

みんなのはんなりPython勉強会

12/6, 2019

---

# Pythonの基本と 最近の動向

---

辻 真吾 (@tsjshg)

Start Python Club



# おまえ、誰よ？

- ❖ 辻 真吾 (つじ しんご)
- ❖ 東京大学先端科学技術研究センターに所属
  - ❖ バイオインフォマティクスから、応用データサイエンスへ
- ❖ 2010年に「Pythonスタートブック」を出版してからいろいろなことが
- ❖ 10月 (黄色) と11月 (緑色) に本を出しました



Python3スキルアップ教科書  
(技術評論社)

Pythonで学ぶアルゴリズムと  
データ構造  
(講談社サイエンティフィク)





# 医療業界がデータ解析できる人を求めています

- ❖ 2020年3月27～29日に開催される国際的なDatathon
- ❖ 東京医科歯科大学（東京・お茶の水）で開催される
- ❖ PythonやRを使って実際の医療現場で得られるデータ解析を体験できる
- ❖ 私もお手伝いする予定

**3rd**  
**Big Data Machine Learning in Healthcare**  
**in Japan@TMDU**

TMDU-MIT-NUS-ANZICS-JSICM Critical Data Workshops and Datathon 2020

- HOME
- WELCOME MESSAGE
- EVENT PROFILE
- PROGRAM
- Steering Committee  
Speakers & Facilitators
- REGISTRATION
- HOTEL  
ACCOMMODATIONS
- SPONSORSHIP
- Photo Album

March 27-29, 2020 / Tokyo Medical and Dental University (TMDU)  
Co-organized by : MIT Critical Data / NUS / ANZICS / JSICM / Datathon-Japan  
Chair: Hidenobu Shigemitsu  
Institute of Global Affairs / Department of Intensive Care Medicine, TMDU























Q: お仕事または趣味でコードを書いている？

Q: Pythonを使ってプログラミングしている？



# IEEE Spectrum

## Language Ranking: IEEE Spectrum

Rank	Language	Type	Score
1	Python	  	100.0
2	Java	  	96.3
3	C	  	94.4
4	C++	  	87.5
5	R		81.5
6	JavaScript		79.4
7	C#	   	74.5
8	Matlab		70.6
9	Swift	 	69.1
10	Go	 	68.0



# PYPL PopularitY of Programming Language

Worldwide, Oct 2019 compared to a year ago:

Rank	Change	Language	Share	Trend
1		Python	29.49 %	+4.5 %
2		Java	19.57 %	-2.4 %
3		Javascript	8.4 %	+0.1 %
4		C#	7.35 %	-0.4 %
5		PHP	6.34 %	-1.2 %
6		C/C++	5.87 %	-0.4 %
7		R	3.82 %	-0.2 %
8		Objective-C	2.6 %	-0.7 %
9		Swift	2.57 %	-0.1 %
10		Matlab	1.87 %	-0.2 %



