みんなのPython勉強会 in 福岡#1 *11/7, 2019*

Pythonの基本と最近の動向

辻 真吾（@tsjshg）

Start Python Club

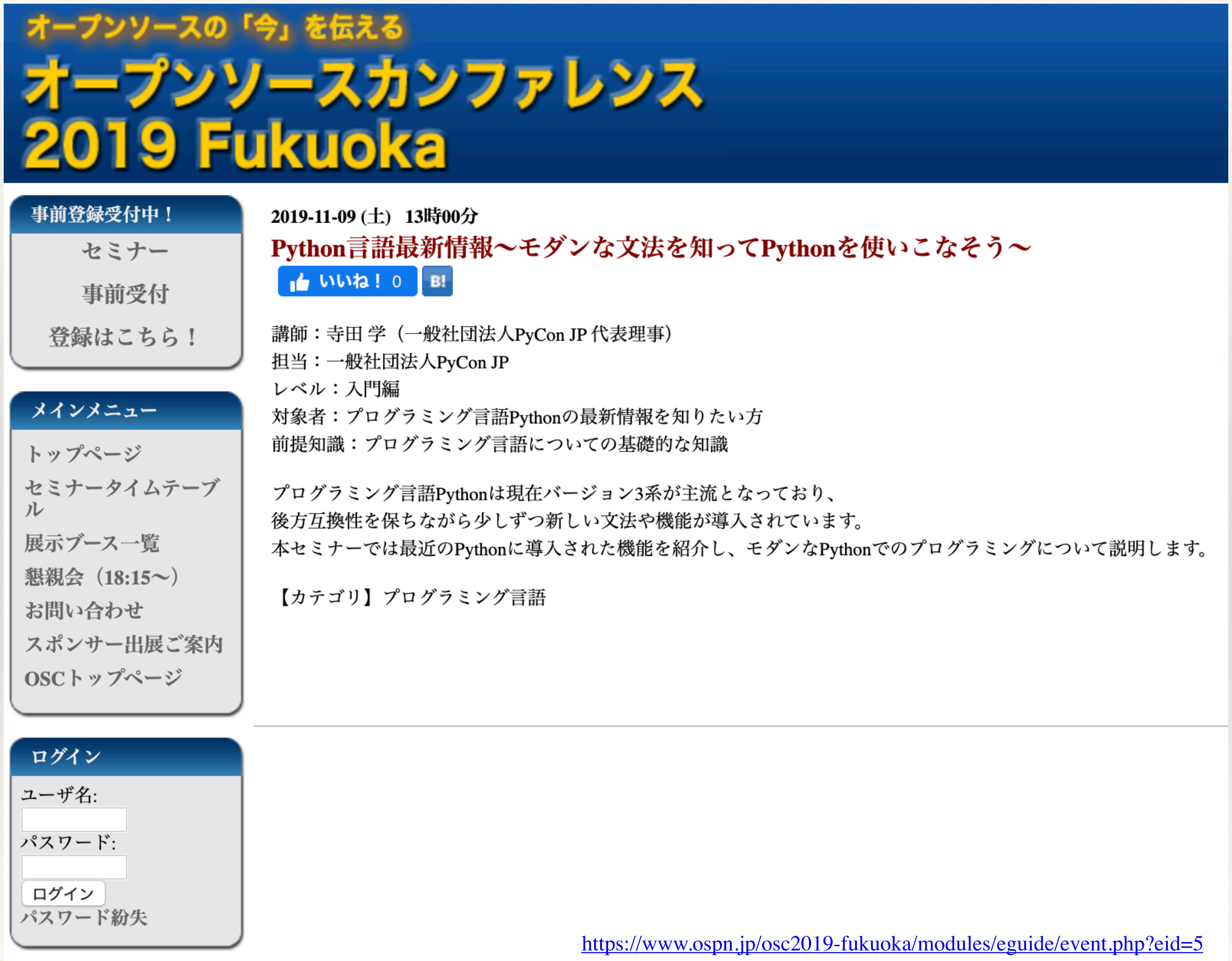
おまえ誰よ？

- 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月に出た本（黄色）と11月に出る本（緑色）

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）

Python界のスゴイ人が来る



より詳しい情報が、PyCon JPのブログページにあります https://pyconjp.blogspot.com/2019/11/pycamp-caravan-osc-fukuoka.html

Python Boot Camp in 福岡2nd

https://pyconjp.connpass.com/event/143786/

Q: お仕事または趣味でコードを書いている？

(さらに言うなら、Python使っている？)

