

# コンピュータリテラシ発展 ～Pythonを学ぶ～

## 第5回：Excel作業を自動化しよう(2)

学びβ 学科 メディア攻  
水 折也 ( [shimizu@info.shonan-it.ac.jp](mailto:shimizu@info.shonan-it.ac.jp) )

## 今回の授業内容

# 今回の授業内容

- 1回の課題
- Excelファイルの編集
- Excelのレイアウトを編集
- 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

(解答例を作成したらURLを貼り付ける)

# Excelファイルを編集する

# Excelファイルを新規作成する

- Excelファイル（ワークブック）を新規作成します

```
import openpyxl as op

wb = op.Workbook()
wb.save('filename.xlsx')
```

- `Workbook` の引数を空にすると新しいワークブックを 読み込みます
- `save()` メソッドで 保存 します
- 保存先を指定したい場合はパスを含めて指定する必要があります

# Excelファイルを新規作成する

- Excelファイル（ワークブック）を新規作成します（パスを含む場合）

```
import openpyxl as op
```

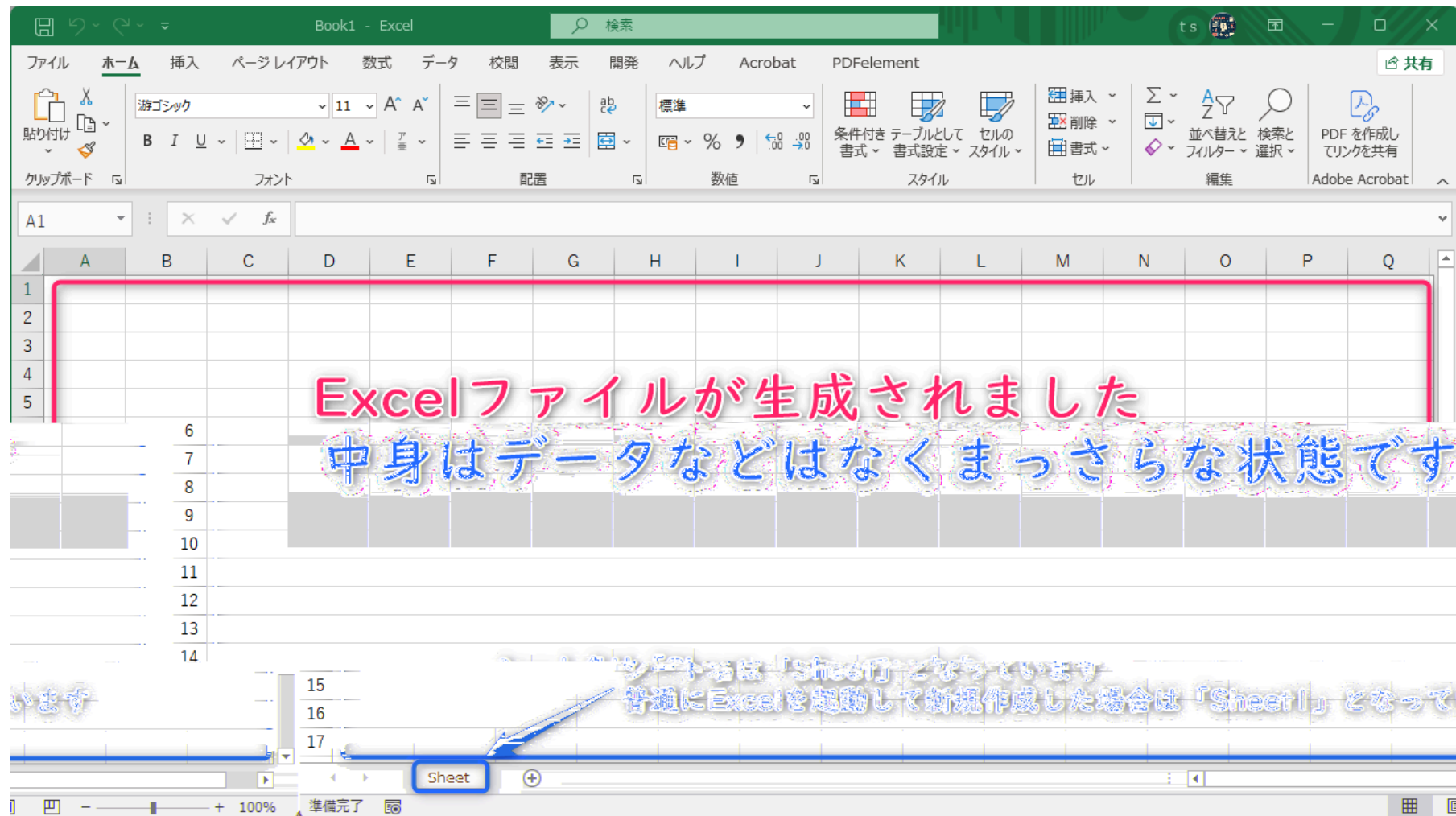
```
wb = op.Workbook()
```

```
wb.save('/content/drive/MyDrive/???/filename.xlsx')
```

