

みんなのPython勉強会#41

1/16, 2019

Pythonの話をしよう

辻 真吾 (Start Python Club)

@tsjshg

shingo.tsuji@gmail.com



GALLERY9

2019.1.15 - 2019.1.21

辻修平（あさくら画廊 オーナー兼アーティスト）作品展『目的地はないんだ 帰り道も忘れたよ』

渋谷ロフト9ギャラリーにて開催中



自己紹介

- ❖ 1975年 東京都足立区出身
- ❖ コンピュータと数学を使ってなんでもやる工学部計数工学科の修士を出たあと、創業間もないITベンチャー「いい生活」に入社
- ❖ 幸いビジネスが軌道に乗った（と思った）ので3年弱で退社
- ❖ 生命科学が学びたくなって、大学にもどりそろそろ15年
- ❖ コンピュータ、数理工学、生命科学にまあまあ詳しい43才に仕上がりました
- ❖ LOGO, BASIC, C, C++, Java, SQL, R, Python

2000年 いい生活時代

- ❖ 不動産情報を含む生活全般の情報サイトを作り運営する
- ❖ Javaを使ったWebアプリ開発
 - ❖ 約3人ですべてを0から作った
- ❖ ハードウェア
 - ❖ Sun Enterprise 420R/220R?
 - ❖ CPU UltraSPARC-II 450MHz、メモリ256MB、HDD9.1GB
(UltraSCSI) でなんと265万円！

サン、『Sun Enterprise 220R』などプロバイダー向けサーバー製品を発売

1999年11月11日 00時00分更新

文●編集部 山本誠志

サン・マイクロシステムズ(株)は、ラックマウント型サーバー『Sun Enterprise 220R』および『Sun Enterprise 420R』、『Netra t 1400』、『Netra t 1405』、ディスクストレージ製品『Netra st D130』を発表した。いずれの製品も、19インチラックへの設置に対応し、省スペース化を図った設計となっている。

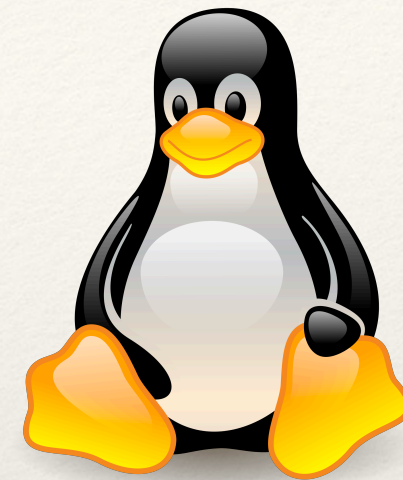


発表会場に陳列された、サンのサーバー製品群。手前の3台が、今回発表された新製品

Sun Enterprise 220RおよびSun Enterprise 420Rは、企業を対象としたサーバー製品で、両機種ともCPUにUltraSPARC-II-450MHzを採用。220RはCPUを最

ソフトウェア

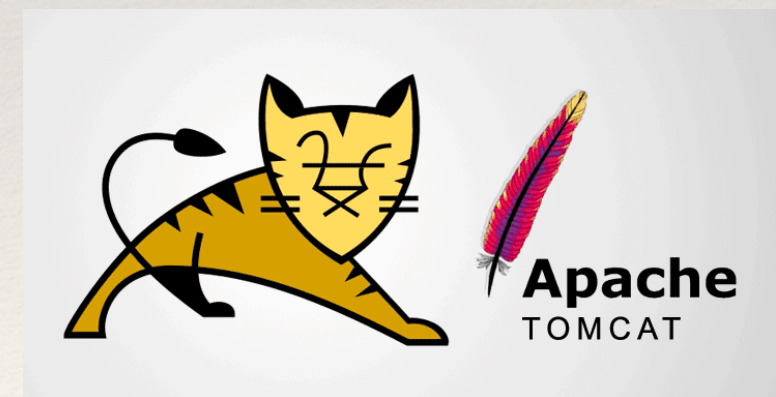
Sun Solaris



Oracle8i



BEA WebLogic



ハードウェア（2台）とソフトウェアのライセンス料で当時2千万を超えていた記憶が・・・

Pythonの話

- ❖ Pythonの基礎知識
- ❖ ありがとうPython2
- ❖ Pythonのインストールと環境設定
- ❖ Guido van Rossumの引退とPEP572