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

https://spectrum.ieee.org/static/interactive-the-top-programming-languages-2019

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 %

http://pypl.github.io/PYPL.html

1位じゃないのもあります

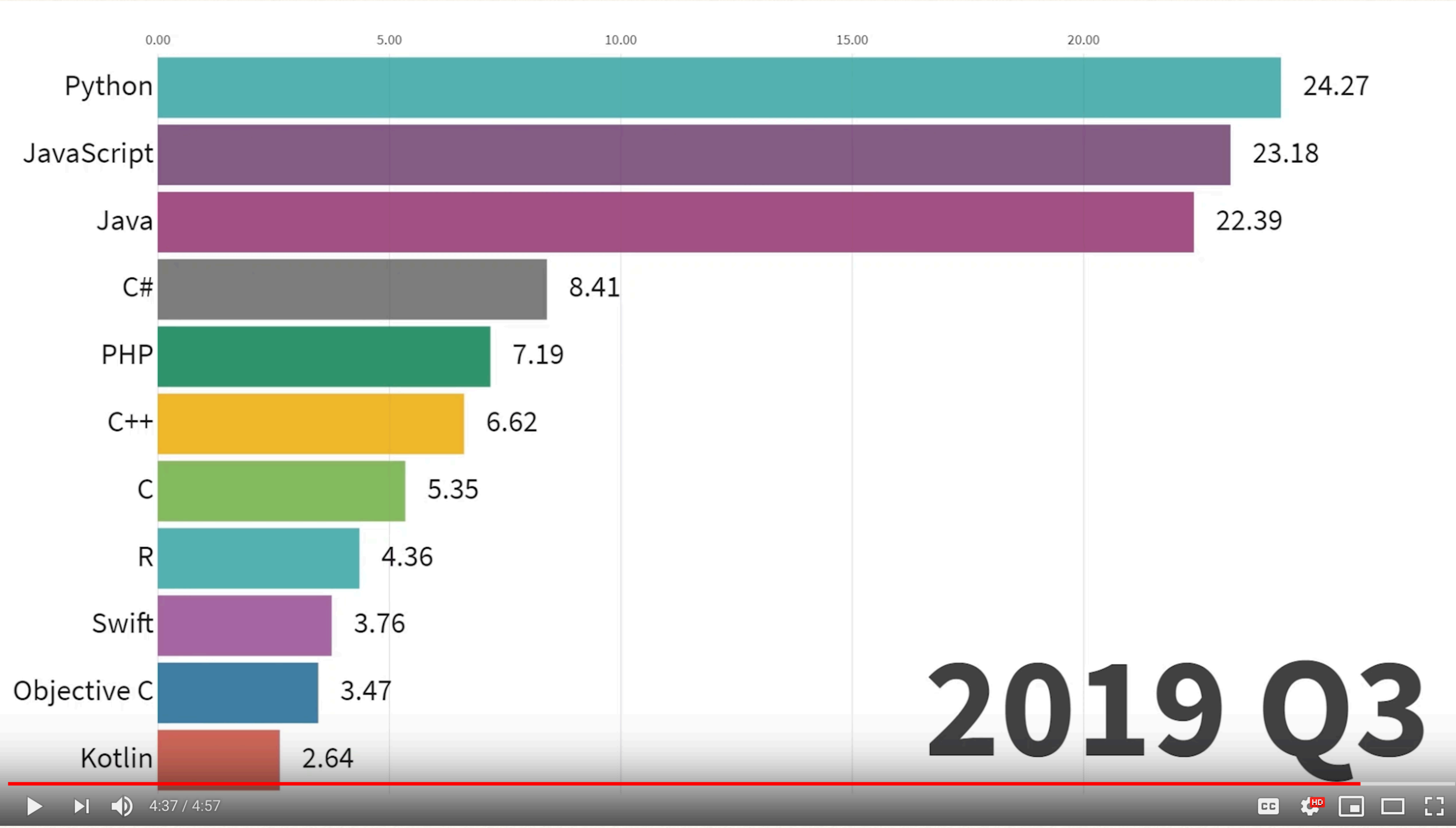
TIOBE Index for November 2019

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

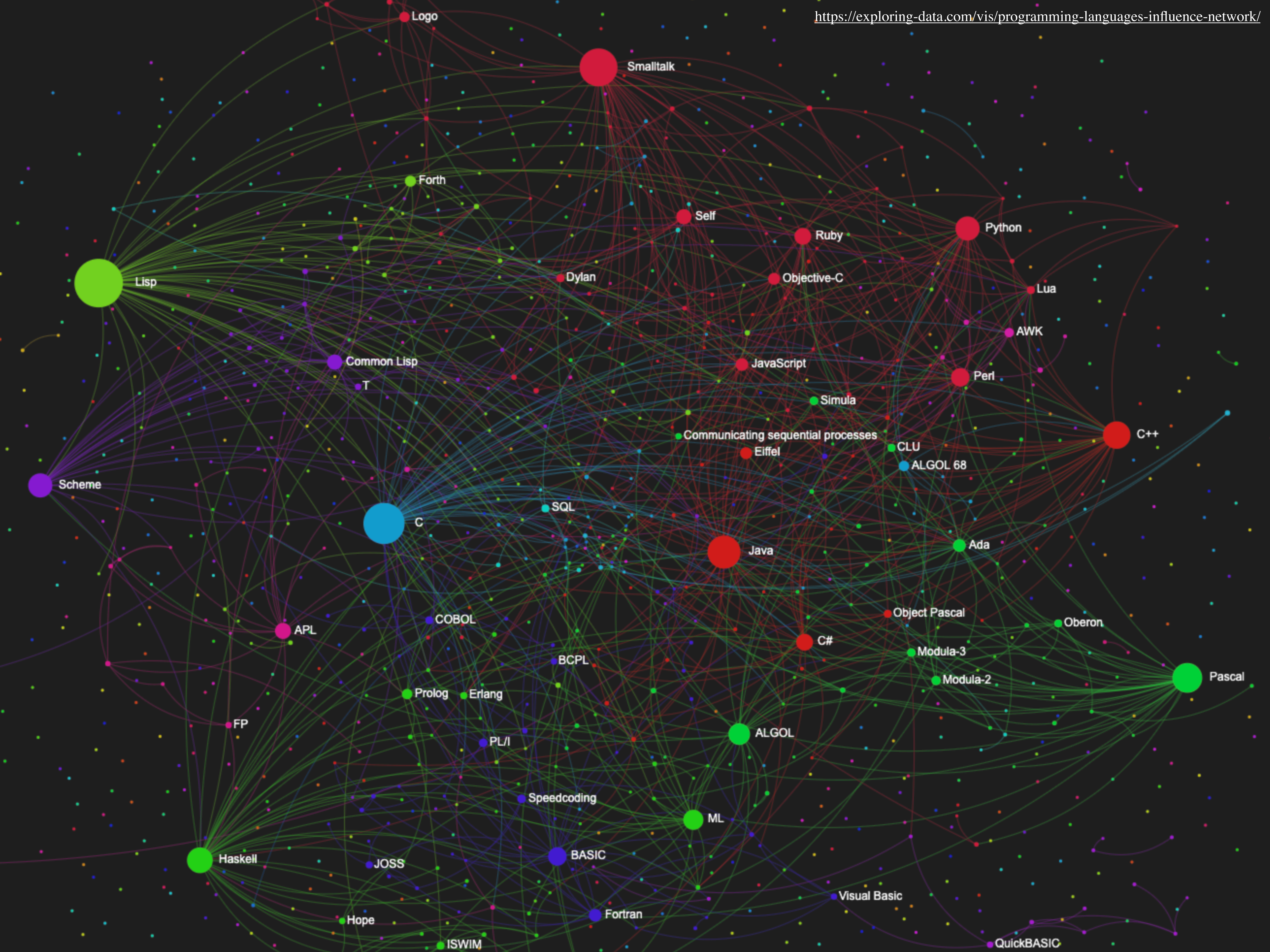
Nov 2019	Nov 2018	Change	Programming Language	Ratings	Change
1	1		Java	16.246%	-0.50%
2	2		C	16.037%	+1.64%
3	4	^	Python	9.842%	+2.16%
4	3	v	C++	5.605%	-2.68%
5	6	^	C#	4.316%	+0.36%
6	5	v	Visual Basic .NET	4.229%	-2.26%
7	7		JavaScript	1.929%	-0.73%
8	8		PHP	1.720%	-0.66%
9	9		SQL	1.690%	-0.15%
10	12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

https://www.tiobe.com/tiobe-index/

過去50年間のプログラミング言語人気の変遷



https://www.youtube.com/watch?v=Og847HVwRSI https://exploring-data.com/vis/programming-languages-influence-network/



"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（していない）

Python is eating the world:

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

By Nick Heath

https://www.techrepublic.com/article/python-is-eating-the-world-how-one-developers-side-project-became-the-hottest-programming-language-on-the-planet/

Python ››› Python Developer's Guide ››› PEP Index ››› PEP 8 -- Style Guide for Python Code

