

みんなのPython勉強会 in 長野#3

3/23, 2019

Pythonを知ろう！

辻 真吾 (Start Python Club)

@tsjshg

shingo.tsuji@gmail.com

自己紹介

- ❖ 1975年 東京都足立区出身 www.tsjshg.info
- ❖ LOGO, BASIC, C, C++, Java, SQL, R, Python



Udemy



Categories

Search for anything



Udemy for Business

Become an instructor



Log In

Sign Up

Development > Programming Languages > Python



Gift This Course



Wishlist

【世界で5万人が受講】実践 Python データサイエンス

データ解析の基本、可視化、統計、機械学習などデータサイエンスに関するあらゆる実践的なスキルがPythonで身に付く！

★★★★★ 4.0 (3,983 ratings) 23,912 students enrolled

Created by Shingo Tsuji, Pierian Data International by Jose Portilla Last updated 3/2016

🗨 Japanese 🗨 Japanese [Auto-generated]



Preview this course

¥1,600 ~~¥12,600~~ 87% off

🕒 6 days left at this price!

Add to cart

Buy now

30-Day Money-Back Guarantee

Includes

- 📺 17.5 hours on-demand video
- 📄 2 articles
- 🌐 Full lifetime access
- 📱 Access on mobile and TV
- 🏆 Certificate of Completion

What you'll learn

- ✓ Pythonを使った基本的なプログラミング
- ✓ IPython notebookを使ったPythonプログラミング
- ✓ numpyを使ったベクトルや行列の操作
- ✓ pandasを使ったデータ処理
- ✓ JSON、HTML、Excelシートなどのデータ形式をPythonで扱う
- ✓ matplotlib、seabornを使ったデータの可視化
- ✓ 応用範囲の広いデータ解析のスキル

今日のもくじ

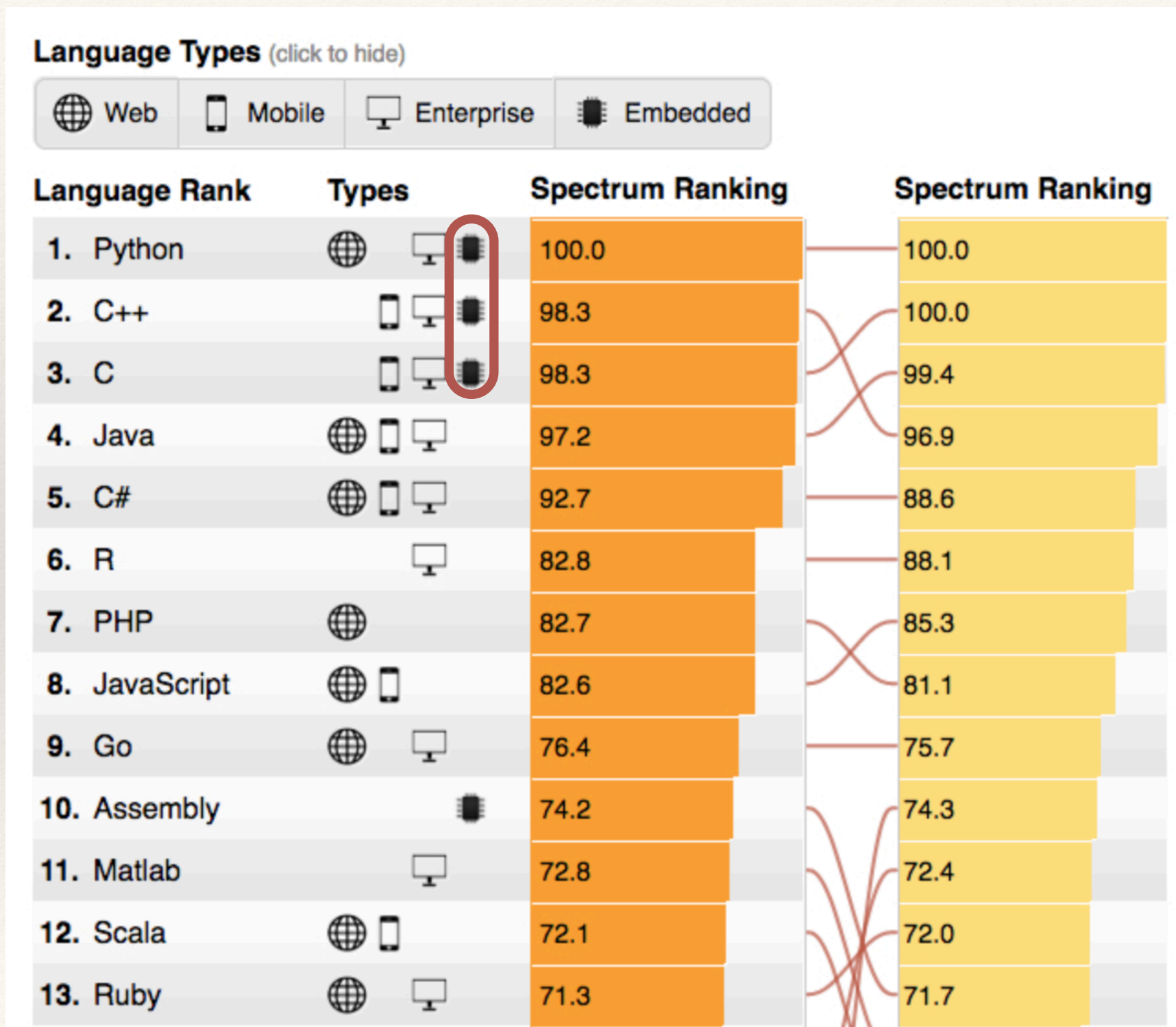
- ❖ Pythonの最新動向
- ❖ インストールと環境構築
- ❖ Guido van Rossumの引退とPEP572
- ❖ Pythonはオープンソース

