コンピュータリテラシ発展 ～Pythonを学ぶ～

第7回：Excel作業の前工程・後工程の自動化

情報学部 情報学科 情報メディア専攻
清水 哲也 (shimizu@info.shonan-it.ac.jp)

今回の授業内容

今回の授業内容

- 前回の課題解説
- フォルダ・ファイル操作
- 課題

前回の課題解説

前回の課題解説

- 前回の課題の解答例を示します
- 解答例について質問があればご連絡ください

解答例

https://colab.research.google.com/drive/1GdzqtL02cuKVY3aU_rJ39R3OpaIZ_HyS?usp=sharing

フォルダ・ファイル操作

絶対パスと相対パス

プログラムで外部ファイルを操作する際には，場所に関する情報を指定する必要があります

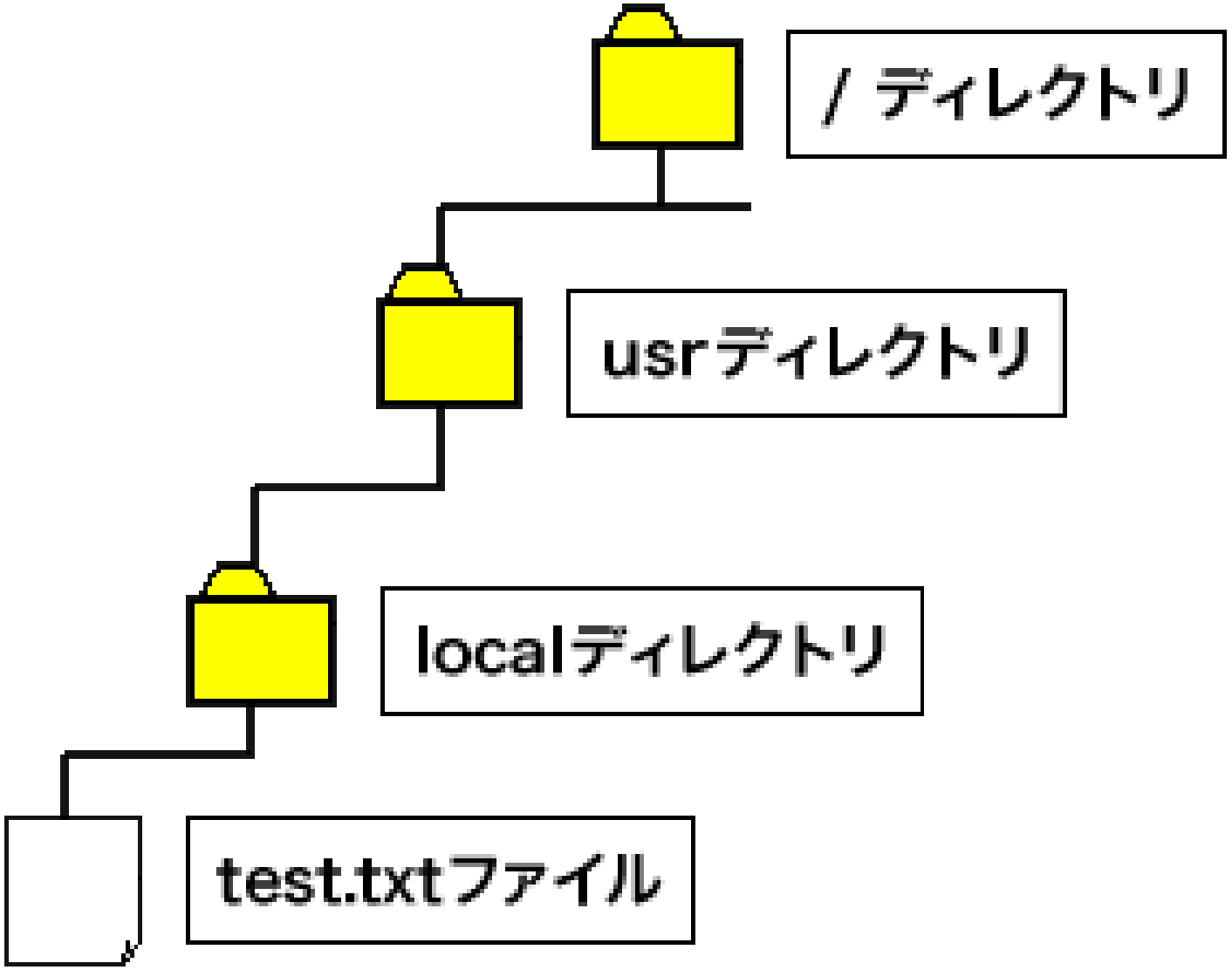
パスの指定には2つの方法があります

- **絶対パス**：絶対的な位置の記述（例：住所）
- **相対パス**：今自分がいる場所を起点にした位置の記述（例：道案内）

絶対パスと相対パス（絶対パスについて）

- OS(コンピュータ)で自分がいる場所に関する情報はツリー構造
- ツリー構造の基準
  - Windowsなら**C**ドライブ
  - LinuxやMacなら**ルート(root)**
- 基準から目的のファイルやフォルダまでのパスを記述するのが絶対パス