# 1位じゃないのもあります

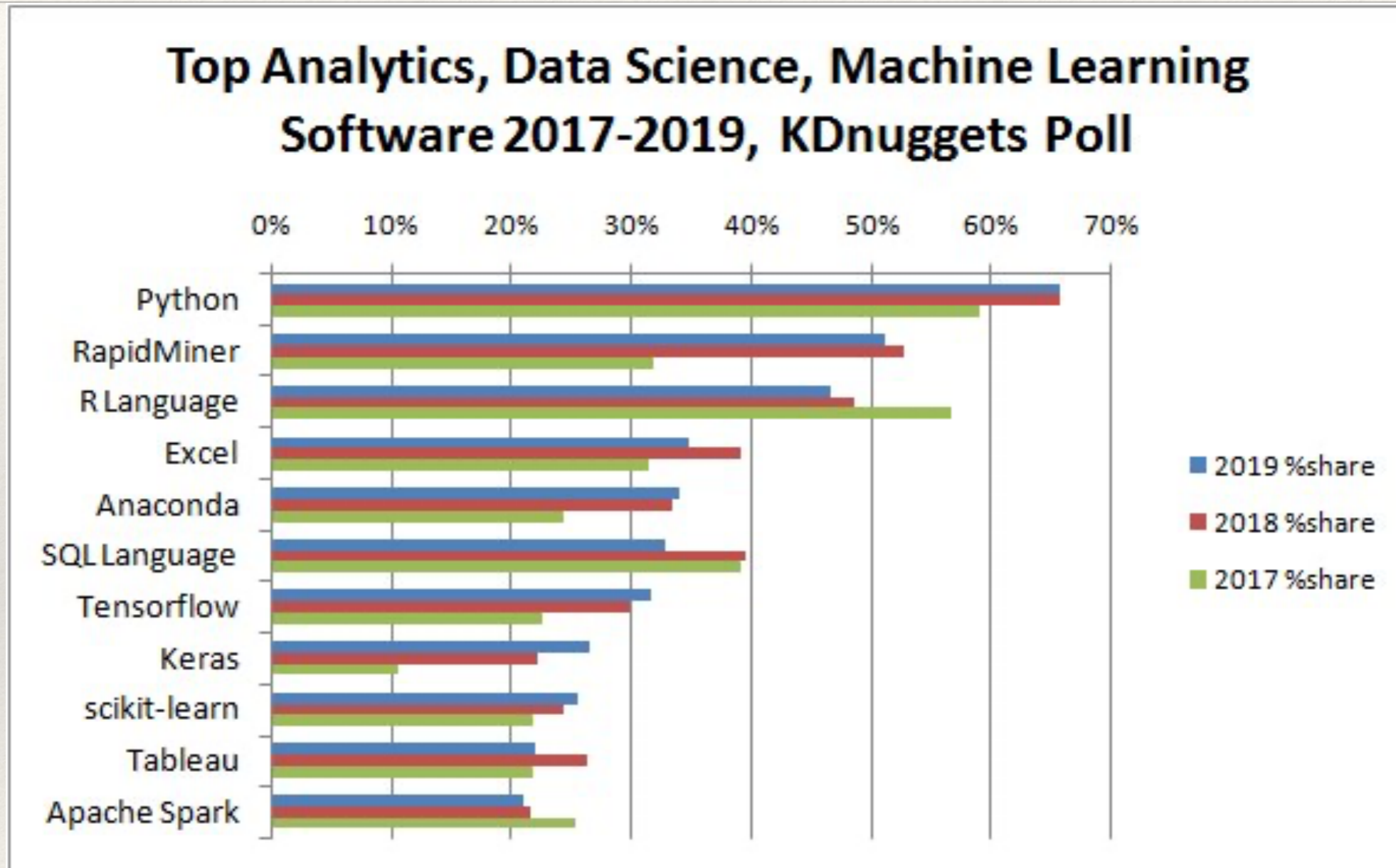
## TIOBE Index for November 2019

November Headline: C getting close to Java, Swift enters top 10 and Rust scores all time high

Nov 2019	Nov 2018	Change	Programming Language	Ratings	Change
1	1		Java	16.246%	-0.50%
2	2		C	16.037%	+1.64%
3	4	▲	Python	9.842%	+2.16%
4	3	▼	C++	5.605%	-2.68%
5	6	▲	C#	4.316%	+0.36%
6	5	▼	Visual Basic .NET	4.229%	-2.26%
7	7		JavaScript	1.929%	-0.73%
8	8		PHP	1.720%	-0.66%
9	9		SQL	1.690%	-0.15%
10	12	▲	Swift	1.653%	+0.20%