“There are only two kinds of languages: the ones people complain about and the ones nobody uses.”

–Bjarne Stroustrup (creator of the C++ programming language.)

完璧なプログラミング言語は存在しないよ。誰も使っていない言語は山ほどあるし、広く使われている言語は常になにかしら文句を言われているでしょ。

IEEE Spectrum



Python と組み込み



IoT (Internet of Things) の発展もあり、小さなコンピュータの需要が増している。
Cとの相性がよいPythonが組み込み系で使われるようになってきている。



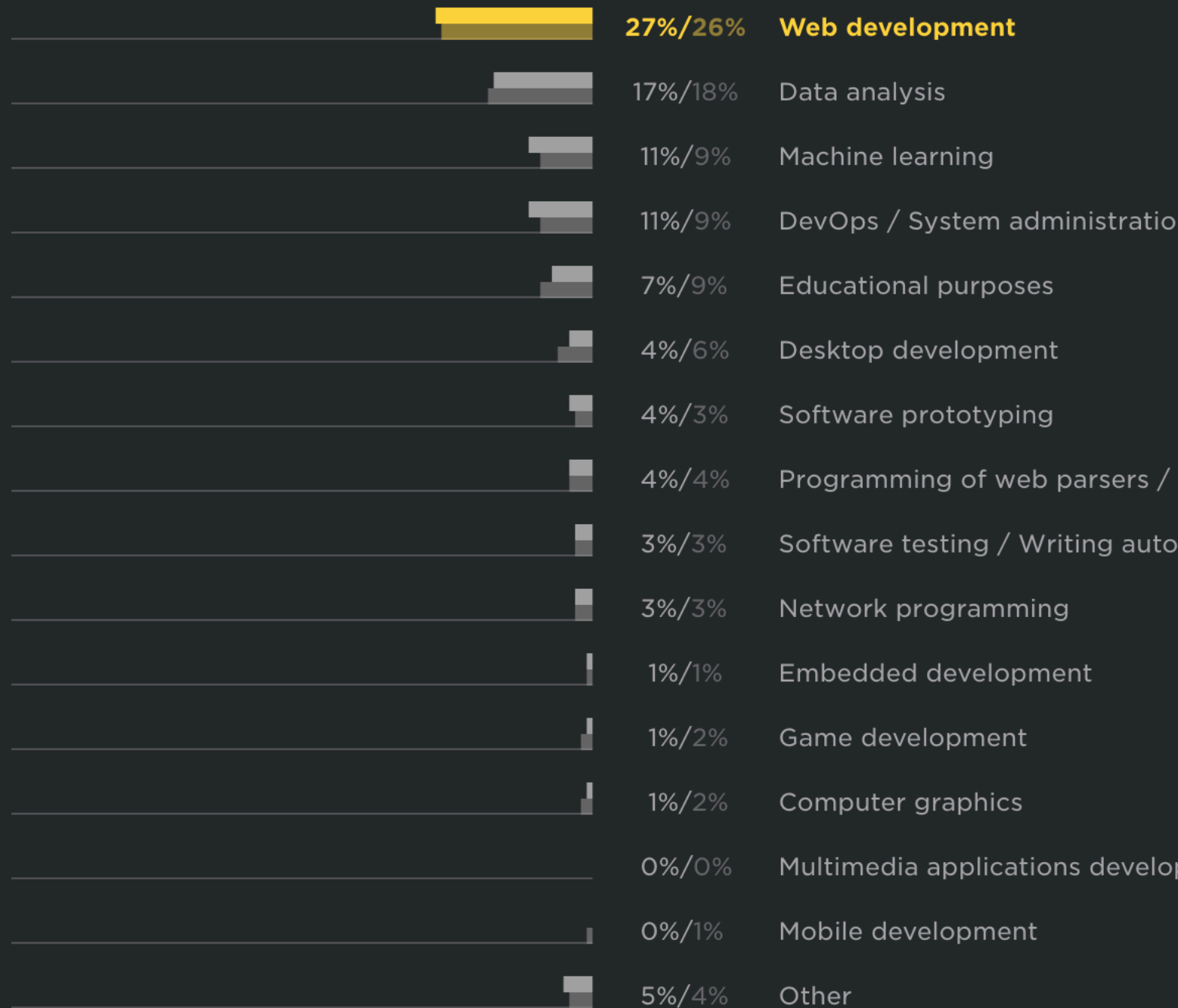
Python Developers Survey 2018 Results

In the fall of 2018, the Python Software Foundation with JetBrains conducted the official annual Python Developers Survey for the second time. Much like the previous survey, we set out to identify the latest trends and gather insight into how the world of Python developers looks in 2018. Over twenty thousand developers from more than 150 different countries participated this year to help us map out an accurate, up-to-date landscape of the Python community.

What do you use Python for the most? (single answer)

- 2018
- 2017

Combined Python is main Python is secondary





買換えに最適タイミング！レッツノート史上最速。2019年春モデル「SV8」レビュー

■ エデュケーション@プログラミング+ — 第13回

プログラミングを始めたい人、必読！

Python入門書22冊を読み比べてみた

2018年06月14日 09時00分更新

