

excelファイルの読み込み必要な データを取得しよう

```
(data_operation) PS C:\Users\小西 翔\python\python7> python data_operation.py
購入金額が1万円を超えている顧客:
  名前                メールアドレス    購入金額
0 山田 太郎          taro@example.com  12000
2 佐藤 花子          hanako@example.com 15000
結果が C:/Users/小西 翔/python/python7\customers10000.xlsx に保存されました。
(data_operation) PS C:\Users\小西 翔\python\python7>
```

この教材でできること

- Pythonのライブラリについて理解できる
- データ収集ができる

どんな教材？

ファイルの読み込みを行って編集や表示などを行う

目次

- ① コードを書く前の準備をしよう
- ② ファイルから必要なデータを取ってこよう
- ③ Pythonでファイルの編集を試してみよう

目次

① コードを書く前の準備をしよう

② ファイルから必要なデータを取ってこよう

③ Pythonでファイルの編集を試してみよう

① コードを書く前の準備をしよう

このページではコードを書く前の準備をしよう

① コードを書く前の準備をしよう

まずは環境構築をしていこうわからなかったら[HTMLに触れてみよう \(koni-prog-online.com\)](http://koni-prog-online.com)の②を見ながら環境設定しよう

今回は「data_operation」の名前にしてるよ

```
Preparing transaction: done
Verifying transaction: done
Executing transaction: done
#
# To activate this environment, use
#
#   $ conda activate file_editing
#
# To deactivate an active environment, use
#
#   $ conda deactivate

(base) PS C:\Users\小西 翔\python\python5> conda activate file_editing
(file_editing) PS C:\Users\小西 翔\python\python5>
```

質問

次は実際にライブラリをインポートしてみよう

excelファイルを使うにはどのライブラリを使ったらいいか調べてみよう

※考えてから次のスライドに進んでね

答え

「pip install pandas openpyxl」

このライブラリをインポートするとエクセルファイルを簡単に編集などができるようになるよ

① コードを書く前の準備をしよう

ターミナルでさっきのライブラリをインポートしてみよう

```
(file_editing) PS C:\Users\小西 翔\python\python5> pip install pandas openpyxl
Collecting pandas
  Downloading pandas-2.0.3-cp38-cp38-win_amd64.whl.metadata (18 kB)
Collecting openpyxl
  Downloading openpyxl-3.1.4-py2.py3-none-any.whl.metadata (2.5 kB)
Collecting python-dateutil>=2.8.2 (from pandas)
  Downloading python_dateutil-2.9.0.post0-py2.py3-none-any.whl.metadata (8.4 kB)
Collecting pytz>=2020.1 (from pandas)
  Downloading pytz-2024.1-py2.py3-none-any.whl.metadata (22 kB)
Collecting tzdata>=2022.1 (from pandas)
  Downloading tzdata-2024.1-py2.py3-none-any.whl.metadata (1.4 kB)
Collecting numpy>=1.20.3 (from pandas)
  Downloading numpy-1.24.4-cp38-cp38-win_amd64.whl.metadata (5.6 kB)
Collecting et_xmlfile (from openpyxl)
  Downloading et_xmlfile-1.1.0-py3-none-any.whl.metadata (1.8 kB)
Collecting six>=1.5 (from python-dateutil>=2.8.2->pandas)
  Downloading six-1.16.0-py2.py3-none-any.whl.metadata (1.8 kB)
```

① コードを書く前の準備をしよう

[customers.xlsx \(live.com\)](#)

