

仮想マシン導入と Python実行環境の整備

- 仮想マシンの導入
- Pythonを実行できる環境整備

- 仮想マシンの導入
- Pythonを実行できる環境整備

そもそも仮想マシンとは？

物理的な環境に囚われず、仮想的に作られるマシン

複数のPCを1台のPC上で動かすことができる！

一人二役ならぬ一つ n 役

さあ導入しよう

今回使用した仮想化ソフト



動かしたいOS



The background features abstract, overlapping green geometric shapes, primarily triangles and polygons, in various shades of green, creating a modern and dynamic visual effect.

とりあえずDLしてきた

ISOファイル容量デカいんだよ...

VBOXで仮想マシンを作ろう！

新規作成ボタンを押していざ作成

名前とオペレーティングシステム

新しい仮想マシンの記述名を指定し、インストールするオペレーティングシステムのタイプを選択してください。入力した名前はVirtualBoxでこのマシンを特定するのに使われます。

名前(N): うぶんとううう

タイプ(T): Linux

バージョン(V): Ubuntu (64-bit)



タイプとバージョンは合わせよう

VBOXで仮想マシンを作ろう

メモリーの容量を決めよう

メモリーサイズ

この仮想マシンに割り当てるメモリー(RAM)の容量をメガバイト単位で選択してください。

必要なメモリーサイズは**1024MB**です。



※ご使用は計画的に

VBOXで仮想マシンを作ろう

あとはポチポチっと進んで...

ストレージの大きさまで決めたら

大卒終了！

VBOXで仮想マシンを作ろう


さあ起動しよう！

ここで事前に落としておいた
Ubuntuのisoファイルを選択

← 起動ハードディスクを選択

開始したい新しい仮想マシンを含むディスクのある、仮想光学ディスクファイルか、ディスクが挿入されている物理光学ドライブを選択してください。

このディスクはコンピューターを起動することができ、仮想マシンにインストールしたいオペレーティングシステムを含んでいなければなりません。このディスクは仮想マシンをオフにした次の回に自動的に取り出されますが、必要であればデバイスメニューから取り出すこともできます。

ubuntu-16.04.3-desktop-amd64.iso (1.4 GiB) 

