シェルでコンピュータを操ろう！

辻 真吾

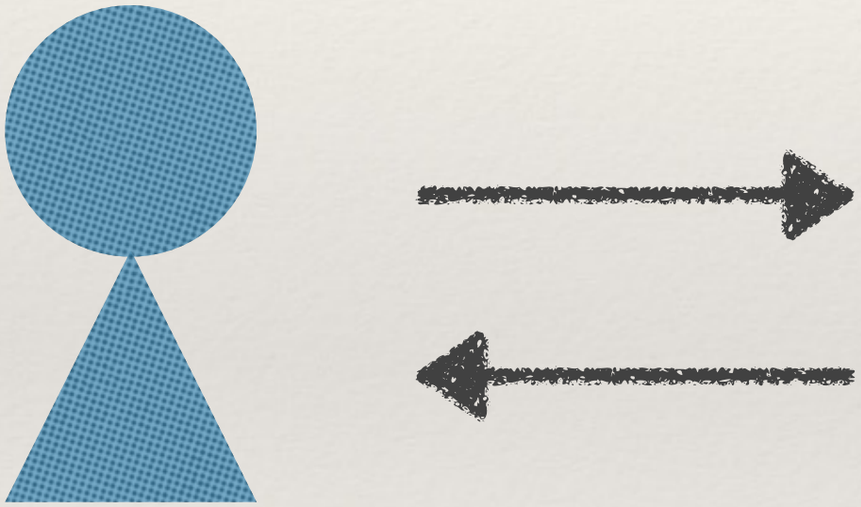
@tsjshg

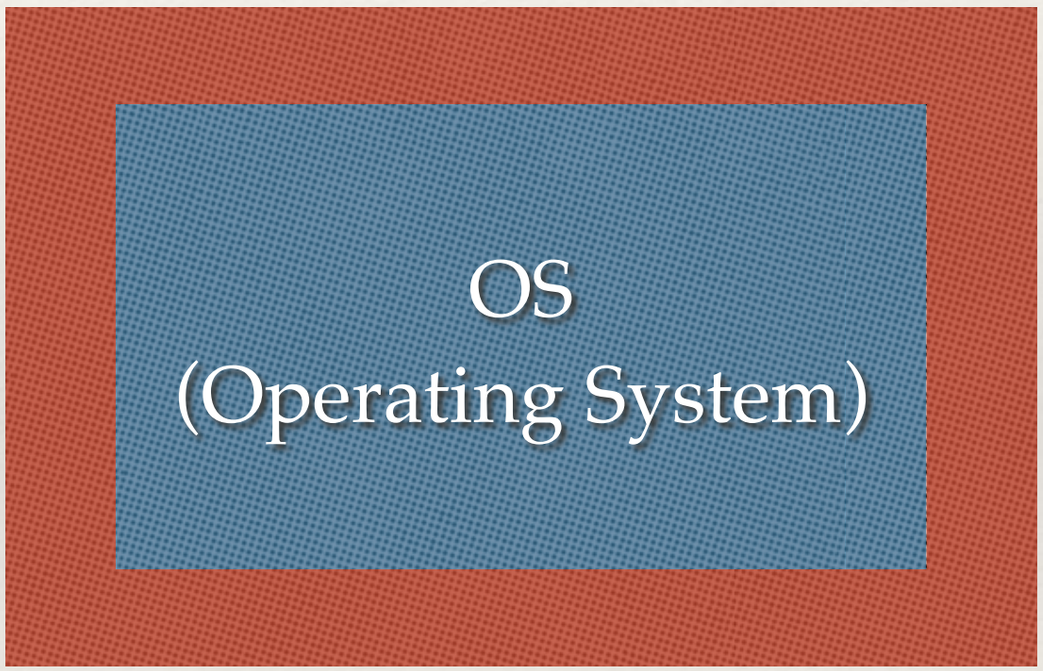
自己紹介

- 1975年生まれ
- 都内のとある大学で研究やってることになってます
  - 研究室のテーマは癌とゲノム
  - 私がやっているのはPythonで主に機械学習を使ったデータ解析
- Udemyで「実践Pythonデータサイエンス」やってます
- AI活用時代にPythonで見る夢@CTC教育サービス
  - http://www.school.ctc-g.co.jp/columns/tsuji/
- http://www.tsjshg.info/

