みんなのPython勉強会#50

10/9, 2019

プログラミング言語Pythonのはなし

辻 真吾（@tsjshg）

おまえ誰よ？

- 東京大学先端科学技術研究センターに所属
- 分野横断的な応用データサイエンスを目指したい
- Pythonが便利過ぎて、ほとんど他の言語を忘れてしまった
- 最近ぬか漬けをはじめました

京都市宝ヶ池公園だった。

お知らせ

- プログラムは書かないと書けるようにならない
  - 節ごとに問題をつけました
- 書名を考えた！
- 懇親会でじゃんけん大会します
  - 2冊もってきましたので、ぜひ懇親会にご参加ください

もくじ

- 技術の進歩とPython
- エラーメッセージのありがたみ
- アルゴリズムとデータ構造の話

Python is eating the world:

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

By Nick Heath

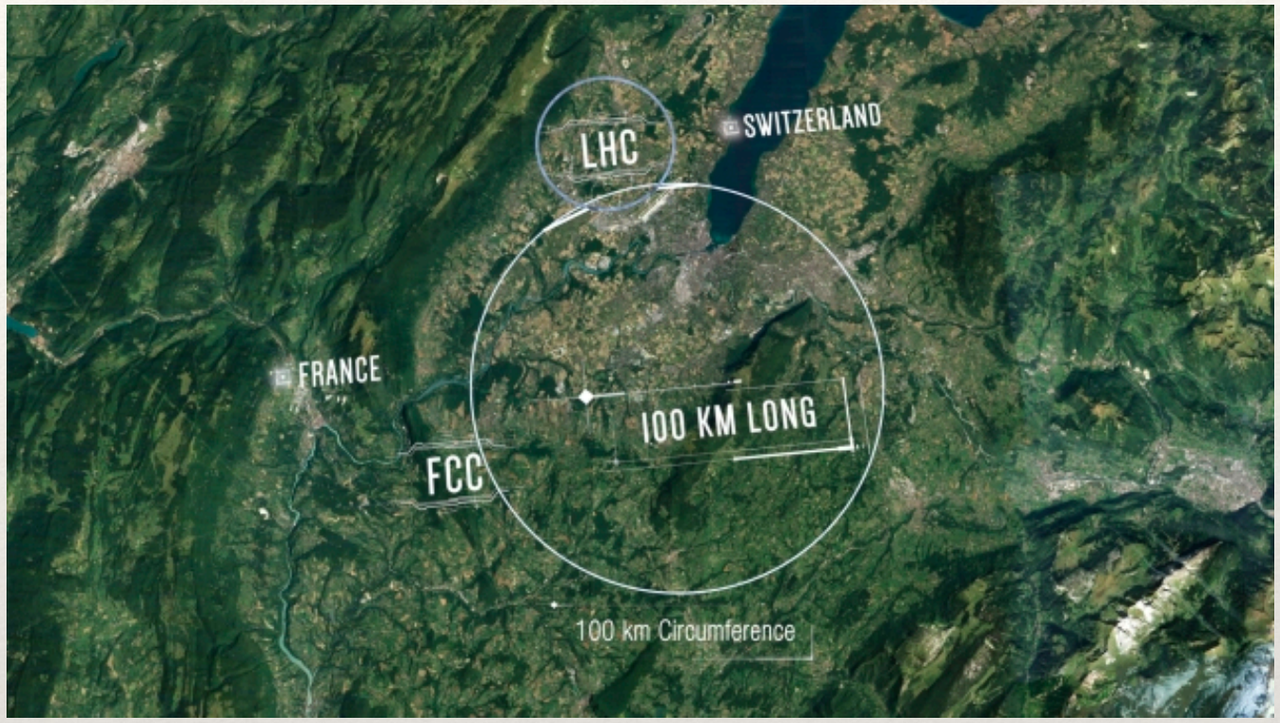
Image: Dan Stroud under the Creative Com