Pythonの基礎知識

- ❖ オランダ出身のGuido van Rossum氏（Benevolent Dictator for Life (BDFL) 慈悲深き終身独裁者）によって始まった
 - ❖ 1990年登場（80年代に着想、89年12月から実装開始）
- ❖ Python2系とPython3系がある
- ❖ PEP（Python Enhancement Proposal）
 - ❖ PEP8はコーディングのスタイルガイド
 - ❖ PEP20はZen of Python（import thisで出てくる）



Pythonのインストール



www.python.orgからダウンロード (3.7)

パッケージ管理はpipコマンド、バージョン管理はvenv

Windows Storeからもインストールできる



米Anaconda社が配布する人気のDistribution

最新はversion2018.12 (Python3.7)

パッケージ管理とバージョン管理が両方できる独自の

condaコマンドが付属

OSごとの詳しい環境構築ガイド <https://www.python.jp/install/install.html>

Dockerを使うのも便利

データサイエンスだと、jupyter/datascience-notebookのイメージがおすすめです

Python 2.7 will retire in...

0

Years

11

Months

21

Days

4

Hours

58

Minutes

48

Seconds

Enable Guido Mode [Huh?](#)

What's all this, then?

Python 2.7 [will not be maintained past 2020](#). Originally, there was no official date. Recently, that date has been updated to [January 1, 2020](#). This clock has been updated accordingly. My original idea was to throw a Python 2 Celebration of Life party at PyCon 2020, to celebrate everything Python 2 did for us. That idea still stands. (If this sounds interesting to you, email pythonclockorg@gmail.com).

Python 2, thank you for your years of faithful service.

Python 3, your time is now.

How do I get started?

If the code you care about is still on Python 2, that's totally understandable. Most of PyPI's popular packages now [work on Python 2 and 3](#), and more are being added every day. Additionally, a number of critical Python projects have [pledged to stop supporting Python 2 soon](#). To ease the transition, the [official porting guide](#) has advice for running Python 2 code in Python 3.

- 2020年1月1日以降はセキュリティパッチも出ない
- もちろん、オープンソースなのでforkして勝手にメンテしてもいいよ

<https://pythonclock.org/>

Pythonのバージョン

- PEP394 The “python” command on Unix-like system
 - python2はpython2.xを、python3はpython3.xを指す
 - pythonとだけ打つならpython2
 - スクリプトのshebang行（#!からはじまる1行目）は、両対応の時にpython
- macOSではpythonコマンドで同梱されたpython2が起動する
 - 標準Pythonをインストールすると、python3コマンドが使えるようになる
 - Anacondaをインストールすると、pythonコマンドのパスが上書きされて、python3が起動する
- Windowsではpythonでpython3が起動するので、今後の動向に注目したい

豊富な外部ライブラリ

164,897 projects

1,186,412 releases

1,664,793 files

294,055 users



The Python Package Index (PyPI) is a repository of software for the Python programming language.

PyPI helps you find and install software developed and shared by the Python community. [Learn about installing packages.](#)

Package authors use PyPI to distribute their software. [Learn how to package your Python code for PyPI.](#)

Mozillaのサポートでpypi.python.orgからpypi.orgになりました

2019はFacebookとOpen Technology Fundの支援を受ける模様



The **Open Technology Fund (OTF)** is a [U.S. Government](#) funded program created in 2012 at [Radio Free Asia](#) to support global [Internet freedom](#) technologies. (Wikipediaより)