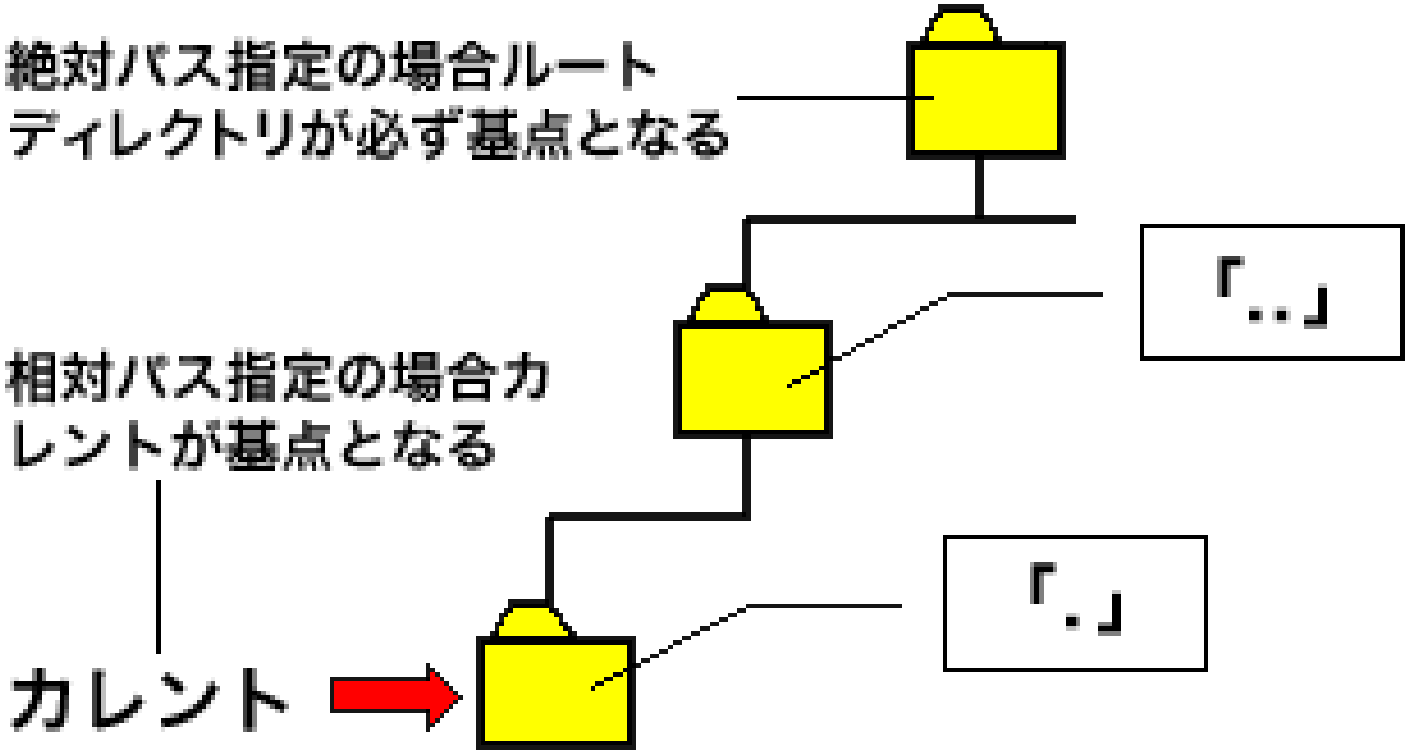
絶対パスと相対パス（絶対パスについて）



絶対パスと相対パス（相対パスについて）

- 自分が現在操作しているフォルダをピリオド1つ「.」で表します
- 1つ上の階層のフォルダならピリオド２つ「..」で表します

絶対パスと相対パス（相対パスについて）



フォルダにあるファイルを一覧表示

- 現在のフォルダの場所を確認します
- `os` モジュールをインポート
  - フォルダやファイルの操作などOSに依存しているさまざまな機能が備わっているモジュールです
- `getcwd()` 関数
  - 現在自分がいるフォルダ（カレントディレクトリ）を絶対パスで表示
  - **get c**urrent **w**orking **d**irectoryの略

フォルダにあるファイルを一覧表示

- 現在のフォルダの場所を確認します
- Colabで実行するので「¥」ではなく「/(スラッシュ)」で表示されます

```
# osモジュールをインポートします
impo   os

# 現在の作業ディレクトリのパスを取得します
path = os.getcwd()
print(path)
```

フォルダにあるファイルを一覧表示

- 現在のフォルダにあるファイルの一覧を表示
- `listdir()` 関数
  - 指定フォルダ内にあるファイルを一覧表示する関数
  - 引数はファイルを表示したいフォルダのパス

使用例

```
# os.listdir()関数を使用して、指定したフォルダの内容をリストとして取得します
os.listdir(path='.')
```

フォルダにあるファイルを一覧表示

- 現在のフォルダにあるファイルの一覧を表示します
- 今自分がいるフォルダ内のファイルを一覧表示します

```
impo    os # osモジュールは一度インポートしていれば再度書かかなくてもよい

# 現在の作業フォルダのパスを取得し、path変数に格納
path = os.getcwd()
# 現在の作業フォルダ内のファイルおよびフォルダの一覧を取得し、files変数に格納
files = os.listdir(path)

print(files)
```

フォルダにあるファイルを一覧表示

- 指定フォルダ内のファイル一覧を表示します
- Google Driveをマウントしてください