# データサイエンスの中心的言語

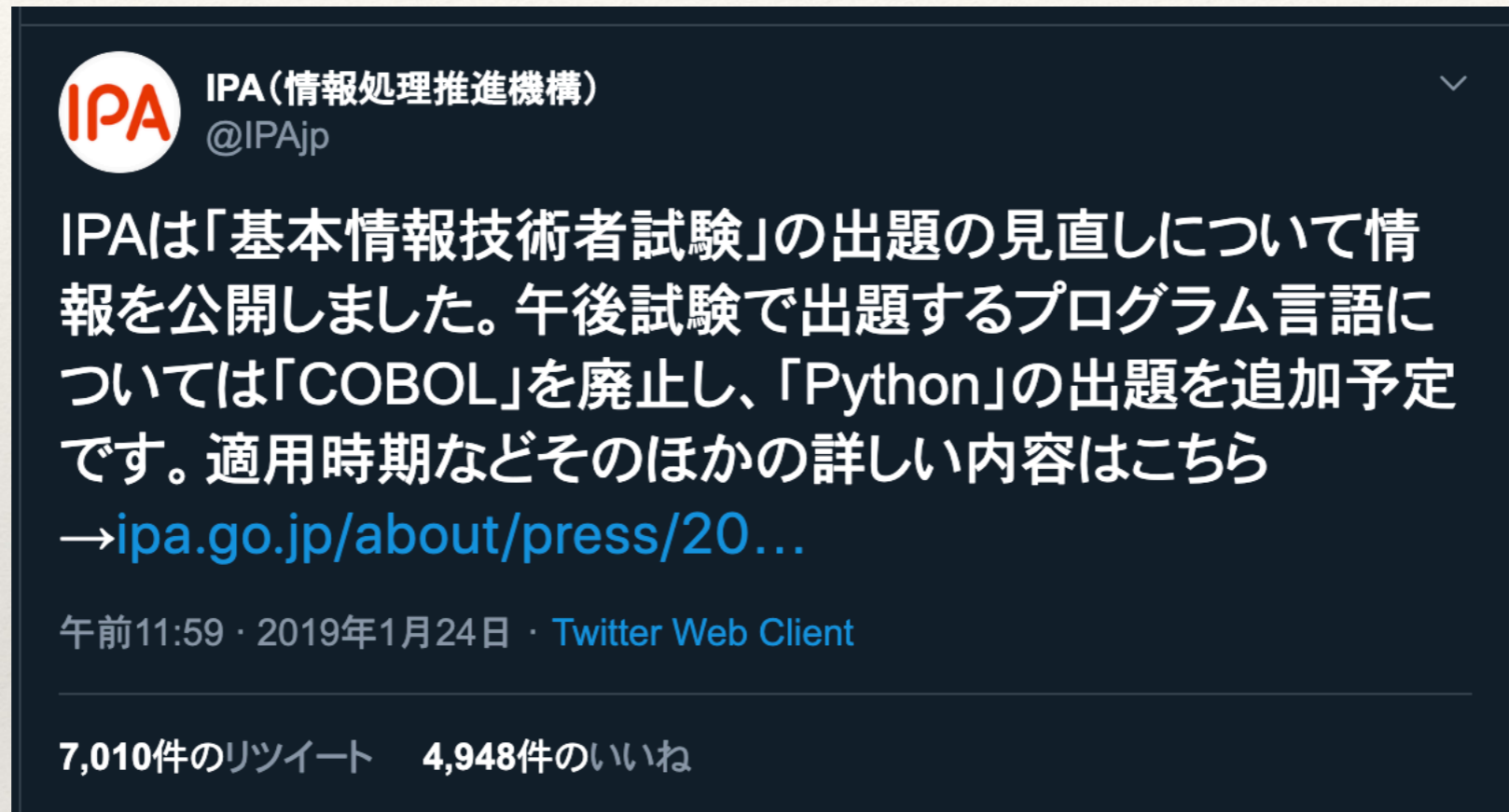



KDnuggetsが毎年行うアンケート

「データサイエンスにどんなツールを使っていますか？」



# 2020年春の試験からPythonが追加



 IPA (情報処理推進機構)  
@IPAjp

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細な内容はこちら  
→[ipa.go.jp/about/press/20...](https://ipa.go.jp/about/press/20190128.html)

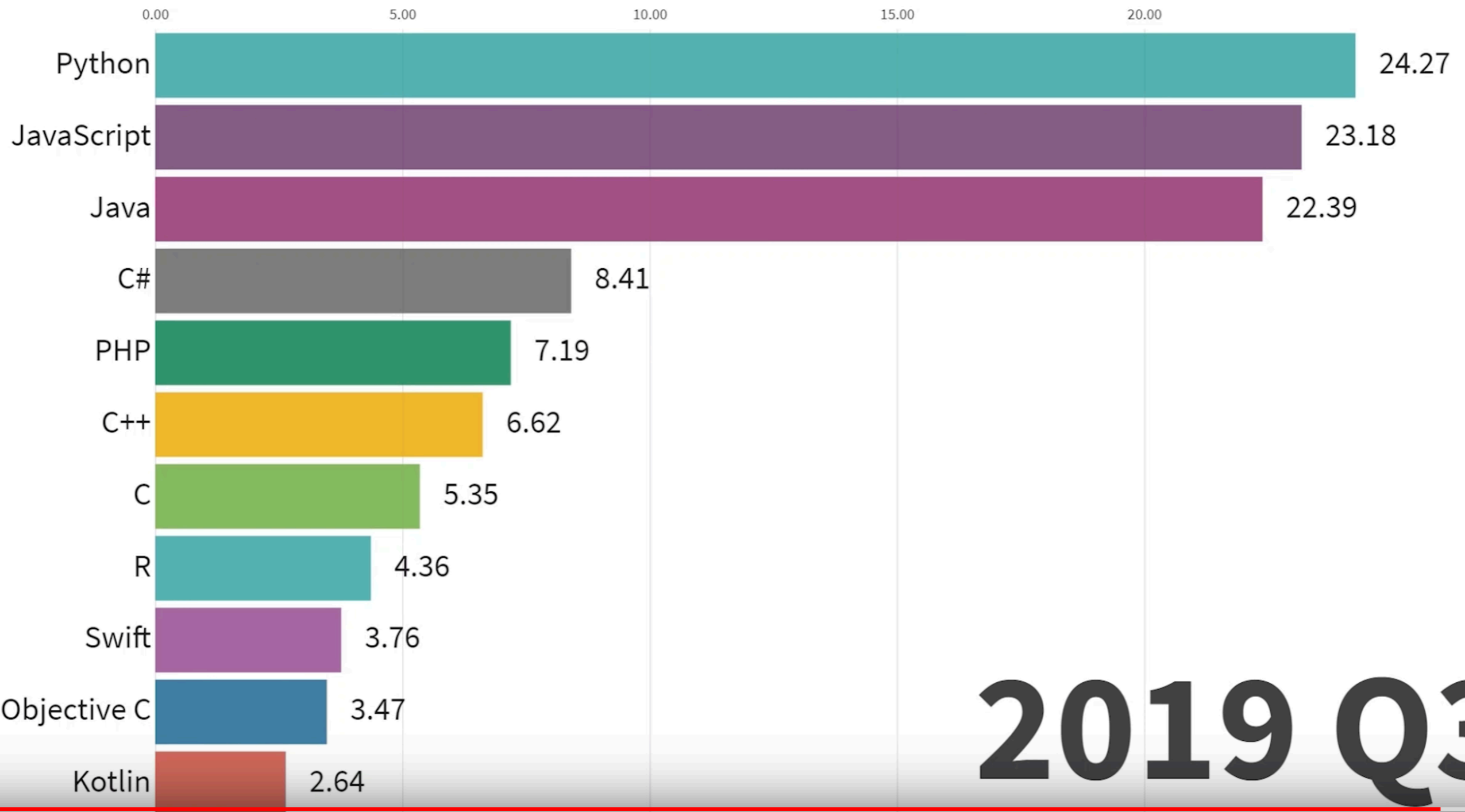
午前11:59 · 2019年1月24日 · Twitter Web Client