今日はシェルの話

シェルとは？



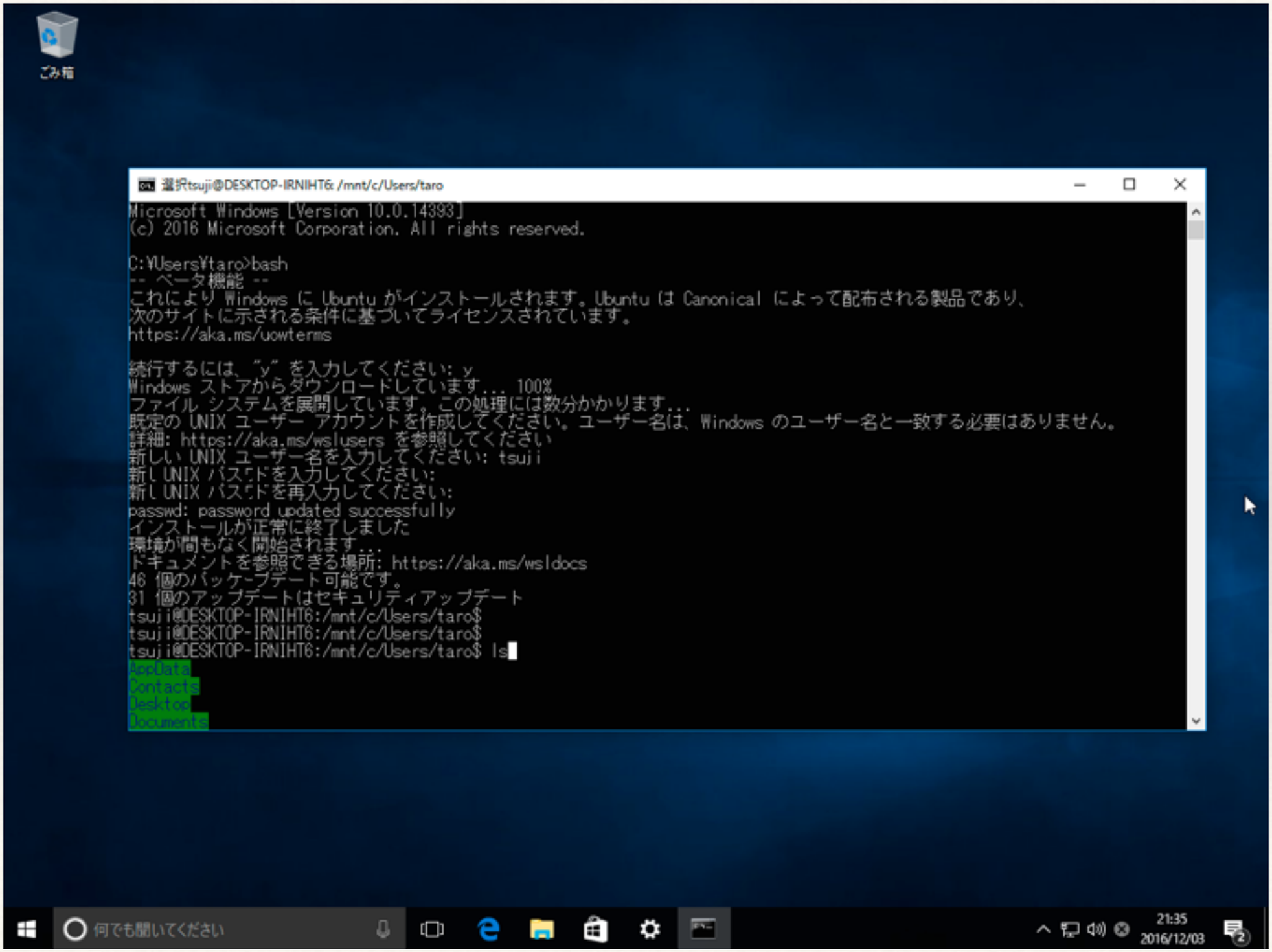


シェル（shell）

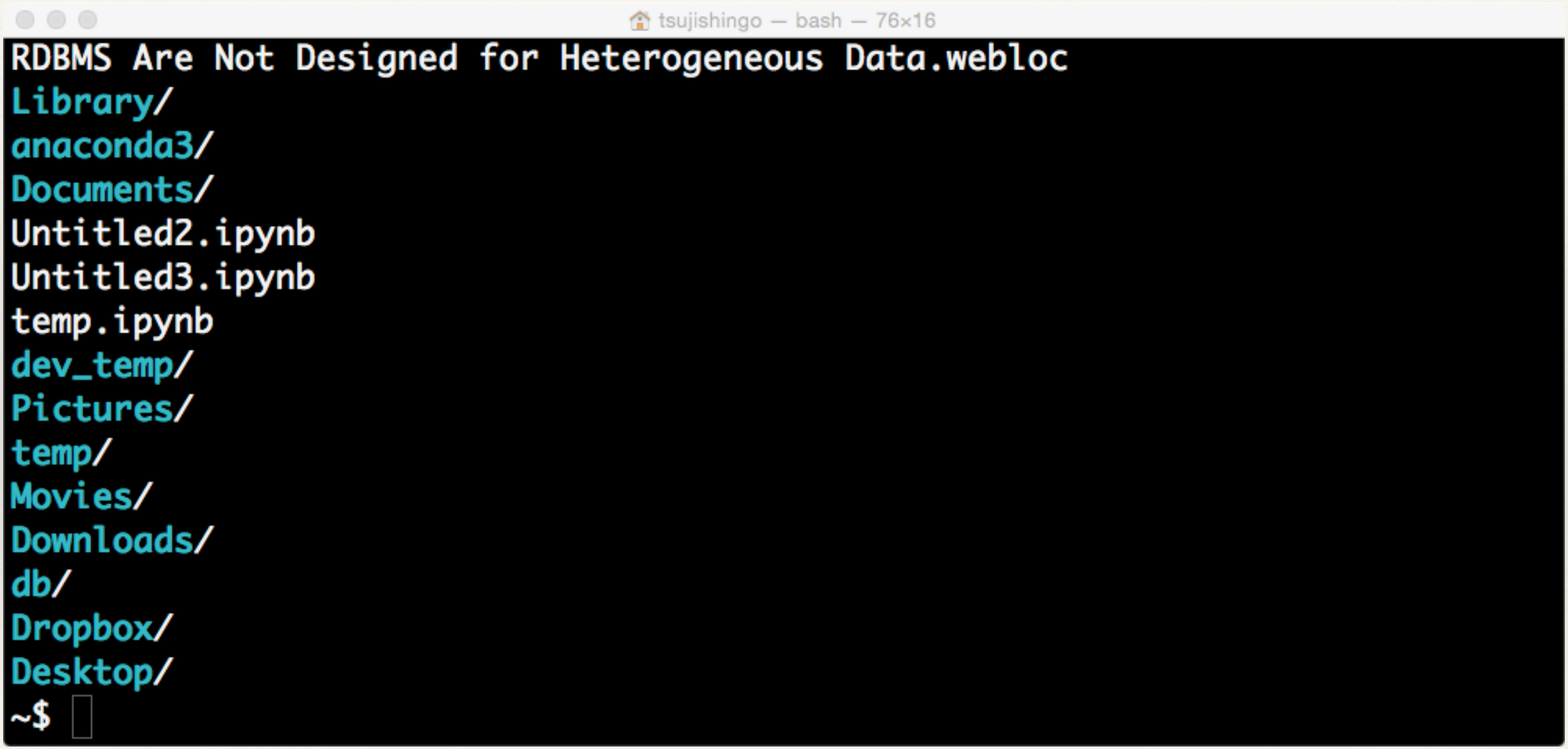
bash

- Bourne Again Shell
- 70年代に普及したBourne Shellの機能強化版
- macOSや多くのLinuxでは、ターミナル
- Windows10から、Microsoft社がbashを供給
  - 開発者モードが必要

Bash on Ubuntu on Windows



ls


tsujishingo — bash — 76×16
RDBMS Are Not Designed for Heterogeneous Data.webloc
Library/
anaconda3/
Documents/
Untitled2.ipynb
Untitled3.ipynb
temp.ipynb
dev_temp/
Pictures/
temp/
Movies/
Downloads/
db/
Dropbox/
Desktop/
~$

まずは、超基本から

ls -lrt

tsujishingo — bash — 76×16

```
d for Heterogeneous Data.webloc
drwx------@  60 tsujishingo  staff    2040 10 10 12:33 Library/
drwxr-xr-x   25 tsujishingo  staff     850 10 10 15:36 anaconda3/
drwx------+  29 tsujishingo  staff     986 11  2 08:41 Documents/
-rw-r--r--    1 tsujishingo  staff     948 11 13 16:54 Untitled2.ipynb
-rw-r--r--    1 tsujishingo  staff     622 11 13 16:55 Untitled3.ipynb
-rw-r--r--    1 tsujishingo  staff  120265 11 20 01:00 temp.ipynb
drwxr-xr-x    4 tsujishingo  staff     136 11 23 23:43 dev_temp/
drwx------+   5 tsujishingo  staff     170 11 27 10:19 Pictures/
drwxr-xr-x   27 tsujishingo  staff     918 11 27 16:18 temp/
drwx------+  35 tsujishingo  staff    1190 12  1 00:01 Movies/
drwx------+ 704 tsujishingo  staff   23936 12  3 19:29 Downloads/
drwx------@  28 tsujishingo  staff     952 12  4 10:29 db/
drwx------@  83 tsujishingo  staff    2822 12  4 10:30 Dropbox/
drwx------+   4 tsujishingo  staff     136 12  4 10:31 Desktop/
~$
```