起動

キャンセル

起動ポチー

VBOXで仮想マシンを作ろう

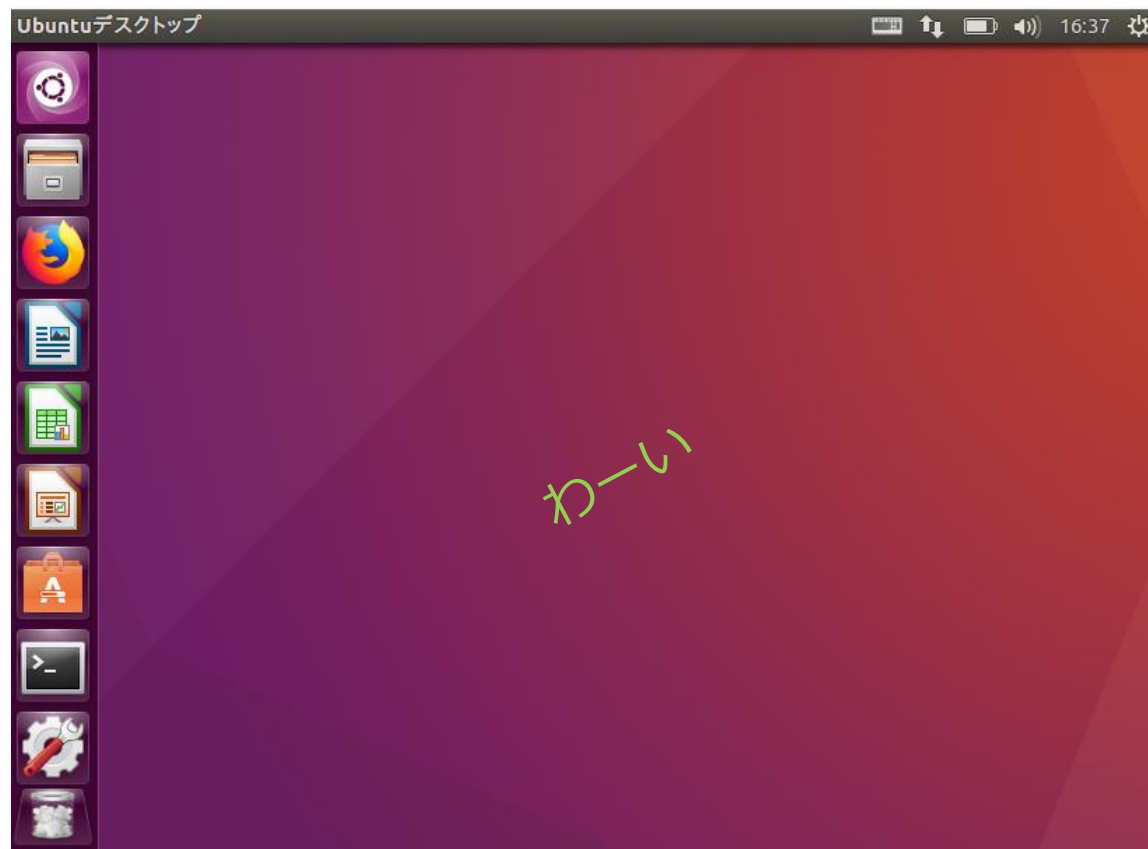
インストールが終わるまで・・・

待機！

待機行動違反は減点7点持っていられるから注意

VBOXで仮想マシンを作ろう

起動したら導入終了！！！！



余談

筆者は1度インスコに失敗しております

原因and改善点

使用できるメモリを増やす

チップセットをPIIX 3 からICH 9 に変更

- 仮想マシンの導入
- Pythonを実行できる環境整備

そもそもPythonとは？

少ないコード量で簡単にプログラムが書けちゃう
ナウい言語

主に

- ・ Webアプリケーション
- ・ 機械学習

に使われている

YoutubeやInstagramもPythonで作られていた...

余談



グイド・バアンロッサム

1956年1月31日（62歳）

オランダ国籍

Pythonは1991年に登場

Python作ったヨ

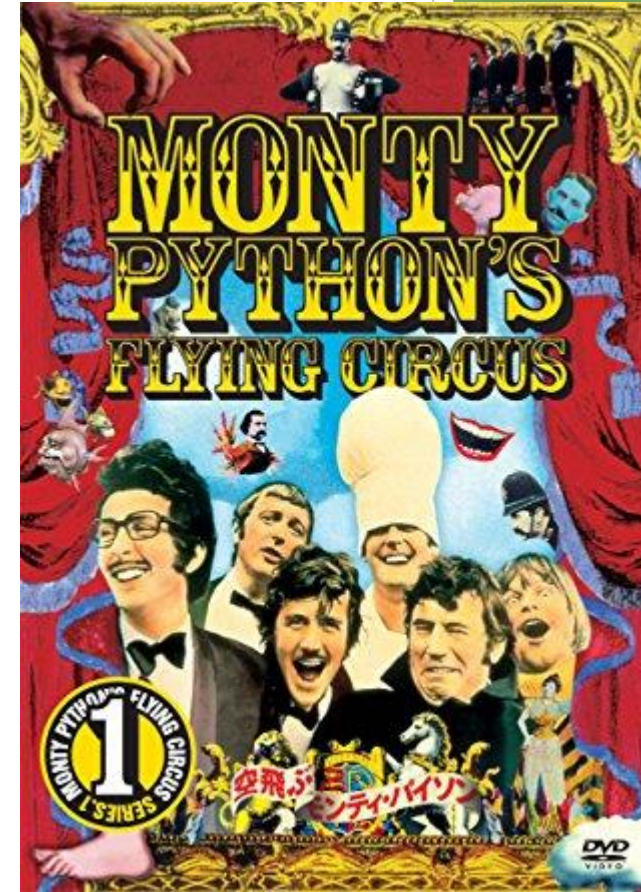
余談

Pythonの名前の由来

空飛ぶモンティ・パイソン

1969年から1974年までBBCで放映
された毒舌コメディ番組

製作者がファンだったらしい



あまりに過激な内容だったためBBCから幾度となく
検閲されたとかナントカ...

Pythonの環境整備！

実は...