- `Workbook` の引数を空にすると新しいワークブックを 読み込みます
- `save()` メソッドで 保存 します
- 場所を指定したい場合はパスを含めて指定する必要があります



# 新規作成したExcelファイルの確認



Workr

- 

名を指定してシートを追加します

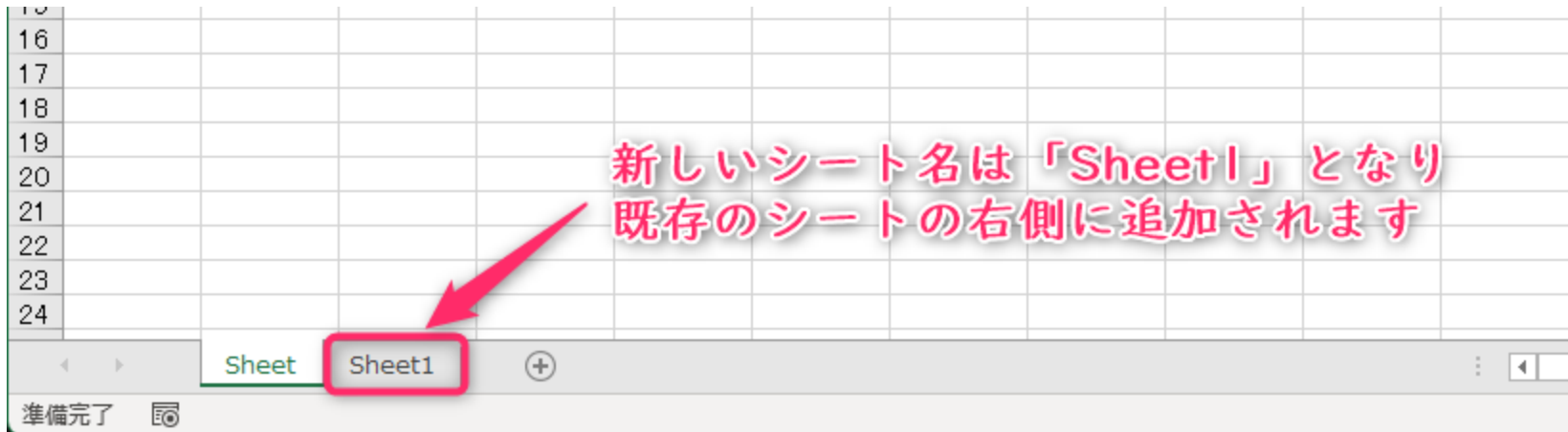
```
create_sheet(index = 数字, title = 'シート名')
```