- 授業用フォルダ内のファイルを一覧表示します
- 絶対パスを利用します

```
impo   os

# Google Driveをマウントして授業用フォルダのpathを書いてください
path = '/content/drive/MyDrive/???'
# 授業用フォルダ内のファイルおよびフォルダの一覧を取得し、files変数に格納
files = os.listdir(path)

print(files)
```

フォルダにあるファイルを一覧表示

- 相対パスを使ってファイルを一覧表示させます
- ColabはLinux系なので「¥」ではなく「/(スラッシュ)」を使います
- 一つ上のフォルダにあるファイルを一覧表示させます
  - パスには「`../`」を指定します

フォルダにあるファイルを一覧表示

- 相対パスを使ってファイルを一覧表示させます
- 一つ上のフォルダにあるファイルを一覧表示させます

```
impo   os

# 現在の作業フォルダの一つ上のフォルダのパスを設定
path = '../'
# 現在の作業フォルダの一つ上のフォルダ内のファイルおよびフォルダの
# 一覧を取得し、files変数に格納
files = os.listdir(path)

print(files)
```

フォルダにあるファイルを一覧表示

- 再帰的にフォルダの中身を表示します
- 出力結果の中にあるフォルダ（サブフォルダ）の中身を表示します
- `glob` モジュール
  - 再帰的なファイル検索を実行します
  - 高度な検索が可能です
- 再帰的とは
  - 自己の行為の結果が自己に戻ってくること

フォルダにあるファイルを一覧表示

- 再帰的にフォルダの中身を表示します
- 出力結果の中にあるフォルダ（サブフォルダ）の中身を表示します
- 表示を見やすくするために `pprint` モジュールを利用します
- 注意：Google Driveに多くのファイルがある場合処理に時間がかかります

```
# globモジュールとpprintモジュールをインポート
impo   glob, pprint

# /content/drive/MyDrive/以下のすべてのファイルとファイルを再帰的に検索し、一覧を取得
files = glob.glob('/content/drive/MyDrive/**', recursive = True)

pprint.pprint(files)
```