PEP 8 -- Style Guide for Python Code

PEP:	8
Title:	Style Guide for Python Code
Author:	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」

```python
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。

- `re` — Regular expression operations
- `difflib` — Helpers for computing deltas
- `textwrap` — Text wrapping and filling
- `unicodedata` — Unicode Database
- `stringprep` — Internet String Preparation
- `readline` — GNU readline interface
- `rlcompleter` — Completion function for GNU readline

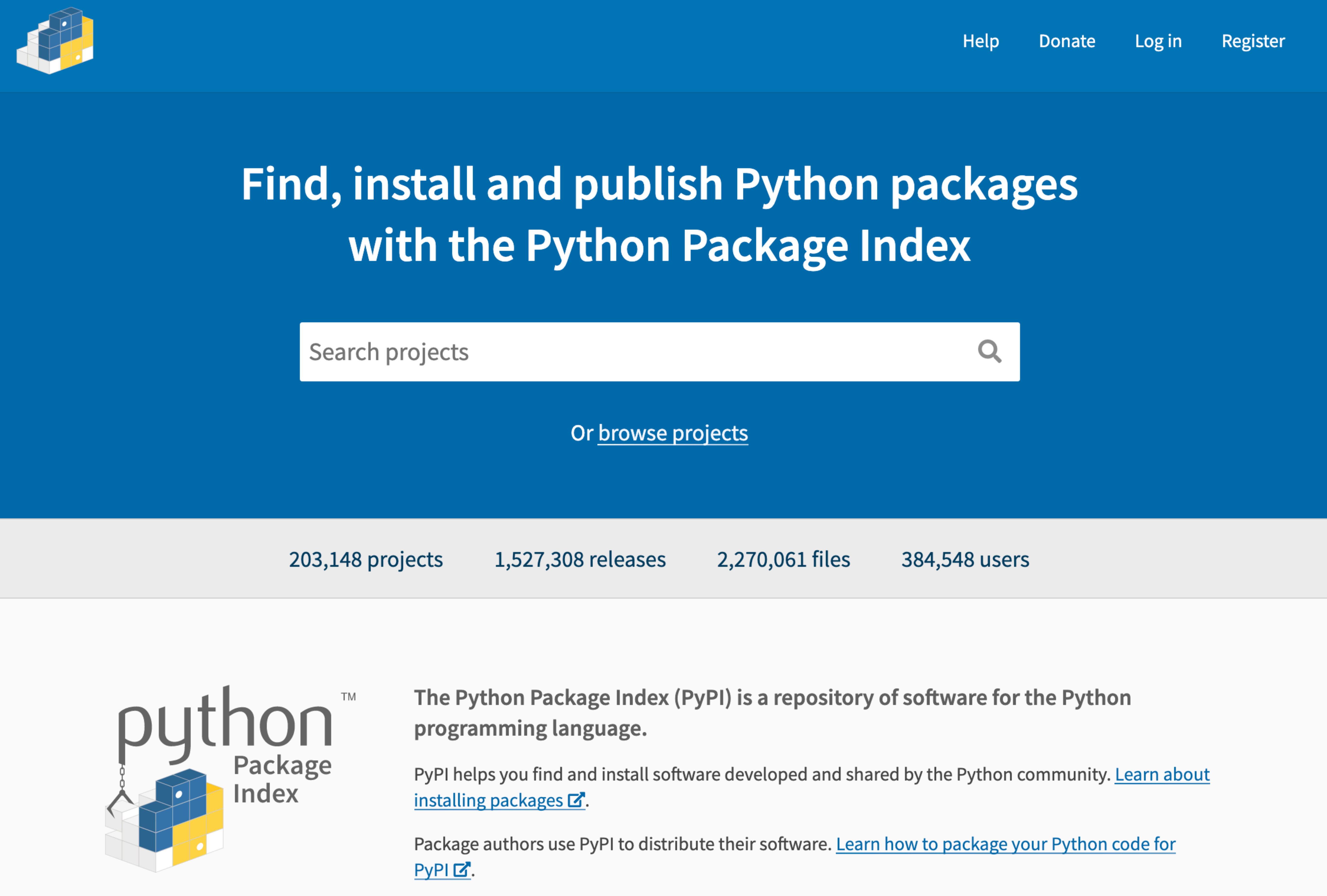
- Binary Data Services
  - `struct` — Interpret bytes as packed binary data
  - `codecs` — Codec registry and base classes
- Data Types
  - `datetime` — Basic date and time types
  - `calendar` — General calendar-related functions
  - `collections` — Container datatypes
  - `collections.abc` — Abstract Base Classes for Containers
  - `heapq` — Heap queue algorithm
  - `bisect` — Array bisection algorithm
  - `array` — Efficient arrays of numeric values
  - `weakref` — Weak references
  - `types` — Dynamic type creation and names for built-in types
  - `copy` — Shallow and deep copy operations
  - `pprint` — Data pretty printer
  - `reprlib` — Alternate `repr()` implementation
  - `enum` — Support for enumerations
- Numeric and Mathematical Modules
  - `numbers` — Numeric abstract base classes
  - `math` — Mathematical functions
  - `cmath` — Mathematical functions for complex numbers
  - `decimal` — Decimal fixed point and floating point arithmetic
  - `fractions` — Rational numbers
  - `random` — Generate pseudo-random numbers

充実した標準ライブラリ

この他にHTTPサーバやSQLiteなど

Butteries Includedと言われる

https://docs.python.org/3/library/index.html



豊富な外部ライブラリ　pipコマンドで簡単に追加可能

整数には大きさの制限がない

