

計算機演習

UNIX(1)

Apr 7, 2021

担当:坂井 郁哉

資料:湯本 航生(2020)、坂井 郁哉(2021)

課題の締め切りについて

今日の課題の期限は環境の違いにより不具合が生じる可能性を懸念し、余裕を持って一週間を設定していますが、**特に問題が生じていない方は今日中に課題を提出してください！**

- *もちろん問題が生じた方は締め切りまでに提出してくれれば減点しません。TAに相談してください。
- *みなさんの状況を把握するために今日一旦提出者の集計を行います。

GUIとCUI

どっちも利点があるので上手く使い分けてください

GUI (Graphical User Interface)

- みんなが普段使ってるPC, スマホ
- 感覚的に操作ができるやつ



CUI (Command-line User Interface)

- 今日演習する「かっこいい」やつ



CUI操作を習得する理由

奇数番号のファイルだけ削除
しておいてね



- test-11132.txt
- test-11133.txt
- test-11134.txt
- test-11135.txt
- test-11136.txt
- test-11137.txt
- test-11138.txt
- test-11139.txt
- test-11140.txt
- test-11141.txt
- test-11142.txt



- 大規模な処理が可能
- 処理の自動化が可能
- 動作が軽く、安定
- テキストデータだけでコンピュータを包括的に操作することが可能
- ネットワーク、サーバーの管理を行う上で必須
- GUIの開発に必要となる場合も

CUI操作を習得する理由

あ、今日もファイル名の書式
間違っちゃった。



- ・大規模な処理が可能
- ・処理の自動化が可能
- ・動作が軽く、安定
- ・テキストデータだけでコンピュータを包括的に操作することが可能
- ・ネットワーク、サーバーの管理を行う上で必須
- ・GUIの開発に必要となる場合も

CUI操作を習得する理由

今までの書類まだできない
の？



- 大規模な処理が可能
- 処理の自動化が可能
- 動作が軽く、安定
- テキストデータだけでコンピュータを包括的に操作することが可能
- ネットワーク、サーバーの管理を行う上で必須
- GUIの開発に必要となる場合も

CUI操作を習得する理由



- ・大規模な処理が可能
- ・処理の自動化が可能
- ・動作が軽く、安定
- ・テキストデータだけでコンピュータを包括的に操作することが可能
- ・ネットワーク、サーバーの管理を行う上で必須
- ・GUIの開発に必要となる場合も

コマンドを打ってみよう

始める前に……

計算機室で演習を行なっている学生は、
以下のコマンドをまず最初に打ってください。

```
~$ ssh asano
```

詳しくは次の講義で説明しますので、今はおまじないだと思ってください。

※自分のPCで演習を行なっている学生はこのコマンドの必要はありません。

コマンドを打ってみよう

Input: x

オプション (ハイフンの後に入力、複数指定可)

ここから入力開始
ですよ、の記号

あなたは今○○
のファイルの中に
いますよ、の記号

```
~$ echo -e 'Hello \n World!'
```

コマンド: f

引数: x

Output: $f(x)$

```
Hello  
World!
```

コマンドの意味を知りたい/どんなオプションがあるか知りたい

manというコマンドの引数にコマンド名を入れればそのコマンドのヘルプを表示できます

- ✓ さっきのechoというコマンドがなんだつたのか、調べてみましょう

Input



Output

The screenshot shows a man page for the 'echo' command. The title is 'ECHO(1) BSD General Commands Manual ECHO(1)'. The page contains the following sections:

- NAME**: echo -- write arguments to the standard output
- SYNOPSIS**: echo [-n] [string ...]
- DESCRIPTION**: The echo utility writes any specified operands, separated by single blank (' ') characters and followed by a newline ('\n') character, to the standard output.
The following option is available:
 - n Do not print the trailing newline character. This may also be achieved by appending '\c' to the end of the string, as is done by iBCS2 compatible systems. Note that this option as well as the effect of '\c' are implementation-defined in IEEE Std 1003.1-2001 ('POSIX.1') as amended by Cor. 1-2002. Applications aiming for maximum portability are strongly encouraged to use printf(1) to suppress the newline character.
- SEE ALSO**: builtin(1), csh(1), printf(1), sh(1)
- STANDARDS**: The echo utility conforms to IEEE Std 1003.1-2001 ('POSIX.1') as amended by Cor. 1-2002.