フォルダの作成

- プログラムでフォルダを作成します
  - 特定の命名規則に従ってフォルダ生成
  - 大量のフォルダを生成
  - いろんな業務上で役に立つ

フォルダの作成

- フォルダを作成します
- `makedirs()` 関数
  - 第1引数：作成するフォルダ名
  - 第2引数：「`exist_ok=True`」とすることですでに作成されていてもエラーが発生しない

実行例

```
os.makedirs('file-name', exist_ok = True)
```

フォルを作成

- フォルダを作成します
- `makedirs()` 関数

```
impo   os

# 'tmp'フォルダを作成．既に存在する場合はエラーを出さずに無視する
os.makedirs('tmp', exist_ok = True)

# 現在の作業フォルダ内のファイルおよびフォルダの一覧を取得し表示
print(os.listdir('.'))
```

ファイルの書き込みと読み込み

- `open()` 関数
  - Pythonに標準で用意されている関数
  - 引数
    - ***file***：ファイル名
    - ***mode***：モードオプション
    - ***encoding***：エンコードオプション
    - ***newline***：改行オプション
    - 上記以外にもオプションがあります

`open()` 関数の使用例

```
open(file, mode='r', encoding=None, newline=None)
```

詳細：https://docs.python.org/ja/3/library/functions.html#open

ファイルの書き込みと読み込み

- `open()` 関数
  - 引数
    - ***file***
      - 作成するファイルの名前
      - ファイルの作成する場所をパスで指定
      - 例：`./file.txt`（カレントディレクトリに「file.txt」を作成）

`open()` 関数の使用例

```
open('file.txt', ...)
```

ファイルの書き込みと読み込み

- `open()` 関数
  - 引数
    - ***mode***
    - ファイルが開かれる際のモードを指定するオプションです
    - デフォルトでは `'r'` で，読み込み用に開きます
    - 読み込み用，書き込み用，新規作成を決めます

`open()` 関数の使用例

```
open('file.txt', mode='r', ...)
```

ファイルの書き込みと読み込み

- `open()` 関数 - 引数 - ***mode***オプション（一部）

モード	解説
`r`	読み込み用に開く（デフォルトのモード）
`w`	書き込み用に開き，ファイルが存在する場合は内容を破棄して上書きする
`x`	生成用に開き，ファイルが存在する場合は失敗する
`a`	書き込み用に開き，ファイルが存在する場合は末尾に追記する
`b`	バイナリモードで開く（画像などのファイル読み込み時に使用）※1

※1：他のモードと組み合わせて，`rb` や `wb` のように使う

ファイルの書き込みと読み込み

- `open()` 関数
  - 引数
    - ***encoding***：エンコードオプション
      - テキストエンコーディングとは，文字や記号などをデジタルデータとして保存するための方法です
      - デフォルトは `None`
      - いろんな規格があります
      - **UTF-8**, **Shift-JIS**など
      - Google Colabは**UTF-8**です

ファイルの書き込みと読み込み

- `open()` 関数
  - 引数
    - ***newline***：オプション
      - 改行コードの制御
      - デフォルトは `None`
      - ここでは説明を省きます

`open()` 関数の使用例

```
open('file.txt', mode='r', encoding='UTF-8', newline=None, ...)
```

ファイルの書き込みと読み込み

- `with`ブロック
  - 開始時と終了時の処理をセットで実行するPythonの構文
  - `open()`でファイルを開いたら，ファイルを閉じる処理`close()`を行う必要があります
  - `with`ブロックを使うことで，ブロックの終了時に自動的にクローズ処理をおこないます
  - 「`with open(...) as 変数：`」
    - 変数はファイルオブジェクト
    - 変数名は任意，一般的にfileの頭文字の「`f`」をよく使います

ファイルの書き込みと読み込み

- ファイルオブジェクト
  - Pythonでファイル操作する際に使用するオブジェクト
  - `write()` メソッド
    - 開いたファイルに書き込むメソッド
    - 引数に書き込み内容を渡します
  - `read()` メソッド
    - `open()` 関数で開いたファイルを読み込みます
    - デフォルトの読み込みモード「`r`」