```
2**216_091 - 1
```

```
7460931030646613436873395794005114895402287540849773288051133049777936627252709687806643956351409557300
0836449415488275742723006299922094081956877578745064819814439468245248674029224495816774248370486483282
9986563153163521212340663734031506157753431814094252985897158828388086813446345123723991685390916879633
3674502254039533266955915753912516109458086972343184814561433194919942423248129480825283814784335269926
5529258003598268623744067114196429865781651682956202815223253794237749860195335965009575222187827232549
8992231767054266246687206726195420071159525576585344363951921893188823602556718780828011691620449083031
6961802141734679873071104841105684443197349401151508334766947900287278402222519455983405494844747255814
5165480704222919872100190703332499487016023504996566702637597998163097809697665710513174505875823725365
0501516660264780270998482777946114795017998269084407457156835180042530198900065756321766511676033520448
5842854748859866388508008474636722119497250239143071556987978887520697320272127679725663472182249170439
5568179987519368032618115788750733048142295502778371184133025576662541021719456411958717742814897270447
2558988328401178756126598945622175271864462227642384450832874510229498246948160879263759022189446710903
5082364534256576780385072138718117910393932418388206378146190438651750718152898894549078360534177575704
1234649337401747972051081068954003382183157863988192144899958656474706720396039781956903521514969675075
9317343832657638511532886057830235887943794233492250267638750397331035062187099099571773992030350934703