7,010件のリツイート 4,948件のいいね

IPAのWebページでPythonのサンプル問題が公開されています



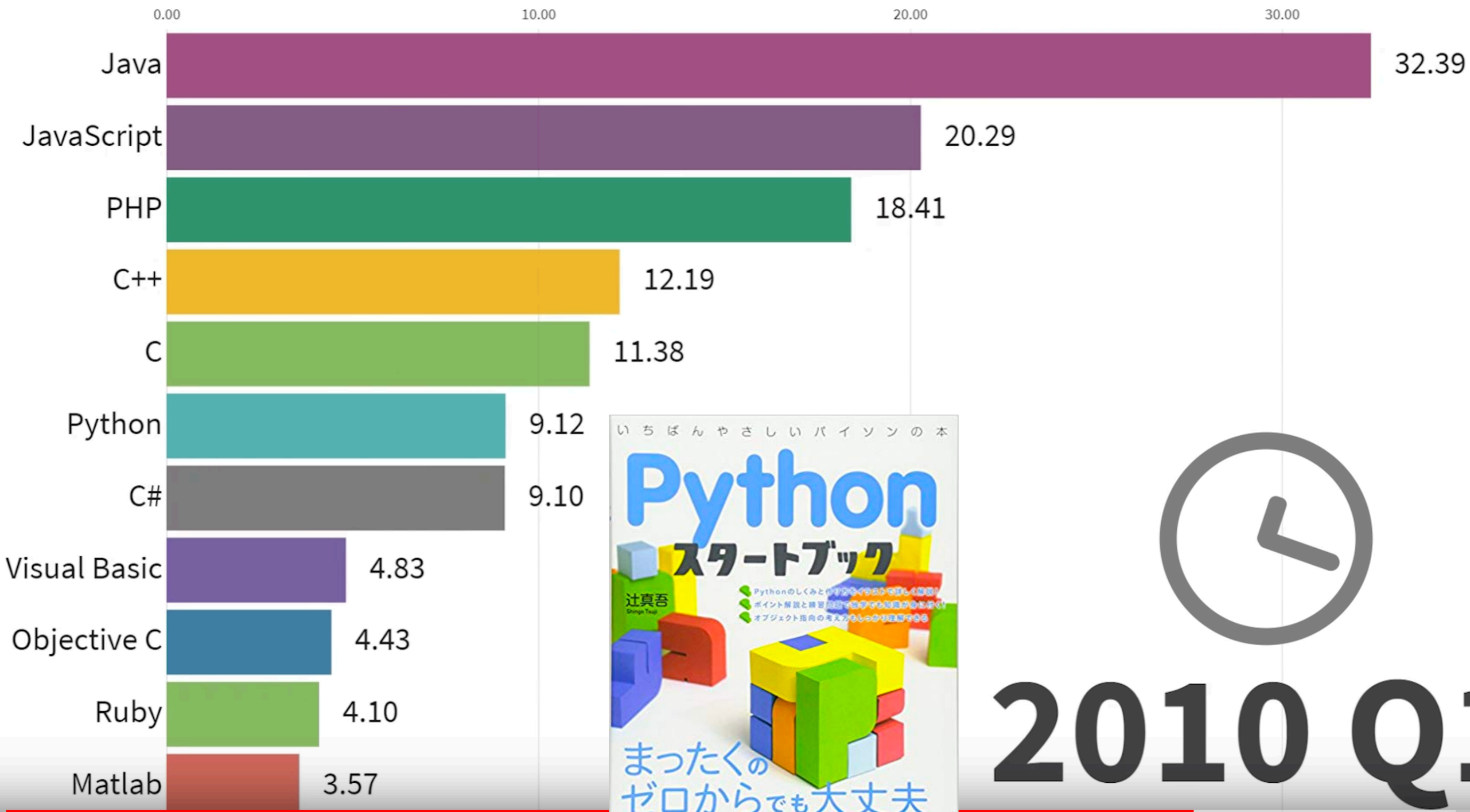
# 過去50年間のプログラミング言語人気の変遷



2019 Q3



# 2010年4月に出版



# 2010 Q1



---

# Start Python Club

---

- ❖ 「Pythonスタートブック」を読んでもくれた阿久津さん (@akucchan\_world) と2015年5月にはじめた活動
- ❖ Pythonと技術を楽しく学べる場を提供
- ❖ 東京新橋の月例会＋不定期に長野、つくば、福岡
- ❖ #stapyでいろいろなツイートあります





## Start Python Club

Pythonでスタートする人たちの集い

開催前イベント

▶ もっと見る

2019/12/11(水) [みんなのPython勉強会 i...](#)

2019/12/11(水) [みんなのPython勉強会#...](#)

イベント

メンバー

資料

B! 4

いいね! 109

メンバーになる

### 次回イベント



開催前 2019/12/11(水) 19:00~

12/20人

[みんなのPython勉強会 in 名古屋#11\(旧:Pythonもくもく会 in 名古屋\)](#)

Takeshi Akutsu 他

愛知県 名古屋市中村区 名駅 4-5-26 ユニモール桜ビル 3F



開催前 2019/12/11(水) 19:00~

47/150人

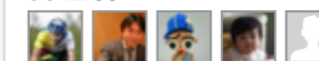
[みんなのPython勉強会#52](#)

Takeshi Akutsu 他

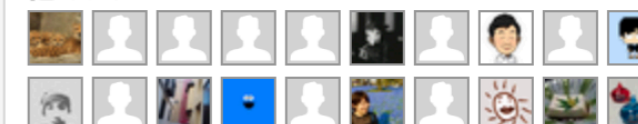
東京都港区新橋4-1-1

### メンバー (5232人)

管理者



他のメンバー



### 資料(267件)

[最近思った機械学習\(PyTorch\)のベストプラクティス](#)

毎回、リーディング・エッジ社（東京・新橋）の会議室を借りています



# Python is eating the world:

How one developer's side project became the hottest programming language on the planet

