

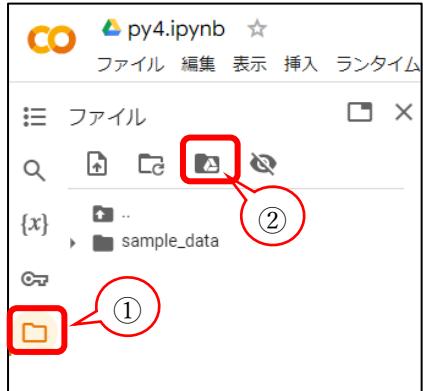
プログラミング演習3(Python)⑤ 画像処理ライブラリ

【Python を使った画像処理】

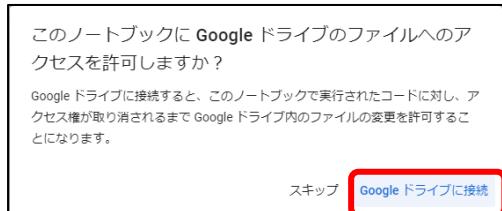
1. GoogleColaboratory を使って画像を表示する

1-1. Google ドライブを接続する

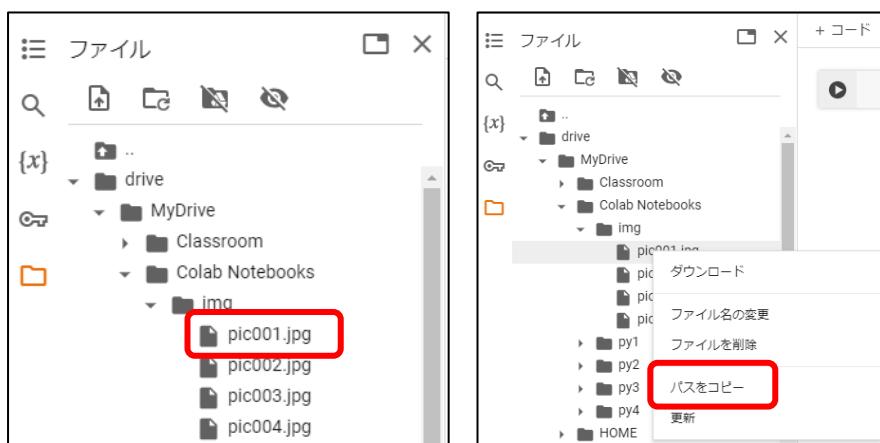
①ファイルを選択して、②ドライブをマウントを押す。



Google ドライブに接続を押す



1-2 Google ドライブに用意した画像ファイルを確認して、右クリックしてパスをコピーを選択する。



1-3 Google Colab で下記のように入力する

「Image.open(“ ”)の後ろは先ほどコピーした画像のパスを張り付けている。

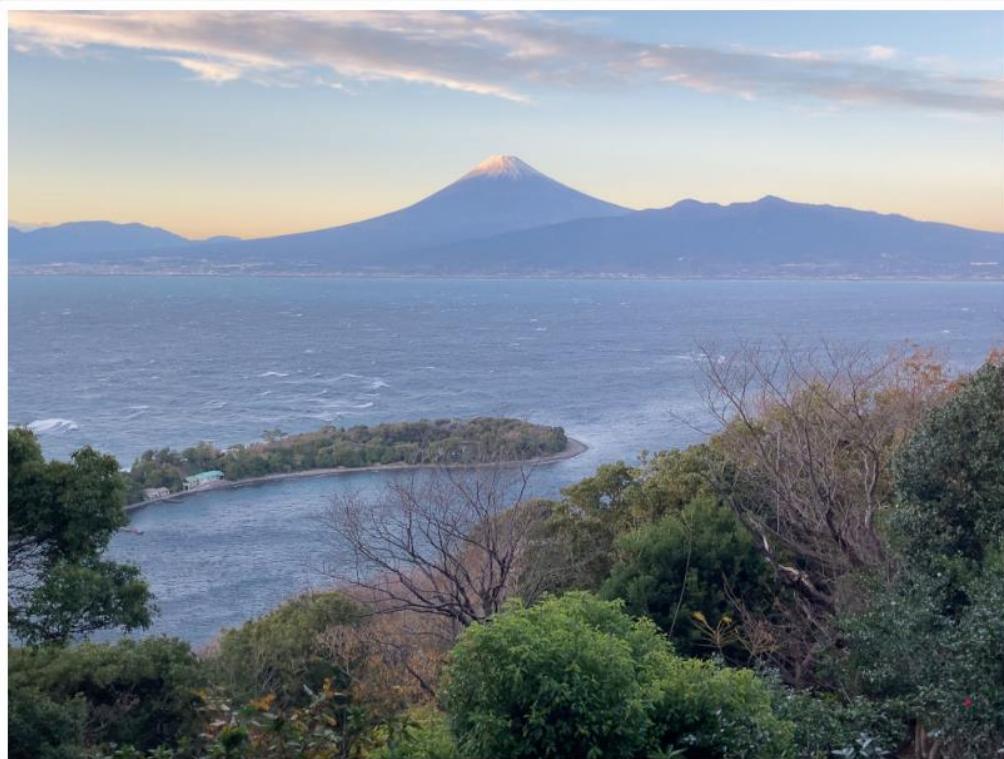
```
▶ from PIL import Image
from IPython.display import display

# 画像を読み込む
img = Image.open("/content/drive/MyDrive/Colab Notebooks/img/pic001.jpg")

# 画像を表示
display(img)
```