8015876743838313631262398836820372683525572784027128351580414889961141919835827473077789146107261505490
8788576288596831632518942157916180880863234011631729313750840661398637165533953131401091422019730741638
2847926081858242171939415725939091644727494895374921519243510707008521866471117451604841600454455598255
5080017122941382037059757505521222536105986960013565000287270491598424613382757860325535339267687280963
5753033323773702776997043677317118035983737434614277099245739972179896605492751645738357060944756326179
```

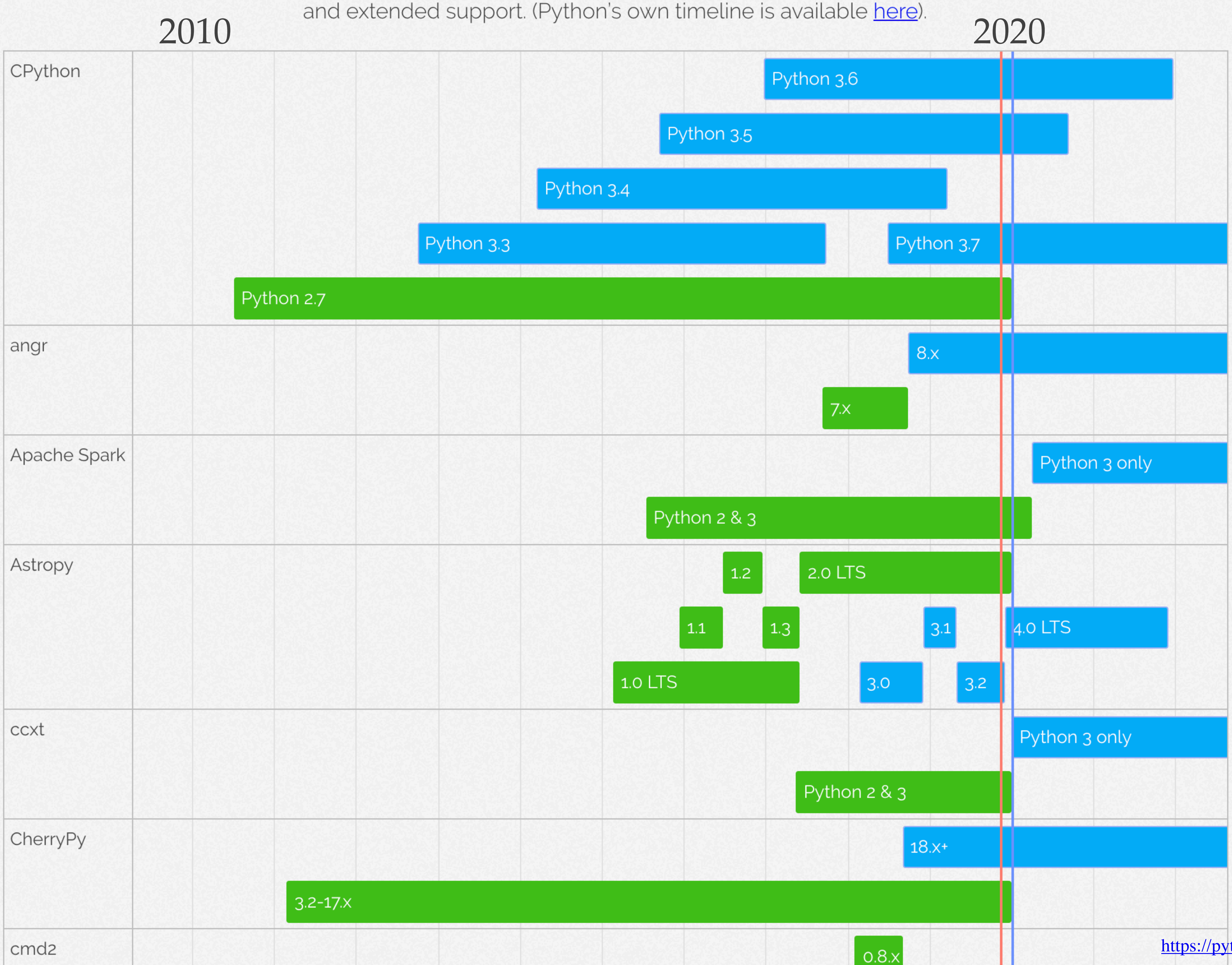
65,050桁の素数（1とその数自身しか約数がない数）

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

Projects Timeline

In the following chart, you can see many projects that have decided to stop supporting Python 2 before 2020. The chart is a guideline to show what versions of each project support Python 2, or not, their release timelines, and extended support. (Python's own timeline is available here).



https://python3statement.org/

Anaconda Distribution

The World's Most Popular Python/R Data Science Platform

Download

The open-source Anaconda Distribution is the easiest way to perform Python/R data science and machine learning on Linux, Windows, and Mac OS X. With over 15 million users worldwide, it is the industry standard for developing, testing, and training on a single machine, enabling *individual data scientists* to:

- Quickly download 1,500+ Python/R data science packages
- Manage libraries, dependencies, and environments with Conda