上のリンクからExcelファイルをダウンロードしよう！

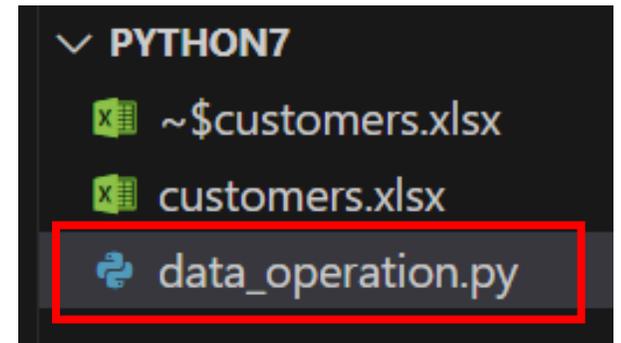
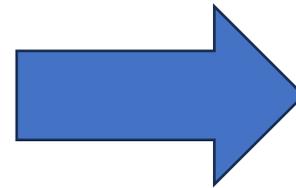
もしダウンロードできなかったら下の表をExcelに書こう

名前	メールアドレス	購入金額
山田 太郎	taro@example.com	12000
鈴木 次郎	jiro@example.com	8000
佐藤 花子	hanako@example.com	15000

① コードを書く前の準備をしよう

ターミナルで「code .」を入力してVscodeでPythonのファイルを生成しよう

```
(base) PS C:\Users\小西 翔\python\python7> conda activate data_operation  
(data_operation) PS C:\Users\小西 翔\python\python7> code .  
(data_operation) PS C:\Users\小西 翔\python\python7>
```



① コードを書く前の準備をしよう

準備はこれで終わりだよ

次からコードを書いていこう

目次

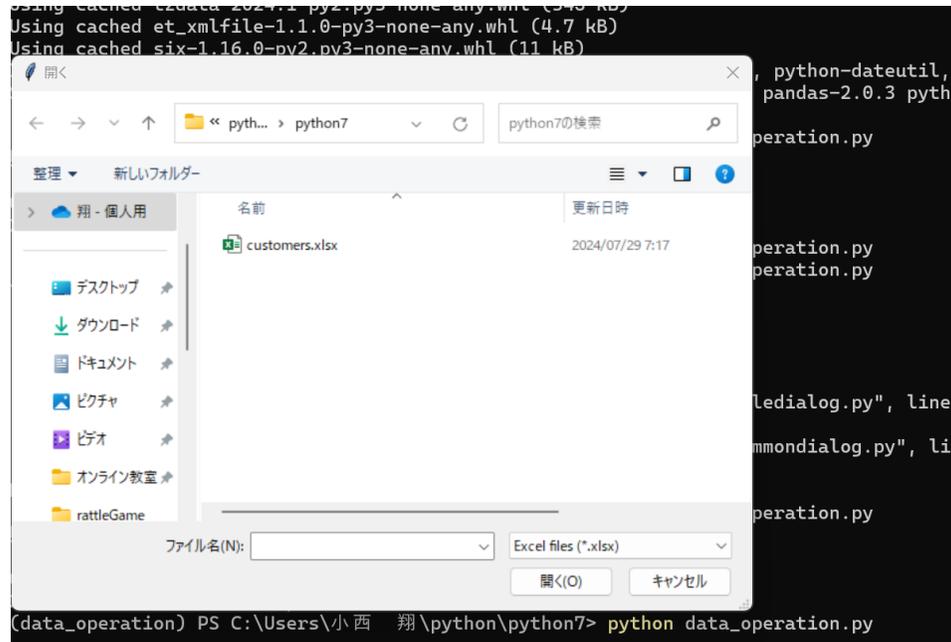
① コードを書く前の準備をしよう

② ファイルから必要なデータを取ってこよう

③ Pythonでファイルの編集を試してみよう

②ファイルから必要なデータを取ってこよう

このページではExcelを読み込んで、必要なデータを取り出すことができるよ



②ファイルから必要なデータを取ってこよう

まずはライブラリをインポートしよう

```
data_operation.py / ...  
1  import pandas as pd  
2
```

②ファイルから必要なデータを取ってこよう

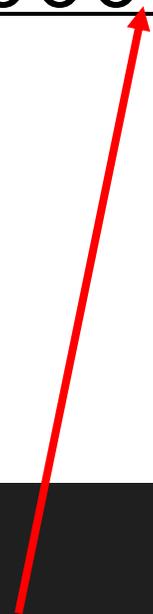
次にExcelのファイルを読み込んで変数dfに代入しよう

```
data_operation.py > ...  
1   import pandas as pd  
2  
3   # 顧客リストのExcelファイルを読み込む  
4   df = pd.read_excel('customers.xlsx')  
5
```