オプション3つ。lで詳しく、tで更新時間順、それをrでひっくり返す

ll

tsujishingo — bash — 76×16

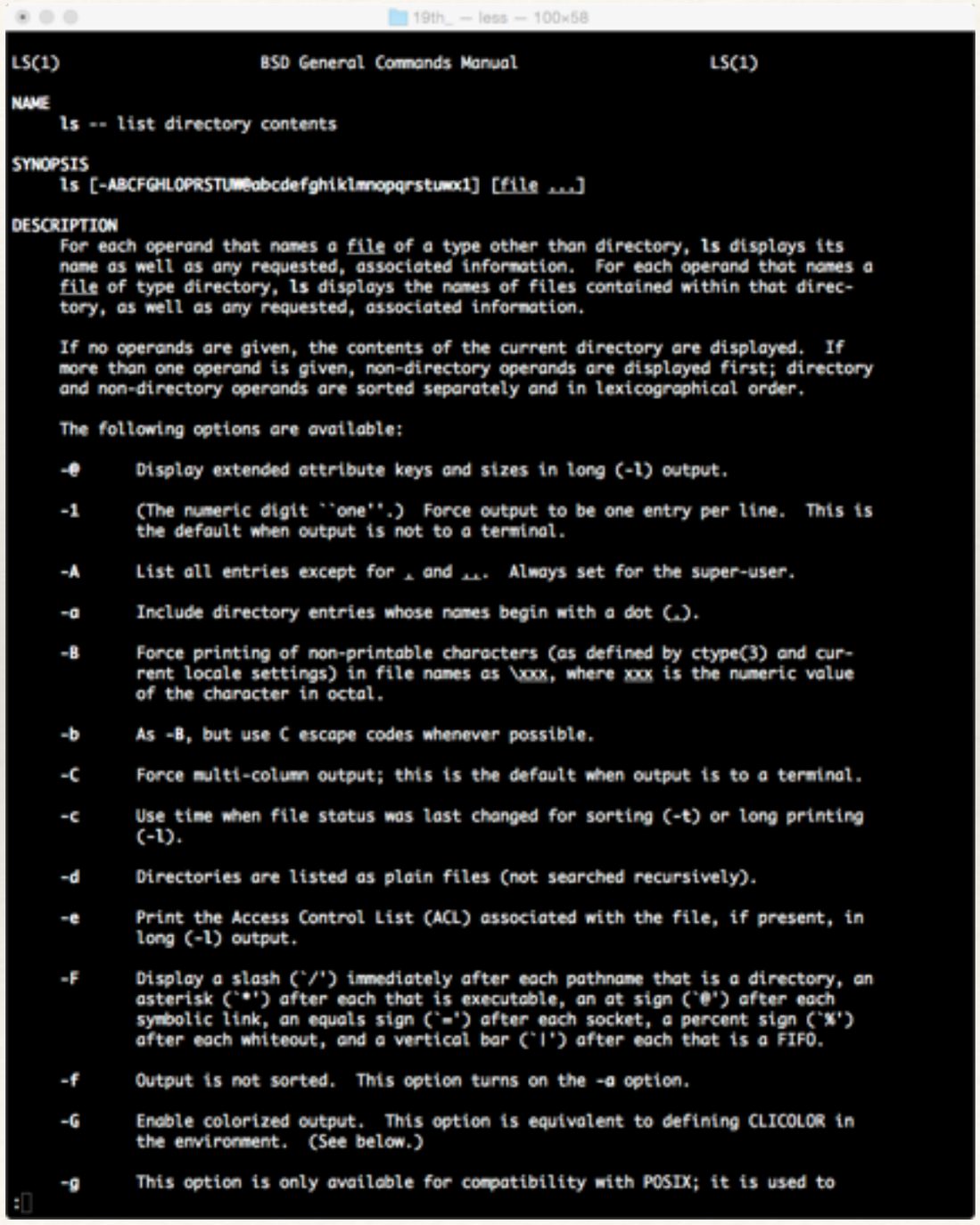
```
drwx------+   5 tsujishingo  staff   170B 11 27 10:19 Pictures/
drwxr-xr-x   27 tsujishingo  staff   918B 11 27 16:18 temp/
drwx------  251 tsujishingo  staff   8.3K 11 29 09:11 .Trash/
drwx------+  35 tsujishingo  staff   1.2K 12  1 00:01 Movies/
drwxr-xr-x    4 tsujishingo  staff   136B 12  2 09:36 .matplotlib/
drwx------+ 704 tsujishingo  staff    23K 12  3 19:29 Downloads/
-rw-------    1 tsujishingo  staff   5.1K 12  4 00:32 .viminfo
-rw-------    1 tsujishingo  staff   8.2K 12  4 02:08 .bash_history
drwx------@  28 tsujishingo  staff   952B 12  4 10:29 db/
drwx------@  83 tsujishingo  staff   2.8K 12  4 10:30 Dropbox/
drwx------+   4 tsujishingo  staff   136B 12  4 10:31 Desktop/
drwxr-xr-x+  88 tsujishingo  staff   2.9K 12  4 12:28 ./
-rw-r--r--@   1 tsujishingo  staff    38K 12  4 12:28 .DS_Store
~$ alias ll
alias ll='ls -rtlhaGv'
~$
```

llという別名（alias）で呼べる。aliasコマンドの引数で詳細を確認できる。

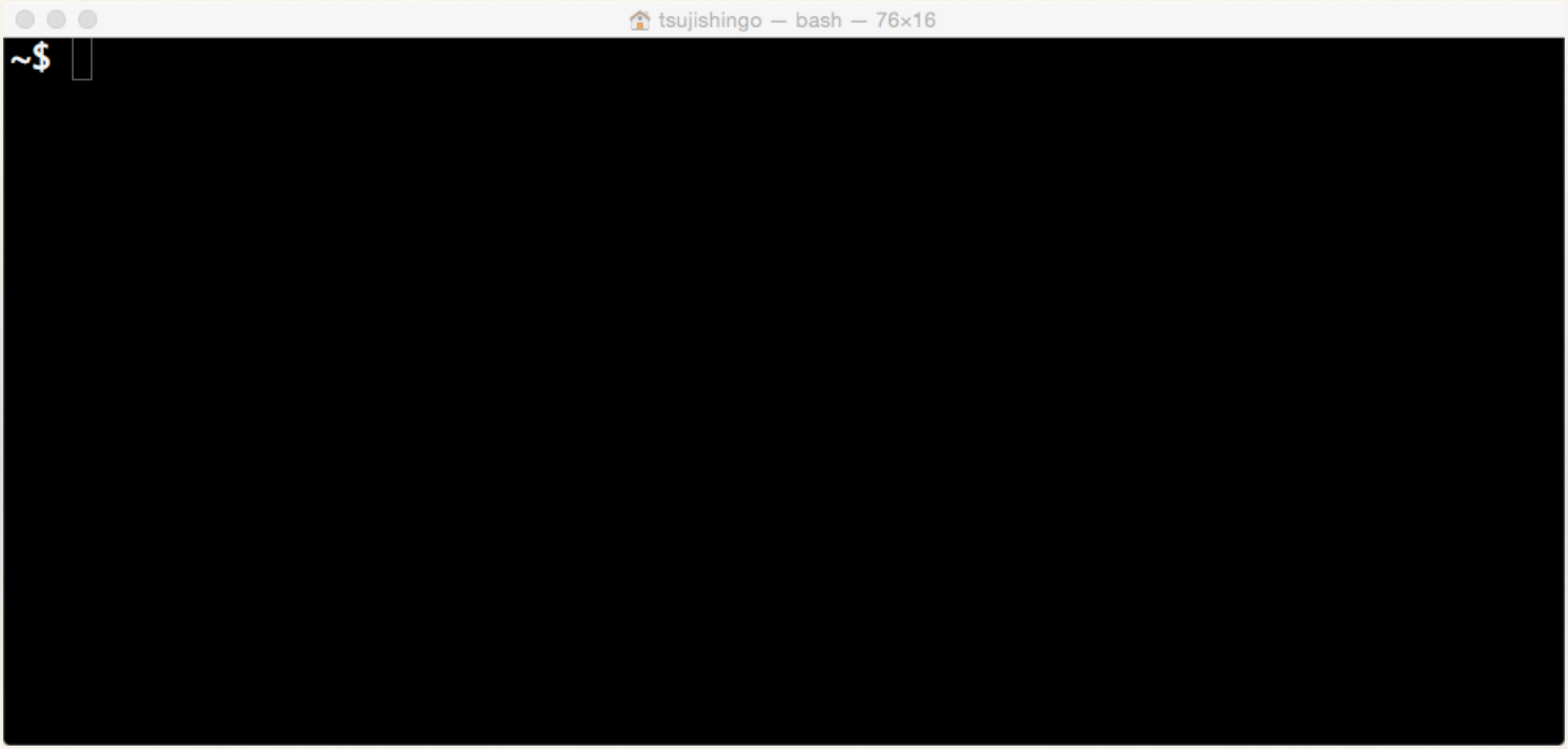
`alias ll='ls -rtlhaGv'` ←— aliasを作りたいときはこんな感じ。