By Nick Heath



Image: Dan Stroud under the Creative Commons



# 最近のGuido先生

- ❖ 2018年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
- ❖ 2018年11月5人の委員会による運営方針を決定 (PEP8016)
  - ❖ Guido先生も入ってます
- ❖ 今年10月勤めていたDropboxを退職し悠々自適？な生活へ



Company

# Thank you, Guido

By Dropbox Team

Published on October 29, 2019

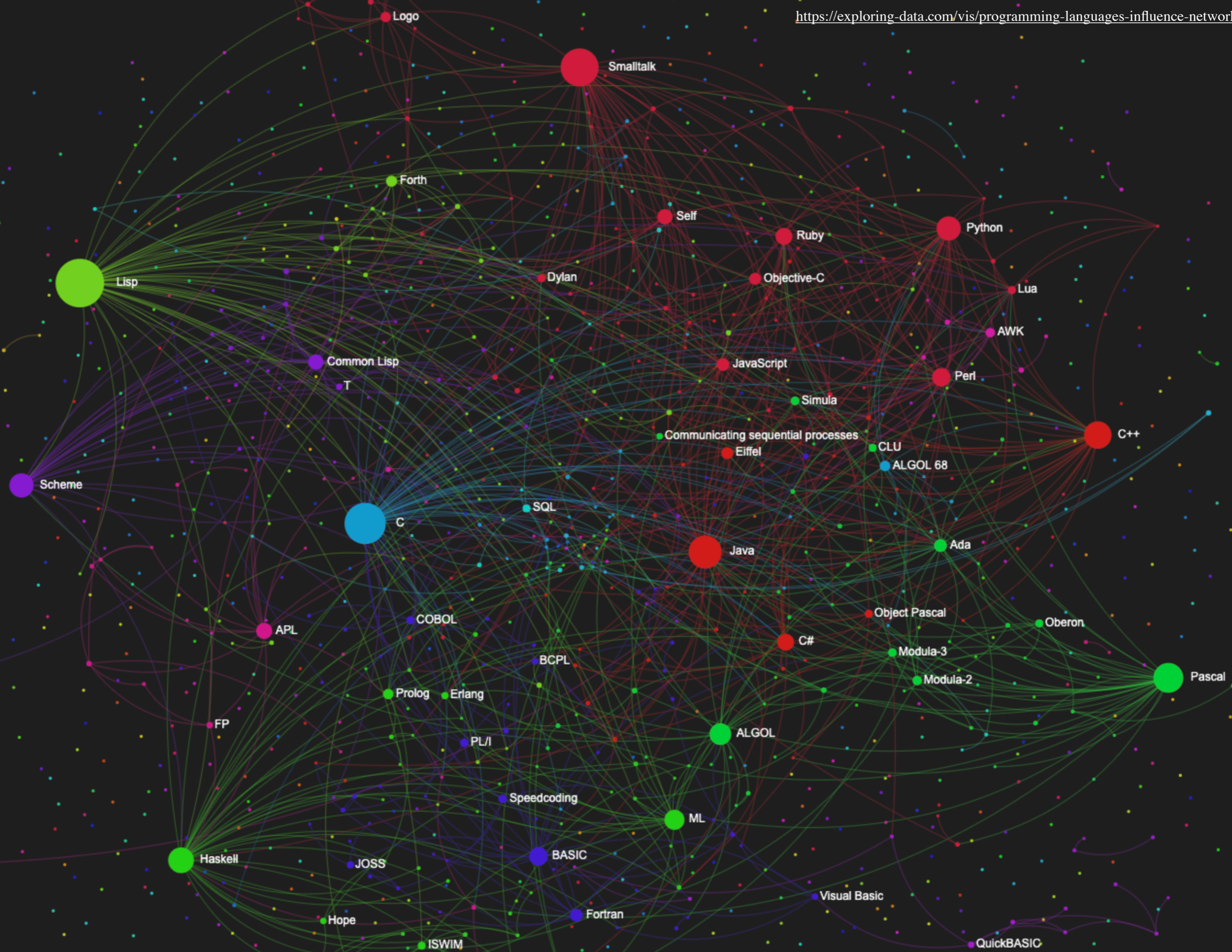


“There are only two kinds of languages: the ones people complain about and the ones nobody uses.”

