea to use directory.
  - ex:　Lecture/BigData/Radiation
    Lecture/BigData/Twitter

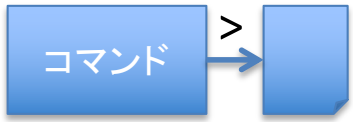
標準入力・標準出力・標準エラー出力
Standard input, output and error output

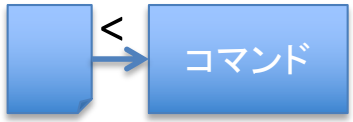
- MacOSやLinux、Windowsには、標準入力、標準出力、標準エラー出力という概念がある。CLIアプリケーションが標準で扱う入出力。
  - 標準入力
    通常の入力。デフォルトではキーボード入力が割り当てられる。
  - 標準出力
    通常の出力。デフォルトではコンソール画面が割り当てられる。
  - 標準エラー出力
    アプリケーションからのエラーメッセージや警告メッセージが出力される。デフォルトではコンソール画面が割り当てられる。

- In MacOS, Linux and Windows, there are standard input, output and error output. CLI Application uses these I/O.
  - Standard input
    Input for general use. In default, keyboard input is assigned.
  - Standard output
    Output for general use. In default, console display is assigned.
  - Standard error output
    Error message, Notification will be delivered to here. In default, console display is assigned.

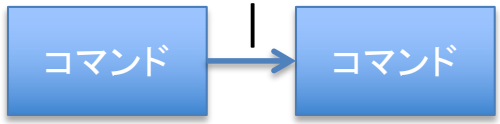
リダイレクト・パイプ
Redirect and Pipe

- リダイレクト
  - 「<」や「>」を使って、標準入出力を切り替えることができる。
  - 「<」を使って標準入力をファイルからに切り替えることができる。
  - 「>」を使って標準出力をファイルに切り替えることができる。

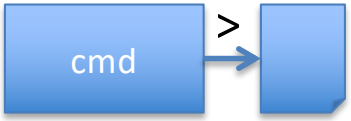


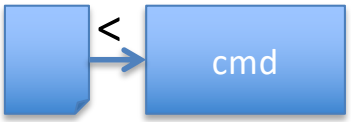


- パイプ
  - 「|」を使って、あるコマンドの標準出力を、次のコマンドの標準入力に接続することができる。

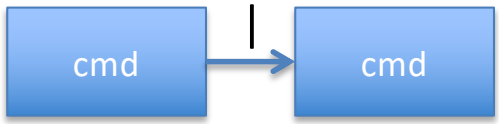


- Redirect
  - Using "<" and ">", you can change the standard input and output.
  - You can change the standard input to a file by using "<".
  - You can change the standard output to a file by using ">".





- Pipe
  - You can connect the output of a command to the input of the other command by using "|".



ファイルの指定
Files

- 1つのファイル / 1 specific file

```
% ls filename
```

- ワーキングディレクトリ内の全てのファイル
  All files in working directory

```
% ls *
```

- ".csv"で終わる全てのファイル / All files end by ".csv"

```
% ls *.csv
```

- "a"で始まるファイル / All files start with "a"

```
% ls a*
```

- aaa.csvとbbb.csv / aaa.csv and bbb.csv

```
% ls {aaa,bbb}.csv
```

ファイル操作
File operation

- ファイルのリストを見る / List files

```
% ls [filename...]
% ls -l [filename...]
```

- ファイル名を変える / Change the name of a file

```
% mv filenaem filename_new
```

- ファイルの中身を見る / View a file

```
% cat filename...
% cat -n filename…
% head filename...
% tail filename...
% less filename...
```

基本的なコマンド
Basic commands

- 文字数のカウント / Counts number of charactors

```
% wc [filename]
```

- 並び替え / Sort

```
% sort
% sort -n
% sort -r
```

- 同じ行の消去 / remove duplicated lines

```
% uniq
```

- パターン検索 / Pattern searcher

```
% grep str
% grep -v str
```

基本的なコマンド
Basic commands

- 文字列の操作 / Manipulation of a line

```
% sed s/pattern/str/g
% sed s/pattern//g
```

