REPUBLIC POLYTECHNIC

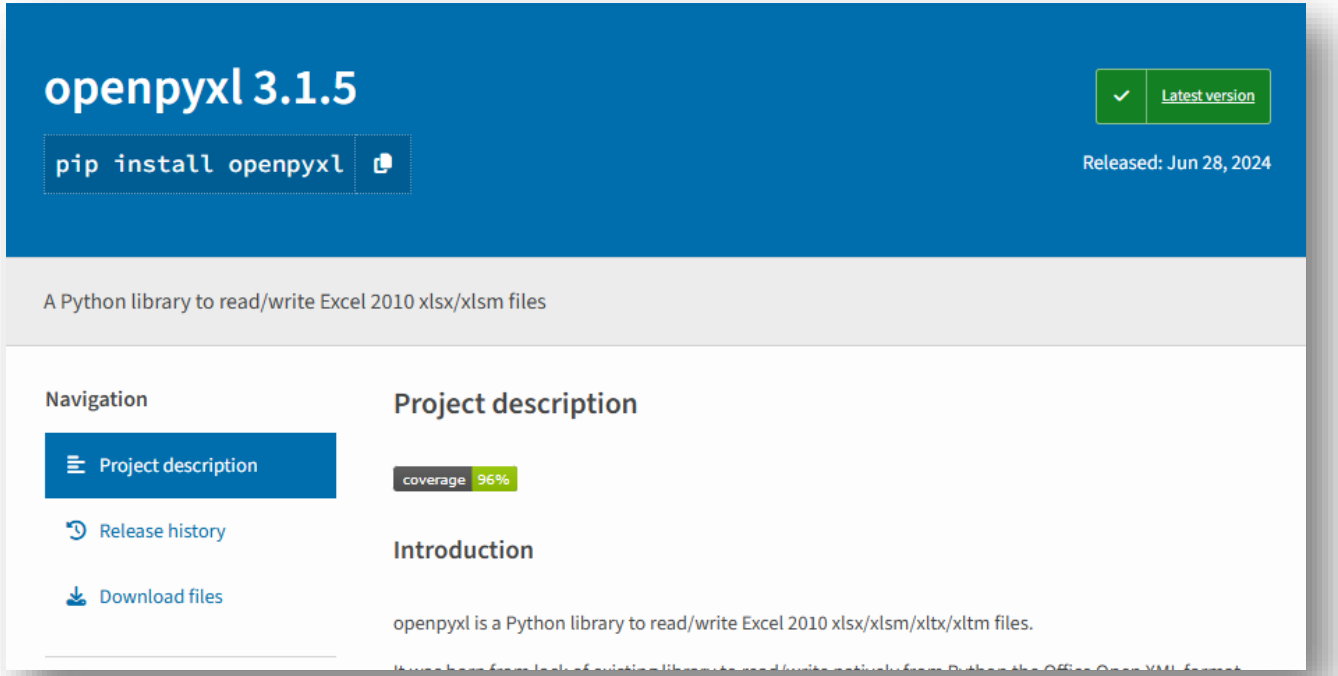
Introductory Programming using Python

Supplement

Excel Spreadsheet Manipulation with Python

Working with Excel

- Install openpyxl module using “pip install openpyxl”
- Full openpyxl documentation: https://openpyxl.readthedocs.io/en/stable/index.html



REPUBLIC POLYTECHNIC

Typical Workflow for Excel Automation

1. You are given some data in a spreadsheet
2. You want to do some or all of the following
   - Analyze the data
   - Manipulate the data
3. Output the processed data in another spreadsheet

Reading Excel file

```
import openpyxl

workbook = openpyxl.load_workbook("bmi.xlsx")
sheet=workbook["Sheet1"]
```

```
name = sheet.cell(row=2, column=1).value
weight = sheet.cell(row=2, column=2).value
height = sheet.cell(row=2, column=3).value
```

```
print("name:%s \tweight: %d \theight: %f " % (name, weight, height))
```