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

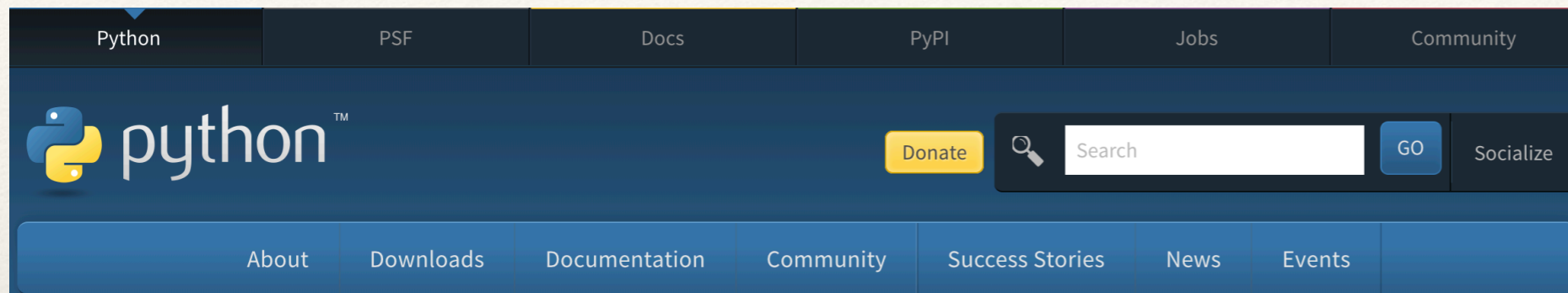
完璧なプログラミング言語はいまだ存在していない







# PEP: Python Enhancement Proposals



Tweets by @ThePSF

**Python Software Foundation** @ThePSF  
Are you looking to get better at Python and do some good this year for Black Friday? @talkpython Training has their everything course bundle on sale for \$50 and some proceeds go to support the #PSF and they'll plant 50 trees / purchase. [training.talkpython.fm/black-friday-2...](https://training.talkpython.fm/black-friday-2...) 12h

**Python Software Foundation** @ThePSF  
"The PSF was North Bay Python's first sponsor. Their early financial support for our mission helped kickstart what has become one of the most well-regarded regional conferences in our community." - Christopher Neugebauer, Conference Organizer @northbaypython

Python >>> Python Developer's Guide >>> PEP Index >>> PEP 0 -- Index of Python Enhancement Proposals (PEPs)

## PEP 0 -- Index of Python Enhancement Proposals (PEPs)

PEP:	0
Title:	Index of Python Enhancement Proposals (PEPs)
Last-Modified:	<a href="#">2019-11-28</a>
Author:	python-dev <python-dev at python.org>
Status:	Active
Type:	Informational
Created:	13-Jul-2000

Contents

SW	<a href="#">422</a>	Simpler customisation of class creation	Coghlan, Urban
IW	<a href="#">426</a>	Metadata for Python Software Packages 2.0	Coghlan, Holth, Stufft
SW	<a href="#">431</a>	Time zone support improvements	Regebro
SS	<a href="#">433</a>	Easier suppression of file descriptor inheritance	Stinner
SR	<a href="#">437</a>	A DSL for specifying signatures, annotations and ...	Krah
SR	<a href="#">439</a>	Inclusion of implicit pip bootstrap in Python ...	Jones
SR	<a href="#">455</a>	Adding a key-transforming dictionary to collections	Pitrou
SW	<a href="#">459</a>	Standard Metadata Extensions for Python Software ...	Coghlan
SW	<a href="#">460</a>	Add binary interpolation and formatting	Pitrou
PW	<a href="#">462</a>	Core development workflow automation for CPython	Coghlan
SR	<a href="#">463</a>	Exception-catching expressions	Angelico
SW	<a href="#">469</a>	Migration of dict iteration code to Python 3	Coghlan
SR	<a href="#">472</a>	Support for indexing with keyword arguments	Borini, Martinot-Lagarde
SR	<a href="#">473</a>	Adding structured data to built-in exceptions	Kreft
PW	<a href="#">474</a>	Creating forge.python.org	Coghlan
PW	<a href="#">481</a>	Migrate CPython to Git, Github, and Phabricator	Stufft
SR	<a href="#">490</a>	Chain exceptions at C level	Stinner

Pythonを良くしていくための提案



# PEP 8 -- Style Guide for Python Code

<b>PEP:</b>	8
<b>Title:</b>	Style Guide for Python Code
<b>Author:</b>	Guido van Rossum <guido at python.org>, Barry Warsaw <barry at python.org>, Nick Coghlan <ncoghlan at gmail.com>

This style guide evolves over time as additional conventions are identified and past conventions are rendered obsolete by changes in the language itself.

Many projects have their own coding style guidelines. In the event of any conflicts, such project-specific guides take precedence for that project.

## A Foolish Consistency is the Hobgoblin of Little Minds

One of Guido's key insights is that code is read much more often than it is written. The guidelines provided here are intended to improve the readability of code and make it consistent across the wide spectrum of Python code. As [PEP 20](#) says, "Readability counts".

A style guide is about consistency. Consistency with this style guide is important. Consistency within a project is more important. Consistency within one module or function is the most important.

However, know when to be inconsistent -- sometimes style guide recommendations just aren't applicable. When in doubt, use

Guido先生の鋭い洞察「コードは書かれるより読まれる時間の方が長い」



```
def insort_right(a, x, lo=0, hi=None):
    """Insert item x in list a, and keep it sorted assuming a is sorted.

    If x is already in a, insert it to the right of the rightmost x.

    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
    a.insert(lo, x)
```

インデントを強制するが、そのためコードが読みやすい。



[Help](#)[Donate](#)[Log in](#)[Register](#)

# Find, install and publish Python packages with the Python Package Index



Or [browse projects](#)

203,148 projects

1,527,308 releases

2,270,061 files

384,548 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](#)

豊富な外部ライブラリ pipコマンドで簡単に追加可能



---

# Pythonのインストール

---

- ❖ 2系と3系がある
  - ❖ 2.7は今年いっぱいでおしまい
- ❖ (方法1) 標準のPythonをインストールして、pipコマンドで外部ライブラリを追加
- ❖ (方法2) Anacondaなど外部ライブラリを同梱したディストリビューションの利用 環境の管理にはcondaを利用



# 落とし穴

- ❖ 環境変数 とくにPATH
- ❖ condaとpipは混ぜないほうがいい
- ❖ 詳しい解説はStapyスタッフでもあるnikkieによる記事  
(<https://qiita.com/ftnext/items/041cf583ba102f69ee45>)
- ❖ Dockerを使うという手もある



@ftnext 2019年09月09日に投稿

...