man ls

- manコマンドの引数に、調べたいコマンド名を書く
- 画面が閲覧モードに切り替わるので、Qで復帰
- 今はググった方がいいか？



clear



画面を綺麗にしてくれる。Ctrl+Lでも同じ事ができます。

移動の基本

tsujishingo — bash — 76×16

```
~$
~$ pwd
/Users/tsujishingo
~$ mkdir demo
~$ cd demo
~/demo$ pwd
/Users/tsujishingo/demo
~/demo$ cd ..
~$ cd de
demo/      dev_temp/
~$ cd demo
~/demo$ cd
~$ pwd
/Users/tsujishingo
~$
```

今居る場所

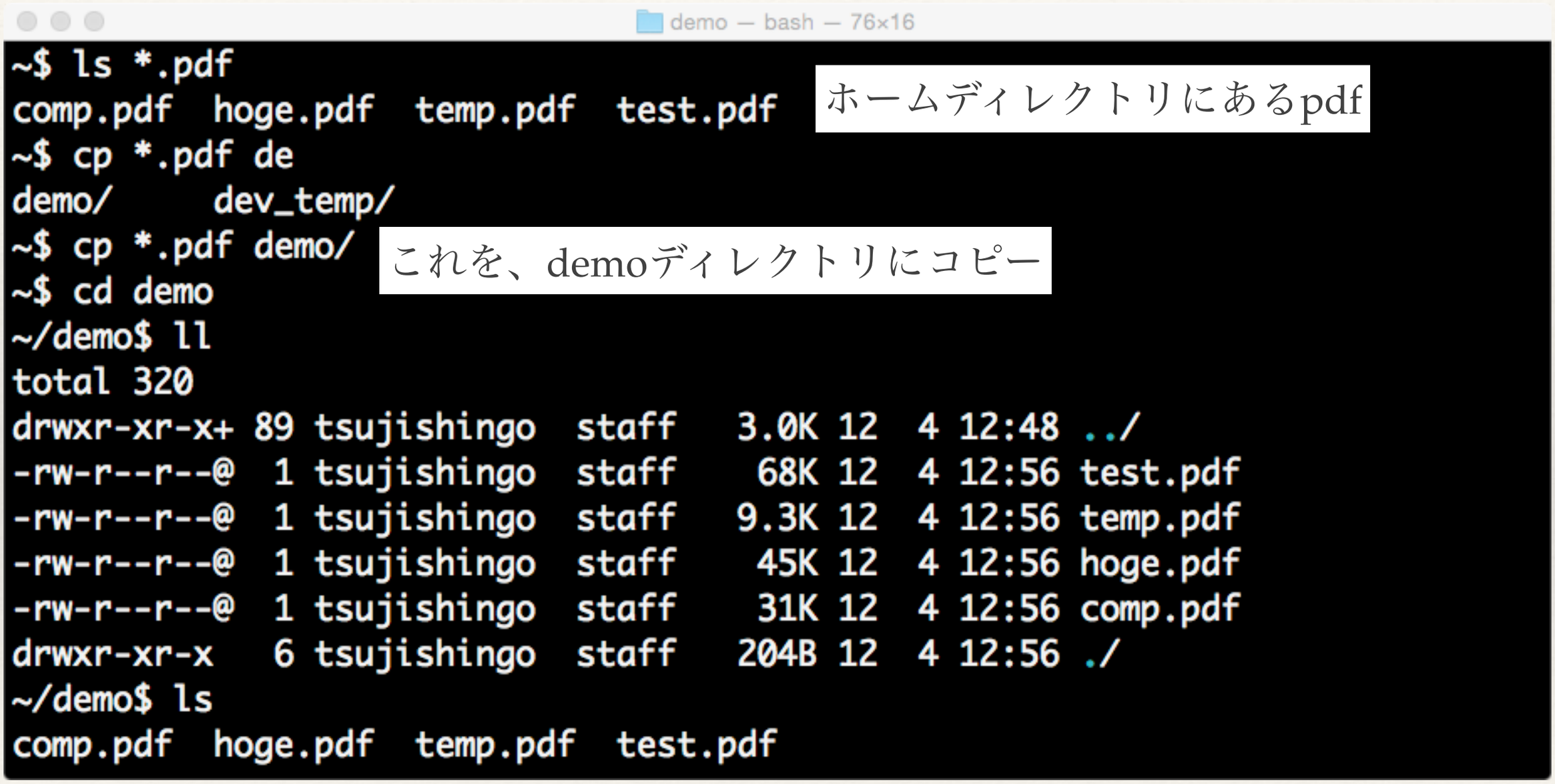
demoディレクトリを作って、cdで移動

..は1つ上

tabで補完。候補一覧の表示

引数なしでホームに帰る

ファイルのコピー



cpのところ、mvだと移動になります。

削除

tsujishingo — bash — 76×16

```
~/demo$
~/demo$ rm hoge.pdf
remove hoge.pdf? y
~/demo$ rm -f te
temp.pdf  test.pdf
~/demo$ rm -f test.pdf
~/demo$ cd
~$ rm -rf demo
~$
```