②ファイルから必要なデータを取ってこよう

変数dfの「購入金額」と書かれている列で10000以上の行を新しい変数に代入するようにしよう

```
5
6 # 購入金額が10000以上の顧客を抽出する
7 high_value_customers = df[df['購入金額'] > 10000]
8
9
```



説明

`[df['購入金額'] > 10000]:df`の行のうち、'購入金額'が10000を超える行だけを選択

```
5
6 # 購入金額が10000以上の顧客を抽出する
7 high_value_customers = df[df['購入金額'] > 10000]
8
9
```

②ファイルから必要なデータを取ってこよう

ここまでで一回Pythonを実行してエラーが出ないか確認してみよう

```
(data_operation) PS C:\Users\小西 翔\python\python7> python data_operation.py  
(data_operation) PS C:\Users\小西 翔\python\python7>
```

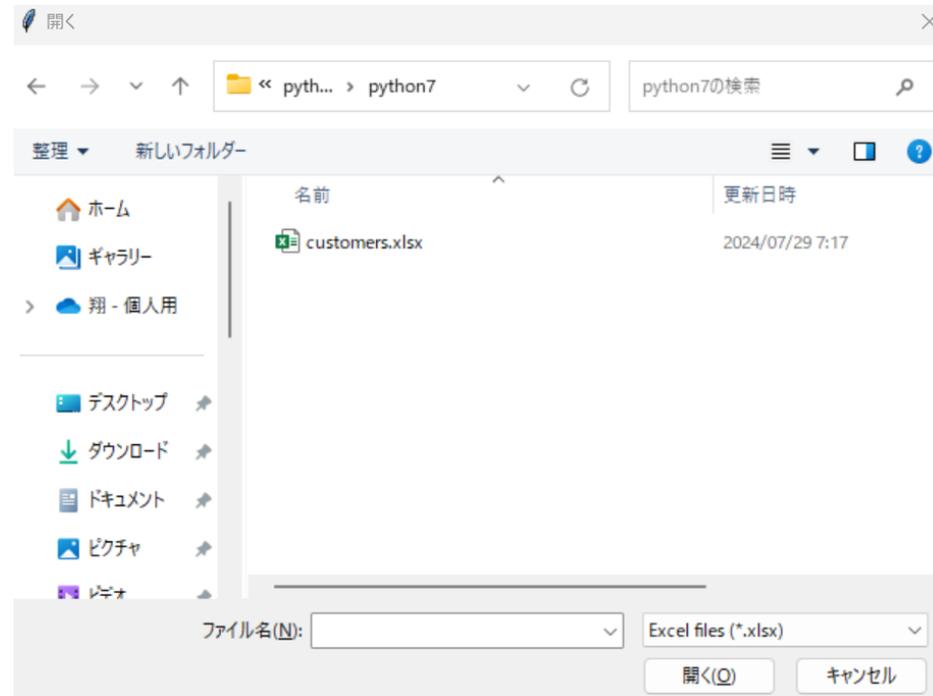
②ファイルから必要なデータを取ってこよう

次にprint文で文字を書いたり、配列を出力するようにしよう

```
6 # 購入金額が10000以上の顧客を抽出する
7 high_value_customers = df[df['購入金額'] > 10000]
8
9 # フィルタリングしたデータをターミナル上に表示する
10 print("購入金額が1万円を超えている顧客:")
11 print(high_value_customers)
12
```