At the bottom, it says 'BSD April 12, 2003 BSD' and has a '(END)' button.

- ✓ もちろんグーグル検索した方が早いケースもあります

練習課題

* qを打てばヘルプ表示を抜けられます

先ほどの `echo -e 'Hello \n World!'` のオプションを `-n` にしたり無しにしてみましょう。また、エスケープシーケンスを `\n` から `\c` や `\v` に変えてみましょう。結果を見てこれらのオプションやエスケープシーケンスの意味を考えて下さい。ターミナルに `man echo` と入力すると、答えが分かります。

補足：歴代TAの教え「タブを使えタブを！」

コマンドラインで「ec」まで入力をしてタブを押すと、
「echo」まで勝手に補完してくれます。

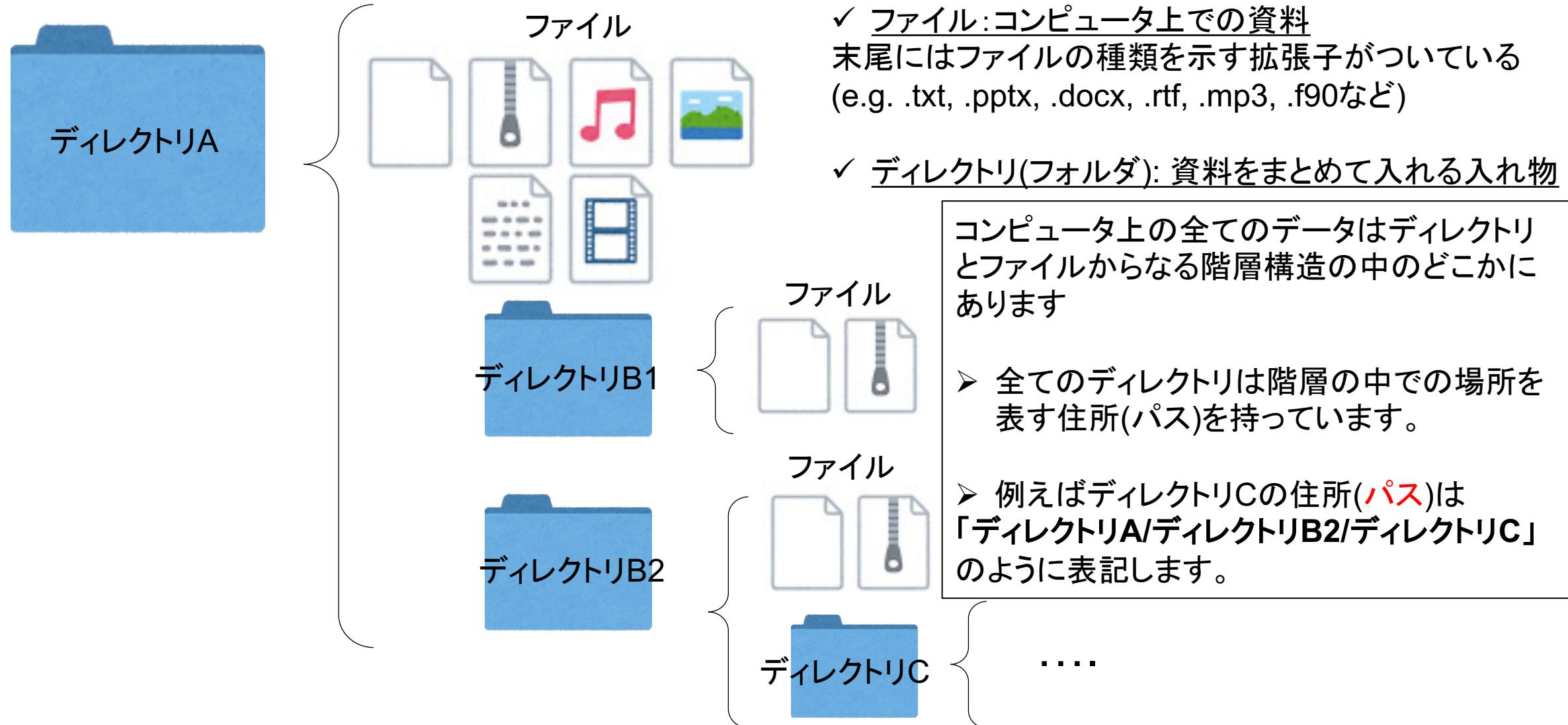
タブお兄さん



怯えながら計算機演習をするみんな

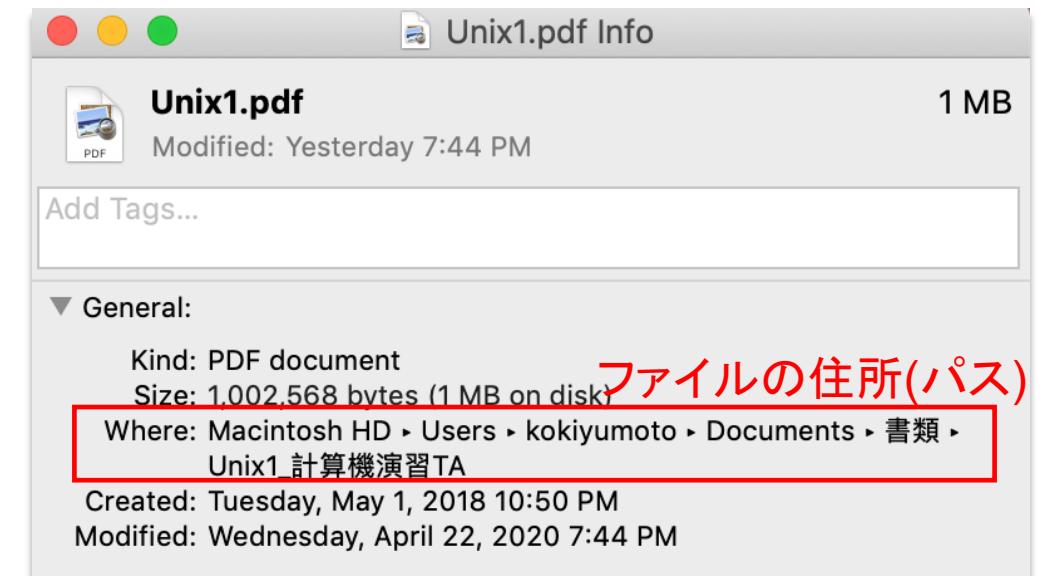
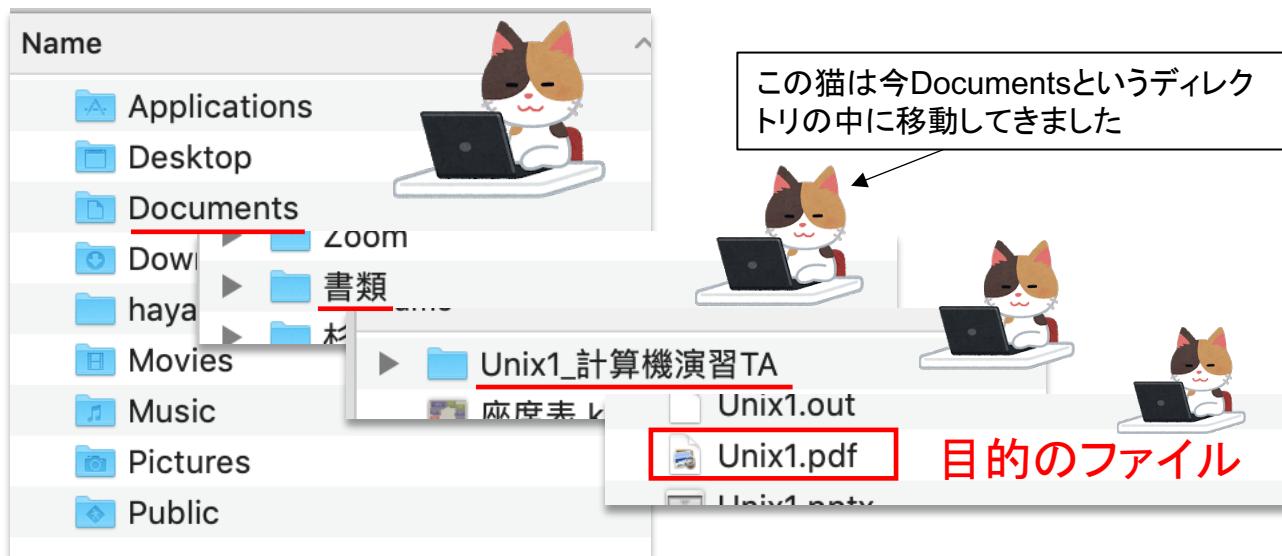
タブキーを使うことでコマンドやプログラムを打つ効率が一気に上がります！