- Develop and train machine learning and deep learning models with scikit-learn, TensorFlow, and Theano
- Analyze data with scalability and performance with Dask, NumPy, pandas, and Numba
- Visualize results with Matplotlib, Bokeh, Datashader, and Holoviews

Windows | macOS | Linux

https://www.anaconda.com/distribution/

Jupyterがすごい

IPython NotebookがJupyter Notebookに名称変更

Python以外のプログラミング言語にも対応（たとえばRなど）

jupyter Jupyter_notebookのデモ Last Checkpoint: 2 minutes ago (autosaved) Logout

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

Code

In [1]:
```
import pandas as pd
```

In [2]:
```
data = pd.read_excel('sample_data.xlsx')
```

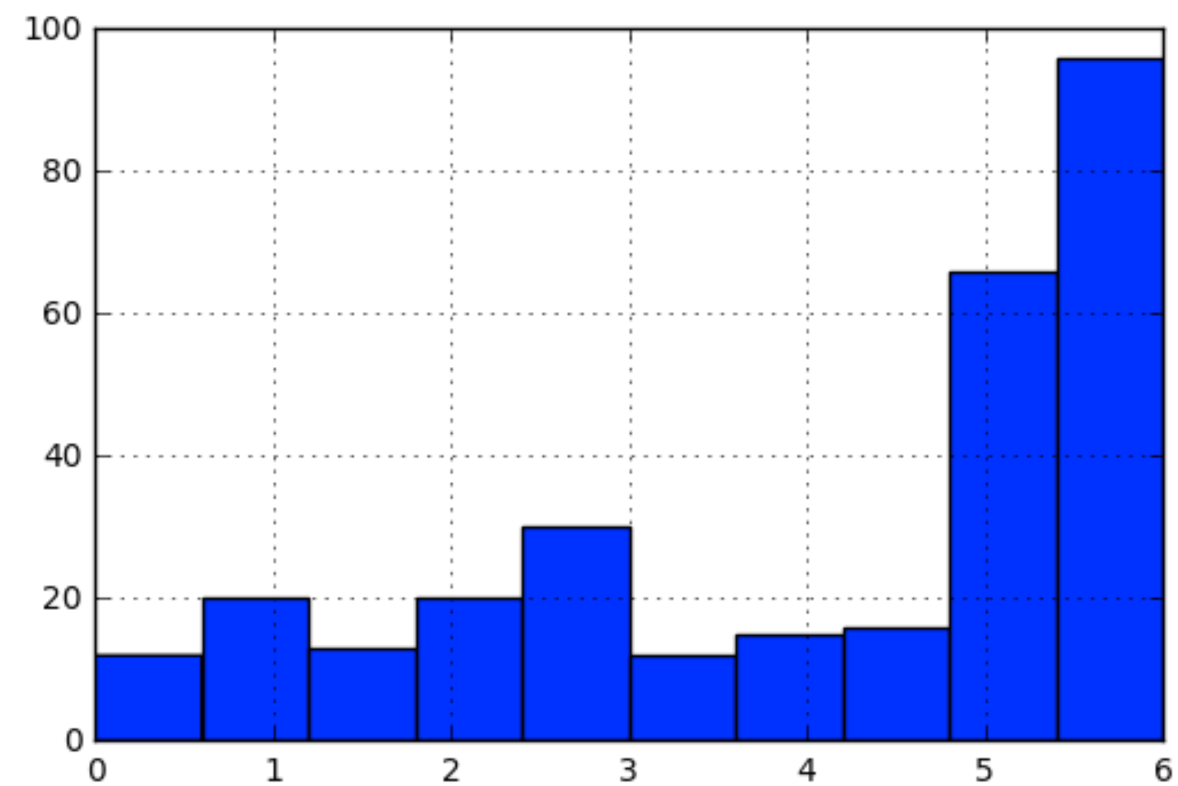
In [3]:
```
data.head()
```

Out[3]:

	sample_0	sample_1	sample_2	sample_3	sample_4	sample_5	sample_6	sample_7	sample_8	sample_9
var_0	1.181546	2.027050	2.307813	1.078164	2.023769	1.088772	0.199090	1.002069	0.715403	1.895127
var_1	0.998445	0.601550	0.054010	0.028059	1.024477	0.737187	2.145073	2.395837	0.594704	0.319478
var_10	5.659790	8.118167	8.338933	8.388697	8.083567	8.349710	8.332364	8.143534	8.277051	8.338494
var_100	2.383074	2.155451	0.948400	2.090212	0.653381	0.897784	1.413244	0.544167	2.484165	0.101348
var_101	5.667063	5.848190	6.075058	5.611951	5.649971	5.808400	5.618007	5.583713	5.629790	5.576541

In [5]:
```
%matplotlib inline
data['sample_0'].hist()
```

Out[5]: <matplotlib.axes._subplots.AxesSubplot at 0x1146362b0>



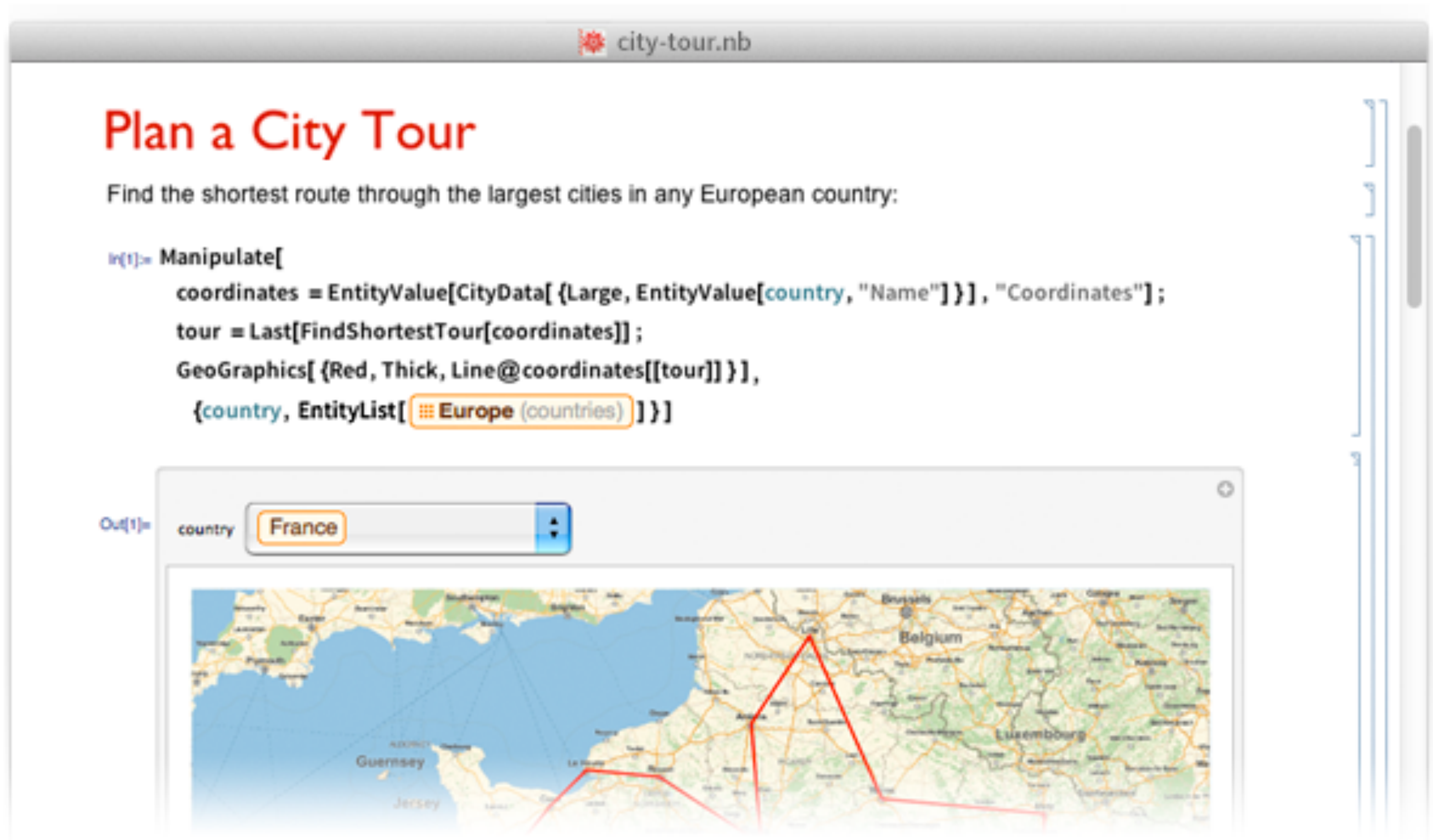
jupyter notebook

エクセルファイルをpandasのDataFrameとして読み込んで、ヒストグラムを描く