Ubuntu16.04にはPython3.5/2.7の2つが
初期導入されているのだ

でも...ほかに何もなくて？
自分で入れるしかなくて？

そのために環境整備をしていきます

Pythonの環境整備！

基本的に必要なもの

- git
- anyenv
- pip

これらを導入していきます

Pythonの環境整備！

まずgitの導入

gitとは...

バージョン管理システム

前のバージョン等に容易に戻したりできる
システム

gitの導入手順

`sudo apt-get install git` コマンドポacher

Pythonの環境整備！

anyenvの導入

anyenvとは...

**env (pyenvとかrbenv...etc) を管理する
複数バージョンの実行環境を管理できる

pyenvの導入

anyenvからpyenvを導入

複数バージョンのpythonをこれで簡単に
切り替えることが可能

Pythonの環境整備！

anyenvの導入手順

```
git clone https://github.com/riywo/anyenv ~/.anyenv
```

上記コマンドでanyenvを入れる

pyenvの導入手順

```
anyenv install pyenv
```

ほかの**envも同じように入れることが可能

簡単にできちゃう優れもの

Pythonの環境整備！

pipの導入

pipとは...

The Python Package Index に公開されている
Pythonパッケージのインストールなどを行う
これで自分が使いたい機能を入れることが
できる

Pythonの環境整備！

pipの導入手順

```
sudo apt-get install python-pip
```

上記コマンドで完了

あとはpip install ~~でほしいパッケージ
を入れるだけ

k 本的なもの入れ終わった！

と、思っていました

Pythonの環境整備！

anyenvがない...(˘ω˘)

```
fuchigami@web-proj414:~$ an
anaconda          anacron           anytopnm
```

```
fuchigami@web-proj414:~$ la
.ICEauthority      .bashrc~          .java             .sudo_a
.PyCharmCE2017.2    .cache            .jupyter          .viminf
.PyCharmCE2017.3    .conda            .lessht           .wget-h
.Xauthority         .config           .local            .xinput
.anyenv             .dbus             .mozc             .xsessi
.tom               .emacs.d          mozilla           .xsessi
.bash_history       .gconf            .pki              .yjs
.bash_logout        .gnome            .presage          anacond
.bash_profile       .gnupg            .profile          bashrc
.bashrc             .ipython          .rbenv            example
```

あるやないかーい

なぜコマンドが出てこない🤔🤔🤔🤔

Pythonの環境整備！

原因？

ファイルあるのに読み込んでくね？

読み込ませるためにはどうするのか *Google先生～*

ここでいきなり登場するのが

.bashrc

Pythonの環境整備！

.bashrcとは

シェルスクリプトで書かれている

いわゆる設定ファイル

起動時に毎回読み込まれる

どうにかしてanyenvを読み込ませたい

Pythonの環境整備！

.bashrcに書き込む

.bashrcに読み込ませる文を書きます

```
export PATH="$HOME/.anyenv/bin:$PATH"
```

```
eval "$(anyenv init -)"
```

Pythonの環境整備！

```
export PATH="$HOME/.anyenv/bin:$PATH"
```

指定するPATH

指定したPATHを有効にする

Pythonの環境整備！

`eval "$(anyenv init -)"` — 標準入出力だよ（という意味らしい）

起動時に読み込みますよー

init系はevalを使う（らしい）

説明のいく文献を見つけきれませんでした...

Pythonの環境整備！

.bashrcに書き込む方法

- ・ 直接書き込む
- ・ echoコマンドを使用する

```
echo 'export PATH="$HOME/.anyenv/bin:$PATH"' >> ~/.bashrc
```


Pythonの環境整備！

.bashrcに書き込んだら

`source ~/.bashrc`

bashrcに変更を適用

`exec $SHELL -l`

シェルを再起動させる

```
fuchigami@web-proj414:~$ any
anyenv      anytopnm
```

✌('ω'✌)≡✌('ω')✌≡(✌'ω')✌

Pythonの環境整った！

Pythonの動作確認

念のためPythonを動かしてみる

emacaでhello.pyを作成

`print("hellow!!")`と書き保存

拡張子注意

python ファイル名で実行...

```
fuchigami@web-proj414:~/pystudy$ python hello.py  
hellow!!
```

実行されたやったね

最後に

仮想環境を整えてpython導入までの軌跡でした

まだまだ序章に過ぎないヨ

難しく考えずに...わからん事だらけは当たり前

つまりGoogle先生は偉大

おわり