ファイル操作 ー ディレクトリ/ファイルとは？



ファイル操作 一ディレクトリ/ファイルの場所

- ❖ GUIだとファイルの階層構造の中をクリックで移動して目的のファイルに辿り着くことができます



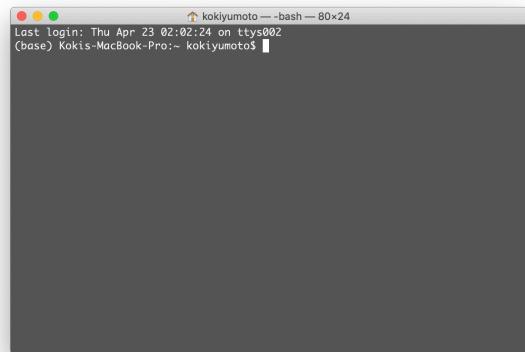
- ❖ CUIだと自分が今どのディレクトリの中にいるのかわからない...?



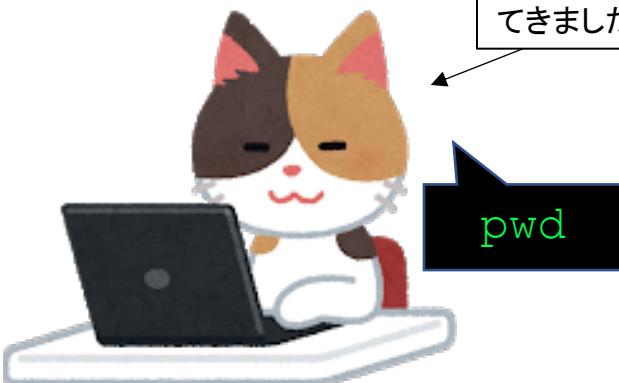
ファイル操作 一ディレクトリ/ファイルの場所

自分が今いるディレクトリを表示するために`pwd`というコマンドを入力してみましょう。

- ✓ ターミナルを新しく起動した時にあなたがいるディレクトリは特別なディレクトリで、「ホームディレクトリ」と呼ばれます。
- ✓ ホームディレクトリは「~」で略記されます。例えば自分のホームディレクトリが「/Users/fumiyasakai/」だと、「/Users/fumiyasakai/directoryA/fileB.txt」というパスは「~/directoryA/fileB.txt」と略記されます
- ✓ 実はコマンドラインの\$の前に自分が今いるディレクトリが表示されています。



```
kokiyumoto — bash — 80x24
Last login: Thu Apr 23 02:02:24 on ttys002
(base) Kokis-MacBook-Pro:~ kokiuyumoto$
```



