

コンピュータリテラシ発展 ～Pythonを学ぶ～

第13回：いろんな業務を自動化してみよう(2)

(shimizu@info.shonan-it.ac.jp)

今回の授業内容

今回の授業内容

-
-
-
-
-
-
-
-

特定のルールに従ってフォルダ構成を整理する

データの準備

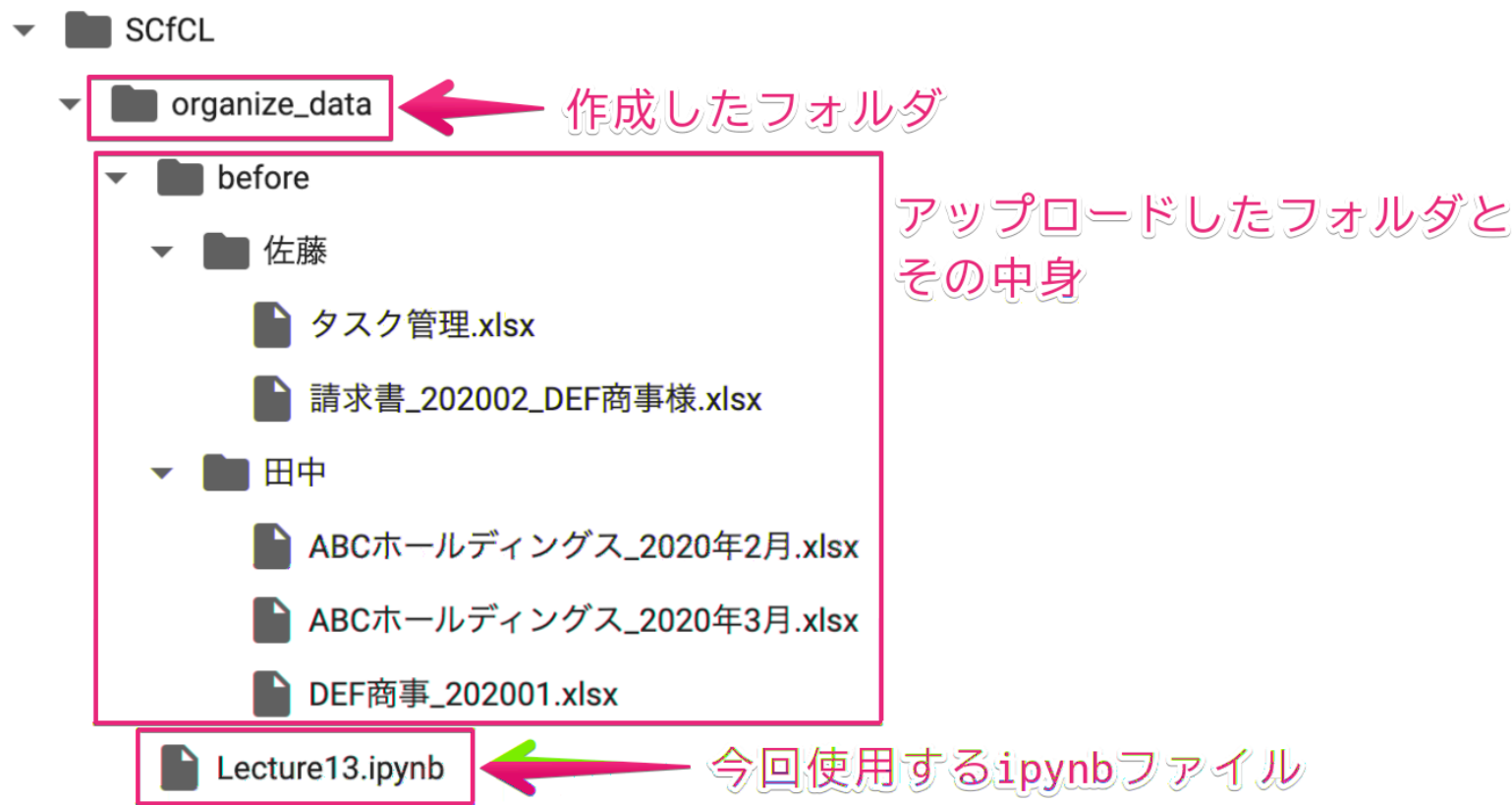
-
- Moodle before.zip
- organize_data before
- organize_data before

python(.ipynb) ファイルの作成

- Python (.ipynb) `organize_data`
-

データの準備

Colab



問題点と目標

問題点

-
-

目標

- - + +YYYY MM
-
-

「佐藤」フォルダの各ファイルの中身

▼図 7-7 佐藤 / 請求書_202002_DEF 商事様 .xlsx

	A	B	C
1		請求書 #105	
2		株式会社DEF商事	
3		住所 XXX	
4		郵便番号、都道府県、市区町村 XXX	
5		電話番号/FAX XXX	
6		日付 2020/02	
7		請求先	内容
8		名前 会社 XXX	XXX
9		住所 XXX	
10		郵便番号、都道府県、市区町村 XXX	
11		電話番号 XXX	