Mathematicaで使われるWolfram言語のドキュメントより

ノートブックドキュメント

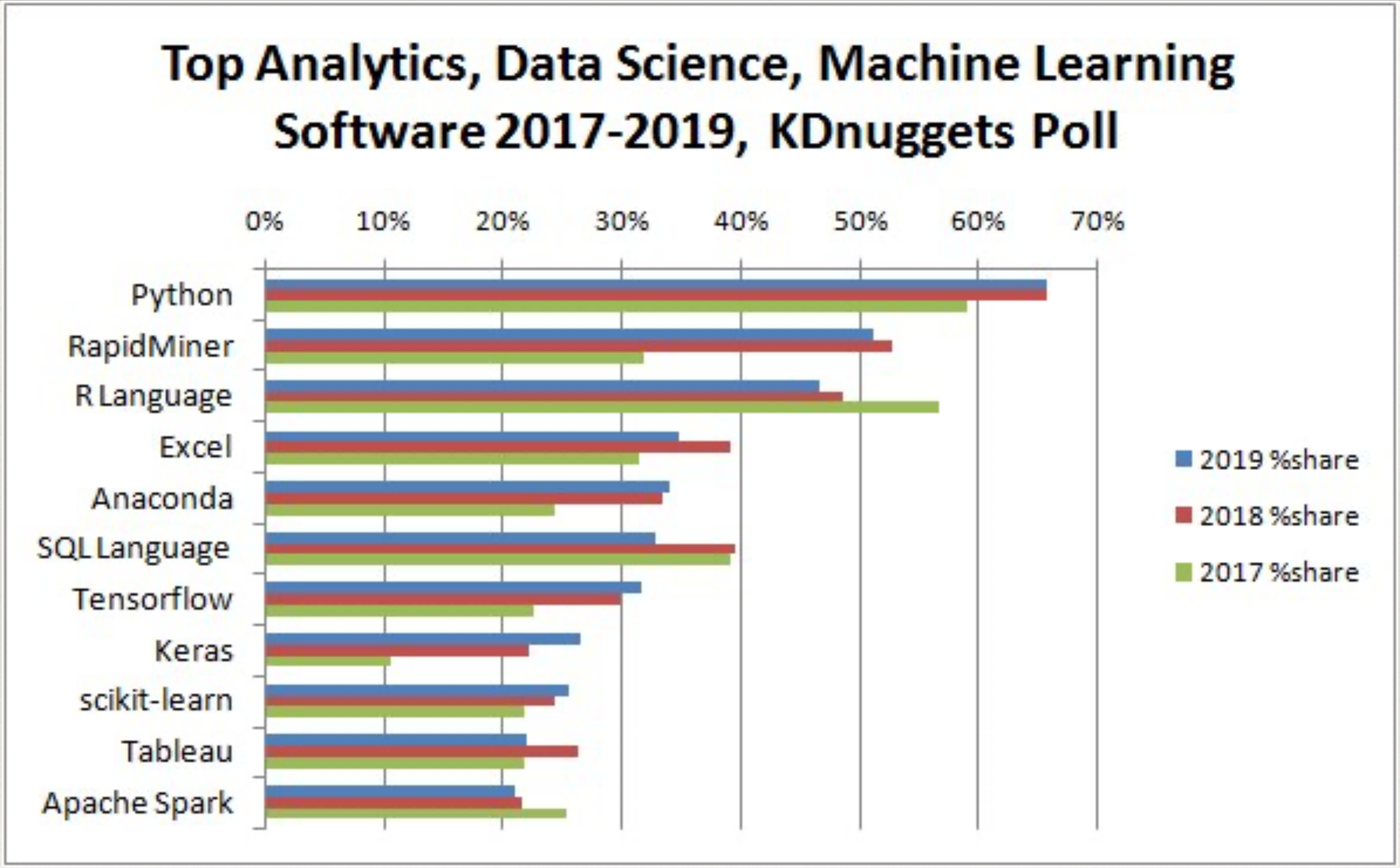
Wolframノートブックは，テキスト，グラフィックス，インターフェース等とコードを組み合せることができ，デスクトップでもWebでも使えます：



PYTHONのプログラマー向けの注意事項
Wolframノートブックは25年以上に渡り常に開発され続けており，Wolfram言語のデスクトップ版とクラウド版のどちらにおいても完全に統合されています．Wolframノートブックは Jupyterのようなノートブックライブラリのきっかけとなっています．

http://www.wolfram.com/language/fast-introduction-for-programmers/ja/notebook-documents/

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？」

https://www.kdnuggets.com/2019/05/poll-top-data-science-machine-learning-platforms.html

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

}
```

```
if (x=0){
    実行されないけどね

}
```

x が0なら「何かの処理」が実行される」

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

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

```python
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。

```python
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

来年は、2020年8月28日・29日に同じ会場で開催予定

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ソフトウエア財団（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 Commitee）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ソフトウエ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

記事をお読みにな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

https://tech.nikkeibp.co.jp/atcl/nxt/news/18/05990/?ST=nnm

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

https://blog.dropbox.com/topics/company/thank-you--guido

まとめ

- Pythonが世界を席巻し始めた
- 読みやすさを目指し、広く受け入れられた
- 充実の標準ライブラリと外部ライブラリ
- 3.8の新機能とGuido先生の引退
- Pythonで、出来ることが広がる世界をめざしたい

ご清聴ありがとうございました！