パッケージ管理

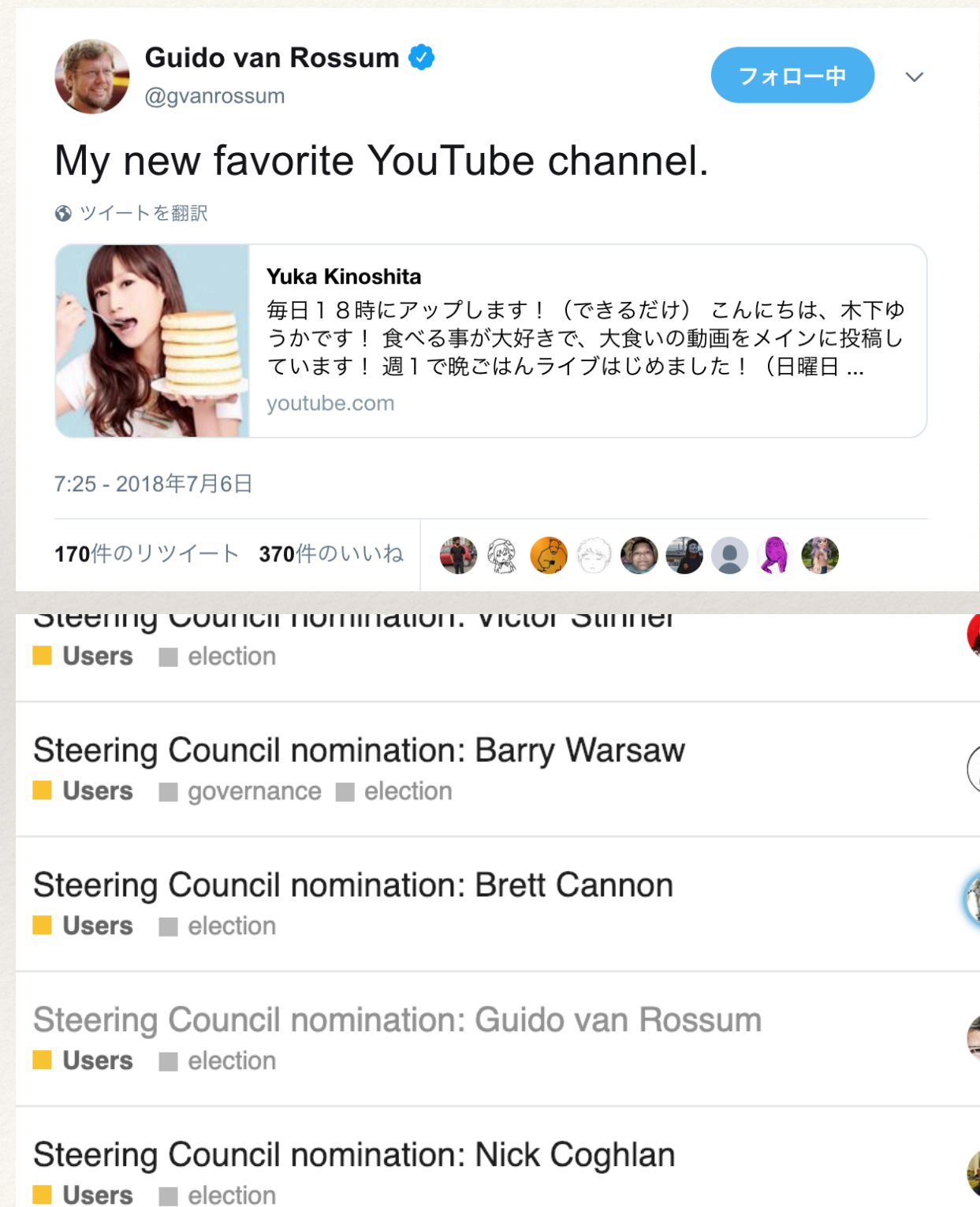
- ❖ `pip` (`pip3`) コマンドでインストール
- ❖ 1つのバージョン内で、仮想環境を作ってインストールする
パッケージのバージョンなどを変えられる
 - ❖ `python -m venv [仮想環境の名前]`
 - ❖ `pyvenv` スクリプトの利用は3.6で非推奨
- ❖ 外部パッケージの`virtualenv`の一部機能が3.3から`venv`として標準ライブラリに組み込まれた
- ❖ `pyenv-virtualenv`は、Pythonのバージョン管理プログラム`pyenv`のプラグイン

conda と pip

- ❖ Anacondaに付属するcondaコマンド
 - ❖ pipとvenvの機能を併せ持つ
 - ❖ `conda create -n [仮想環境名] python=2.7 anaconda`
- ❖ 併用は可能だが、やめたほうがいい
- ❖ Anacondaを入れて、venvを使って仮想環境を作り、そこはpipでパッケージ管理する世界にするのがいいか？

2018年のPythonニュース

- ❖ 7月Guido van Rossum氏BDFLを引退
- ❖ PEP572を巡る議論に疲れたというのが理由らしい
- ❖ 11月5人の委員会による運営方針を決定 (PEP8016)
- ❖ Guido先生も立候補
- ❖ 投票は来週21日から2/4まで



The image shows a screenshot of a tweet from Guido van Rossum (@gvanrossum) and a list of Steering Council nominations. The tweet, dated July 6, 2018, at 7:25, says "My new favorite YouTube channel." and features a video by Yuka Kinoshita. Below the tweet, there is a list of Steering Council nominations with filters for "Users" and "election".