▼図 7-8 佐藤 / タスク管理 .xlsx

	A	B	C	D	E	F	G
1		タスクリスト					
2		マイ タスク	開始日	期日	達成率	メモ	
3		メール送信	日付	日付	0%		
4		テレアポ	日付	日付	50%		
5		請求書送付	日付	日付	100%		

「田中」フォルダの各ファイルの中身

▼図 7-9 田中 /DEF 商事_202001.xlsx

	A	B	C
1		請求書 #98	
2		株式会社DEF商事	
3		住所 XXX	
4		郵便番号、都道府県、市区町村 XXX	
5		電話番号/FAX XXX	
6		日付 2020/01	

▼図 7-10 田中 /ABC ホールディングス_2020 年 2 月.xlsx

	A	B	C
1		請求書 #100	
2		株式会社ABCホールディングス	
3		住所 XXX	
4		郵便番号、都道府県、市区町村 XXX	
5		電話番号/FAX XXX	
6		日付 2020/02	

▼図 7-11 田中 /ABC ホールディングス_2020 年 3 月

	A	B
1		請求書 #101
2		株式会社ABCホールディングス
3		住所 XXX
4		郵便番号、都道府県、市区町村 XXX
5		電話番号/FAX XXX
6		日付 2020/03
7		請求先
8		名前 会社 XXX
9		住所 XXX
10		郵便番号、都道府県、市区町村 XXX
11		電話番号 XXX

新しいフォルダにすべてのファイルをコピーする

新しいフォルダにすべてのファイルをコピーする

-

-

after

-

-

shutil

copytree()

-

2

2

-

新しいフォルダにすべてのファイルをコピーする

```
# shutilモジュールをインポート
import shutil

# organize_dataフォルダパスを指定
org_path = '/content/drive/MyDrive/???/organize_data/'

try:
    # 「before」フォルダを「after」フォルダにコピー
    shutil.copytree(org_path + 'before', org_path + 'after')

# FileExistsErrorが発生した場合のエラーハンドリングを行う
# ここでは「after」フォルダがすでに存在する場合にこのエラーが発生
except FileExistsError as e:
    # 「after」フォルダがすでに存在する場合のエラーメッセージを表示
    print('すでにafterフォルダが存在します')
```

organize_data

after

佐藤

タスク管理.xlsx

請求書_202002_DEF商事様.xlsx

田中

ABCホールディングス_2020年2月.xlsx

ABCホールディングス_2020年3月.xlsx

DEF商事_202001.xlsx

before

佐藤

タスク管理.xlsx

請求書_202002_DEF商事様.xlsx

田中

ABCホールディングス_2020年2月.xlsx

ABCホールディングス_2020年3月.xlsx

DEF商事_202001.xlsx

SCfCL_13-Lecture-Memo.ipynb

新たに作成されたフォルダとその中身

新しいフォルダに
すべてのファイルを
コピーする

すべてのファイルを取得する

すべてのファイルを取得する

- after

-

- `os.listdir()`

- `glob`

すべてのファイルを取得する

```
# globモジュールをインポート
import glob

# 「after」フォルダ内のすべてのファイルとディレクトリを再帰的に取得
files = glob.glob(org_path + 'after/**', recursive=True)

# 取得したファイルとディレクトリのリストを表示
files
```

取得したファイルが請求書ファイルかどうか判別する

取得したファイルが請求書ファイルかどうか判別する

-

-

Excel

-

OpenPyXL

拡張子が「.xlsx」かどうかの確認

- OpenPyXL .xlsx
- OpenPyXL

```
# Excelファイルかどうかをチェックする関数を定義
def check_excel_file(file):
    # ファイル名に '.xlsx' が含まれているか进行检查
    if '.xlsx' in file:
        return True # Excelファイルである
    else:
        return False # Excelファイルではない

# ファイルリストをループで処理
for file in files:
    # ファイルがExcelファイルかどうか进行检查
    if check_excel_file(file):
        # Excelファイルの場合のメッセージを表示
        print('「' + file + '」はExcelです。')
    else:
        # Excelファイルではない場合のメッセージを表示
        print('「' + file + '」はExcelではありません。')
```