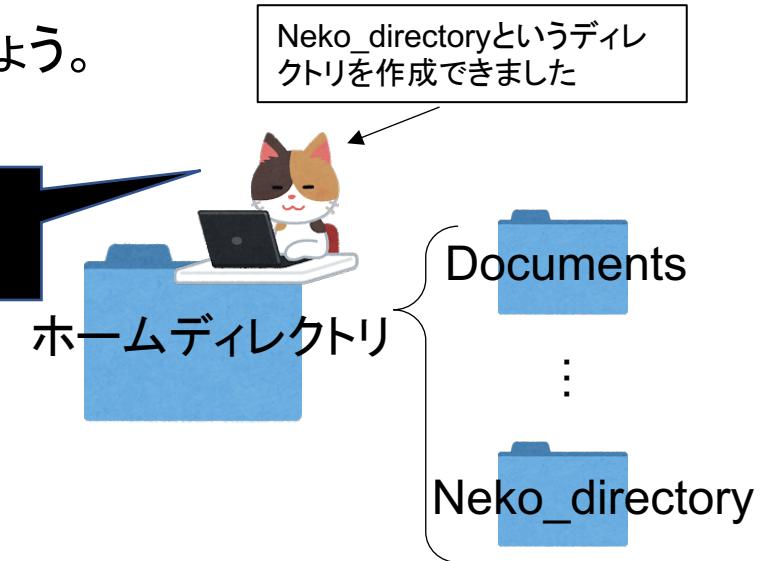
\$の左に~がついているので自分が
ホームディレクトリにいることがわかつ
てきました。

ファイル操作 一ディレクトリの作成

自分が今いるディレクトリの中に新しいディレクトリを作つてみましょう。

- 「`mkdir ディレクトリ名`」で作成することができます。

```
mkdir Neko_directory  
ls
```



正しく作成できたのでしょうか？

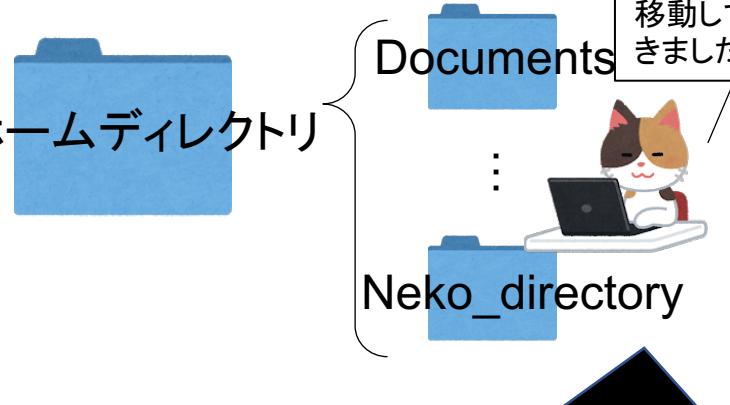
- 自分がいるディレクトリの中にはどんなファイル/ディレクトリがあるのか調べてみるため、「`ls`」というコマンドを入力してみましょう。
- 先ほど作ったディレクトリが表示されていれば無事ディレクトリが作成できました！
- この`ls`というコマンドは無限回使えます

ファイル操作 一ディレクトリの移動

相対パス
を使って
Neko_directoryに
移動して
きました

次に、今作ったディレクトリの中に移動してみましょう。

- 「cd 移動先のディレクトリの住所(パス)」で移動することができます。 ホームディレクトリ



このとき、移動先のディレクトリのパスの指定方法には**絶対パス**と
相対パスという方法があります。

- ✓ 絶対パス: 大元のディレクトリの名前から順次列挙してパスを記載する方法
例) /Users/fumiyasakai/Documents/DirectoryA/DirectoryB

cd ./Neko_directory

または、ホームディレクトリ(/Users/fumiyasakai/)を「~」で略記して、
~/Documents/DirectoryA/DirectoryB

- ✓ 相対パス: 自分が今いるディレクトリを「.」その親ディレクトリを「..」で略記してパスを記載する方法
例) もし自分が/Users/fumiyasakai/Documents/の中にいる場合、上記のディレクトリの相対
パスは