Guido van Rossum @gvanrossum
フォロー中

My new favorite YouTube channel.

ツイートを翻訳

Yuka Kinoshita
毎日18時にアップします！(できるだけ) こんにちは、木下ゆうかです！ 食べる事が大好きで、大食いの動画をメインに投稿しています！ 週1で晩ごはんライブはじめました！（日曜日 ...
youtube.com

7:25 - 2018年7月6日

170件のリツイート 370件のいいね

Steering Council nomination: Victor Stinner
Users election

Steering Council nomination: Barry Warsaw
Users governance election

Steering Council nomination: Brett Cannon
Users election

Steering Council nomination: Guido van Rossum
Users election

Steering Council nomination: Nick Coghlan
Users election

PEP572とは？

❖ assignment expression（代入式）の追加

例えばC言語の場合

```
if (x==0){  
    何かの処理  
}
```

xが0なら「何かの処理」
が実行される」

```
if (x=0){  
    実行されないけどね  
}
```

xが何であれ0が代入される。
さらに、この式が評価され0となるので
if文の中身は実行されない。

```
>>> x = 0  
>>> if x=0:  
File "<stdin>", line 1  
    if x=0:  
      ^  
SyntaxError: invalid syntax  
>>>
```

Pythonでは文法的に許されていない。



Victor Stinner

@VictorStinner

フォローする



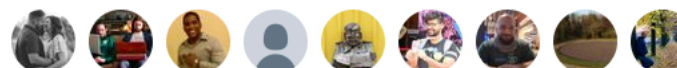
PEP 572: I wrote a pull request to show how assignment expressions could be use in the Python standard library github.com/python/cpython ... I don't propose to merge this PR. It's just to open a discussion on where they are appropriate or not.

🌐 ツイートを翻訳

448 - while True:	446 + while (line := fp.readline()):
449 - line = fp.readline()	447 + if (m := define_rx.match(line)):
450 - if not line:	
451 - break	
452 - m = define_rx.match(line)	
453 - if m:	
454 - n, v = m.group(1, 2)	448 n, v = m.group(1, 2)
455 - try:	449 try:
456 - v = int(v)	450 v = int(v)
457 - except ValueError:	451 except ValueError:
458 - pass	452 pass
459 - vars[n] = v	453 vars[n] = v
460 - else:	454 + elif (m := undef_rx.match(line)):
461 - m = undef_rx.match(line)	455 + vars[m.group(1)] = 0
462 - if m:	
463 - vars[m.group(1)] = 0	
464 return vars	456 return vars

6:45 - 2018年7月6日

63件のリツイート 180件のいいね



💬 15

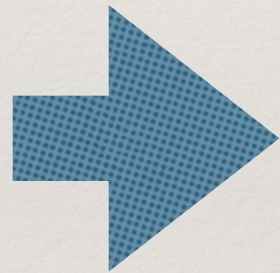
↺ 63

❤️ 180



<https://twitter.com/VictorStinner/status/1014988580282912770>

Pythonの哲学に反する？



```
[In [8]: import this]
The Zen of Python, by Tim Peters

Beautiful is better than ugly.
Explicit is better than implicit.
Simple is better than complex.
Complex is better than complicated.
Flat is better than nested.
Sparse is better than dense.
Readability counts.
Special cases aren't special enough to break the rules.
Although practicality beats purity.
Errors should never pass silently.
Unless explicitly silenced.
In the face of ambiguity, refuse the temptation to guess.
There should be one-- and preferably only one --obvious way to do it.
Although that way may not be obvious at first unless you're Dutch.
Now is better than never.
Although never is often better than *right* now.
If the implementation is hard to explain, it's a bad idea.
If the implementation is easy to explain, it may be a good idea.
Namespaces are one honking great idea -- let's do more of those!
```

ある目的を達成するためのコードは、できれば1つの書き方になることが望ましい。

`y := func(x)` のような普通の代入での利用は禁止されるので、なんとかセーフな気がする。

まとめ

- ❖ Python2系は2019年いっぱいで終了
- ❖ インストール：標準のPython、Anaconda、Dockerもおすすめ
- ❖ 仮想環境
 - ❖ 標準モジュールのvenv
 - ❖ Anacondaのconda
- ❖ 代入式は反対意見も多いけど便利そう
 - ❖ 3.8のリリースは今年10月の予定みたいです