②ファイルから必要なデータを取ってこよう

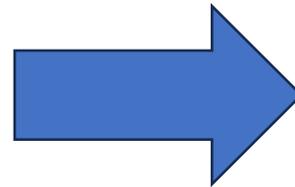
最後にファイルの選択をダイアログ形式で選択できるようにしよう



②ファイルから必要なデータを取ってこよう

ライブラリを増やそう

```
data_operation.py / ...  
1 import pandas as pd  
2
```



```
data_operation.py / ...  
1 import pandas as pd  
2 import tkinter as tk  
3 from tkinter import filedialog  
4
```

②ファイルから必要なデータを取ってこよう

ダイアログでファイルを自分で選択できるような関数を作ろう

```
6 # ファイル選択ダイアログを表示するための関数
7 def select_file():
8     root = tk.Tk()
9     root.withdraw() # メインウィンドウを表示しない
10    file_path = filedialog.askopenfilename(
11        filetypes=[("Excel files", "*.xlsx"), ("All files", "*.*")]
12    )
13    return file_path
```

説明

`root = tk.Tk():Tkinter(Pythonの標準GUIライブラリ)のメインウィンドウを作成します。`

`root.withdraw():メインウィンドウを表示しないようにする`

説明

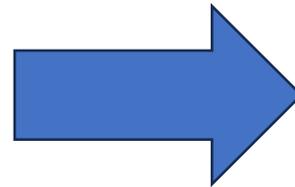
```
filedialog.askopenfilename(  
    filetypes=[("Excel files", "*.xlsx"), ("All files", "*.*")]  
)
```

ファイル選択をダイアログで開き、ファイルタイプはExcelを選択するようにする

②ファイルから必要なデータを取ってこよう

ファイル選択はさっき作った関数で読み込むようにしよう

```
data_operation.py > ...  
1  import pandas as pd  
2  
3  # 顧客リストのExcelファイルを読み込む  
4  df = pd.read_excel('customers.xlsx')
```



```
14  
15  # Excelファイルを選択して読み込む  
16  file_path = select_file()  
17
```

②ファイルから必要なデータを取ってこよう

ファイルがちゃんと読み込まれた場合と読み込まれなかった
場合でコードを書こう

読み込んでいる場合は変数dfにデータを代入しよう

```
14
15 # Excelファイルを選択して読み込む
16 file_path = select_file()
17
18 if file_path:
19     df = pd.read_excel(file_path)
20
21 else:
22     print("ファイルが選択されませんでした。")
23
```

質問

データを読み込んだ後のコードはさっきまで書いてたコードを追加すればいいよ！

※考えてから次のスライドに進んでね

```
17
18 if file_path:
19     df = pd.read_excel(file_path)
20
21     # 購入金額が10000以上の顧客を抽出する
22     [REDACTED]
23
24     # フィルタリングしたデータをターミナル上に表示する
25     [REDACTED]
26
27
28 else:
29     print("ファイルが選択されませんでした。")
```