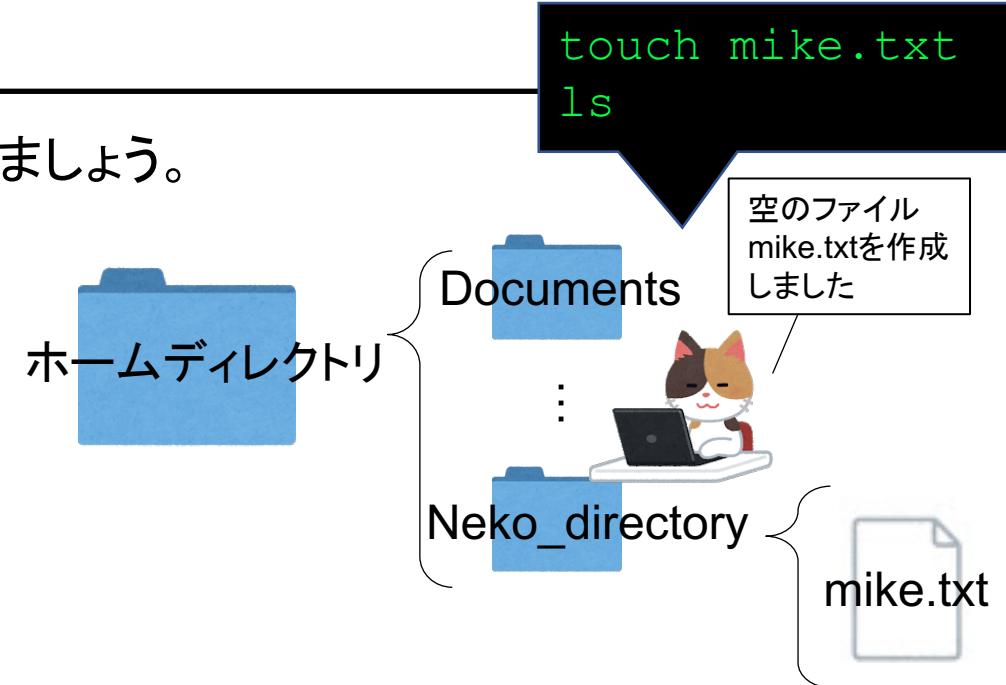
./DirectoryA/DirectoryB

もし自分が/Users/fumiyasakai/Documents/DirectoryA/の中にいる場合、
./DirectoryB
となり、自分が今いるディレクトリに応じて相対パスは変わります。

ファイル操作

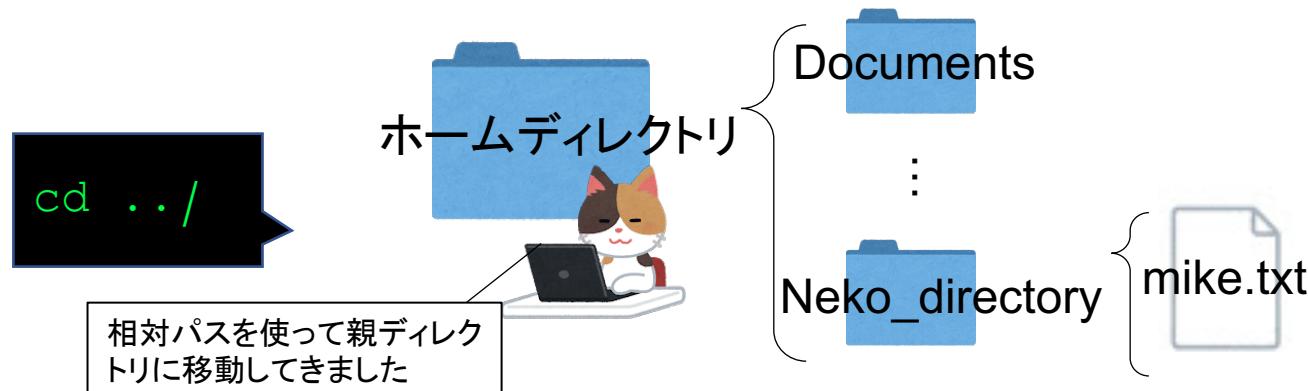
新しいディレクトリが作成できたらその中にファイルを作成してみましょう。

- ファイルの作成は「**touch ファイル名**」で行うことができます。



では、ホームディレクトリに帰ってきてください。

- 帰り方は絶対パスで「cd ~」、または相対パスで親ディレクトリに移動「cd ..」で帰ってこれます。



ファイル操作

rmで消したファイル/ディレクトリはゴミ箱には行かず、失われます。
特にrm -rやrm -fを使うときは気をつけてください！

せっかく作ったディレクトリですが、削除してみましょう。

- ファイルやディレクトリを削除するときは「rm」というコマンドを使います。
- 削除できませんか？ manコマンドでrmのヘルプを表示してみましょう。
正しくオプションを指定するとディレクトリの中身を再帰的に削除できます。