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

現代は科学技術の時代

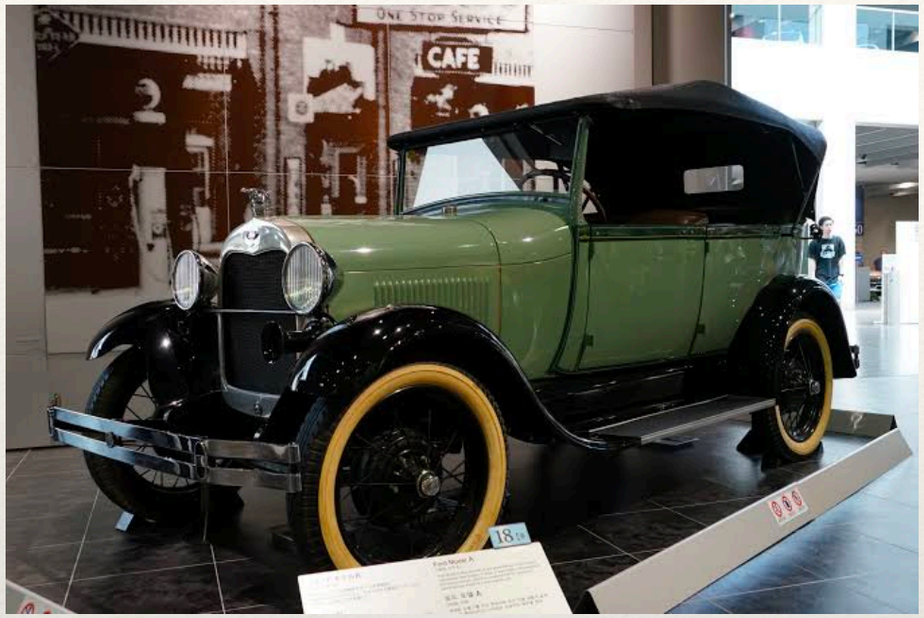
科学は進歩すると大型化

- CERN（欧州原子核研究機構）が計画する次世代加速器
  - 総工費1兆円以上！
- ヒトゲノム計画（2003年）
  - 約30億文字のDNA配列
  - 当時費やしたお金、数千億円
  - 今は10万以下（技術の進歩）



https://home.cern/news/press-release/accelerators/international-collaboration-publishes-concept-design-post-lhc

技術は進歩すると便利になる



https://time-space.kddi.com/digicul-column/bunka/20160614/

プログラミング言語も同じ

- コンピュータの性能が向上
  - 速いCPUと潤沢なメモリ
- C言語はツライ・・・
- 軽量言語（Lightweight programming Language）
  - Perl, Ruby, Python
- 技術の進歩でプログラミング言語がみんなのものに

2つのコマンドライン引数を足す　C言語

```c

    #include <stdio.h>
    #include <stdlib.h>

    int main(int argc, char *argv[])
    {
        int num;
        num = atoi(argv[1]) + atoi(argv[2]);
        printf("%d\n", num);
        return 0;
    }
```

実行例とエラー

2つ目の引数がないとき、segmentation faultとか言われて終わる・・・。

2つのコマンドライン引数を足す　Python

```python
import sys

print(int(sys.argv[1]) + int(sys.argv[2]))
```

```
20191009 — -zsh ▸ -zsh — 80×24
~/D/p/s/20191009 ❯❯❯ python test.py 2 4
6
~/D/p/s/20191009 ❯❯❯ python test.py 2
Traceback (most recent call last):
  File "test.py", line 4, in <module>
    print(int(sys.argv[1]) + int(sys.argv[2]))
IndexError: list index out of range
~/D/p/s/20191009 ❯❯❯
```

