Python入門
アプリケーションを作る

アプリケーションを作る前に

- 何をしたいのか
  - 前提、手順、得られる結果
- どういう手順なのか
  - 手順を細かい手順に分解
  - ライブラリでは何ができるか
- どういうエラーがあり得るか

ファイル中の各単語の出現回数を数える

- ファイルを開く：open()
- アルファベット以外の文字で分割
  - 正規表現を利用
- 辞書構造を使って、単語と出現回数を登録

```
import re

def getWords(filename):
	with open(filename) as inputFile:#ファイルを開く
		text = inputFile.read()#ファイルからの読み込み
	#単語に分割：正規表現を利用
	return re.split('[¥s,.;:]',text)

def countwords(wordList):
	dic = {}
	count = 0
	for s in wordList:#単語の出現回数をカウント
		ss = s.lower()#小文字化
		count += 1
		if ss in dic.keys():
			n = dic[ss]+1
			dic[ss]=n
		else:
			dic[ss]=1
	return (dic,count)

filename = 'ConstitutionOfJapan.txt'
words = getWords(filename)
(dic,count) = countwords(words)
for k,v in sorted(dic.items()):#結果出力
	print(k+':'+str(v))
print(str(count)+'語')
```

CSVファイルの分析

- ライブラリcsvの利用
- csv.reader(file)で読み込み
  - reader型オブジェクト：一行毎に読み込む
  - 各行のカラムを指定して値読み出し
- 事故の内容や類型の辞書を使って、件数をカウント

```
import csv

#福岡県交通事故状況
def main(filename):
    with open(filename,'r') as f:
        reader = csv.reader(f)#CSVファイルを読む
        header = next(reader)#一行目を読み飛ばす：各カラムのラベル
        levels,cat = getDataFromCSV(reader)
    return levels,cat

def getDataFromCSV(reader):
    levels={}
    cat={}
    for row in reader:
        sv = row[3]#4カラム目：事故内容
        count(sv,levels)

        c = row[10]#11カラム目：事故類型
        count(c,cat)
    return levels,cat
```

```
def count(k,data):#件数カウント
    if k in data.keys():
        n = data[k] + 1
        data[k] = n
    else:
        data[k] = 1

levelsSum,cat =main('h29.1-12.csv')
for l in levelsSum:
    print(l+' '+str(levelsSum[l])+' 件')
print()
for c in cat:
    print(c+' '+str(cat[c])+' 件')
```

Pandasを使ってxlsxファイルを読む

- 表のように構造化されたデータを扱う外部ライブラリ
  - DataFrameという二次元データ構造
    - ラベルでデータを取り出せる
  - Seriesという一次元データ構造
- データ分析でよくつかわれる

```
import pandas

def main(filename):
    with pandas.ExcelFile(filename) as f:#エクセルを開く
        data = pandas.read_excel(f)
        levels,cat = getDataFromSheet(data)
    return levels,cat

def getDataFromSheet(dataFrame):#データ分析
    levels={}
    cat={}
    for sv in dataFrame['事故内容']:
        count(sv,levels)

    for c in dataFrame['事故類型']:
        count(c,cat)
    return levels,cat
```

```
def count(k,data):
    if k in data.keys():
        n = data[k] + 1
        data[k] = n
    else:
        data[k] = 1

filename='h29.1-12.xlsx'
levelsSum,cat=main(filename)
for l in levelsSum:
    print(l+' '+str(levelsSum[l])+' 件')
print()
for c in cat:
    print(c+' '+str(cat[c])+' 件')
```

ネットワーク上にあるXLSXファイルを読む

- URLを開く
  - ライブラリurllibを利用
  - urllib.request.urlopen(url)でurlにあるファイルを取得できる。
- XLSXファイルを読む
  - 外部ライブライpandasを利用
  - google colaboratoryではインストール不要

URL上のxlsxを読み
集計結果をxlsxへ保存

- pandas.DataFrame構造のデータを作成
- エクセルファイルへ保存

```
import pandas
import urllib.request

def main(url):
    with urllib.request.urlopen(url) as f:
        data = pandas.read_excel(f)
        outdata = getDataFromSheet(data)
    return outdata

def getDataFromSheet(dataFrame):
    outdata = pandas.DataFrame({'車両相互':[0,0,0],
                                '人対車両':[0,0,0],
                                '車両単独':[0,0,0],
                                '列車':[0,0,0]},
                                index=['軽傷','重傷','死亡'])
    for index,ser in dataFrame.iterrows():
        c = ser.get('事故内容')
        s = ser.get('事故類型')
        v = outdata[s][c]
        v += 1
        outdata[s][c] = v

    return outdata
```

```
url='https://ckan.open-governmentdata.org/dataset/'¥
+'b1b1dbba-12b9-4a30-a80c-2d8abec5598c/resource/'¥
+'c0362010-afe5-41e3-a6e7-492d80917de7/download/h29.1-12.xlsx'
outdata = main(url)
outdata.to_excel('output.xlsx')
```