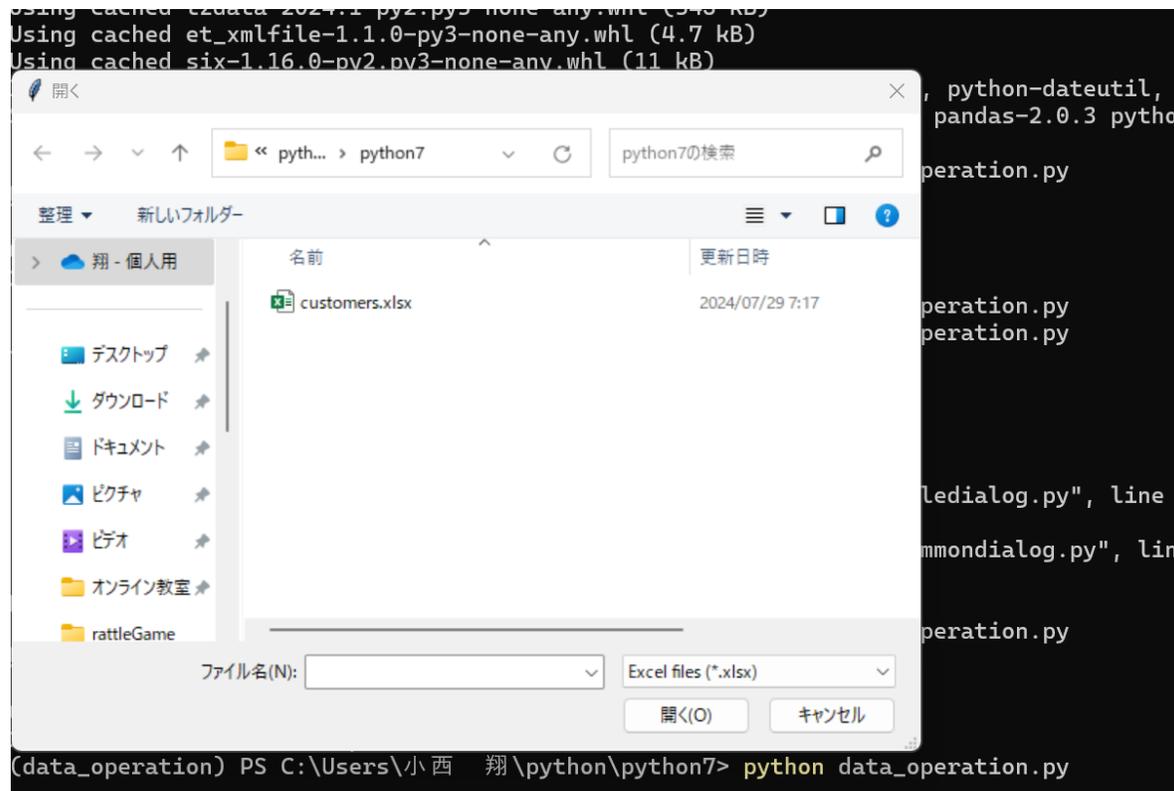
答え

このようにできたかな？

```
17
18 if file_path:
19     df = pd.read_excel(file_path)
20
21     # 購入金額が10000以上の顧客を抽出する
22     high_value_customers = df[df['購入金額'] > 10000]
23
24     # フィルタリングしたデータをターミナル上に表示する
25     print("購入金額が1万円を超えている顧客:")
26     print(high_value_customers)
27
28 else:
29     print("ファイルが選択されませんでした。")
```