ます

# Excelシートを追加/削除します

- 新しいシートを追加します

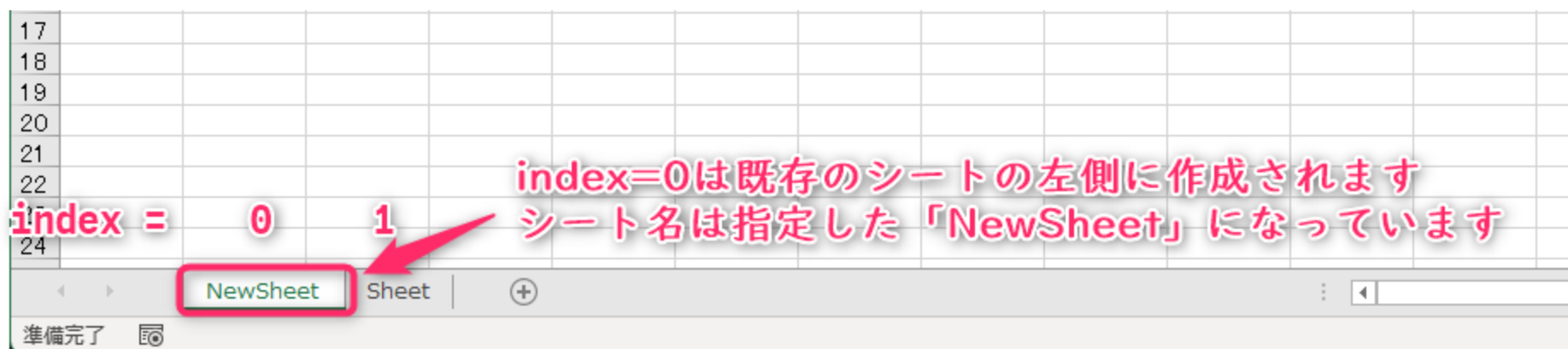
```
wb = op.Workbook()  
  
wb.create_sheet()  
print(wb.sheetnames)  
wb.save('/content/drive/MyDrive/???/create_sheet.xlsx')
```



# Excelシートを追加/削除します

- 入位置とシート名を指定して新しいシートを追加します

```
wb = op.Workbook()  
  
wb.create_sheet(index = 0, title = 'NewSheet')  
print(wb.sheetnames)  
wb.save('/content/drive/MyDrive/???/create_sheet.xlsx')
```



# Excelシートを追加/削除します

- シートを追加して既 のシートを削 します

```
wb = op.Workbook()  
  
wb.create_sheet()  
print(wb.sheetnames)  
  
wb.remove(wb['Sheet'])  
print(wb.sheetnames)
```

## セルの値を編集します

- 指定したセルを編集します
  - 文字列を入力します : `Worksheetオブジェクト[セル] = '文字列'`
  - 数値を入力します : `Worksheetオブジェクト[セル] = 数値`
  - 数式を入力します : `Worksheetオブジェクト[セル] = '=数式'`

## セルの値を編集する

```
wb = op.Workbook()
sheet = wb.active

sheet['B2'] = '文字列'           # 文字列
sheet['B3'] = '10'               # 文字列としての数字
sheet['B4'] = 10                 # 数字
sheet['B5'] = 20                 # 数字
sheet['B6'] = '=sum(B4:B5)'     # Excelの関数

wb.save('/content/drive/MyDrive/???/cell.xlsx')
```

## セルの値を編集します（結果）

	A	B	C	D	E
1					
2		文字列			
3		10			
4		10			
5		20			
6		30			

Diagram illustrating the result of editing cell values in a spreadsheet. The cells B2, B3, B4, B5, and B6 are highlighted with a red border. Red arrows point from labels to the corresponding cells:

- Text (文字列) points to B2.
- Number (数値) points to B4.
- Formula (数式) points to B6.

The spreadsheet interface shows the active cell is B3, containing the value 10. The status bar at the bottom indicates the active sheet is "Sheet".