ファイルを1つ消す

確認しない

確認しないで、ディレクトリまるごと消す

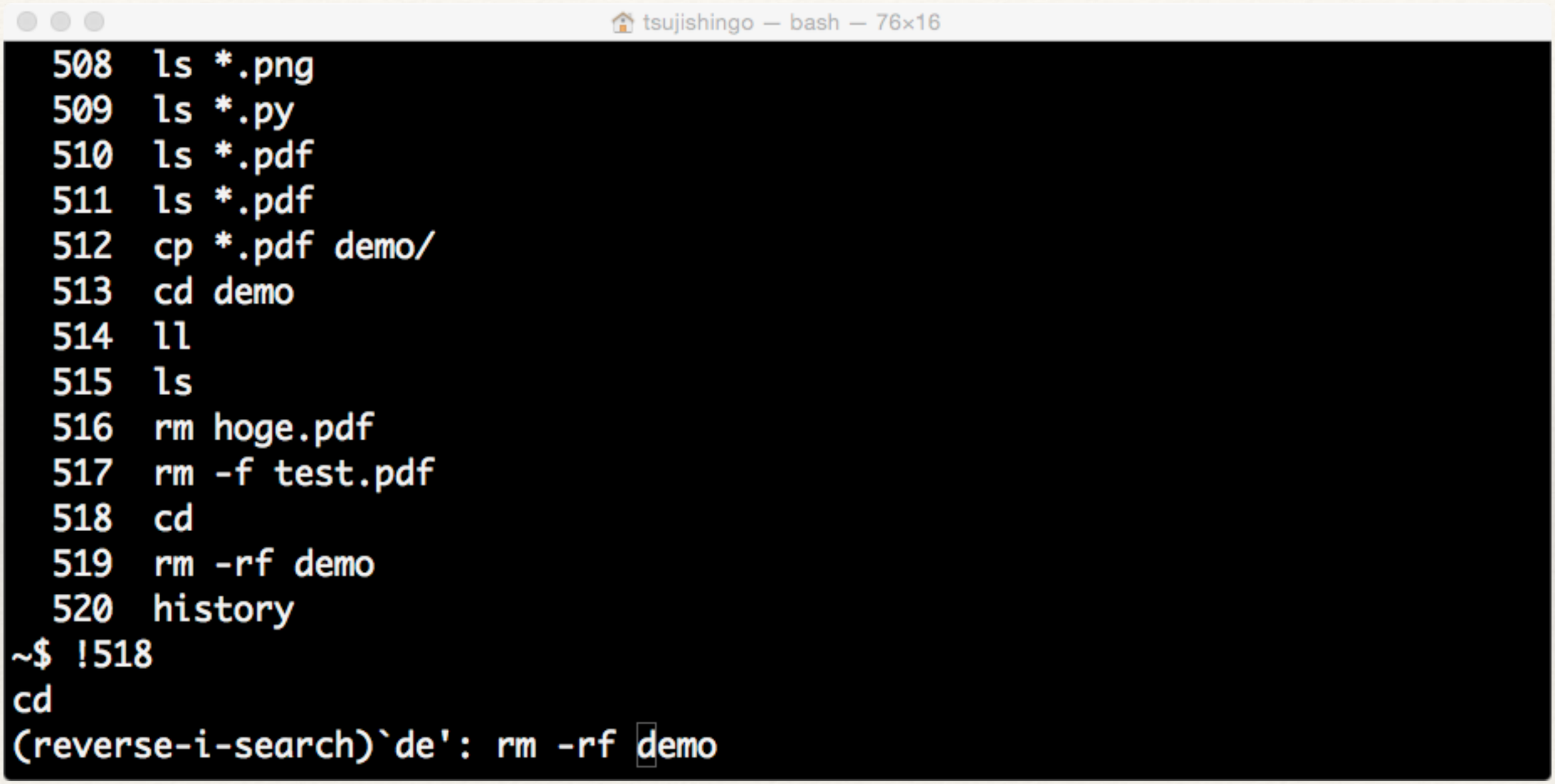
history

tsujishingo — bash — 76×16

```
  508  ls *.png
  509  ls *.py
  510  ls *.pdf
  511  ls *.pdf
  512  cp *.pdf demo/
  513  cd demo
  514  ll
  515  ls
  516  rm hoge.pdf
  517  rm -f test.pdf
  518  cd
  519  rm -rf demo
  520  history
~$ !518
cd
~$
```

