

この文書はEclairを試用したい人向けのサンプルプログラム、`reduction.ipynb`と`fitsget.ipynb`の使用法、使用する際の注意点などを記したものです。

Eclair:

天体画像処理をGPU上で高速に行うためのPythonモジュール

`reduction.ipynb`:

MITSuMEの画像の一次処理をEclairを使用してGoogle Colaboratory上で実装したサンプルプログラム

`fitsget.ipynb`:

`reduction.ipynb`で使用するサンプルデータをGoogle Driveへダウンロードするためのプログラム

目次

基本事項

最初は必ず読むべし

1. Google Colaboratory
2. 事前準備
3. 基本操作
4. 全体の流れ
5. Google Driveのマウント

追記事項

必要に応じて読むべし

6. Drive上のパス
7. 枚数を増やしたい場合
8. Colaboratoryの実行環境

基本事項

最初は必ず読むべし

1. Google Colaboratory

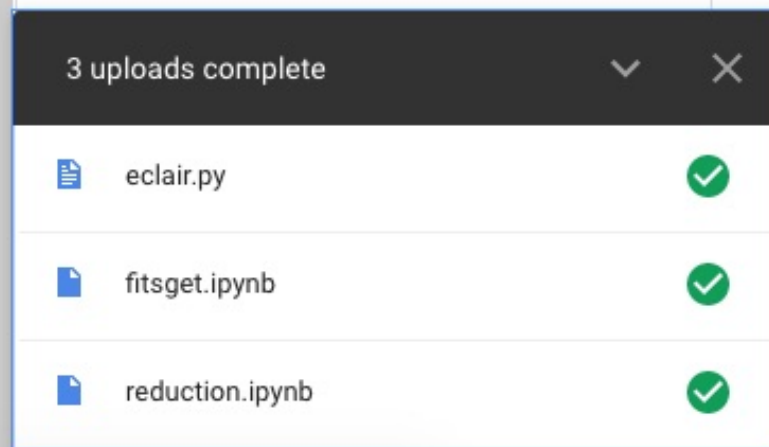
- クラウド上で実行されるJupyter Notebook環境
- **無料**
- GPUとそれに関するPythonライブラリを使用できる



2. 事前準備

eclair.py, fitsget.ipynb, reduction.ipynbの3つのファイルを github(<https://github.com/MNiwano/Eclair>)より入手し、Google Driveにアップロードする

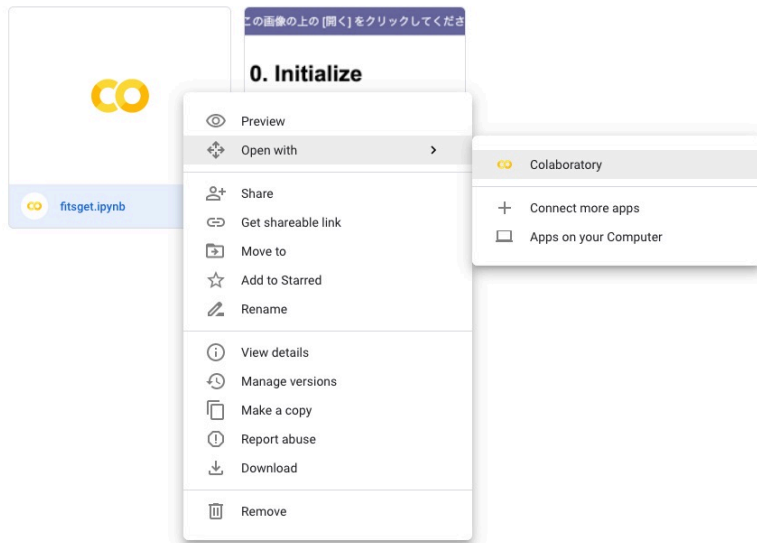
※ アップロード先ディレクトリはどこでもOK



3. 基本操作

• ipynbの開き方

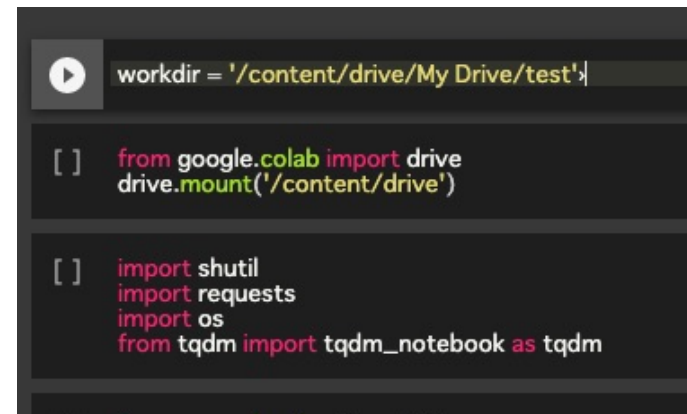
Google Drive上でファイルを右(左)クリックしメニューを表示させ、Open withからColaboratoryを選ぶ。



