今年あらためて気付いたPythonのいいところ

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
- 最近読んで面白かった本
  - 伴名練「なめらかな世界と、その敵」
  - 加藤文元「宇宙と宇宙をつなぐ数式 IUT理論の衝撃」

https://www.hayakawa-online.co.jp/shopdetail/000000014286/

https://www.kadokawa.co.jp/product/321802000140/

関数の一部の引数を固定する

- 3章「Pythonで実践する機械学習（コードだけGithubにあります）」の中で利用

```
from functools import partial

# end引数のデフォルト値は改行(\n)
print('Hello', end='---')
```

```
Hello---
```

```
ends_with_Pe = partial(print, end='Pe')

ends_with_Pe('Hello!')
```

```
Hello!Pe
```

https://github.com/tsjshg/sd201903

https://gihyo.jp/magazine/SD

pathlib

- ファイルパスをオブジェクトとして扱える
  - もちろんwin, mac共通

```
import pathlib
```

```
# ルートディレクトリの取得
root = pathlib.Path('/')
root
```

```
PosixPath('/')
```

```
# /演算子を使ったパスの構築
my_home = root / 'Users' / 'shingo'
my_home
```

```
PosixPath('/Users/shingo')
```

```
list(my_home.glob('*.pdf'))
```

```
[PosixPath('/Users/shingo/TSUJI_20191206.pdf')]
```

f-string

- 文字列のformatメソッドよりやっぱりかなり便利

```
# Python3.6から
name = 'taro'
age = 3
f'{name}は{age}歳です。'
```

```
'taroは3歳です。'
```

```
import math
math.pi
```

浮動小数点数（float）のπ

```
3.141592653589793
```

```
f'円周率は{math.pi:.2f}です。'
```

```
'円周率は3.14です。'
```

3.8の新機能

```
# デバッグように変数の内容を出力
print(f'name = {name} age = {age}')
```

```
name = taro age = 3
```

```
#これでいい
print(f'{name = } {age = }')
```

```
name = 'taro' age = 3
```

実はこれが1番3.8でよく使う便利機能では？という意見もあるくらい

いくつかの変数を同時に画面出力したい時には重宝しそう

atsuoishimoto’s diary https://atsuoishimoto.hatenablog.com/entry/2019/09/10/085430

Positional-only parameter （new in 3.8）

```
def x_minus_y(x, y):
    return x - y
```

```
x_minus_y(7, 3)
```

```
4
```

```
x_minus_y(y=3, x=7)
```

こう書いてもいい

```
def x_minus_y(x, y, /):
    return x - y
```

xとyは位置だけで指定できる引数になる

```
x_minus_y(7, 3)
```

```
4
```

```
x_minus_y(y=3, x=7)
```

これはエラーになる

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
<ipython-input-34-dd46ed9a92c1> in <module>
----> 1 x_minus_y(y=3, x=7)

TypeError: x_minus_y() got some positional-only arguments passed as keyword arguments: 'x, y'
```

リストのスライス記法

```
my_list = list(range(10, 0, -1))
my_list
```

```
[10, 9, 8, 7, 6, 5, 4, 3, 2, 1]
```

```
my_list[-3:]
```

```
[3, 2, 1]
```

```
# 行き過ぎを許容してくれる
my_list[-100:]
```

```
[10, 9, 8, 7, 6, 5, 4, 3, 2, 1]
```

```
my_list[8:]
```

```
[2, 1]
```

```
# 外れたら無視してくれる
my_list[20:]
```

```
[]
```

ちょっとわき道　Ellipsis

```
Ellipsis
```

```
Ellipsis
```

```
...
```

```
Ellipsis
```

```
def mystery_func():
    Ellipsis
```

passがよく使われる

```
mystery_func()
```

```
bool(Ellipsis)
```

passは文、Ellipsisは式

```
True
```

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

https://www.kspub.co.jp/book/detail/5178034.html

強敵が控えていた・・・

- 2020年1月発売予定
- 拙著の前半部分を400ページに拡大して解説している感じか？
- 望洋先生の本で学生時代C++を学んだ身としてはむしろ光栄
- 動的計画法やラビン・ミラー素数判定法など知りたい方はぜひ私の本を！

https://www.sbcr.jp/product/4815603199/

3.8から使えるpowの新機能