↑↓で、このコマンド履歴を辿っていけます。

Ctrl+R

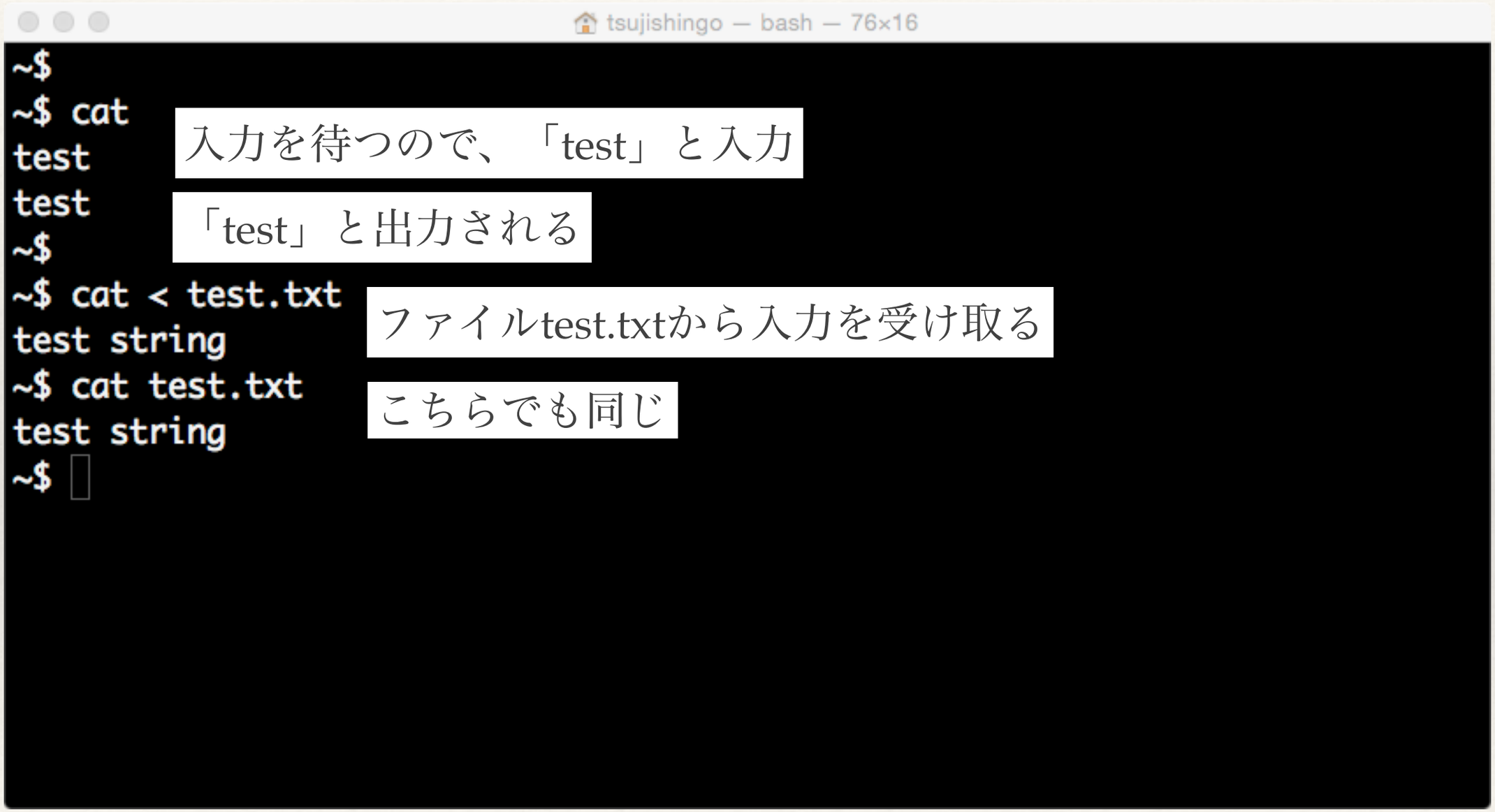


コマンドの履歴を検索できる。Ctrl+Rのあと「de」と入力したところ。

Emacsキーバインド

- Ctrl+F 右へ、Ctrl+B 左へ
- Ctrl+P 上へ、Ctrl+N 下へ
- Ctrl+A 行の先頭、Ctrl+E 行の最後
- Ctrl+D 右1文字を消す、Ctrl+H 左一文字を消す、Ctrl+K ここから最後まで消す
- Aの横をCtrlにすると、カーソルキーをほとんど使わなくてよくなる
  - macOSだと多くのアプリで、このキーバインドが使えます

標準入出力


tsujishingo — bash — 76×16
~$
~$ cat
test
test
~$
~$ cat < test.txt
test string
~$ cat test.txt
test string
~$
入力を待つので、「test」と入力
「test」と出力される
ファイルtest.txtから入力を受け取る
こちらでも同じ

標準エラー出力

```
# 標準入力から受け取った文字列を、標準出力に返す

temp = input()
print(temp)
```

```
# これも同じ

import sys
temp = input()
sys.stdout.write(temp)
```

```
# 標準エラー出力

import sys
temp = input()
sys.stderr.write(temp)
```

出力先の違い

```
import sys

sys.stdout.write('標準出力\n')
sys.stderr.write('標準エラー出力\n')
```

```
demo — bash — 76×16
~/demo$ python std_err.py
標準出力
標準エラー出力
~/demo$ python std_err.py > out.txt
標準エラー出力
~/demo$ cat out.txt
標準出力
~/demo$ python std_err.py >> out.txt
標準エラー出力
~/demo$ cat out.txt
標準出力
標準出力
~/demo$
```

両方画面にでる

＞はリダイレクト。
標準出力のみファイルに書き出される

＞＞は追加モード
相変わらず、標準エラー出力は画面

時々使う 2>&1

demo — bash — 76×16

```
~/demo$
~/demo$ python std_err.py > out.txt 2>&1
~/demo$
~/demo$ cat out.txt
標準エラー出力
標準出力
~/demo$
```

> out.txt
標準出力はファイルへリダイレクト

2>&1
標準エラー出力（2番）を1番と同じ場所へ

Mac限定

- ターミナルから、「open .」でFinderが起動します
  - しかも、カレントディレクトリが開く
- Finderからファイルを投げ込むと、フルパスをターミナルの文字列として展開してくれます

bashの設定

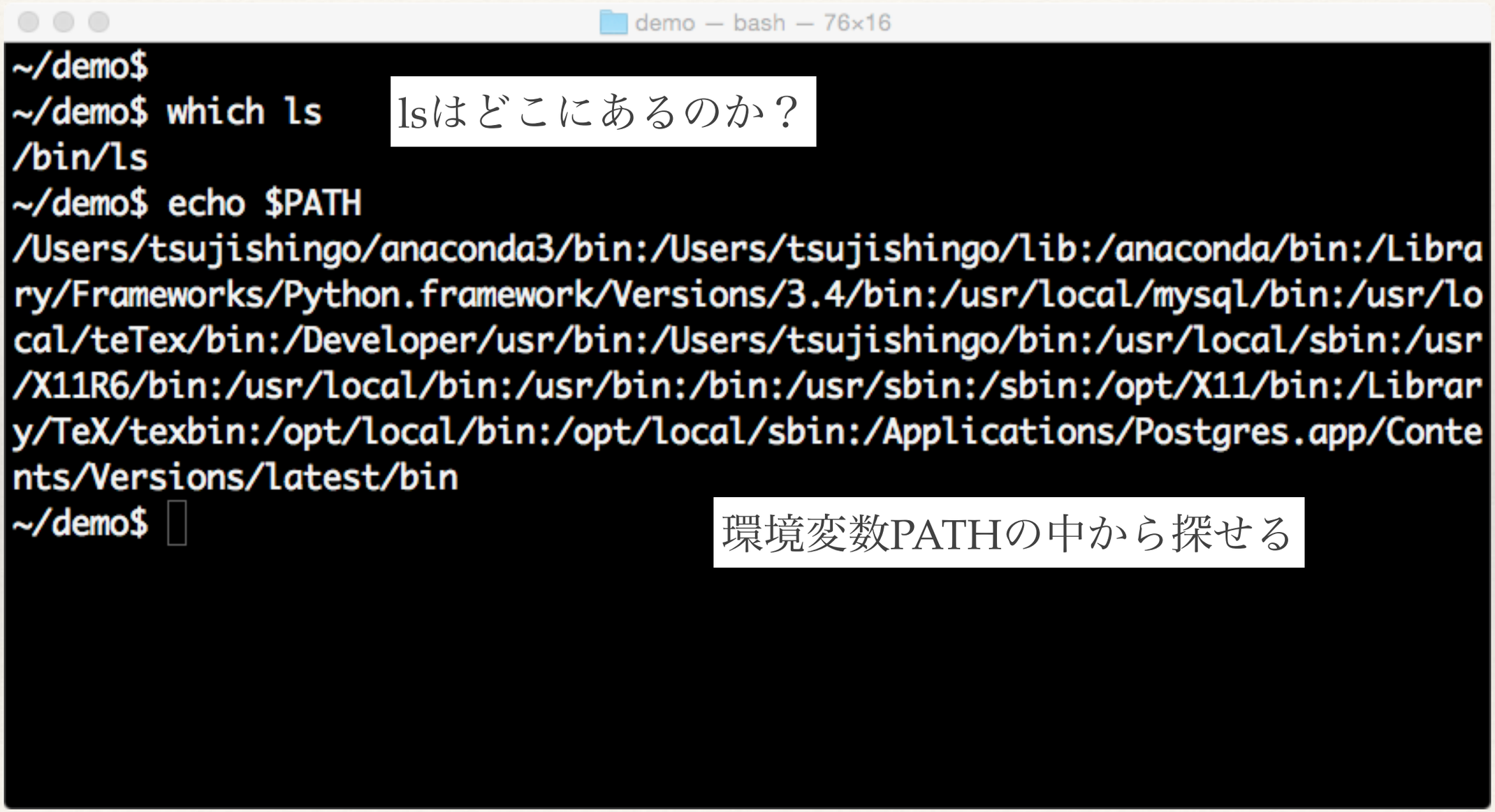
- ホームディレクトリに.（ドット）から始まるファイルがある
  - ls -aとしないと通常、見えません
- 基本は.bash_profile、.bashrc
  - .bash_profileがログイン時、.bashrcがシェルの起動時に読み込まれる