ファイルの書き込みと読み込み

- ファイルの作成と書き込み
  - `open()` 関数, `with` ブロック, `write()` メソッド
    - カレントディレクトリに「**file.txt**」ファイルをモード「**x**」で開きます
    - `with` ブロックでクローズ処理します
    - `write()` メソッドで「**file.txt**」に「**sample text**」と書き込みます
    - インデントに注意！！

```
# 'file.txt'という名前のファイルを新規作成して開く
# 'x'モードは新規作成専用で、ファイルが既に存在する場合はエラーを発生させる
 i h open ('./file.txt', 'x') a  f:
  # ファイルに'sample text'という文字列を書き込む
  f.write('sameple text')
```

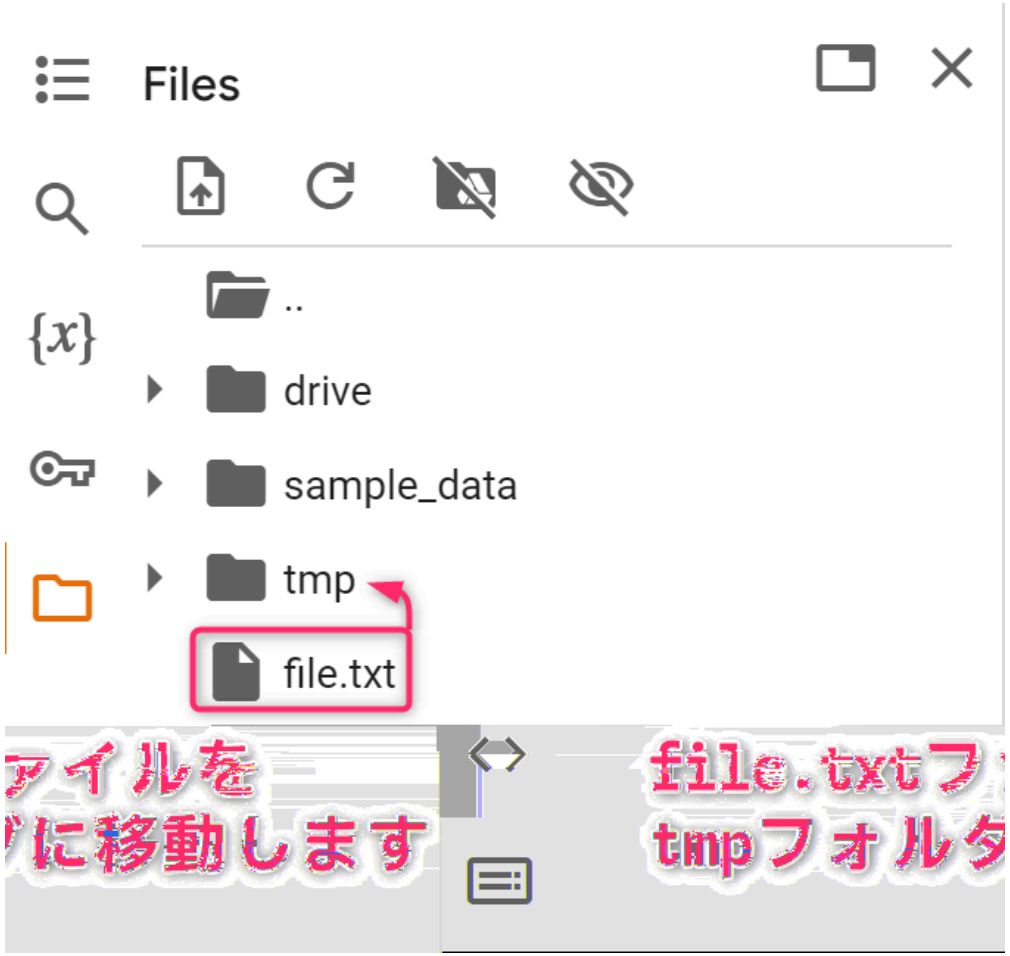
ファイルの書き込みと読み込み

- ファイルを開き，ファイルの内容を読み込む
- `open()` 関数，`with` ブロック，`read()` メソッド
  - カレントディレクトリにある「**file.txt**」ファイルを開きます
  - `with` ブロックでクローズ処理します
  - `read()` メソッドで「**file.txt**」の内容を読み込みます

```
# 'file.txt'という名前のファイルを読み取りモードで開く
# デフォルトでは'r'モード（読み取り専用モード）でファイルを開く
 i h open('./file.txt') a  f:
  # ファイルの内容を全て読み取り、text変数に格納
  text = f.read()
  print(text)
```