1) Import openpyxl library
2) Load Excel content into “workbook“ object by specifying the file name
3) Get the worksheet named "Sheet1“
4) Get the value of each cell by specifying the row and column
5) Display the retrieved values, only for a row

excel_read_bmi.py

Reading Excel file

- The typical workflow for reading excel file is to use a for loop to go through each row to read the data

```
import openpyxl

workbook = openpyxl.load_workbook("bmi.xlsx")
sheet=workbook["Sheet1"]


max_row = sheet.max_row # get number of rows

#loop through every row
for i in range(2, max_row + 1):

    #read cell
    name = sheet.cell(row=i, column=1).value
    weight = sheet.cell(row=i, column=2).value
    height = sheet.cell(row=i, column=3).value

    print("name:%s \tweight: %d \theight: %f " % (name, weight, height))
```

1) Get the number of rows and columns
2) Use For loop to go through every row
3) Get the value of each cell by specifying the row and column
4) Display the retrieved values, for all rows

excel_read_bmi.py

REPUBLIC POLYTECHNIC

Update Excel file

```
import openpyxl

workbook = openpyxl.load_workbook("bmi.xlsx")
sheet=workbook["Sheet1"]

max_row = sheet.max_row # get number of rows

# add a column header for bmi
sheet.cell(row=1, column=4).value = "bmi"

#loop through every row
for i in range(2, max_row + 1):

    #read cell
    name = sheet.cell(row=i, column=1).value
    weight = sheet.cell(row=i, column=2).value
    height = sheet.cell(row=i, column=3).value

    bmi = weight / (height * height)

    sheet.cell(row=i, column=4).value = bmi

    print("name:%s \tBMI: %f" % (name, bmi))

#save the file
workbook.save("bmi_update.xlsx")
```

1) Access and read the Excel
2) Adds a header at the 4th column
3) Perform calculation with values taken from the excel files
4) Add calculated value to cell
5) Save the spreadsheet to a new file name

excel_update_bmi.py

Create Excel file

- If you have data in nested python list, you can write the data into an excel file.

```
import openpyxl

workbook = openpyxl.Workbook()

#get the default sheet
sheet=workbook["Sheet"]

#create a list of tuples as data source
data = [
    [225.7,'Gone with the Wind','Victor Fleming'],
    [194.4, 'Star Wars', 'George Lucas'],
    [161.0, 'ET: The Extraterrestrial', 'Steven Spielberg']
]

for row in data:
    sheet.append(row)

#save the spreadsheet
workbook.save("movies.xlsx")
```

1) Some data in nested list
2) Using for loop to add each row of data into the excel sheet
3) Save the spreadsheet

excel_create.py

Excel file formatting

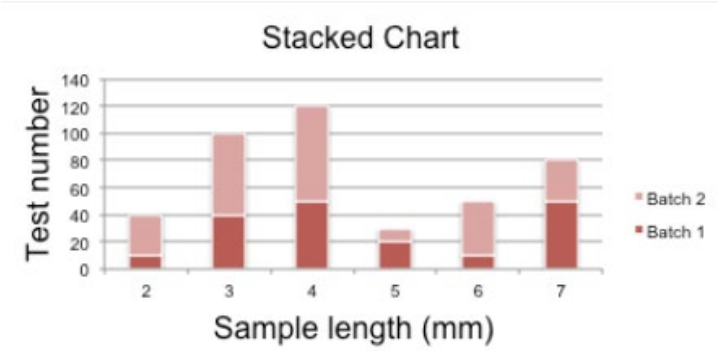
```
import openpyxl
from openpyxl.styles import Font, PatternFill, Border, Side

workbook = openpyxl.load_workbook("bmi.xlsx")
sheet=workbook["Sheet1"]

#define the colors to use for styling
BLUE = "0033CC"
LIGHT_BLUE = "E6ECFF"
WHITE = "FFFFFF"

#define styling
header_font = Font(name="Tahoma", size=14, color=WHITE)
header_fill = PatternFill("solid", fgColor=BLUE)

# format header
for row in sheet["A1:c1"]:
  for cell in row:
    cell.font = header_font
    cell.fill = header_fill

#define styling
white_side = Side(border_style="thin", color=WHITE)
blue_side = Side(border_style="thin", color=BLUE)
alternate_fill = PatternFill("solid", fgColor=LIGHT_BLUE)
border = Border(bottom=blue_side, left=white_side, right=white_side)

# format rows
for row_index, row in enumerate(sheet["A2:C3"]):
  for cell in row:
    cell.border = border
    if row_index %2 :
      cell.fill = alternate_fill

workbook.save("bmi_format.xlsx")
```