本当に削除できているか「ls」コマンドで確認してみましょう。

コマンドのまとめ

コマンド	意味
pwd	現在のパスを表示
mkdir <i>PATH</i>	<i>PATH</i> というディレクトリを作成
touch <i>PATH</i>	<i>PATH</i> というファイルを作成
cd <i>PATH</i>	<i>PATH</i> というディレクトリに移動
ls <i>PATH</i>	<i>PATH</i> というディレクトリの中身を表示
rm <i>PATH</i>	<i>PATH</i> というファイル/ディレクトリを削除



ファイル操作

pwdを実行した時に分かったと思いますが、ホームディレクトリにも更に親ディレクトリがあります。色々なディレクトリを探検してみてください。

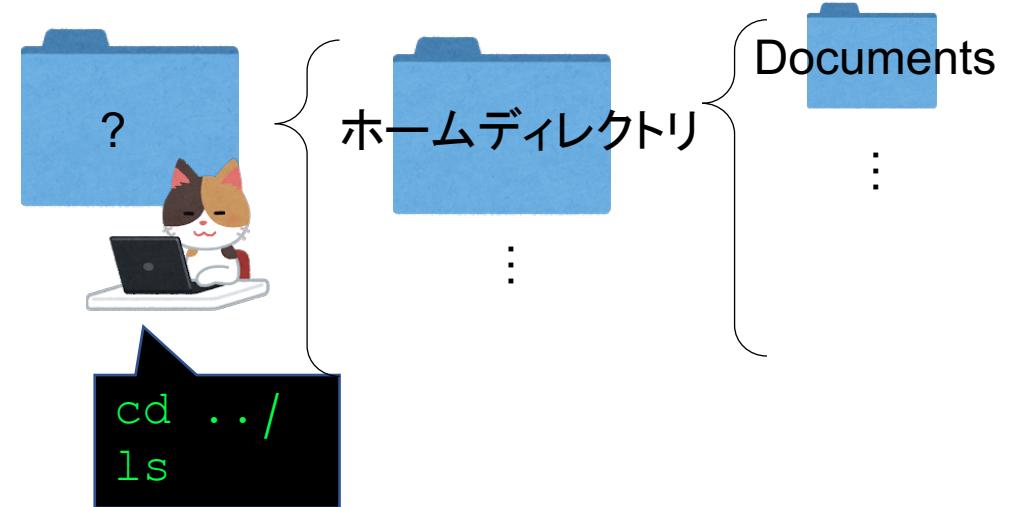
- 迷っても「cd ~」でいつでもホームディレクトリに帰ってこれます
- 「ls -a」で「.」から始まる隠しファイルを表示できます
- システム上大切なファイルを消してしまう危険があるのでrmコマンドの使用は控えてください

練習課題

まず、カレントディレクトリを確認してみましょう。

続いて cd を用いて様々なディレクトリに移動し、pwd でどこにいるのかを確認してみましょう。cd は別のディレクトリを移動するコマンド (Change Directory) で、"cd <ディレクトリ名>" のように使います (cd ./や cd ../とするとどこに移動するでしょう)。

色々移動して遊んだら、最後に絶対パスでディレクトリ名を指定して最初にいたディレクトリに帰ってきましょう。おかえりなさい。



ワイルドカード「*」、「?」

ファイル名に「？」や「*」を使うのは避けましょう。
どうしてもこのような特殊文字を単なる文字として扱いときはこの前にバックスラッシュをつけることでエスケープできます

「ls -a /etc」と打ってみましょう。
このディレクトリの中にはたくさんのファイルがあります。

- 中でも.confで終わるファイルのみlsで表示させたいときは、「ls -a /etc/*.conf」と入力してみましょう。「*」という記号が**任意の長さの任意の文字列**として認識されます。
- さらにこの中でも.confの前が三文字であるファイルのみ表示させたいときは、「ls -a /etc/???.conf」と入力してみましょう。「?」という記号が**任意の1文字**として認識されます。

