

Python in Excel:

****Python for Excel Cheat Sheet****

****1. Installing Required Libraries:****

- Install `pandas` library for data manipulation.
- Install `openpyxl` library for Excel file manipulation.

```
```python  
pip install pandas openpyxl
```
```

****2. Importing Libraries:****

```
```python  
import pandas as pd
from openpyxl import load_workbook
```
```

****3. Reading Excel Files:****

- Read an Excel file into a Pandas DataFrame.

```
```python  
df = pd.read_excel('filename.xlsx')
```
```

****4. Writing to Excel:****

- Create a Pandas DataFrame and write it to an Excel file.

```
```python  
df = pd.DataFrame(data)
df.to_excel('output.xlsx', index=False)
```
```

****5. Excel Sheet Manipulation:****

- Load an existing Excel file for manipulation.

```
```python  
wb = load_workbook('filename.xlsx')
```
```

****6. Accessing Excel Sheets:****

- Access a specific sheet in the Excel file.

```
```python  
sheet = wb['SheetName']
```
```

****7. Accessing Cells:****

- Access a cell in the Excel sheet.

```
```python  
value = sheet['A1'].value
```
```

- Modify cell values.

```
```python  
sheet['A1'] = 'New Value'
```
```

****8. Adding New Sheets:****

- Add a new sheet to the Excel file.

```
```python  
new_sheet = wb.create_sheet('NewSheet')
```
```

****9. Applying Formulas:****

- Apply Excel formulas using openpyxl.

```
```python  
sheet['B1'] = '=SUM(A1:A5)'
```
```

****10. Formatting Cells:****

- Apply formatting to cells (font, alignment, borders, etc.).

```
```python  
from openpyxl.styles import Font, Alignment

cell = sheet['A1']
cell.font = Font(bold=True)
cell.alignment = Alignment(horizontal='center', vertical='center')
```
```

****11. Data Analysis with Pandas:****

- Utilize powerful data analysis capabilities of Pandas.

```
```python  
mean = df['column_name'].mean()
filtered_df = df[df['column_name'] > 10]
```
```

****12. Data Visualization:****

- Use libraries like `matplotlib` or `seaborn` for data visualization.

```
```python  
import matplotlib.pyplot as plt
df.plot(kind='bar', x='x_column', y='y_column')
plt.show()
```
```

****13. Advanced Tasks:****

- Advanced operations like merging, pivoting, and more can be performed using Pandas.

****14. Saving Excel Changes:****

- Save the changes made to the Excel file.

```
```python  
wb.save('filename.xlsx')
```
```

Remember to refer to official documentation for detailed explanations and advanced features of both Pandas and openpyxl libraries.