文● 坂川慎二 / ASCII

B! 107

シェア 1,087

ツイート

一覧

お気に入り

本文印刷



CPU大賞

配信元：@PRESS

🕒 2019.03.04

※ 配信会社から提供された企
たします。

CPU大賞 書籍部門：

全国主要書店173店舗のコンピュータ書籍ご担当者様に投票いただき、評価された新刊書籍です。

(※2018年1月1日から12月31日に発売された加盟各社の書籍を対象とし、全国の書店様からの投票を元に選定
しています)

【書店員さんが選んだ第1位】 _____

画像1: https://www.atpress.ne.jp/releases/178395/img_178395_1.jpg

『ドラゴンクエストXを支える技術—大規模オンラインRPGの舞台裏』

(ISBN：978-4-297-10174-9)

発売：株式会社技術評論社

著者：青山 公士

【書籍部門】受賞書籍に対する書店担当者さまの声

- ・大ヒットゲームを題材とすることで、一般の読者も手に取りやすかった印象。書店としても展開しやすかった。
- ・ゲーム制作者のみならず、プレイヤーにも訴求する内容で幅広い層に支持された本。技術者のみならず、普段コンピュータ書を買わないようなドラクエファンも多数購入してくれた。
- ・企画力、切り口の面白さ。PC書の可能性を広げるきっかけとなるであろう1冊。

【書店員さんが選んだ第2位】 _____


『スラスラ読める Pythonふりがなプログラミング』

(ISBN 978-4-295-00386-1)

発売：株式会社インプレス

著者：リブワークス／株式会社ビープラウド 監修

基本情報技術者試験

 **IPA (情報処理推進機構)** @IPAjp フォロー中

IPAは「基本情報技術者試験」の出題の見直しについて情報を公開しました。午後試験で出題するプログラム言語については「COBOL」を廃止し、「Python」の出題を追加予定です。適用時期などそのほかの詳しい内容はこちら→[ipa.go.jp/about/press/20...](https://ipa.go.jp/about/press/20190124)

11:59 - 2019年1月24日

7,222件のリツイート 5,102件のいいね

43 7,222 5,102

鈴木たかのりさんの紹介で、3月から試験委員をすることになりました。

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 2.7 will retire in...