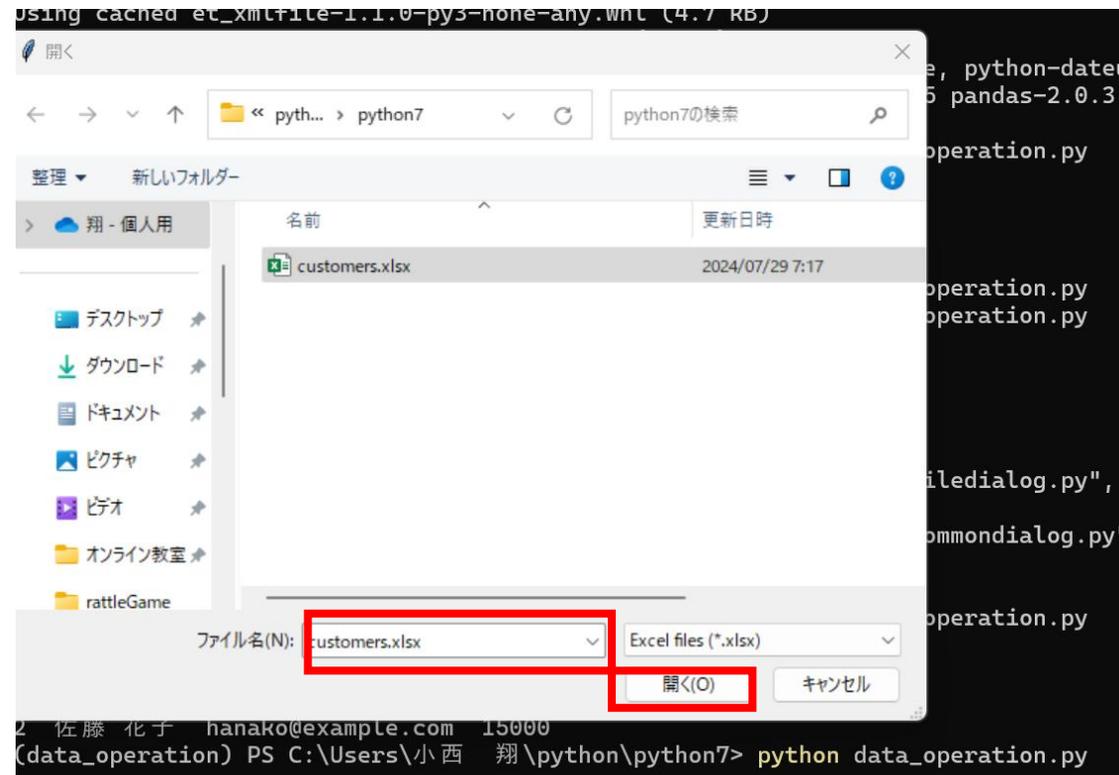
②ファイルから必要なデータを取ってこよう

実行してみてダイアログが出てきたかな？



②ファイルから必要なデータを取ってこよう

さっきのExcelファイルを選択して「開く」を押してみよう



②ファイルから必要なデータを取ってこよう

うまく情報が出力されたかな？

```
2 佐藤 花子 hanako@example.com 15000
(data_operation) PS C:\Users\小西 翔\python\python7> python data_operation.py
購入金額が1万円を超えている顧客:
名前          メールアドレス  購入金額
0 山田 太郎    taro@example.com 12000
2 佐藤 花子    hanako@example.com 15000
(data_operation) PS C:\Users\小西 翔\python\python7>
```

目次

- ① コードを書く前の準備をしよう
- ② ファイルから必要なデータを取ってこよう
- ③ Pythonでファイルの編集を試みよう

③Pythonでファイルの編集を試みよう

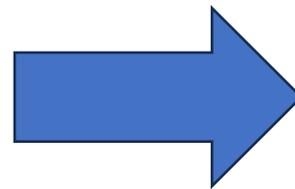
このページでは取ってきたデータをExcelファイルに格納して新しくエクセルファイルを保存するようにできる

```
(data_operation) PS C:\Users\小西 翔\python\python7> python data_operation.py
購入金額が1万円を超えている顧客:
  名前                メールアドレス    購入金額
0  山田 太郎          taro@example.com  12000
2  佐藤 花子          hanako@example.com 15000
結果が C:/Users/小西 翔/python/python7\customers10000.xlsx に保存されました。
(data_operation) PS C:\Users\小西 翔\python\python7>
```

③Pythonでファイルの編集を試みよう

ライブラリを増やそう

```
data_operation.py > ...  
1 import pandas as pd  
2 import tkinter as tk  
3 from tkinter import filedialog  
4
```



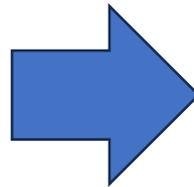
```
data_operation.py > select_file  
1 import pandas as pd  
2 import tkinter as tk  
3 from tkinter import filedialog  
4 from openpyxl import load_workbook  
5 import shutil  
6 import os  
7
```