どこでどんなエラーが起きたか、詳細に報告してくれる！

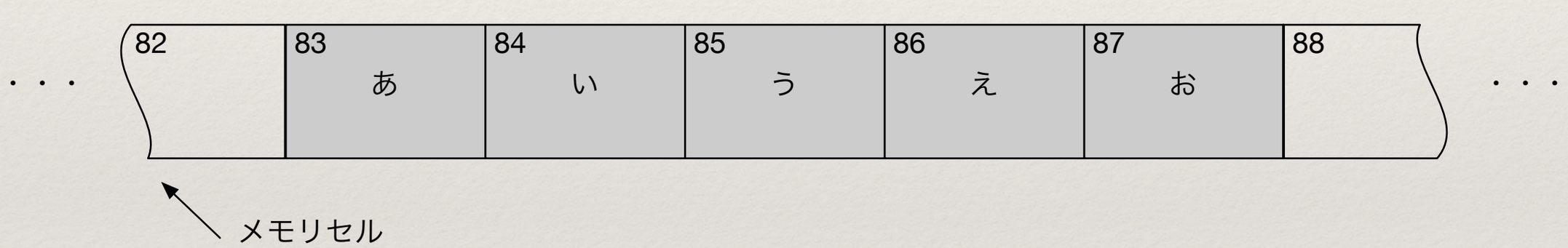
プログラミングの民主化

- 技術の進歩でプログラミング言語が便利に
- LL言語であれば気軽に入門できる
  - すべての人がいまはじめるべき（と思う）
- Pythonは世界的に優秀な開発者を惹き付けている気がする
  - データサイエンス、機械学習、Webアプリケーションフレームワークなど各分野で外部パッケージが充実

データ構造の話

Pythonのリストとcの配列

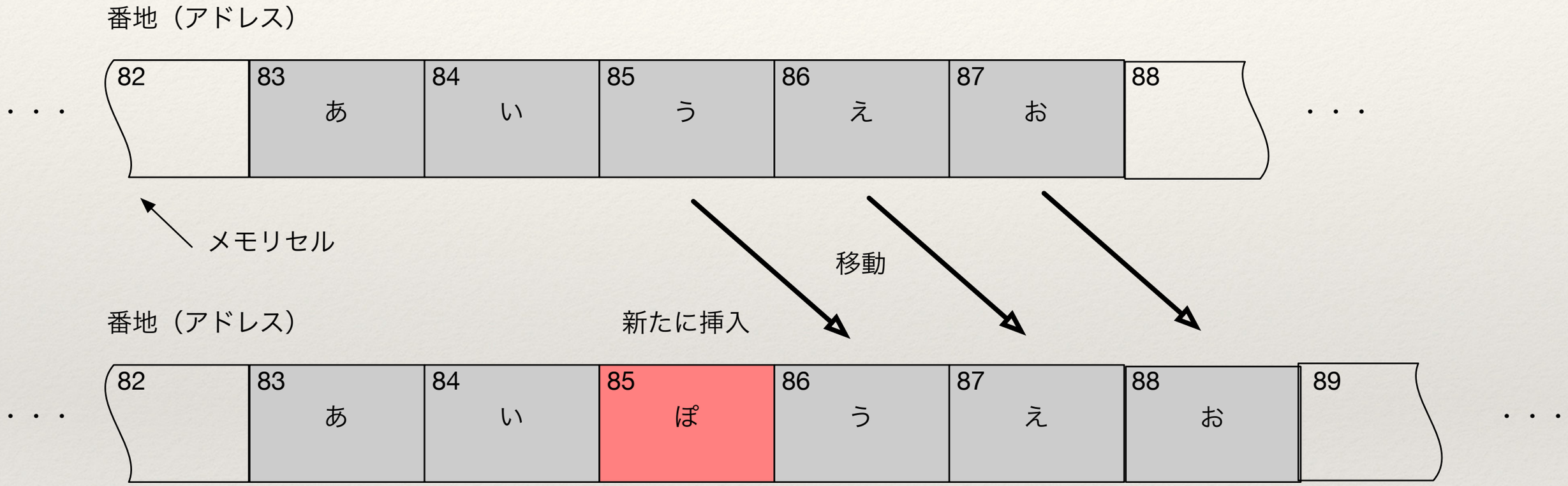
['あ', 'い', 'う', 'え', 'お']