シート名が「請求書」かどうかを確認

-
- Excel

sheetnames

シート名が「請求書」かどうかを確認

```
# openpyxlモジュールをインポート
import openpyxl

# チェックするシート名を指定
invoice_sheet_name = '請求書'

# 指定されたExcelファイルに請求書シートがあるかどうかをチェックする関数を定義
def check_invoice_excel_file(wb):
    # ワークブックのシート名一覧に '請求書' が含まれているかを確認
    if invoice_sheet_name in wb.sheetnames:
        return True # '請求書' シートがある
    else:
        return False # '請求書' シートがない

# ファイルリストをループで処理
for file in files:
    # ファイルがExcelファイルかどうかを確認
    if check_excel_file(file):
        # Excelファイルを読み込む
        wb = openpyxl.load_workbook(file)
        # ワークブックに請求書シートがあるかどうかを確認
        if check_invoice_excel_file(wb):
            print('「' + file + '」は請求書です。') # 請求書シートがある場合のメッセージを表示
        else:
            print('「' + file + '」は請求書ではありません。') # 請求書シートがない場合のメッセージを表示
    else:
        print('「' + file + '」はExcelではありません。') # Excelファイルではない場合のメッセージを表示
```

新しいファイル名とフォルダ名を取得する

新しいファイル名とフォルダ名を取得する

- $\text{Date} = \text{Date}(\text{Year} + \text{Year_Offset}, \text{Month}, \text{Day})$
- $\text{Date} = \text{Date}(\text{Year}, \text{Month}, \text{Day})$
- Excel: $\text{Date} = \text{B2} + \text{B5}$

会社名を取得する

- OpenPyXL

B2

name

```
# チェックするセルの位置を指定
corporate_name_cell = 'B2'

# 指定されたExcelファイルに請求書シートがあるかどうかをチェックする関数を定義
def check_invoice_excel_file(wb):
    # ワークブックのシート名一覧に '請求書' が含まれているかを確認
    if invoice_sheet_name in wb.sheetnames:
        return True # '請求書' シートがある
    else:
        return False # '請求書' シートがない

# ワークブックから請求書シートの企業名を取得する関数を定義
def get_invoice_corporate_name(wb):
    # '請求書' シートの指定されたセルの値を取得
    name = wb[invoice_sheet_name][corporate_name_cell].value
    return name # 取得した企業名を返す

# ファイルリストをループで処理
for file in files:
    # ファイルがExcelファイルかどうかを確認
    if check_excel_file(file):
        # Excelファイルを読み込む
        wb = openpyxl.load_workbook(file)
        # ワークブックに請求書シートがあるかどうかを確認
        if check_invoice_excel_file(wb):
            # 請求書シートの企業名を取得
            name = get_invoice_corporate_name(wb)
            # 企業名を表示
            print(name)
```

請求書の発行年月を取得する

- Excel B5
- B5 YYYY/MM
- \d\d\d\d/\d\d YYYY/MM
date

請求書の発行年月を取得する(1/3)

```
import re  # reモジュールをインポート（正規表現操作を行うためのモジュール）

# チェックするセルの位置を指定
invoice_created_date_cell = 'B5'

# 指定されたExcelファイルに請求書シートがあるかどうかをチェックする関数を定義
def check_invoice_excel_file(wb):
    # ワークブックのシート名一覧に '請求書' が含まれているかをチェック
    if invoice_sheet_name in wb.sheetnames:
        return True  # '請求書' シートがある
    else:
        return False  # '請求書' シートがない
```

請求書の発行年月を取得する(2/3)

```
# ワークブックから請求書シートの作成日を取得する関数を定義
def get_invoice_created_date(wb):
    # '請求書' シートの指定されたセルの値を取得
    value = wb[invoice_sheet_name][invoice_created_date_cell].value

    # 作成日を抽出するための正規表現パターンを定義
    invoice_created_date_regex = re.compile(r'\d\d\d\d/\d\d')
    # 正規表現パターンにマッチする部分を検索
    invoice_created_date_match = invoice_created_date_regex.search(value)

    # マッチした部分を取得
    date = invoice_created_date_match.group()
    return date # 取得した日付を返す
```