0 Years 9 Months 9 Days 9 Hours 51 Minutes 2 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

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、pipでだけパッケージ管理する世界を作るのがいいかも

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
- ❖ 結果は後ほど

The image shows a screenshot of a tweet from Guido van Rossum (@gvanrossum) dated July 6, 2018. The tweet reads: "My new favorite YouTube channel." and includes a link to a YouTube channel by Yuka Kinoshita. Below the tweet is a list of Steering Council nominations, each with a bar chart showing the number of users and election votes.

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

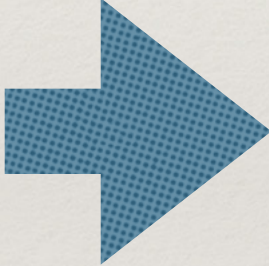
63

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。



Python Software

@ThePSF

フォロー中



The Python Core Developers have elected their first Steering Council! [@pumpichank](#), [@brettsky](#), [@WillingCarol](#), [@gvanrossum](#), and [@ncoghlan_dev](#) will lead the project under the specifications in PEP 13.
python.org/dev/peps/pep-0...

[mail.python.org/pipermail/pyth ...](https://mail.python.org/pipermail/pyth...)

ツイートを翻訳



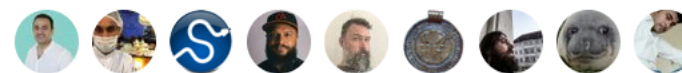
PEP 13 -- Python Language Governance

The official home of the Python Programming Language

python.org

0:59 - 2019年2月5日

154件のリツイート 340件のいいね



8

154

340



PEP 13 -- Python Language Governance

PEP:	13
Title:	Python Language Governance
Author:	The Python core team and community
Status:	Active
Type:	Informational
Created:	2018-12-16

Current steering council

The current steering council consists of:

- Barry Warsaw
- Brett Cannon
- Carol Willing
- Guido van Rossum
- Nick Coghlan

Per the results of the vote tracked in [PEP 8100](#).

python / steering-council

Watch 29 Star 19 Fork 2

Code Issues 0 Pull requests 0 Projects 0 Wiki Insights

Communications from the Steering Council

python cpython governance

12 commits 1 branch 0 releases 2 contributors

Branch: master New pull request

Create new file Upload files Find File Clone or download

brettcannon Use the right person's name	Latest commit 65140b5 17 days ago
process	edits to best practices 29 days ago
updates	Use the right person's name 17 days ago
.gitignore	Initial commit a month ago
CONTRIBUTING.md	initial documents a month ago
README.md	add update from 2-26 meeting 21 days ago

<https://github.com/python/steering-council> で継続的な情報公開がされている

二分探索アルゴリズム

```
1 import random
2 my_array = [random.randint(0,99) for i in range(20)]
3 my_array.sort()
4 my_array
```

```
[0, 1, 4, 5, 5, 16, 22, 24, 38, 38, 39, 43, 49, 56, 59, 61, 61, 66, 69, 81]
```

```
1 import bisect
2 bisect.bisect(my_array, 3)
```

```
2
```

`bisect.bisect_right(a, x, lo=0, hi=len(a))`

`bisect.bisect(a, x, lo=0, hi=len(a))`

`bisect_left()` と似ていますが、`a` に含まれ
点を返します。

返された挿入点 i は、配列 `a` を二つに分け
`a[i:hi]` が右側になるようにします。

```
def bisect_right(a, x, lo=0, hi=None):
    """Return the index where to insert item x in list a, assuming a is sorted.
```

The return value i is such that all e in `a[:i]` have $e \leq x$, and all e in `a[i:]` have $e > x$. So if x already appears in the list, `a.insert(x)` will insert just after the rightmost x already there.

Optional args `lo` (default 0) and `hi` (default `len(a)`) bound the slice of `a` to be searched.

```
"""
if lo < 0:
    raise ValueError('lo must be non-negative')
if hi is None:
    hi = len(a)
while lo < hi:
    mid = (lo+hi)//2
    if x < a[mid]: hi = mid
    else: lo = mid+1
return lo
```

まとめ

- ❖ Python2系は2019年いっぱいまで終了
- ❖ 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
- ❖ Pythonのソースでアルゴリズムの勉強ができる（現在本を執筆中）

ご静聴ありがとうございました