時系列データをプロット

- matplotlibライブラリで作図できる
  - 日本語表示は面倒
- pandas.Seriesを活用
- 佐賀県の人口推移をプロット

```
import pandas
import matplotlib.pyplot as plt

#エクセル中の和暦ラベルと西暦の対応付け
wareki={'大正9年':1920, '昭和5年':1930, '10年':1935, '15年':1940,
        '20年':1945, '25年':1950, '30年':1955, '35年':1960,'40年':1965,
        '45年':1970, '50年':1975, '55年':1980, '60年':1985, '平成2年':1990,
        '7年':1995, '12年':2000, '17年':2005, '22年':2010,'27年':2014}

def main(filename):
    data = pandas.read_excel(url,header=3)
    evalPop(data,2).plot(label='Total')#全体
    evalPop(data,3).plot(label='Cities')#市部
    evalPop(data,4).plot(label='Counties')#郡部
    plt.title('Population in Saga Prefecture')#図タイトル
    plt.xlabel('year')
    plt.ylabel('population (thousand)')
    plt.legend(loc='best')#凡例
    plt.savefig('output.png')#画像保存
    plt.show()#画像表示
```

```
def evalPop(data,j):
    vList=[]
    iList=[]
    for k in wareki.keys():
        p = data[k][j]
        i = wareki[k]
        vList.append(p)
        iList.append(i)
    return pandas.Series(vList,index = iList)


url = 'http://data.bodik.jp/dataset/'¥
        '77e0cc66-c15d-4473-b3df-2664fe8e2e63/resource/'¥
        '8dc71515-526a-4168-866c-05d2cc8dad7b/download/jinkou.xlsx'
outdata = main(url)
```


Population in Saga Prefecture
Total
Cities
Counties
population (thousand)
1000
800
600
400
200
0
1920
1940
1960
1980
2000
year

発展

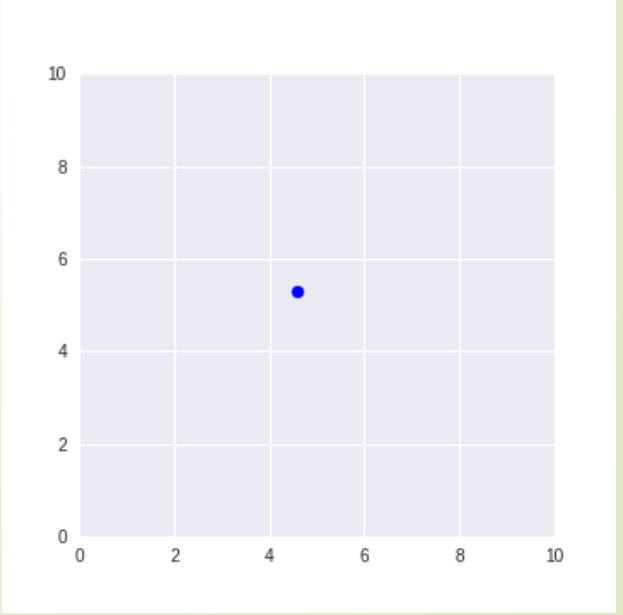
- 佐賀県の人口推移
  - 2行目：全県
  - 3行目：市部
  - 4行目：郡部
- 別の行をやってみよう：年齢構成の推移
  - 15行目：15歳未満
  - 16行目：15歳以上、65歳未満
  - 17行目：65歳以上

アニメーション

- 二つの方法
  - 画像のリストを順に表示
  - 画像を作成する関数を繰り返し呼び出し

例：酔歩

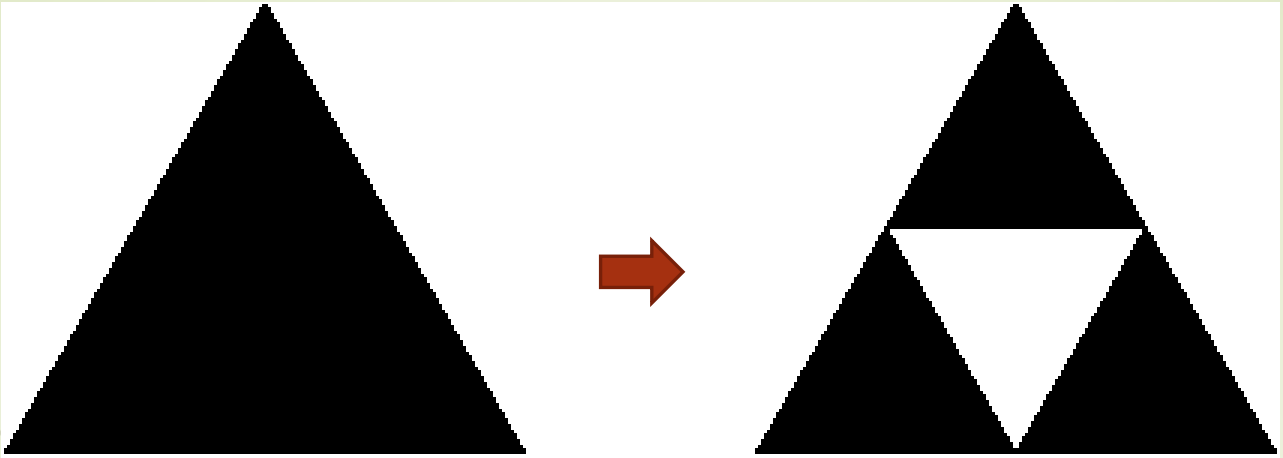
- 粒子が位置(x, y)に居るとき、次の位置をランダムに選ぶ
  - $(x + a, y + b), -0.5 \leq a, b > 0.5$