1-4 実行する

Shift+Enter を押すと pic001.jpg が表示される



参考

<https://invisiblepotato.com/google-colaboratory01/>

2-1 画像をリサイズする

Image オブジェクト.thumbnail((幅,高さ))
でリサイズできる。

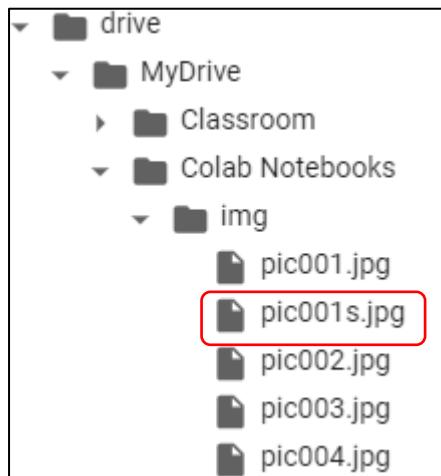
引数「幅」にはリサイズ後の幅、引数「高さ」にはリサイズ後の高さの値をピクセル単位で指定する。縦横比を固定した状態で、指定した値に収まるようにリサイズされる。

引数には(幅,高さ)という、値が「,」で区切られたタプルの形式で入れるため、かっこを入る形になる。

例

```
▶ from PIL import Image  
  
img = Image.open('/content/drive/MyDrive/Colab Notebooks/img/pic001.jpg')  
img.thumbnail((300, 200))  
img.save('/content/drive/MyDrive/Colab Notebooks/img/pic001s.jpg')
```

ファイル「pic001s.jpg」が生成される



2-2. 生成した画像を表示させて、img.size で画像サイズを確認する

```
▶ img = Image.open("/content/drive/MyDrive/Colab Notebooks/img/pic001s.jpg")
display(img)
img.size
```



(267, 200)

(1000, 750)から(267,200)になっており、高さに合わせて小さくなっている。なお、幅に合わせると(300,225)になり、少し大きな画像になる。

同様に、縦長の写真 002.jpgをリサイズして、サイズを表示させると

```
▶ img = Image.open("/content/drive/MyDrive/Colab Notebooks/img/pic002s.jpg")
display(img)
img.size
```



(150, 200)

(150, 200)になり、高さに合わせていることがわかる。

2-3. 画像のファイルサイズを取得する

ファイル容量を取得する関数は `os.path.getsize`
os モジュールの関数で `import os` を宣言する必要がある

書式は

`os.path.getsize(ファイル名)`

となり、ファイル名は `PIL.Image.open` と同じ要領で指定する

実行すると、ファイルの容量が数値として返される。

それを `print` で表示させる

```
[1] import os
print(os.path.getsize('/content/drive/MyDrive/Colab Notebooks/img/pic001.jpg'))
170323
```

デスクトップにファイルをダウンロードして、右クリックしてプロパティで確認する



170323バイトであることが確認できる

2-4 条件式を書く。ここではファイルサイズ 200KB だったら、リサイズする

条件式の IF 文は

If 条件式1:

 処理1

elif: 条件式2

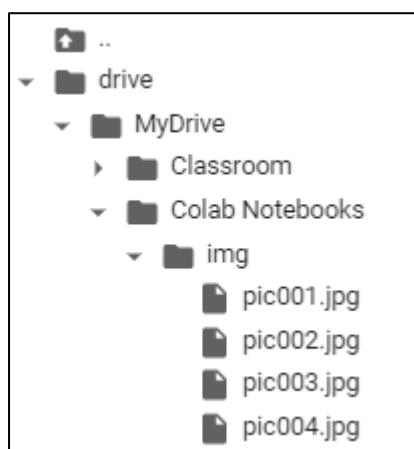
 処理2

else:

 不成立時の処理

と書く。elif 以降は必要がないなら省略する

条件式は



```
▶ import os
  print(os.path.getsize('/content/drive/MyDrive/Colab Notebooks/img/pic001.jpg'))
  import os
  print(os.path.getsize('/content/drive/MyDrive/Colab Notebooks/img/pic002.jpg'))

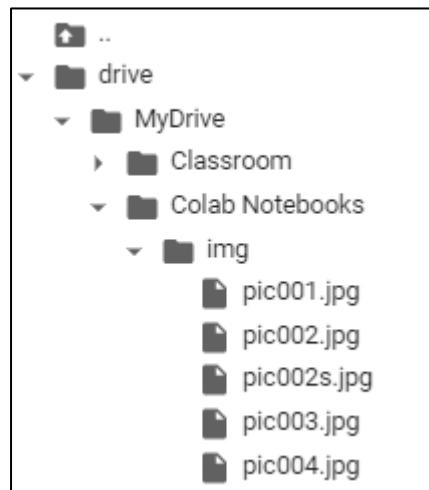
⇒ 170323
  310113
```

pic001.jpg は 170323 バイトで、pic002.jpg は 310113 バイトであることが分かる。この時、200KB 以上の pic002.jpg がリサイズできるようにする

```
[8] import os
    from PIL import Image

    if os.path.getsize('/content/drive/MyDrive/Colab Notebooks/img/pic001.jpg')>=204800:
        img=Image.open('/content/drive/MyDrive/Colab Notebooks/img/pic001.jpg')
        img.thumbnail((300, 200))
        img.save('/content/drive/MyDrive/Colab Notebooks/img/pic001s.jpg')
    if os.path.getsize('/content/drive/MyDrive/Colab Notebooks/img/pic002.jpg')>=204800:
        img=Image.open('/content/drive/MyDrive/Colab Notebooks/img/pic002.jpg')
        img.thumbnail((300, 200))
        img.save('/content/drive/MyDrive/Colab Notebooks/img/pic002s.jpg')
```

実行すると



pic002.jpg の方だけファイルが生成されている。

2-5 ファイル一覧を取得する

os.listdir 関数を使うと、ファイルの一覧が取得できる

os.listdir(フォルダ名)

ファイル名を取得するのと同じ要領でフォルダのパスをコピーする

例

```
▶ import os
  print(os.listdir('/content/drive/MyDrive/Colab Notebooks/img'))
  ['pic001.jpg', 'pic002.jpg', 'pic003.jpg', 'pic004.jpg']
```

[]で囲まれ[,]で分けられたリスト形式で表示される

2-6 要素番号を指定して取得する

filename にファイル一覧を入れ、要素番号2の3番目のファイル名を表示させる

例

```
[▶] import os
  filename = os.listdir('/content/drive/MyDrive/Colab Notebooks/img')
  print(filename[2])
  ↗ pic003.jpg
```

2-7 for文で一覧を表示させる

例

```
import os

filename = os.listdir('/content/drive/MyDrive/Colab Notebooks/img')

for i in filename:
    print(i)

pic001.jpg
pic002.jpg
pic003.jpg
pic004.jpg
```

2-8 画像のパスの一覧を作る

画像ファイルを指定するために、フォルダのパスとつなげ合わせる必要がある。

書式は

os.path.join('フォルダのパス',ファイル名)