いろいろと簡略化していますがポイントは

・メモリは同一サイズの細かい領域に分割されている
・配列はメモリ空間上の連続した領域（どこから始まって幾つ確保するか）

配列は挿入や削除に弱い

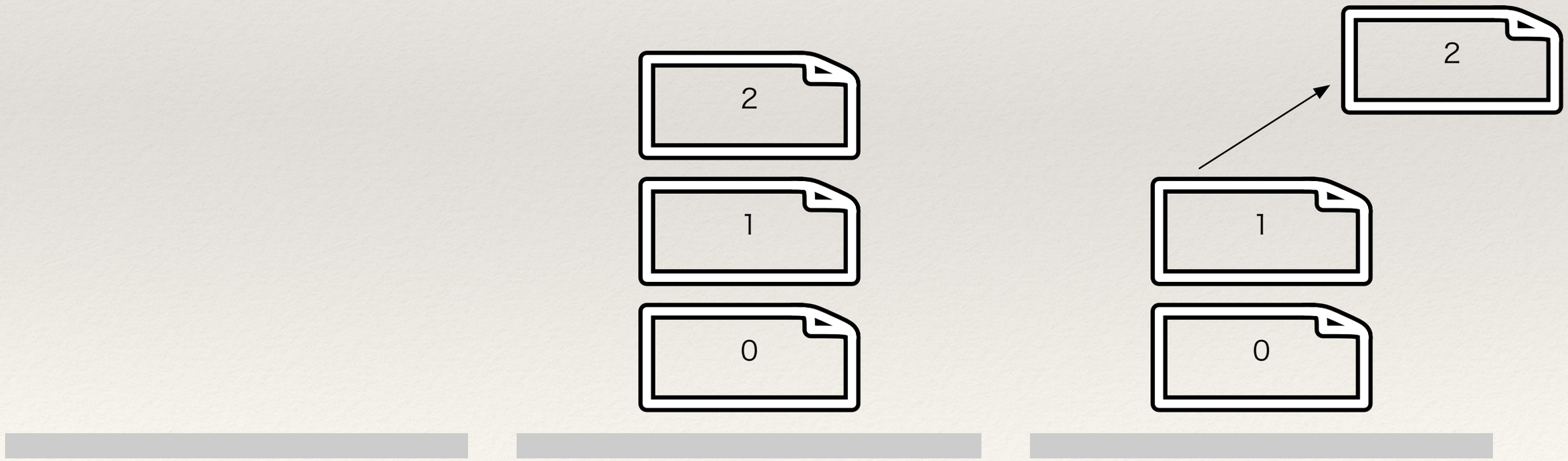


挿入（削除）された場所より後ろが全員引っ越し

一番後ろはいいけど、先頭やられたら被害甚大

スタック

- スタック（Stack）は積み上げ
  - LIFO（last in first out）
  - 後から入ったものが先に出る



（A）空のスタック　（B）データが溜まると積み上がる　（C）データは上から順に処理される

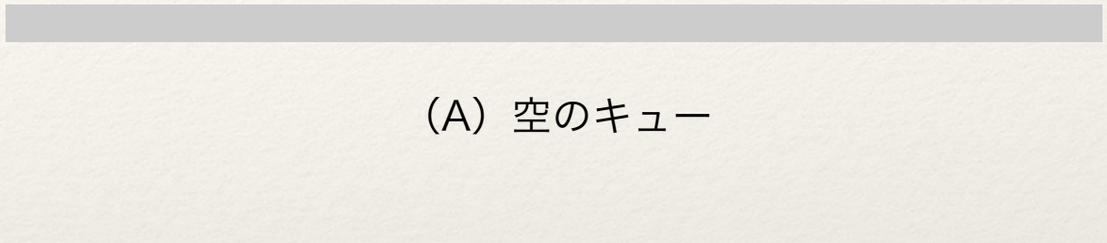
リストはそのままスタックとして使える

In [1]:
```
my_stack = []

# スタックに要素を追加
my_stack.append('あ')
my_stack.append('い')
my_stack.append('う')

# スタックから要素を取り出す
my_stack.pop()
```