• コードの実行方法

コードセルを選択した状態でShift+Enter(Return)か左のボタンをクリック

※ 一部セルは実行に時間がかかります



4. 全体の流れ

1. fitsget.ipynbを実行する

- Google Drive上に作業ディレクトリが作成され、サンプルデータ(FITS画像13枚)がそこに保存される。

2. reduction.ipynbを実行する

- Eclairの機能を利用してサンプルデータの一次処理が行われる。

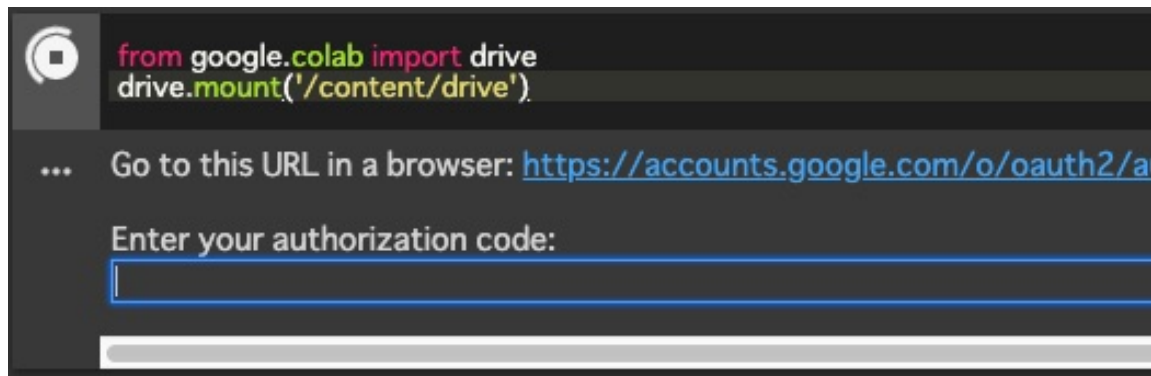
これらプログラムが動作するには、ColaboratoryからGoogle Drive上のデータにアクセスできなければならず、このためにGoogle Driveのマウントを行う。

5.1. Google Driveのマウント(1)

1. fitsget.ipynb, reduction.ipynbの中には
このようなコードセルがあり、実行すると

```
[ ] from google.colab import drive  
    drive.mount('/content/drive')
```