基本的な書き方

- export 変数名=“値”
- $変数名
  - 変数の中身を表示
- 区切りは「：」
- 結構ユルい（気がする）
- 追記したら、
  - source .bashrc

```
DISPLAY=:0.0
export DISPLAY

# Source global definitions
if [ -f /private/etc/profile ]; then
    . /private/etc/profile
fi
if [ -f /private/etc/bashrc ]; then
    . /private/etc/bashrc
fi
if [ -f /private/etc/profile-osxws ]; then
    . /private/etc/profile-osxws
fi

###
### User specific aliases and functions
###

#stty -ixon

# path setting
if ! echo $PATH | /usr/bin/grep -q "/usr/X11R6/bin" ; then
    PATH="/usr/X11R6/bin:${PATH}"
fi

if ! echo $PATH | /usr/bin/grep -q "/usr/local/sbin" ; then
    PATH="/usr/local/sbin:${PATH}"
fi

if ! echo $PATH | /usr/bin/grep -q "/usr/local/bin" ; then
    PATH="/usr/local/bin:${PATH}"
fi

if ! echo $PATH | /usr/bin/grep -q "${HOME}/bin" ; then
    PATH="${HOME}/bin:${PATH}"
fi

if ! echo $MANPATH | /usr/bin/grep -q "/usr/local/share/man" ; then
    MANPATH="/usr/local/share/man:/usr/share/man:/usr/X11R6/man"
fi
PATH="/usr/local/mysql/bin:/usr/local/teTex/bin:/Developer/usr/bin:${PATH}"
MANPATH="/usr/local/teTex/man:${MANPATH}"
export PATH MANPATH

# for Postgresql
export PATH=$PATH:/Applications/Postgres.app/Contents/Versions/latest/bin


# set LANG
#export LANG=ja_JP.eucJP
#export LC_ALL=ja_JP.eucJP
export LANG=ja_JP.UTF-8
export LC_ALL=ja_JP.UTF-8

# 'ls' color setting
#export LSCOLORS="ExFxCxDxBxEGEDABAGACAD"
export LSCOLORS=Gxdxcxdxbxegedabagacad
```

コマンドサーチパス


demo — bash — 76×16
~/demo$
~/demo$ which ls
/bin/ls
~/demo$ echo $PATH
/Users/tsujishingo/anaconda3/bin:/Users/tsujishingo/lib:/anaconda/bin:/Library/Frameworks/Python.framework/Versions/3.4/bin:/usr/local/mysql/bin:/usr/local/teTex/bin:/Developer/usr/bin:/Users/tsujishingo/bin:/usr/local/sbin:/usr/X11R6/bin:/usr/local/bin:/usr/bin:/bin:/usr/sbin:/sbin:/opt/X11/bin:/Library/TeX/texbin:/opt/local/bin:/opt/local/sbin:/Applications/Postgres.app/Contents/Versions/latest/bin
~/demo$
lsはどこにあるのか？
環境変数PATHの中から探せる

環境変数

- env
  - 設定されている環境変数が全部みられます

- conda create などでPythonの仮想環境を作る
  - 実体は、環境変数を調整して、適切なバージョンのPythonやライブラリが動くようにしている

ファイル処理

demo — bash — 76×16

```
~/demo$
~/demo$ cat data.txt
ID      name    age
1       たろう  22
2       はなこ  21
3       じろう  19
4       ようこ  20
~/demo$ head -2 data.txt
ID      name    age
1       たろう  22
~/demo$ tail -2 data.txt
3       じろう  19
4       ようこ  20
~/demo$
```

全体を表示

先頭2行

最後の2行

並べ替え

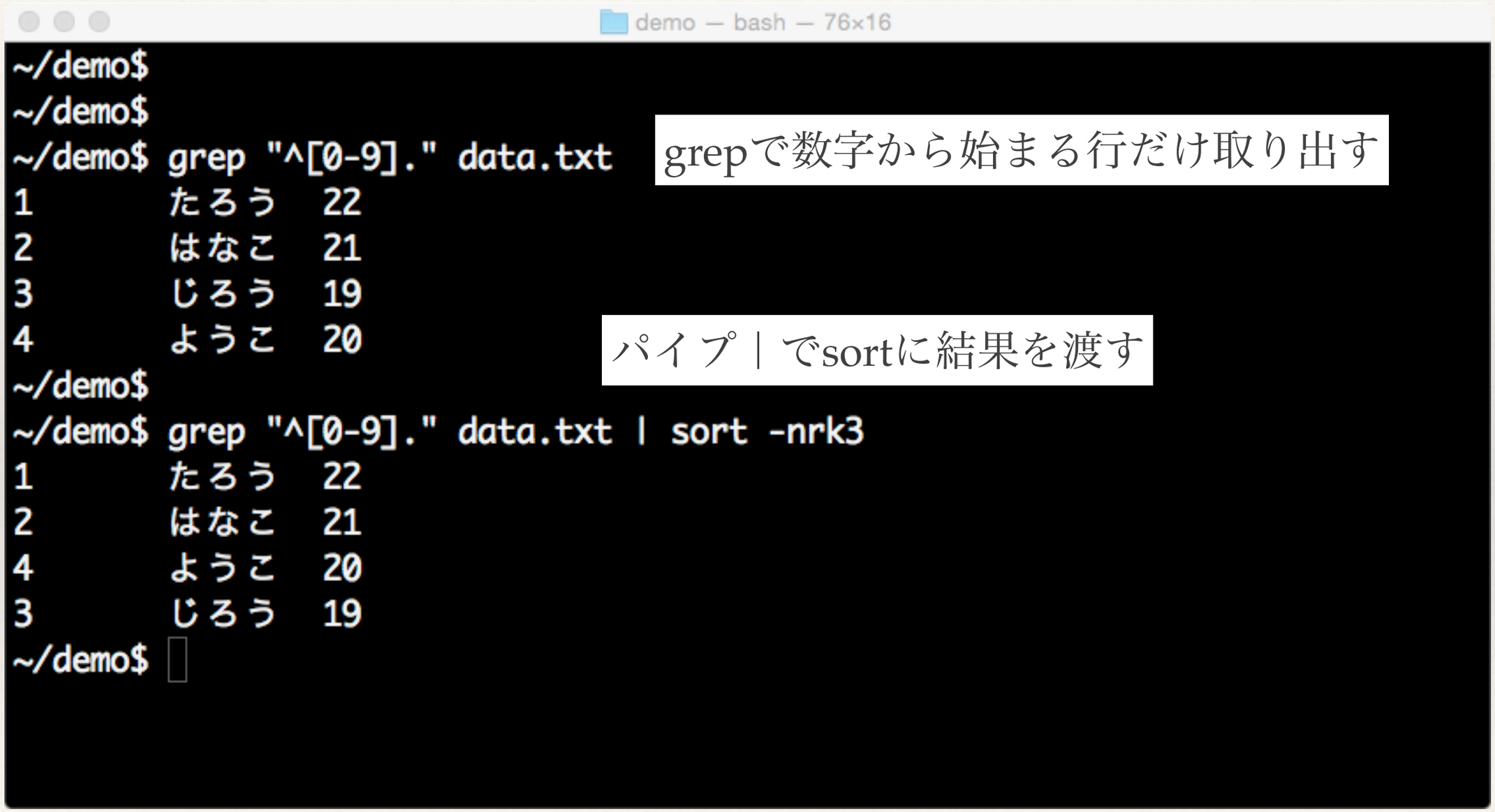
demo — bash — 76×16

```
~/demo$
~/demo$ sort data.txt
1       たろう   22
2       はなこ   21
3       じろう   19
4       ようこ   20
ID      name     age
~/demo$ sort -nk3 data.txt
ID      name     age
3       じろう   19
4       ようこ   20
2       はなこ   21
1       たろう   22
~/demo$
```