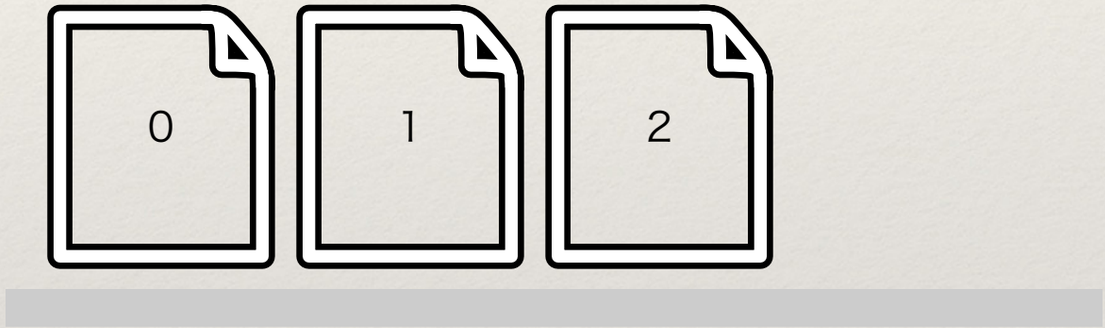
Out[1]: 'う'

キュー

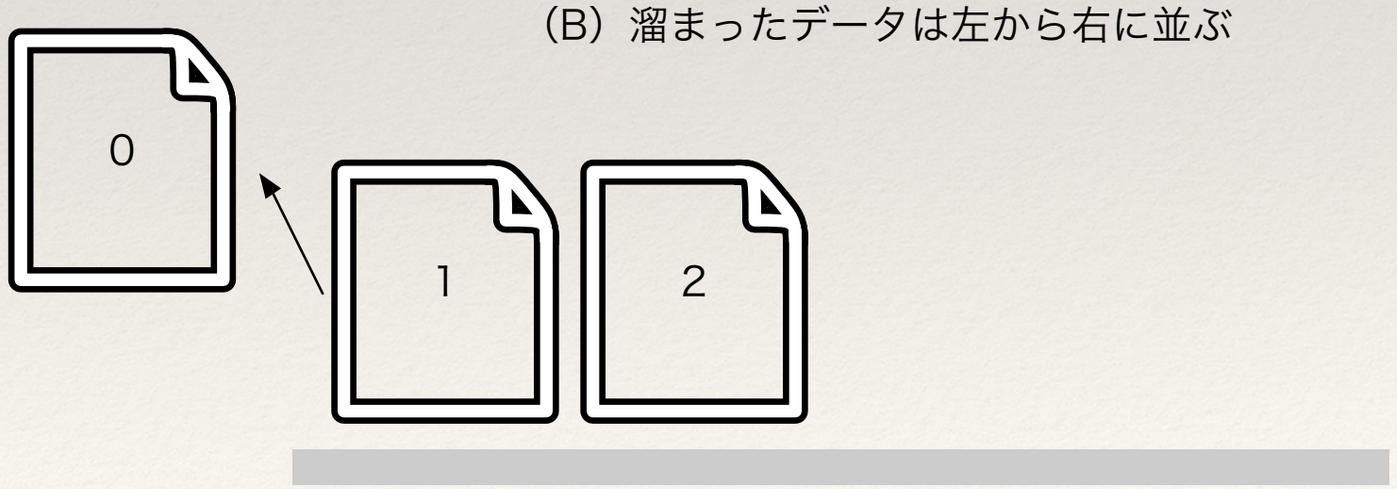
- キュー（Queue）は待ち行列
  - FIFO（first in fast out）
    - 先に入ったものが先に出る
  - スーパーのレジ



（A）空のキュー



（B）溜まったデータは左から右に並ぶ



（C）データは先頭（古い物）から順に処理される

リストをキューに（ダメ？）

In [2]:
```
my_queue = []

# キューに要素を追加
my_queue.append('あ')
my_queue.append('い')
my_queue.append('う')

# キューから要素を取り出す
my_queue.pop(0)
```

Out[2]: 'あ'