# フォントを設定します

- `Font()` 関数：セルのフォントを設定する関数です
- `Font()` 関数を使用するには `op.styles.fonts.Font` と書く必要があります
- 書くのが大変なので関数を指定してインポートしておきます

```
import openpyxl as op
from openpyxl.styles.fonts import Font
```

- `Font` 関数の基本的な使い方です

`Font(キーワード引数1=値, キーワード引数2=値・・・)`

細： <https://openpyxl.readthedocs.io/en/stable/styles.html>

# フォントを設定します

- Font 関数の基本的な使い方です

```
Font(キーワード引数1=値, キーワード引数2=値・・・)
```

-  : フォントサイズを18pt, 太  にする  です

```
Font(size=18, bold=True)
```

-  : フォントサイズを24pt,  にする  です

```
Font(size=24, italic=True)
```

## Font() 関数の代表的な引数

引数名	データ型	解説
name	文字列型	フォント名を指定します (例: name='メイリオ')
size	整数型	フォントsizeを変更します (例: size=18)
bold	ブール型	True で太字になります (例: bold=True)
italic	ブール型	True でイタリック( )になります (例: italic=True)
color	文字列型	カラーコードで 表 の色を指定します (例: color='FF0000')

## フォントを設定します

```
wb = op.Workbook()  
sheet = wb.active  
  
sheet['B2'] = '18pt bold'  
sheet['B2'].font = Font(size=18, bold=True)  
  
sheet['B4'] = '24pt 下線'  
sheet['B4'].font = Font(size=24, underline='single')  
  
wb.save('/content/drive/MyDrive/???/font.xlsx')
```

## フォントを設定します（結果）

	A	B	C	D	E	
2		<b>18pt bold</b>				
3						
4		<b><u>24pt 下線</u></b>				
5						
6						
7						

## Excelのレイアウトを編集

# Excelの行高と列幅を設定する

- Excelの行高と列幅を設定する方法
  - 行高：行番号を指定して高さの数（ポイント）を入力します

```
Worksheetオブジェクト.row_dimensions[行番号].height = 高さの数値
```

- 列幅：列番号を指定して幅の数（文字数）を入力します

```
Worksheetオブジェクト.column_dimensions[列番号].width = 幅の数値
```

行高と列幅で単位が異なるので 注意が必要です

# Excelの行高と列幅を設定する

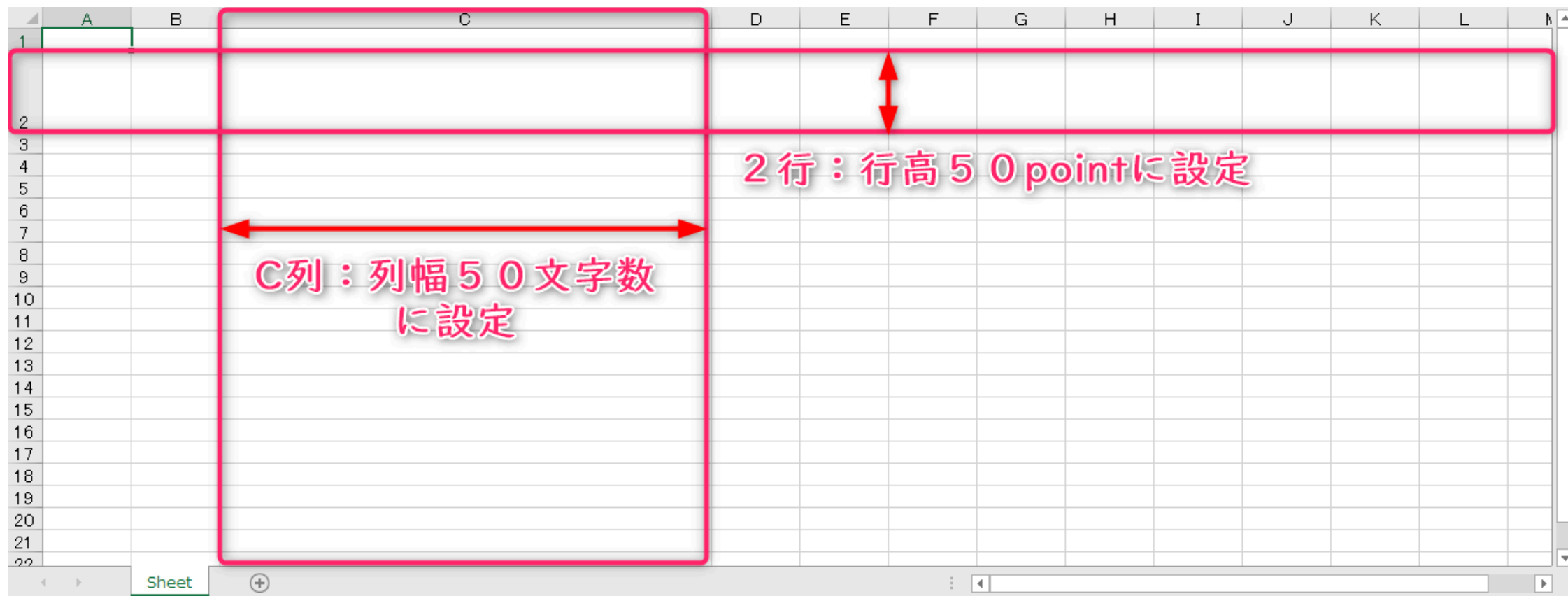
- Excelの行高と列幅を設定する例です
  - 行高の設定：2行目を「50」にします
  - 列幅の設定：C列目を「50」にします

```
wb = op.Workbook()
sheet = wb.active

sheet.row_dimensions[2].height = 50
sheet.column_dimensions['C'].width = 50
wb.save('/content/drive/MyDrive/???/row_column.xlsx')
```