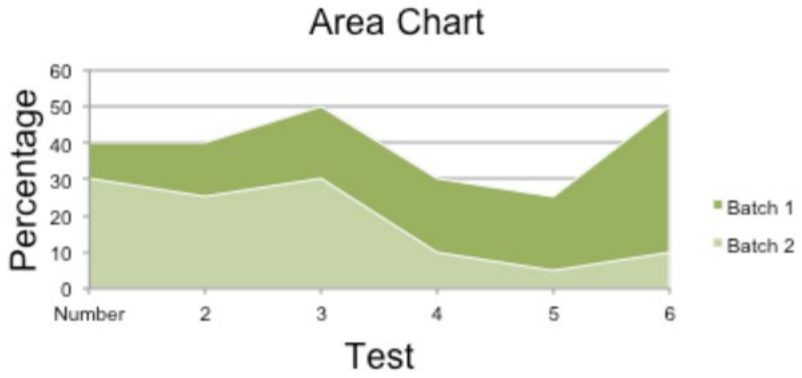
1) Import necessary functions
2) Setup colors and styles
3) Loop through cell and set properties
4) Apply required styles

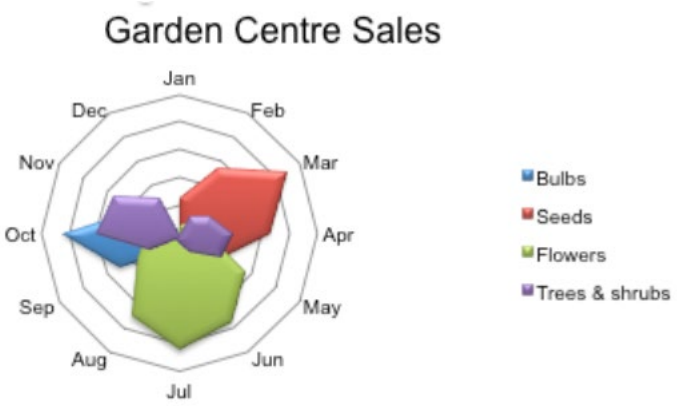
excel_format.py

Excel Charts

- Openpyxl supports the creation of many types of charts
  - Area Charts
  - Bar and Column Charts
  - Bubble Charts
  - Line Charts
  - Scatter Charts
  - etc







https://openpyxl.readthedocs.io/en/stable/charts/introduction.html

Create Excel Chart

```
import openpyxl
from openpyxl.chart import BarChart, Reference, Series

workbook = openpyxl.load_workbook("bmi.xlsx")
sheet=workbook["Sheet1"]


chart = BarChart()

labels = Reference(sheet, min_col=1, min_row=2, max_row=3)
data = Reference(sheet, min_col=3, min_row=1, max_row=3)

chart.add_data(data, titles_from_data=True)
chart.set_categories(labels)
chart.title = "Height"

sheet.add_chart(chart, 'E1')
workbook.save('bmi_chart.xlsx')
```

1) Import necessary functions
2) Load the data from excel file
3) Specify the label and the data range
4) Add the chart to the sheet, and save the file in another excel file.