## Anacondaと付き合うための情報を整理する (Pythonのインストール方法とパッケージ管理 方法を中心に)

Python 環境構築 Anaconda

### はじめに

PyCon JP 2019で「Anaconda環境運用TIPS」という15分のトークをします。

ドキュメントを当たったり、Anacondaの環境で実験したりして得た知見全てを15分のトークに収めることが難しかったので、別途記事にすることにしました。

正確性を重視したいので、誤った記載を見つけた場合は、コメントなどでご指摘いただけるとありがたいです。

### 前提

### この記事で扱うこと

- Pythonのインストール方法 (Anaconda中心)
- AnacondaでPythonをインストールした場合のパッケージ管理方法の特徴



# 今年の話題

3.8のリリリースとPyConJP



# What's New In Python 3.8

**Editor:** Raymond Hettinger

This article explains the new features in Python 3.8, compared to 3.7. For full details, see the [changelog](#).

Python 3.8 was released on October 14th, 2019.

## Summary – Release highlights

### New Features

#### Assignment expressions

There is new syntax `:=` that assigns values to variables as part of a larger expression. It is affectionately known as “the walrus operator” due to its resemblance to [the eyes and tusks of a walrus](#).

In this example, the assignment expression helps avoid calling `len()` twice:

```
if (n := len(a)) > 10:
    print(f"List is too long ({n} elements, expected <= 10)")
```

A similar benefit arises during regular expression matching where match objects are needed twice, once to test whether a match occurred and another to extract a subgroup:

```
discount = 0.0
if (mo := re.search(r'(\d+)% discount', advertisement)):
    discount = float(mo.group(1)) / 100.0
```

The operator is also useful with while-loops that compute a value to test loop termination and then need that same value again in the body of the loop:

```
# Loop over fixed length blocks
while (block := f.read(256)) != "":
    process(block)
```

Another motivating use case arises in list comprehensions where a value computed in a filtering condition is also needed in the expression body:

```
[clean_name.title() for name in names
 if (clean_name := normalize('NFC', name)) in allowed_names]
```



# 代入式 (PEP572) とは？

## ❖ assignment expression (代入式) の光と影

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かった。



# たしかに便利な気はする

```
my_list = list(range(10))
```

```
if len(my_list) >= 5:  
    n = len(my_list)  
    print(f'あなたのリスト長さは{n}なので長すぎます。')
```

あなたのリスト長さは10なので長すぎます。

```
if (n := len(my_list)) >= 5:  
    print(f'あなたのリスト長さは{n}なので長すぎます。')
```

あなたのリスト長さは10なので長すぎます。

:=はセイウチ演算子と呼ばれている





# 密かに話題の3.8の新機能

```
a, b, c = 6, 28, 496
```

```
# 長いプログラムの途中でデバッグ用に出力したい  
print(f'a = {a} / b = {b} / c = {c}')
```

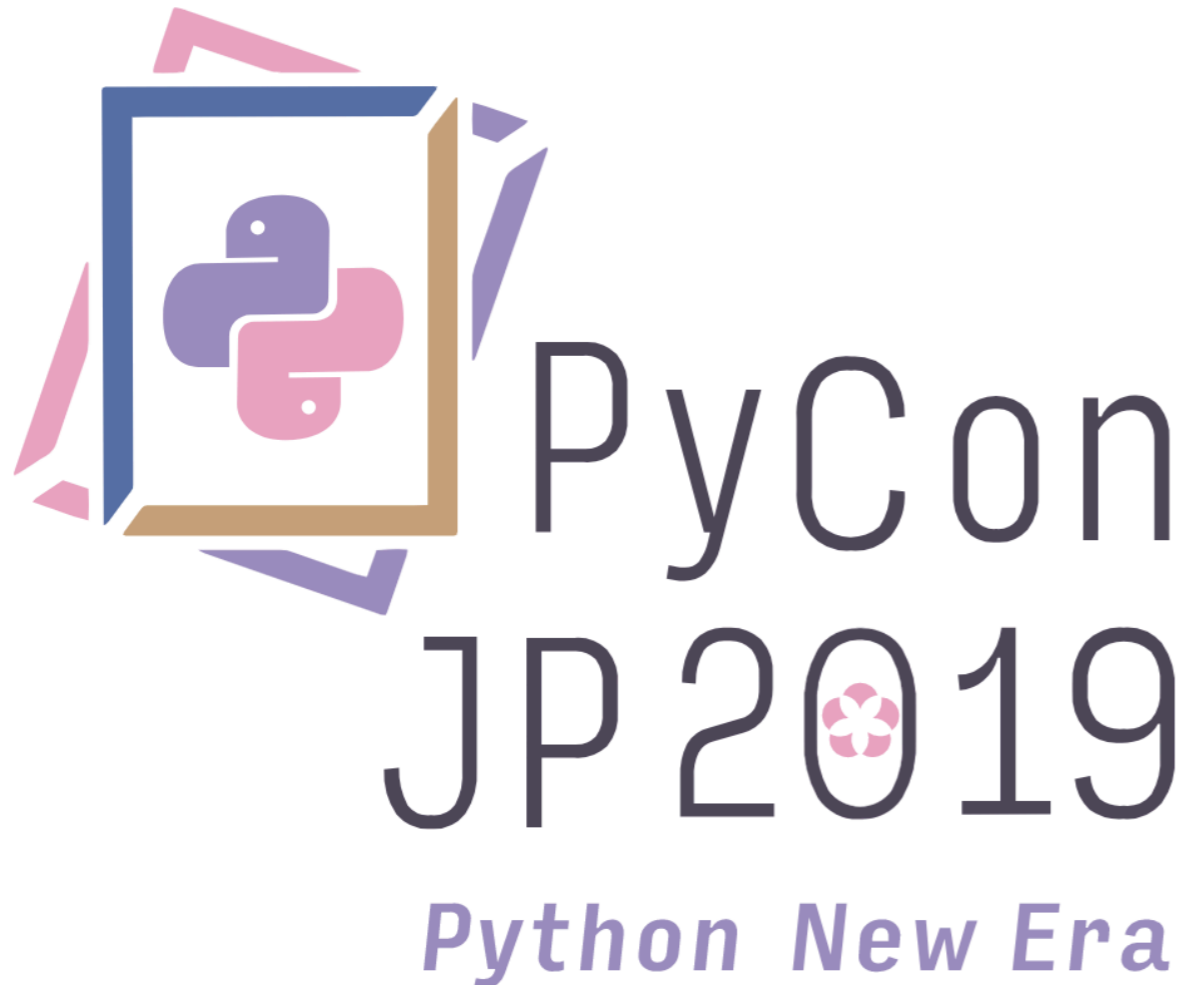
```
a = 6 / b = 28 / c = 496
```

```
# これでいい  
print(f'{a = } / {b = } / {c = }')
```

```
a = 6 / b = 28 / c = 496
```