③Pythonでファイルの編集をしてみよう

ファイルを新しく作り、その中にデータを書き込むようにしよう

まずは新しいファイル(customers10000.xlsx)を作ろう

```
23
24     # フィルタリングしたデータをターミナル上に表示
25     print("購入金額が1万円を超えている顧客:")
26     print(high_value_customers)
27
28     else:
29         print("ファイルが選択されませんでした。")
```

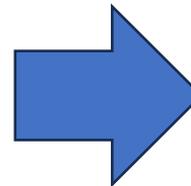


```
25     # フィルタリングしたデータをターミナル上に表示する
26     print("購入金額が1万円を超えている顧客:")
27     print(high_value_customers)
28
29     # コピー先のファイル名を設定
30     new_file_path = os.path.join(os.path.dirname(file_path), "customers10000.xlsx")
31
32     else:
33         print("ファイルが選択されませんでした。")
```

③Pythonでファイルの編集をしてみよう

元のファイルを新しく作ったファイルにコピーしよう

```
25 # フィルタリングしたデータをターミナル上に表示する
26 print("購入金額が1万円を超えている顧客:")
27 print(high_value_customers)
28
29 # コピー先のファイル名を設定
30 new_file_path = os.path.join(os.path.dirname(file_path), "customers10000.xlsx")
31
32 else:
33     print("ファイルが選択されませんでした。")
```



```
28
29 # コピー先のファイル名を設定
30 new_file_path = os.path.join(os.path.dirname
31
32 # 元のファイルを新しいファイルにコピー
33 shutil.copy(file_path, new_file_path)
34
35 else:
36     print("ファイルが選択されませんでした。")
37
```

③Pythonでファイルの編集を試みよう

Pythonを実行したら新しくExcelのファイルが生成されて、
中もコピーされているか確認してみよう

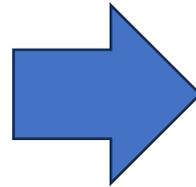
 customers.xlsx	2024/07/29 7:17	Microsoft Excel ワーク...	9 KB
 data_operation.py	2024/07/30 12:58	Python ソースファイル	2 KB
 customers10000.xlsx	2024/07/30 12:58	Microsoft Excel ワーク...	9 KB

	A	B	C	D
1	名前	メールアドレス	購入金額	
2	山田 太郎	taro@example.com	12000	
3	鈴木 次郎	jiro@example.com	8000	
4	佐藤 花子	hanako@example.com	15000	
5				
6				

③Pythonでファイルの編集を試みよう

コードに戻って、次はコピーされたファイルを開こう！（画面
上では何も起きないよ）

```
28
29     # コピー先のファイル名を設定
30     new_file_path = os.path.join(os.path.dirna
31
32     # 元のファイルを新しいファイルにコピー
33     shutil.copy(file_path, new_file_path)
34
35 else:
36     print("ファイルが選択されませんでした。")
37
```

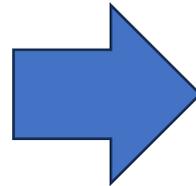


```
31
32     # 元のファイルを新しいファイルにコピー
33     shutil.copy(file_path, new_file_path)
34
35     # コピーされたファイルを開く
36     wb = load_workbook(new_file_path)
37     ws = wb.active
38
39 else:
40     print("ファイルが選択されませんでした。")
41
```

③Pythonでファイルの編集を試みよう

ファイルの1行目E列に見出しを書こう！

```
31
32 # 元のファイルを新しいファイルにコピー
33 shutil.copy(file_path, new_file_path)
34
35 # コピーされたファイルを開く
36 wb = load_workbook(new_file_path)
37 ws = wb.active
38
39 else:
40     print("ファイルが選択されませんでした。")
41
```



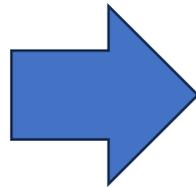
```
35 # コピーされたファイルを開く
36 wb = load_workbook(new_file_path)
37 ws = wb.active
38
39 # 1行目に見出しを書く
40 ws['E1'] = "購入金額が1万円を超えている顧客"
41
42 else:
43     print("ファイルが選択されませんでした。")
44
```