excel_chart.py

Create Excel Chart

```
import openpyxl
from openpyxl.chart import BarChart, Reference, Series

workbook = openpyxl.load_workbook("bmi.xlsx")
sheet=workbook["Sheet1"]


chart = BarChart()

# first column is used as label, starting from row 2
labels = Reference(sheet, min_col=1, min_row=2, max_row=3)

# first row is used for header, that is why min_row is 1
data = Reference(sheet, min_col=3, min_row=1, max_row=3)

chart.add_data(data, titles_from_data=True)
chart.set_categories(labels)
chart.title = "Bar Chart"
chart.x_axis.title = "Name"
chart.y_axis.title = "Height"
chart.series[0].SeriesLabel = "height"

sheet.add_chart(chart, 'E1')
workbook.save('bmi_chart.xlsx')
```

1) Import necessary functions
2) Load the data from excel file
3) Specify the label and the data range
4) Specify values for title x-axis and y-axis for the chart
5) Add the chart to the sheet, and save the file in another excel file.

excel_chart.py

More Explanation on Chart Reference

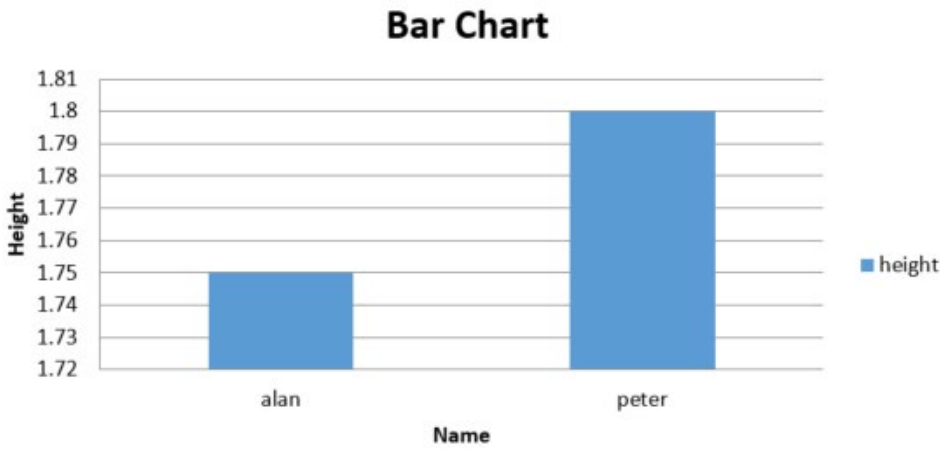
data = Reference(sheet, min_col=3, min_row=1, max_row=3)

	A	B	C
1	name	weight	height
2	alan	65	1.75
3	peter	70	1.8

min_col = 3 ← height

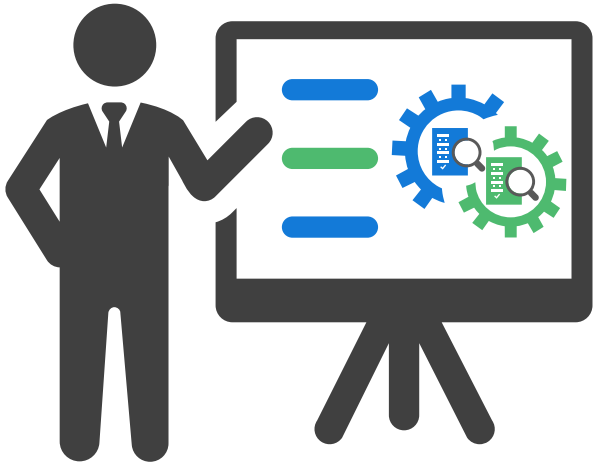
min_col = 2 ← weight

labels = Reference(sheet, min_col=1, min_row=2, max_row=3)



REPUBLIC POLYTECHNIC

Thank you



Learning material & source code:
https://bit.ly/IPP-July2025

Email us at:

jason_lim@rp.edu.sg