# Excelの行高と列幅を設定する（結果）



# Excelの行や列を非表示にする

## 行や列を非表示にする **hidden** 属性

行や列を指定して非表示設定にします

```
Worksheetオブジェクト.row_dimensions[行番号].hidden = True  
Worksheetオブジェクト.column_dimensions[列番号].hidden = True
```

非表示の行や列を表示します

```
Worksheetオブジェクト.row_dimensions[行番号].hidden = False  
Worksheetオブジェクト.column_dimensions[列番号].hidden = False
```

# Excelの行や列を非表示にする

- 3行目とB,D列を非表示にする例です

```
wb = op.Workbook()
sheet = wb.active

#セルA1～E5に1～25の数字を入れる
for i in range(1,6):
    for j in range(1,6):
        sheet.cell(j,i).value = i * j

sheet.row_dimensions[3].hidden = True
sheet.column_dimensions['B'].hidden = True
sheet.column_dimensions['D'].hidden = True
wb.save('/content/drive/MyDrive/???.row_column_hidden.xlsx')
```

# 3行目を非表示  
# B列を非表示  
# D列を非表示

# Excelの行や列を非表示にする

- 3行目とB,D列を非表示にする例です

The image shows an Excel spreadsheet with columns A through G and rows 1 through 9. Row 3 is hidden, and columns B and D are hidden. The visible data is as follows:

	A	C	E	F	G
1					
2	2	6	10		
4	4	12	20		
5	5	15	25		
6					
7					
8					
9					

Annotations in the image:

- Red boxes highlight the column headers for B and D, with arrows pointing to the hidden state.
- A blue box highlights the row number 3, with an arrow pointing to the hidden state.
- Blue text with arrows pointing to the hidden areas: "3行目が非表示" (Row 3 is hidden) and "B列とD列が非表示" (Columns B and D are hidden).

# Excelの行と列を固定表示にする

## Excelの行と列を固定表示します

- 固定する行や列の下もしくは右，右下のセルを指定します
- 例：1行目のみ固定する場合：指定するセルは「A2」
- 例：B列まで固定する場合：指定するセルは「C1」
- 例：2行目とC列まで固定する場合：指定するセルは「D3」

```
Worksheetオブジェクト.freeze_panes = セル
```