とする。

なので、前のプログラムを書き替えて、

```
import os

filename = os.listdir('/content/drive/MyDrive/Colab Notebooks/img')

for i in filename:
    print(os.path.join('/content/drive/MyDrive/Colab Notebooks/img',i))

→ /content/drive/MyDrive/Colab Notebooks/img/pic001.jpg
   /content/drive/MyDrive/Colab Notebooks/img/pic002.jpg
   /content/drive/MyDrive/Colab Notebooks/img/pic003.jpg
   /content/drive/MyDrive/Colab Notebooks/img/pic004.jpg
```

とする

2-9 リサイズの操作をすべての画像に対して行う

1つのファイルの処理が下記のプログラムでできるので、

```
[8] import os
    from PIL import Image

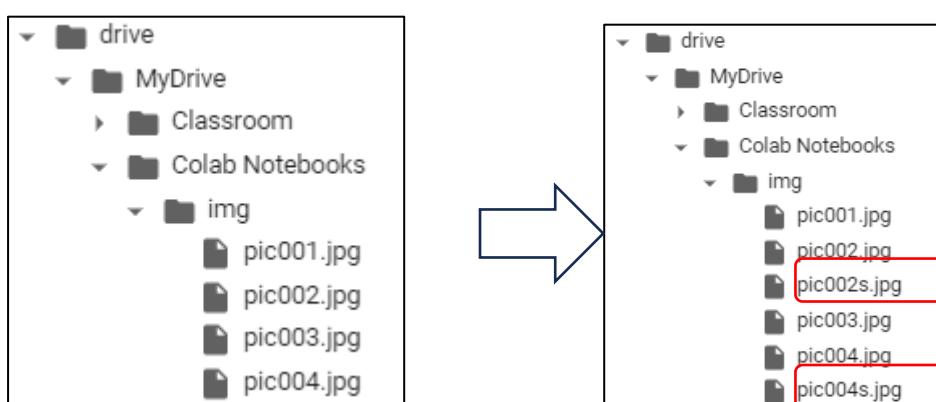
    if os.path.getsize('/content/drive/MyDrive/Colab Notebooks/img/pic001.jpg')>=204800:
        img=Image.open('/content/drive/MyDrive/Colab Notebooks/img/pic001.jpg')
        img.thumbnail((300, 200))
        img.save('/content/drive/MyDrive/Colab Notebooks/img/pic001s.jpg')
```

2-8のprintの代わりにリサイズをする処理をすればいいのだから、画像のパスを変数jに入れた後、画像ファイルと拡張子を分けるos.path.splitext()を使って、ファイル名と拡張子の間にsを付けて保存するようにすると下記のようになる。

```
[8] import os
    from PIL import Image
    filename = os.listdir('/content/drive/MyDrive/Colab Notebooks/img')

    for i in filename:
        j=os.path.join('/content/drive/MyDrive/Colab Notebooks/img',i)
        root, ext = os.path.splitext(j)
        if os.path.getsize(j)>=204800:
            img=Image.open(j)
            img.thumbnail((300, 200))
            img.save(root+ 's' + ext)
```

実行すると、200KB 以上の pic002.jpg と pic004.jpg が変換される



2-10 設定の値を前の方で変更できるようにする

```
import os
from PIL import Image

imgDir='/content/drive/MyDrive/Colab Notebooks/img'
filename = os.listdir(imgDir)
appStr='s'
maxSize=204800
maxX=300
maxY=200

for i in filename:
    j=os.path.join(imgDir,i)
    root, ext = os.path.splitext(j)
    if os.path.getsize(j)>=maxSize:
        img=Image.open(j)
        img.thumbnail((maxX, maxY))
        img.save(root+ appStr + ext)
```

■参考 立山 秀利著 「図形！Python のツボとコツがゼッタイにわかる本「プログラミング実践編」」(秀和システム)

■参考 BIZ CODE 【業務効率化】Python で画像サイズとファイル名を一括変更・連番付ける方法

<https://www.bizzcode.net/%E6%A5%AD%E5%8B%99%E6%94%B9%E5%96%84python%E3%81%A7%E7%94%BB%E5%83%8F%E3%82%B5%E3%82%A4%E3%82%BA%E3%81%A8%E3%83%95%E3%82%A1%E3%82%A4%E3%83%AB%E5%90%8D%E3%82%92%E4%B8%80%E6%8B%AC/>