③Pythonでファイルの編集をしてみよう

10000を超えている顧客を見出しの下に書いていこう！

ここではfor文を使うよ

```
35     # コピーされたファイルを開く
36     wb = load_workbook(new_file_path)
37     ws = wb.active
38
39     # 1行目に見出しを書く
40     ws['E1'] = "購入金額が1万円を超えている顧客"
41
42 else:
43     print("ファイルが選択されませんでした。")
44
```



```
39     # 1行目に見出しを書く
40     ws['E1'] = "購入金額が1万円を超えている顧客"
41
42     # 2行目からデータを書き込む
43     for i, (_, row) in enumerate(high_value_customers.iterrows(), start=2):
44         ws[f'E{i}'] = row['名前']
45         ws[f'F{i}'] = row['メールアドレス']
46         ws[f'G{i}'] = row['購入金額']
47
48 else:
49     print("ファイルが選択されませんでした。")
50
```

説明

```
for i, (_, row) in enumerate(high_value_customers.iterrows(), start=2):
```

```
    ws[f'E{i}'] = row['名前']
```

```
    ws[f'F{i}'] = row['メールアドレス']
```

```
    ws[f'G{i}'] = row['購入金額']
```

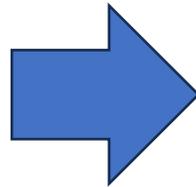
iは行番号（今回は2から始まる）、**row**はその行のデータ

`high_value_customers.iterrows()`は各行についてrowに1行ずつイテレート（各要素を順番に処理すること）する

③Pythonでファイルの編集をしてみよう

最後に新しいファイルを保存して保存されたことをprint文で出力しよう

```
39 # 1行目に見出しを書く
40 ws['E1'] = "購入金額が1万円を超えている顧客"
41
42 # 2行目からデータを書き込む
43 for i, (_, row) in enumerate(high_value_customers.iterrows(), start=2):
44     ws[f'E{i}'] = row['名前']
45     ws[f'F{i}'] = row['メールアドレス']
46     ws[f'G{i}'] = row['購入金額']
47
48 else:
49     print("ファイルが選択されませんでした。")
50
```



```
42 # 2行目からデータを書き込む
43 for i, (_, row) in enumerate(high_value_customers.iterrows(), start=2):
44     ws[f'E{i}'] = row['名前']
45     ws[f'F{i}'] = row['メールアドレス']
46     ws[f'G{i}'] = row['購入金額']
47
48 wb.save(new_file_path)
49 print(f"結果が {new_file_path} に保存されました。")
50
51 else:
52     print("ファイルが選択されませんでした。")
53
```

③Pythonでファイルの編集を試みよう

実行してみよう

```
(data_operation) PS C:\Users\小西 翔\python\python7> python data_operation.py
購入金額が1万円を超えている顧客:
  名前                メールアドレス    購入金額
0  山田 太郎          taro@example.com  12000
2  佐藤 花子          hanako@example.com 15000
結果が C:/Users/小西 翔/python/python7\customers10000.xlsx に保存されました。
(data_operation) PS C:\Users\小西 翔\python\python7>
```

③Pythonでファイルの編集を試みよう

実行した結果新しいファイル(customers_10000)はどうなったか見てみよう！

	A	B	C	D	E	F	G	H
1	名前	メールアドレス	購入金額		購入金額が1万円を超えている顧客			
2	山田 太郎	taro@example.com	12000		山田 太郎	taro@example.com	12000	
3	鈴木 次郎	jiro@example.com	8000		佐藤 花子	hanako@example.com	15000	
4	佐藤 花子	hanako@example.com	15000					
5								
6								
7								

お疲れさまでした

テキストは終了です。
あとは自分なりにアレンジを付け加えていこう！