# Excelの行と列を固定表示にする

- 2行目までを固定にします

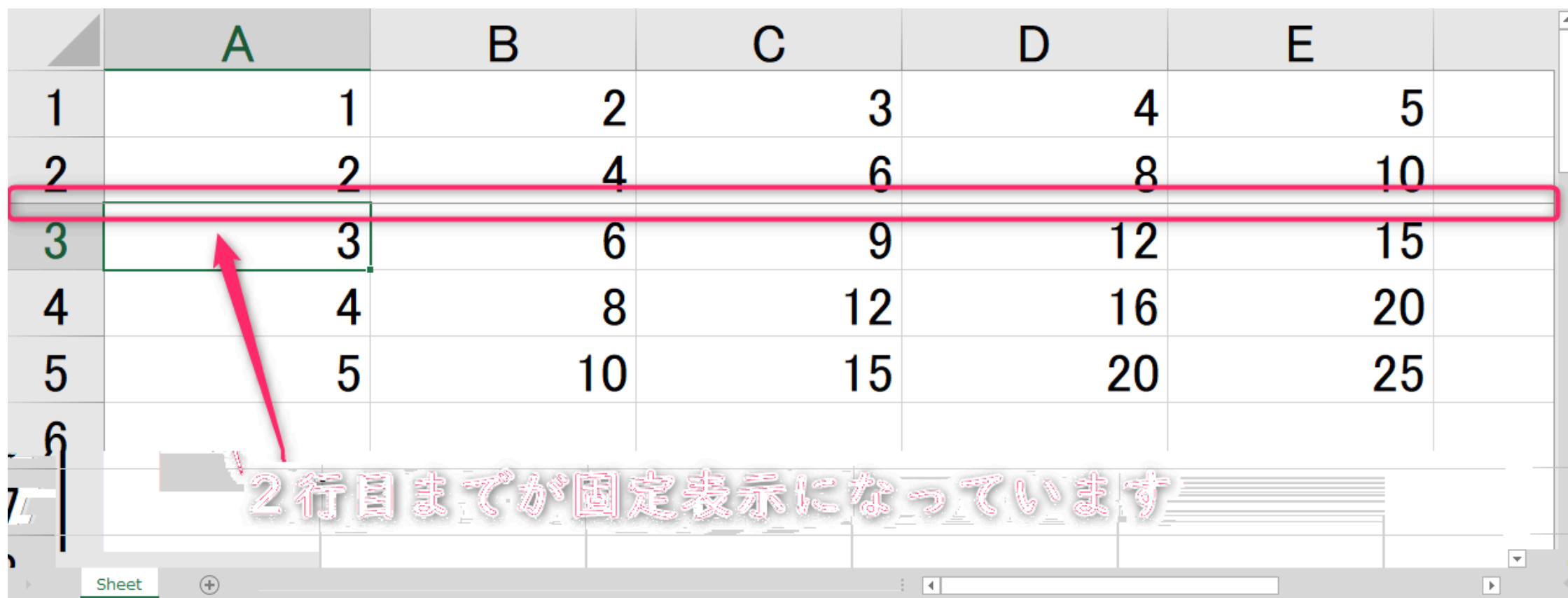
```
wb = op.Workbook()
sheet = wb.active

#セルA1～E5に1～25の数字を入れる
for i in range(1,6):
    for j in range(1,6):
        sheet.cell(j,i).value = i * j

sheet.freeze_panes = 'A3'          # 2行目までを固定:セルA3を指定

wb.save('/content/drive/MyDrive/???/freeze-panes.xlsx')
```

## Excelの行と列を固定表示にする(結果)



The image shows an Excel spreadsheet with columns A through E and rows 1 through 6. Row 2 is highlighted with a red border, indicating it is frozen. A red arrow points from a text box at the bottom to row 2. The text box contains the Japanese text "2行目までが固定表示になっています" (Rows 1 and 2 are frozen).

	A	B	C	D	E
1	1	2	3	4	5
2	2	4	6	8	10
3	3	6	9	12	15
4	4	8	12	16	20
5	5	10	15	20	25
6					

2行目までが固定表示になっています

# 課題



## 課題

- Moodleにある「SCfCL-05-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 と作成した **Excelファイル3つ** をMoodleに 出して ください
- 提出期限は **10月24日(木) 20時まで** です