文字 / Char.	日本語説明	Description in English
.	任意の1文字	Any 1 character
*	直前の文字の0回以上の繰り返し	Matches the preceding element zero or more times
^	先頭	Beginning
$	後尾	Trailing

- パターンベースの操作 / Pattern based processing language

```
% awk '{print $1}'
% awk -F, '{print $2}'
```

- Webからのファイルの取得 / Get a file from Web

```
% wget http://scanningtheearth.org/data/29
% wget -O - http://scanningtheearth.org/data/29
% curl http://scanningtheearth.org/data/29
% curl -o 29.csv http://scanningtheearth.org/data/29
```

トレーニング
Training

- 下記のURLにある全てのファイルをダウンロードしなさい。
  http://scanningtheearth.org/data/

- Please download all files located in
  http://scanningtheearth.org/data/

シェルによる繰り返しファイル処理
File operation using Loop in shell

sh

csh

- csh

```
% foreach f ( * )
foreach? mv $f $f.csv
foreach? end
```

- sh

```
$ for f in *
> do
> mv $f $f.csv
> done
```

フィアル名の操作
Operation of Filename

CSH

演算子 Operator	説明 Description
変数:e Var:e	拡張子だけを取り出す。 Get extension
変数:h Var:h	上位のディレクトリ構成部のみを取り出す。 Get Path of directory
変数:r Var:r	ファイル名のベース部のみを取り出す。上位のディレクトリ構成部も残る。 Get base of filename with path of directory
変数:t Var:t	上位のディレクトリ構成部以外を取り出す。 Get except of Path of directory

SH

演算子 Operator	説明 Description
${変数#パターン} ${Var#pattern}	先頭から最短一致した部分を取り除く。 Remove beginning shortest matching part
${変数##パターン} ${Var##pattern}	先頭から最長一致した部分を取り除く。 remove beginning longest matching part
${変数%パターン} ${Var%pattern}	末尾から最短一致した部分を取り除く。 remove trailing shortest matching part
${変数%%パターン} ${Var%pattern}	末尾から最長一致した部分を取り除く。 remove trailing longest matching part

クォーテーションとエスケープ
Quotation and Escape

- 文字列をクオートする方法には2つある。
  - シングルクオート（'）を使う: 変数は展開されない
  - ダブルククオートを（"）使う: 変数が展開される
- エスケープは、shとcshあるいはbash、tcsh等によって違いがあるので注意。

- There're two choices to quote a string.
  - Use of single quote ('): variable is not expanded
  - Use of double quote ("): variable is expanded
- Please be careful that there're differences of effect of escape among sh, csh, bash, tcsh and so on.

トレーニング
Training

- 地震の情報をダウンロードしてExcelやCSVを作る
  Download earthquake info, make Excel or CSV
  http://www.data.jma.go.jp/svd/eqev/data/daily_map/20160414.html

- CSVをコマンドラインで扱うためのツール群
  Tools to manipulate CSV
  - https://csvkit.readthedocs.org/

```
% pip install csvkit
```

- データサイエンスコマンドラインツール
  - https://github.com/jeroenjanssens/data-science-at-the-command-line

```
% git clone https://github.com/jeroenjanssens/data-science-at-the-command-line.git
```

CSVtools

- Convert Excel to CSV:
```
% in2csv data.xls > data.csv
```
- Convert JSON to CSV:
```
% in2csv data.json > data.csv
```
- Print column names:
```
% csvcut -n data.csv
```
- Select a subset of columns:
```
% csvcut -c column_a,column_c data.csv > new.csv
```
- Reorder columns:
```
% csvcut -c column_c,column_a data.csv > new.csv
```
- Find rows with matching ells:
```
% csvgrep -c phone_number -r 555-555-¥d{4}" data.csv > matching.csv
```
- Convert to JSON:
```
% csvjson data.csv > data.json
```
- Generate summary statistics:
```
% csvstat data.csv
```
- Query with SQL:
```
% csvsql --query "select * from stdin where M > 5" < earthquake.csv
```

Reference

- Jeroen Janssens, “Data Science at the Command Line”, August 20, 2018
  https://www.datascienceatthecommandline.com
- https://github.com/jeroenjanssens/data-science-at-the-command-line