このようにある文字、文字列を置換する記号「*」や「?」を**ワイルドカード**といいます。



パイプ

コマンドと出力は x と $f(x)$ の関係にあります。

しかし、**パイプ**を用いることで出力に対して更にコマンドを適用した出力 $g(f(x))$ をコマンド1行で得ることができます。

今度は、次のようにして `/etc/` ディレクトリの中身を表示させて下さい。

コマンド: f 引数:x

```
~$ ls -al /etc/
```

中身が多すぎて表示し切れません。このような時こそ、標準入出力の仕組みが役立つときです。次のようにすると、`ls -al` の標準出力を `less` の標準入力に送り込んで、少しずつ読むことができます。`less` はファイル内容を1ページずつ見るようなコマンドです。`less` から抜けるときは`q`を押します。

```
~$ ls -al /etc/ | less
```

コマンド: f 引数:x

f(x)

→ コマンド: g

リダイレクト

これまでコマンドの出力が画面に表示されるだけ(標準出力)でした。
得られた出力のあるファイルに書き込んで欲しい場合は、**リダイレクト(>または、>>)**を用います。

```
~$ ls
```

: 出力はターミナルの標準出力へ

```
~$ ls > ls.log
```

: 出力はls.logへ
*ファイルが既に存在している場合は**上書き**

```
~$ ls >> ls.log
```

: 出力はls.logへ
*ファイルが既に存在している場合は**後ろに
続けて追記**

よく使うコマンド

世の中には数々のコマンドがあります。

基本的には必要に応じてグーグルやmanコマンドで調べることで使い方を覚えていきます。

中でも私(先代TA)がこれまでによく使ってきたコマンドを以下にまとめておくのでよければ参考にしてください。

コマンド	意味
pwd	現在のパスを表示
mkdir	ディレクトリを作成
touch	ファイルを作成
cd	ディレクトリに移動
ls	ディレクトリの中身を表示
rm	ファイル/ディレクトリを削除
cp	ファイル/ディレクトリを別の場所へ複製
mv	ファイル/ディレクトリを別の場所へ移動
chmod	ファイル/ディレクトリのパーミッション変更
chown	ファイル/ディレクトリの所有者変更
cat	ファイルの内容を結合して出力に表示
find	ファイル名を検索
grep	ファイル中の文字列を検索

提出課題

課題 1. — man —

シンボリックリンク (symbolic link) とは何かを調べ、シンボリックリンクを作成するにはなんというコマンドを用いればよいのかと共に、簡単な説明を書いて下さい。

ヒント: `man -k` あるいは `apropos`

課題 2. — ファイルの操作 —

表にあるコマンドを使って次の操作を行い、どのようなコマンドを入力したかを記入して下さい。但し、日本語で説明を加える必要はありません。1行目にカレントディレクトリの名称を（絶対パスで）書き、2行目以降に `ls` や `cd ???/` 等、入力したコマンドを1行ずつ書いて下さい。

1. ホームディレクトリに移動する。
2. ディレクトリ `dir/` を作成する。
3. 存在するファイルの一覧を表示して作成した `dir/` が存在することを確かめる。
4. `dir/` に移る。
5. 隠しファイルも表示するオプションをつけてファイルの一覧を表示して「`.`」と「`..`」しかないことを確かめる。
6. `hoge.txt` という空ファイルを作成する。
7. 存在するファイルの一覧を表示して作成した `hoge.txt` があることを確認する。
8. 親ディレクトリに移る。
9. ディレクトリ `dir/` を削除する。失敗した場合はどうして失敗したか、どうすれば削除できるか考えてみる。
(ヒント: `rm` に適切なオプションをつける)



計算機演習TA

提出課題

課題 3. — パイプ —

`/etc` ディレクトリに `.conf` で終わる名前のファイルがいくつあるかを、コマンドライン一行で数えるにはどうしたらよいか考え、その説明を書いて下さい。そして、その方法で数えた結果を 2 行目に書いて下さい。

但し、"一行" というのはコマンドが "List" ではなく、"Simple Command" または "Pipeline" という意味です。

ヒント: `ls`, `wc`, パイプ、ワイルドカード