請求書の発行年月を取得する(3/3)

```
# ファイルリストをループで処理
for file in files:
    # ファイルがExcelファイルかどうかをチェック
    if check_excel_file(file):
        # Excelファイルを読み込む
        wb = openpyxl.load_workbook(file)
        # ワークブックに請求書シートがあるかどうかをチェック
        if check_invoice_excel_file(wb):
            # 請求書シートの作成日を取得
            date = get_invoice_created_date(wb)
            # 作成日を表示
            print(date)
```

請求書ファイル名をフォーマット通りの形にする

- `format()`
- `get_invoice_created_date()` 2020/02
2020 02
-

請求書ファイル名をフォーマット通りの形にする(1/2)

```
# ワークブックから新しい請求書ファイル名を生成する関数を定義
def get_new_invoice_file_name(wb):
    # 企業名と作成日を取得
    invoice_corporate_name = get_invoice_corporate_name(wb)
    invoice_created_date = get_invoice_created_date(wb)

    # 作成日をフォーマットする
    formatted_date = '{0}年{1}月'.format(invoice_created_date[0:4], invoice_created_date[5:7])

    # 新しいファイル名を生成する
    file_name = '請求書_{0}様_{1}'.format(invoice_corporate_name, formatted_date)
    file_name_with_ext = file_name + '.xlsx'
    return file_name_with_ext, invoice_corporate_name # 新しいファイル名と企業名を返す
```

請求書ファイル名をフォーマット通りの形にする(2/2)

```
# ファイルリストをループで処理
for file in files:
    # ファイルがExcelファイルかどうかをチェック
    if check_excel_file(file):
        # Excelファイルを読み込む
        wb = openpyxl.load_workbook(file)
        # ワークブックに請求書シートがあるかどうかをチェック
        if check_invoice_excel_file(wb):
            # 新しい請求書ファイル名を生成する
            new_file_name, new_dir_name = get_new_invoice_file_name(wb)
            # 新しいファイル名と企業名を表示する
            print(new_file_name, new_dir_name)
```


新しいフォルダを作成する

新しいフォルダを作成する

-
-
-

`os`

`makedirs()`

`exist_ok`

`True`

新しいフォルダを作成する

```
import os

# 新しい請求書ディレクトリを作成する関数を定義
def make_new_invoice_dir(invoice_corporate_name):
    # 新しいディレクトリのパスを組み立てる
    dir_path = org_path + 'after/' + invoice_corporate_name
    # 新しいディレクトリを作成（存在する場合はエラーを無視）
    os.makedirs(dir_path, exist_ok=True)
    return dir_path # 新しいディレクトリのパスを返す

# ファイルリストをループで処理
for file in files:
    # ファイルがExcelファイルかどうかをチェック
    if check_excel_file(file):
        # Excelファイルを読み込む
        wb = openpyxl.load_workbook(file)
        # ワークブックに請求書シートがあるかどうかをチェック
        if check_invoice_excel_file(wb):
            # 新しい請求書ファイル名を生成する
            new_file_name, new_dir_name = get_new_invoice_file_name(wb)
            # 新しい請求書ディレクトリを作成する
            new_dir_path = make_new_invoice_dir(new_dir_name)

# 新しいディレクトリの内容をリストで取得
new_dir = os.listdir(org_path + 'after/')
new_dir # 取得したディレクトリの内容を表示
```

ファイル名変更とフォルダ移動を行う

ファイル名変更とフォルダ移動を行う

-

shutil

move()

```
# ファイルリストをループで処理
for file in files:
    # ファイルがExcelファイルかどうかをチェック
    if check_excel_file(file):
        # Excelファイルを読み込む
        wb = openpyxl.load_workbook(file)
        # ワークブックに請求書シートがあるかどうかをチェック
        if check_invoice_excel_file(wb):
            # 新しい請求書ファイル名を生成する
            new_file_name, new_dir_name = get_new_invoice_file_name(wb)
            # 新しい請求書ディレクトリを作成する
            new_dir_path = make_new_invoice_dir(new_dir_name)
            # ファイルを新しいディレクトリに移動し、名前を変更する
            shutil.move(file, new_dir_path + '/' + new_file_name)
```

プログラムをまとめる

- Google Colab
- python 1
-
- <https://colab.research.google.com/drive/1TiiPqPtIjqFAEM4EHKoGd1lZQ8sJr3gc?usp=sharing>

課題

課題13

-
- File > Download > Download .ipynb .ipynb
- .ipynb Moodle
- 7 18 () 20