

Ubuntu講習

5s 野口史遠



Altairu



@Flying___eagle



Ubuntuとは？

ubuntu

- オープンソースのLinuxディストリビューション
- デスクトップ、サーバ、クラウドなど幅広い用途に対応
- **Canonical社**が開発、サポート

基本コマンド

- **ls:** ディレクトリの内容を表示

```
ls
```

- **cd:** ディレクトリを変更

```
cd /path/to/directory
```

基本コマンド

- **cp:** ファイルをコピーして貼り付け

```
cp /path/to/sourcefile /path/to/destination/
```

- 例: `cp /home/user/file.txt /home/user/backup/`
- 指定した `sourcefile` を `destination` ディレクトリにコピーします。

基本コマンド

- **mv**: ファイルを移動または名前を変更

```
mv oldname newname
```

- 例: `mv /home/user/file.txt /home/user/backup/file.txt`
- ファイルを指定した場所に移動するか、名前を変更します。

基本コマンド

- **rm:** ファイルを削除

```
rm filename
```

- **例:** `rm /home/user/file.txt`
- 指定したファイルを削除します。

基本コマンド

- **mkdir**: ディレクトリを作成

```
mkdir newdirectory
```

- **例**: `mkdir /home/user/newfolder`
- 新しいディレクトリを作成します。

基本コマンド

- **rmdir**: ディレクトリを削除

```
rmdir directoryname
```

- 例: `rmdir /home/user/oldfolder`
- 指定したディレクトリを削除します（空である必要があります）。

基本コマンド

- **rm -r**: ディレクトリとその内容を再帰的に削除

```
rm -r directoryname
```

- 例: `rm -r /home/user/oldfolder`
- 指定したディレクトリとその中の全てのファイルおよびサブディレクトリを削除します。

基本コマンド

- **touch**: 空のファイルを作成、または既存のファイルのタイムスタンプを更新

```
touch filename
```

- **例:** `touch /home/user/newfile.txt`
- 指定した名前の空のファイルを作成します。

基本コマンド

- **gedit**: テキストエディタを使用してファイルを編集

```
gedit filename
```

- 例: `gedit /home/user/newfile.txt`
- GNOMEデスクトップ環境のテキストエディタで指定したファイルを開きます。

基本コマンド

- USB確認

```
ls -l /dev/serial/by-id/
```

- USB承認

```
sudo chmod 777 /dev/ttyUSB0
```

シェルスクリプトの例

簡単なスクリプト

```
#!/bin/bash  
echo "Hello, World!"
```

- **#!/bin/bash**: スクリプトの先頭に記述し、シェルの種類を指定
- **echo**: メッセージを表示

Pythonスクリプトの実行

- Pythonスクリプトの拡張子は通常 **.py**
- 実行方法

```
python3 scriptname.py
```

例: Hello Worldプログラム

```
print("Hello, World!")
```

自作コマンドの作り方

- シェルスクリプトを作成して実行可能にすることで、自作コマンドを作成できます。

ステップ1: スクリプトを作成

- 任意のエディタでシェルスクリプトファイルを作成

```
nano mycommand.sh
```

ステップ2: スクリプトにコマンドを記述

- 例: `mycommand.sh` の内容

```
#!/bin/bash  
echo "This is my custom command!"
```

ステップ3: スクリプトを実行可能にする

- 実行権限を追加

```
chmod +x mycommand.sh
```


ステップ4: スクリプトをパスに追加

- `/usr/local/bin` などにコピー

```
sudo cp mycommand.sh /usr/local/bin/mycommand
```

ステップ5: 自作コマンドを実行

- どこからでも実行可能

```
mycommand
```

自作コマンド

[Ubuntuで自作コマンド?を作ろう](#)
Qiitaの記事

スクリプトファイルの作成

まずは、スクリプトファイルを作成。
altairって名前のコマンドにします。

```
nano ~/bin/altair
```

スクリプトの内容を編集

```
#!/bin/bash
# altairコマンド

function show_help {
    echo "用法: altair [オプション]"
    echo "オプション:"
    echo "  -h, --help      このヘルプメッセージを表示"
    echo "  txt             geditでサンプルのテキストファイルを開く"
    echo "  pdf [filename]  evinceでPDFファイルを開く"
    echo "  py [filename]   /home/user/Documents/python からPythonファイルを検索して実行する"
    echo "  mkpy [filename] /home/user/Documents/python にPythonファイルを作成してgeditで開く"
    echo "  oppy [filename] /home/user/Documents/python からPythonファイルを検索してgeditで開く"
}
```

スクリプトの内容を編集

```
function open_text {  
    gedit ~/sample.txt  
}  
  
function open_pdf {  
    if [ -z "$1" ]; then  
        echo "ファイル名を指定してや"  
    else  
        evince "$1"  
    fi  
}
```

スクリプトの内容を編集

```
function run_python {  
    if [ -z "$1" ]; then  
        echo "ファイル名を指定してや"  
    else  
        python3 "/home/user/Documents/python/$1"  
    fi  
}  
  
function make_python {  
    if [ -z "$1" ]; then  
        echo "ファイル名を指定してや"  
    else  
        touch "/home/user/Documents/python/$1"  
        gedit "/home/user/Documents/python/$1"  
    fi  
}
```

スクリプトの内容を編集

```
function open_python {  
    if [ -z "$1" ]; then  
        echo "ファイル名を指定してや"  
    else  
        gedit "/home/user/Documents/python/$1"  
    fi  
}
```

スクリプトの内容を編集

```
case "$1" in
  -h|--help)
    show_help
    ;;
  txt)
    open_text
    ;;
  pdf)
    open_pdf "$2"
    ;;
  py)
    run_python "$2"
    ;;
  mkpy)
    make_python "$2"
    ;;
  oppy)
    open_python "$2"
    ;;
  *)
    echo "不正なオプションやで"
    show_help
```


実行権限を付与

作成したスクリプトに実行権限を付与

```
chmod +x ~/bin/altair
```

PATHにディレクトリを追加

自作コマンドをどこからでも実行できるようにするために、スクリプトのディレクトリをPATHに追加。 `~/ .bashrc` に以下の行を追加。

```
echo 'export PATH=$PATH:~/bin' >> ~/ .bashrc  
source ~/ .bashrc
```

コマンドの実行

これで自作コマンド「altair」が完成。

ヘルプを表示

```
altair -h
```

テキストファイルを開く

```
altair txt
```

PDFファイルを開く

```
altair pdf filename.pdf
```

Pythonファイルを実行

```
altair py script.py
```

Pythonファイルを作成して開く

```
altair mkpy new_script.py
```

Pythonファイルを開く

```
altair oppy existing_script.py
```