配列の先頭を削除する計算にコストがかかる。

collections.deque

In [4]:
```
from collections import deque
my_queue = deque()

# キューに要素を追加
my_queue.append('あ')
my_queue.append('い')
my_queue.append('う')

# キューから要素を取り出す
my_queue.popleft()
```

Out[4]: 'あ'

右側からデータを追加して、左側から取り出すキューの動きを実装

dequeはdouble ended queueの略なので、どちら方向からでも使える

プログラムが書けるとは？

だいたいのことはすぐできる

In [9]:
```
import random

my_list = [random.randint(1, 10) for i in range(20)]
my_list
```

Out[9]: [1, 1, 8, 2, 3, 2, 3, 8, 1, 5, 9, 9, 2, 9, 5, 3, 2, 6, 10, 2]

In [10]:
```
sorted(my_list)
```

Out[10]: [1, 1, 1, 2, 2, 2, 2, 2, 3, 3, 3, 5, 5, 6, 8, 8, 9, 9, 9, 10]

クイックソートの実装

```
def quick_sort(array):
    # 空の配列はそのまま返す
    if not array:
        return array
    # 最後の要素をpivotにする。
    p = array[-1]
    left = []
    right = []
    pivots = []
    # pivotとの関係で要素を分割する
    for v in array:
        if v < p:
            left.append(v)
        elif v == p:
            pivots.append(v)
        else:
            right.append(v)
    # 左と右は再び関数を適用して返す。
    return quick_sort(left) + pivots + quick_sort(right)
```

プログラミングのスキル

- ライブラリ全盛の時代
  - ほとんどのことはライブラリを呼べばできる
  - また、それだけでもかなりのことができる
- 自分で書けるようになるには？
- アルゴリズムの知識が必要
  - 条件分岐（if）と繰り返し（for）の組み合わせだけで問題を解くスキル

まとめ

- Pythonが世界を席巻し始めた
- 技術の進歩がプログラミングをみんなのものに
- エラーは友達
- データ構造とアルゴリズムが重要
  - Pythonはこれらを学ぶのにもよい環境を提供する

「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より11月26日発売予定！

書籍情報 | 購入方法 | ダウンロード | 会社概要 | 採用情報 | お問い合わせ

Home ＞ 書籍情報 ＞ データサイエンス入門シリーズ

書籍情報

検索

ジャンル

- 科学一般
- 科学英語
- 数学・情報科学
- 物理学
- 工学
- 化学
- 生物学・生物科学
- 医学・薬学・臨床検査・看護
- 生活科学
- 獣医学・畜産学・農学・水産学
- 地球環境科学
- 健康科学・スポーツ医科学
- 社会科学
- その他

» 実験ノート

シリーズ

- データサイエンス入門シリーズ
- 実践Data Scienceシリーズ
- イラストで学ぶシリーズ
- 栄養科学シリーズＮＥＸＴ
- エキスパート応用化学テキスト
- 絵でわかる

今、必要とされる人材を育てる。超注目シリーズ

データサイエンス入門シリーズ

［シリーズ編集委員］

竹村彰通（滋賀大学、編集委員長）
狩野裕（大阪大学）
駒木文保（東京大学）
清水昌平（滋賀大学）
下平英寿（京都大学）
西井龍映（長崎大学、九州大学名誉教授）
水田正弘（北海道大学）

［本シリーズの特徴］

- 「数理・データサイエンス教育強化拠点コンソーシアム」のスキルセットに依拠
- 具体的、体験的に学べる応用例、練習問題を収録
- フルカラーで見やすい構成

［近刊］（第2期：11月26日刊行予定）

『統計モデルと推測』
松井秀俊・小泉和之（著）竹村彰通（編）
A5・224頁・予価2400円・ISBN 978-4-06-517802-7

『Pythonで学ぶアルゴリズムとデータ構造』
辻真吾（著）下平英寿（編）
B5変・208頁・予価2400円・ISBN 978-4-06-517803-4

https://www.kspub.co.jp/book/series/S137.html

ご清聴ありがとうございました！