1行を1つの単語として、文字列としてソート

3列目を数字としてソート

grepとパイプ｜


demo — bash — 76×16
~/demo$
~/demo$
~/demo$ grep "^[0-9]." data.txt
1 たろう 22
2 はなこ 21
3 じろう 19
4 ようこ 20
~/demo$
~/demo$ grep "^[0-9]." data.txt | sort -nrk3
1 たろう 22
2 はなこ 21
4 ようこ 20
3 じろう 19
~/demo$
grepで数字から始まる行だけ取り出す
パイプ｜でsortに結果を渡す

awk

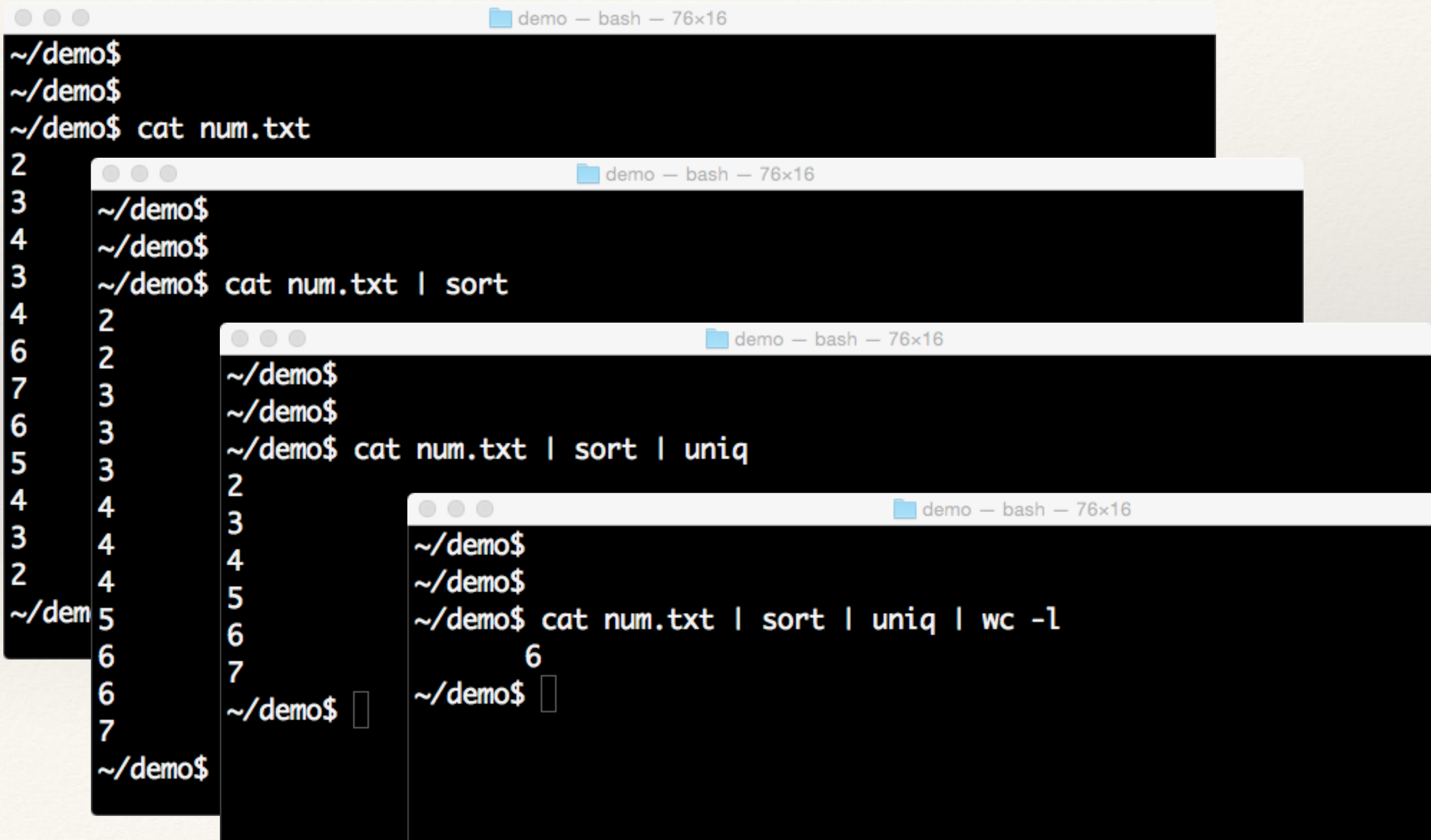
demo — bash — 76×16

```
~/demo$
~/demo$ awk '{print $2,$3}' data.txt
name age
たろう 22
はなこ 21
じろう 19
ようこ 20
~/demo$
~/demo$ awk 'BEGIN{OFS="\t"} {print $2,$3}' data.txt
name    age
たろう  22
はなこ  21
じろう  19
ようこ  20
~/demo$
```

2列目と3列目だけを表示

OFS(output field separator)をタブに設定し直す

UNIXの考え方


demo — bash — 76×16
~/demo$
~/demo$
~/demo$ cat num.txt
2
3
4
3
6
7
6
5
4
3
2
~/dem
demo — bash — 76×16
~/demo$
~/demo$
~/demo$ cat num.txt | sort
2
2
3
3
3
4
4
4
5
6
6
7
~/demo$
demo — bash — 76×16
~/demo$
~/demo$
~/demo$ cat num.txt | sort | uniq
2
3
4
5
6
7
~/demo$
demo — bash — 76×16
~/demo$
~/demo$
~/demo$ cat num.txt | sort | uniq | wc -l
6
~/demo$

まとめ

- どこでも使えるbashを紹介
- ワイルドカード＊を使ったファイルの処理
- 環境変数やaliasの設定
- 便利なEmacsキーバインド
- 小さなコマンドを組み合わせて、大きな仕事をする