以下のような出力が表示されるので、
表示されたURLのリンク先を開く



5.2. Google Driveのマウント(2)

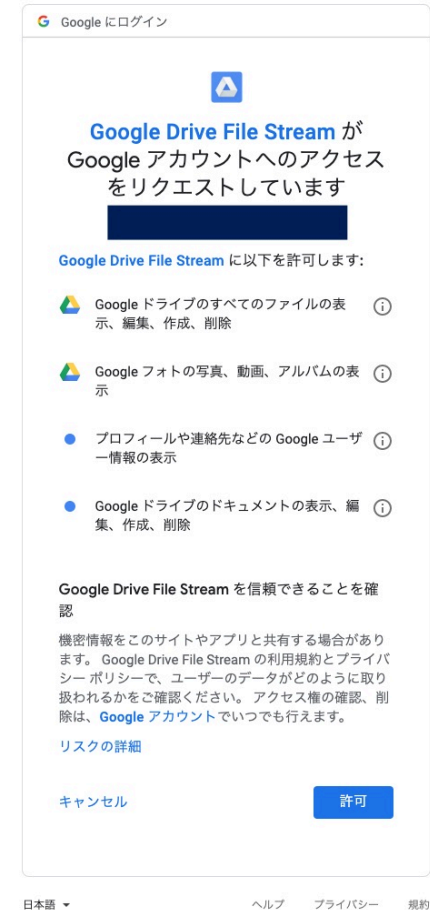
2. Driveを使用するGoogleアカウントを選択する



5.3. Google Driveのマウント(3)

3. Google Drive File Streamの Googleアカウントへのアクセス を許可する

※ 許可しなければ、ColaboratoryからDrive上の
データにアクセスできません



5.4. Google Driveのマウント(4)

4. 表示されたコード
をクリップボードへ
コピーする



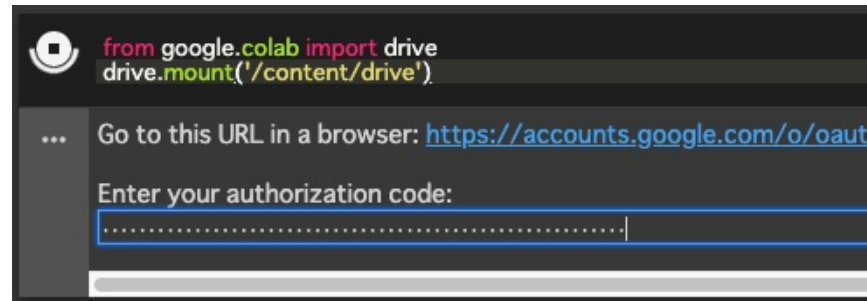
ログイン

このコードをコピーし、アプリケーションに切り替えて貼り付けてください。

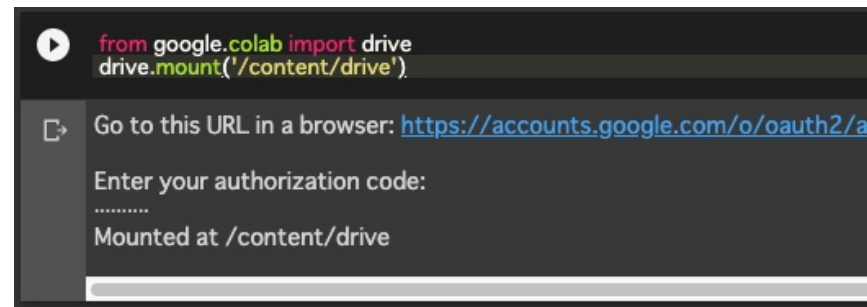
4/YAFPW3FdS8PMkCOAK69NyrvEfyWPAXtxWIV2m-SxfUs9pZ-AQxvtKbU 

5.5. Google Driveのマウント(5)

5. コピーしたコードを入力フォームにペーストしてEnter(Return)



出力が以下のようなになればマウントは完了
以降のセルを実行します



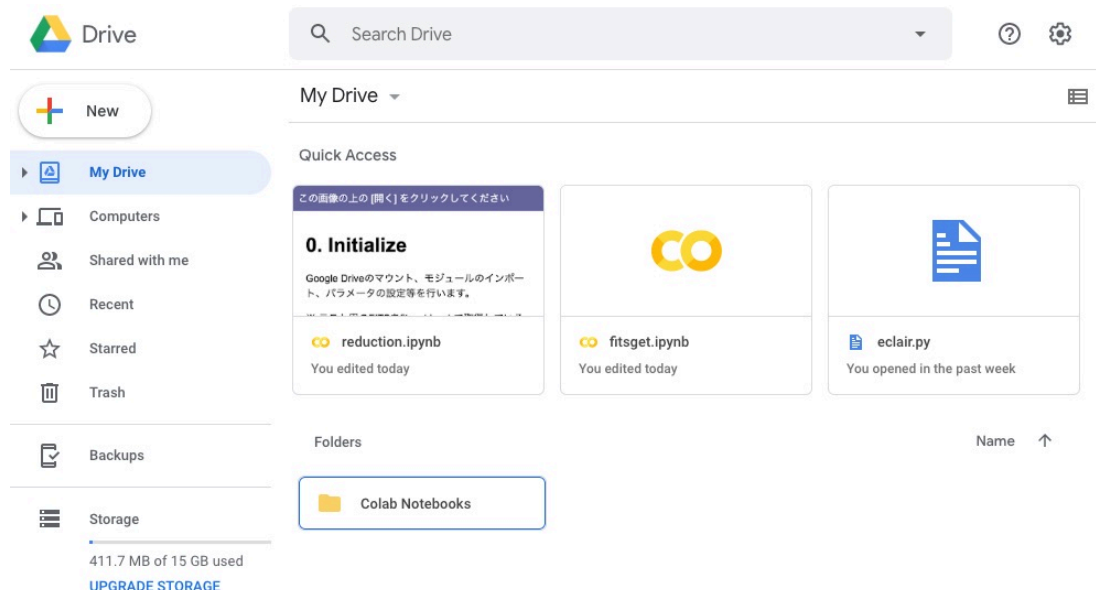
追記事項

必要に応じて読むべし

6. Drive上のパス

- My Drive or マイドライブが/content/drive/My Driveに対応する

※ Colaboratoryではpwd、cdといったUNIXコマンドが使用できる



7. 枚数を増やしたい場合

変数**fitslist**で読み込む**FITS**を指定しているので、これを整数倍すれば、擬似的に枚数が多い状況を再現できる。

※ 増やしすぎると**VRAM**がオーバーフローするので注意

```
[ ] fitslist = ['raw%02d.fits'%i for i in range(10)]
# 枚数が多い場合のベンチマークは以下のように擬似的に枚数を増やして行います
# fitslist *= 10
# 増やしすぎるとVRAMがオーバーフローします

dark = 'dark.fits'
flat = 'flat.fits'
bpmask = 'bpmask.fits'

output = 'combine.fits'
```

```
[ ] fitslist = ['raw%02d.fits'%i for i in range(10)]
# 枚数が多い場合のベンチマークは以下のように擬似的に枚数を増やして行います
fitslist *= 10
# 増やしすぎるとVRAMがオーバーフローします

dark = 'dark.fits'
flat = 'flat.fits'
bpmask = 'bpmask.fits'

output = 'combine.fits'
```

8. Colaboratoryの実行環境

	製品名	スペック
CPU	Intel Xeon	2.6GHz 1C 2T
Memory	-	12.6GB
Storage (GPUあり)	-	360GB
GPU	NVIDIA Tesla T4	2560C(CUDA),320C(Tensor) 16GB GDDR6 8.1TFLOPS

- OS: Ubuntu 18.04.2 LTS
- アイドル状態が90分続くと停止
- 連続使用は最大12時間
- Notebookサイズは最大20MB

※ 2019/06/21時点