ファイルの移動

- ファイルの移動
- 「**file.txt**」を「**tmp**」フォルダに移動することを体験します



ファイルの移動

- `shutil` モジュールを利用します
  - インストール済み，インポート必要
  - `move()` 関数
    - 第1引数：移動対象となるファイルパス
    - 第2引数：移動先のフォルダパス
    - 第1引数はファイルそのもので，第2引数は移動先のフォルダへのパスになります

`move()` 関数の使用例

`dir1` フォルダにある `file.txt` を `dir1` フォルダ内にある `dir2` フォルダに移動

```
shutil.move('dir1/file.txt', 'dir1/dir2')
```

ファイルの移動

- 「**file.txt**」を「**tmp**」フォルダに移動します
- `shutil` モジュールの `move()` 関数を使用します

```
# shutilモジュールをインポート
impo   shutil

# 'file.txt'を'tmp/'フォルダに移動し、新しいパスをnew_path変数に格納
new_path = shutil.move('./file.txt', 'tmp/')

print(new_path)
```

ファイルのコピー

- ファイルをコピーする方法を学びます
- `shutil` モジュールの `copy()` 関数を利用します
  - 第1引数：コピー対象となるファイルパス
  - 第2引数：コピーしたファイルを作成する場所のパス

`copy()` 関数の使用例

`dir1` フォルダにある `file.txt` を `dir1` フォルダ内にある `dir2` フォルダにコピー

```
shutil.copy('dir1/file.txt', 'dir1/dir2')
```

ファイルのコピー

- 「**tmp/file.txt**」をコピーしてカレントディレクトリに作成する
- `shutil` モジュールの `copy()` 関数

```
# shutilモジュールをインポート
impo   shutil

# 'tmp/file.txt'を作業フォルダにコピー
shutil.copy('tmp/file.txt', './')
```

フォルダのコピー

- フォルダをコピーする方法を学びます
- `shutil` モジュールの `copytree()` 関数を利用します
  - フォルダ内の中身ごとコピーすることができます
  - 第1引数：コピー対象となるフォルダパス
  - 第2引数：コピーしたフォルダを作成する場所のパス
  - 第2引数はフォルダの新規作成も可能です

フォルダのコピー

- フォルダのコピー
- 「**tmp**」フォルダをコピーしてカレントディレクトリに「**tmp-copy**」フォルダを作成します
- `shutil` モジュールの `copytree()` 関数を利用する

```
# shutilモジュールをインポート
impo   shutil

# 'tmp'フォルダをコピーして、新しいフォルダ'tmp-copy'を作成
shutil.copytree('tmp', 'tmp-copy')
```

ファイル名の変更

- ファイル名を変更する方法を学びます
- `os` モジュールの `rename()` 関数を利用します
  - 第1引数：名前変更対象となるファイルパス
  - 第2引数：変更後の名前とパス

ファイル名の変更

- ファイル名の変更する方法を学びます
- 「**file.txt**」を「**file2.txt**」に変更します
- `os` モジュールの `rename()` 関数を利用します

```
# osモジュールをインポート
impo   os

# 'tmp/file.txt'のファイル名を'file2.txt'に変更
os.rename('tmp/file.txt', 'tmp/file2.txt')
```

課題

課題

- Moodleにある「SCfCL_07_prac.ipynb」ファイルをダウンロードしてColabにアップロードしてください
- 課題が完了したら「File」>「Download」>「Download .ipynb」で「.ipynb」形式でダウンロードしてください
- ダウンロードした **.ipynbファイル** とをMoodleに提出してください

提出期限は 11月14日(木) 20時まで です