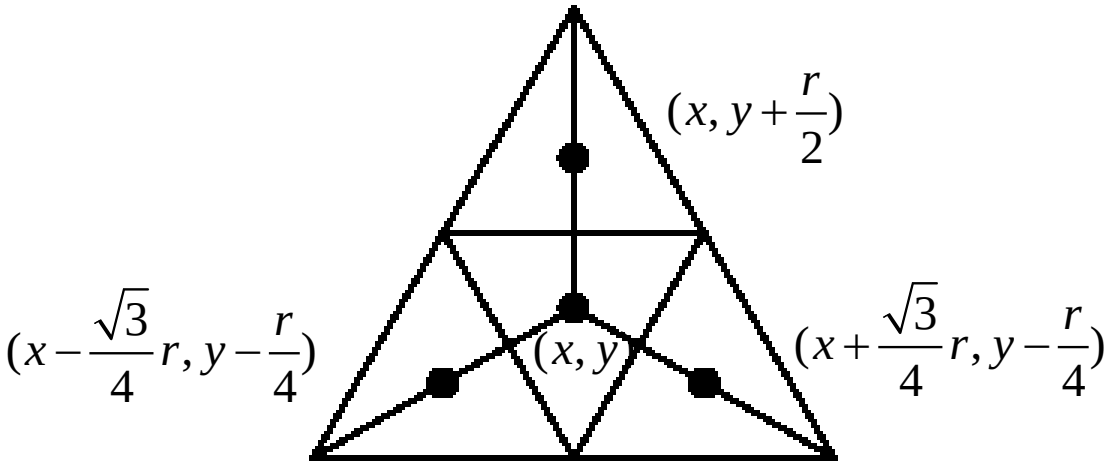
```
import random
import matplotlib.pyplot as plt
from matplotlib.animation import ArtistAnimation
from matplotlib import rc

def oneStep():
  global p
  x = (p[0]+r*(random.random()-0.5)+L)%L
  y = (p[1]+r*(random.random()-0.5)+L)%L
  im = ax.scatter(x,y,c='blue')
  p=(x,y)
  artists.append([im])

L=10
r = 1
p = (L/2.,L/2.)
artists = []
fig, ax = plt.subplots(figsize=(5,5))
ax.set_xlim(0,L)
ax.set_ylim(0,L)
tm = 500
for i in range(tm):
  oneStep()
anim = ArtistAnimation(fig,artists,interval=100)
rc('animation',html='jshtml')
anim
```



再帰的に分割を繰り返す
・サイズを半分にする
・中心をずらす



```
import math
import matplotlib.pyplot as plt
import matplotlib.patches as pts
import matplotlib.animation as animation

#一つの三角形を塗りつぶす
def drawTriangle(center,radius):
    return pts.RegularPolygon(center,3,radius,fill=True,color='blue')

#再帰的に三角形を生成
def oneStep(center,radius,iter,ax):
    if iter == 0:
        ax.add_patch(drawTriangle(center,radius))
        return
    #サイズが1/2の三角形へ分割
    r = radius/2.
    p0 = (center[0]-math.sqrt(3)*r/2,center[1]-r/2)
    oneStep(p0,r,iter-1,ax)
    p1 = (center[0]+math.sqrt(3)*r/2,center[1]-r/2)
    oneStep(p1,r,iter-1,ax)
    p2 = (center[0],center[1]+r)
    oneStep(p2,r,iter-1,ax)
```

```
fig = plt.figure(figsize=(10,10))
plt.xlim(-10,10)
plt.ylim(-10,10)
r = 10
center = (0,-1)
imgs=[]
tm = 8
for i in range(tm):
    ax = fig.subplots()
    ax.set_xlim(-10,10)
    ax.set_ylim(-10,10)
    oneStep(center,r,i,ax)
    im = ax.get_children()
    imgs.append(im)
ani = animation.ArtistAnimation(fig,imgs,interval=1000)
rc('animation',html='jshtml')
ani
```

HTMLから必要な情報を取り出す

```
import requests
import bs4

url = 'https://www.cc.saga-u.ac.jp'
with requests.get(url,timeout=3) as f:
  f.encoding='utf-8'
  soup = bs4.BeautifulSoup(f.text)
elements = soup.select('#securityList li')
for entry in elements:
  print(entry.text)
```