```
pow(2, 3)
```

```
8
```

2の3乗なので8

```
pow(2, 3, 5)
```

```
3
```

2の3乗を5で割ったあまりなので3

```
pow(23, -1, 5)
```

```
2
```

23に掛けて5で割ると1余る数を返してくれる（new in 3.8）

つまり、23x≡1 (mod 5)を解いてくれる

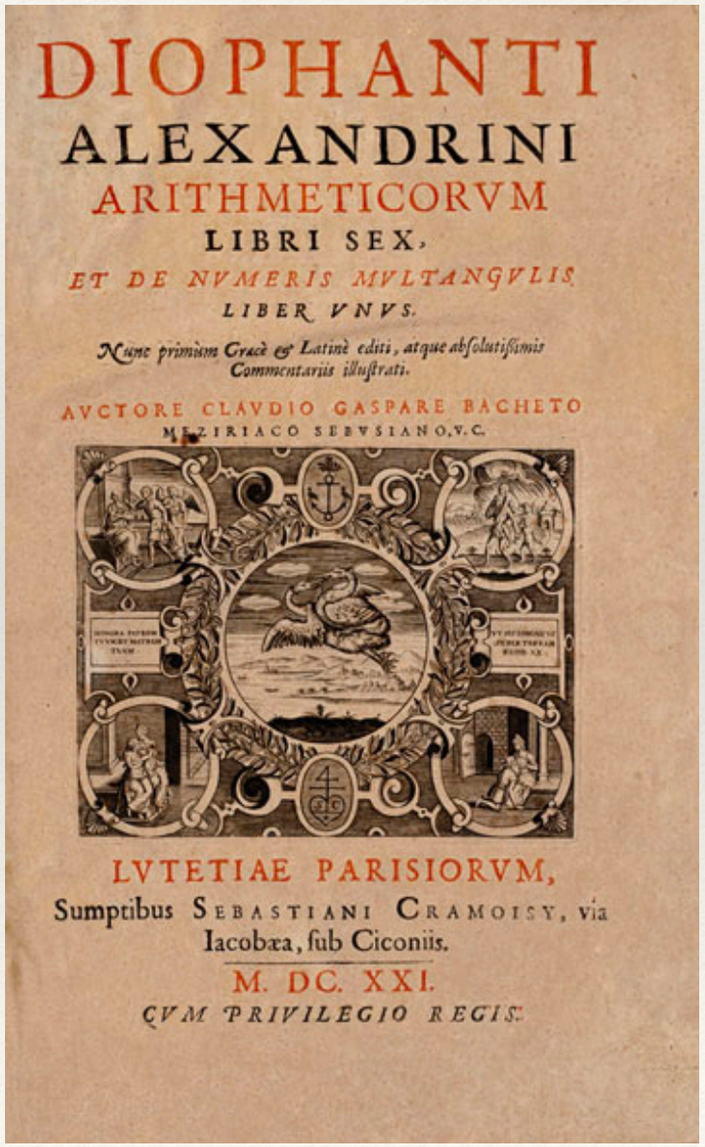
何の役に立つのか？それはもちろんディオファントス方程式を解くため。

Modular inverses arise in the solution of linear Diophantine equations. For example, to find integer solutions for $4258x + 147y = 369$, first rewrite as $4258x \equiv 369 \pmod{147}$ then solve:

```
>>> x = 369 * pow(4258, -1, 147) % 147
>>> y = (4258 * x - 369) // -147
>>> 4258 * x + 147 * y
369
```

ディオファントス

- 西暦200年代の数学者
- 「代数学の父」の異名を持つ
- 「算術」という古典が有名
  - フェルマーの書き込み
- 84歳まで生きたとされ、80歳から代数の研究をはじめた（らしい）



https://www.storyofmathematics.com/hellenistic_diophantus.html

https://upload.wikimedia.org/wikipedia/commons/6/60/Diophantus-cover.jpg

今年も1年ありがとうございました

2020年もはりきって行きましょー