# PyCon JP



2019.09.14 - 17

 #pyconjp

## Conference (終了しました)

カンファレンス

09.16 月・祝 - 09.17 火

 大田区産業プラザ PiO

タイムテーブル >

セッション一覧 >

Youtubeでセッションを見る >

発表資料を見る >

YouTubeにアーカイブがあります

[https://www.youtube.com/channel/UCxNoKygeZIE1AwZ\\_NdUCkhQ](https://www.youtube.com/channel/UCxNoKygeZIE1AwZ_NdUCkhQ)

来年は、2020年8月28日・29日に同じ会場で開催予定



お知らせ **!** 「秋得」キャンペーン！今なら月額プランが12月末まで無料！ 詳細は→案内ページ

🏠 > 日経ニューメディア > 読者限定サービス

2019/09/17 20:30

ニュース

## Python商標登録に新たな動き、米国本家が「適切な方法で対処」と表明

白井 良 = 日経 xTECH

日経 **xTECH**



プログラミング言語Pythonの開発を支援する米Pythonソフトウェア財団（Python Software Foundation、以下PSF）は2019年9月17日、日本国内で「Python」が商標登録されている事象に対し、行動を取る予定があると明らかにした。Pythonの「本家」とも言えるPSFで商標委員会（Trademark Committee）の議長を務めるマーク・アンドレ・レンブルグ氏が、国内最大のPythonカンファレンス「PyCon JP 2019」のクロージングセッションに登壇して表明した。



Pythonソフトウェア財団の商標委員会の議長を務めるマーク・アンドレ・レンブルグ氏

（出所：PyCon JP）

[\[画像のクリックで拡大表示\]](#)

### 本誌アーカイブ

1999年1月4日号以降の「日経ニューメディア」本誌に掲載された記事を検索、閲覧できます。

キーワード

検索

[アーカイブ一覧](#)

### 読者限定サービスについて

記事をお読みにするには購読申し込みとは別にユーザーIDとパスワードの登録が必要です。

### 登録／会員登録情報の変更

### お申し込み

### 日経ニューメディアとは

### 定期購読お申し込み

### イベント・セミナー

### 関連サイト

### 日経 xTECH



# Pythonで学ぶアルゴリズムとデータ構造

- ❖ 講談社データサイエンス入門シリーズ
- ❖ 配列、ソート、探索、グラフなど古典的な内容
- ❖ 動的計画法
  - ❖ ナップサック問題
- ❖ 問題の難しさ PとNP
- ❖ 数論、乱択アルゴリズム、RSA暗号（公開鍵暗号）
- ❖ すべてのコードと練習問題の解答
  - ❖ <https://github.com/tsjshg/pyalgdata>





---

# アルゴリズムを学ぶ意味

---

- ❖ 「まえがき」と「あとがき」にも書きましたが、結構苦悩が・・・
- ❖ アルゴリズムを知る必要があるのか？
- ❖ ソートならsortedかsortメソッドを使えばいいだけ
- ❖ グラフもNetworkXなど優れたライブラリが多数
- ❖ プログラミングスキルを磨くよい練習題材になる



```
def insort_right(a, x, lo=0, hi=None):
    """Insert item x in list a, and keep it sorted assuming a is sorted.

    If x is already in a, insert it to the right of the rightmost x.

    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
    a.insert(lo, x)
```

## 二分探索の実装

Pythonはオープンソースなのでコードを読んで勉強できる



---

# まとめ

---

- ❖ Pythonが世界を席卷し始めた
- ❖ 読みやすい≒書きやすい
- ❖ いろいろな言語があり、Pythonも変化し続けている
- ❖ アルゴリズムはコーディングスキルアップのためのよい教材になる



ご清聴ありがとうございました！