REPUBLIC POLYTECHNIC

Introductory Programming Using Python

Day 2

Republic Polytechnic

Tea Break

Vending Machine Code

851475

Morning:
9.30 am – 11.00am

Afternoon:
2.30pm – 4.30pm

Programme Day Two

Morning	Afternoon
• Read and writing files • Copying, moving and deleting files and folders • Working with Excel	• Image Processing • Connecting to the Web • Sending emails

Outline for the day

Time	Agenda
9.00am	Welcome and admin matters
9.15am – 10.30am	
10.30am – 10.45am	Break
10.45am – 12.30pm	
12.30pm – 1.30pm	Lunch
1.30pm – 3.15pm	
3.15pm – 3.30pm	Break
3.30pm – 4.30pm	
4.45pm – 5.00pm	Wrap up, Q&A

File Paths

Absolute file paths are notated by a **leading forward slash or drive label**.
For example,
`/home/example_user/example_directory`

or

`C:/system32/cmd.exe`

An absolute file path describes how to access a given file or directory, starting from the root of the file system.

Relative file paths are notated by a **lack of a leading forward slash**.
For example,
`example_directory`.

A relative file path is interpreted from the perspective your current working directory. If you use a relative file path from the wrong directory, then the path will refer to a different file than you intend, or it will refer to no file at all..

Read files

```
1  # make sure you have a hello.txt in your current working director
2  # same directory as your python script
3  helloFile = open("hello.txt")
4  content = helloFile.read()
5  print(content)
6  helloFile.close()
7
8  # make sure you have a hello.txt in the specified director
9  # same directory as your python script
10 helloFile = open("hello.txt")
11
12 content = helloFile.readlines()
13 print(content)
14
```

Search | Stack Data

Search:
Replace:
Case sensitive | Whole words | In Selection

Debug I/O | Python Shell

Commands execute without debug. Use arrow keys for history. Options

```
3.7.4 (tags/v3.7.4:e09359112e, Jul  8
Python Type "help", "copyright", "cre
>>> [evaluate file_read_01.py]
THis is also another line
Hello world again

['THis is also another line\n', 'Hello world again\n']
```

Open() will return a file object which has reading and writing related methods

Pass ‘r’ (or nothing) to open() to open the file in read mode.

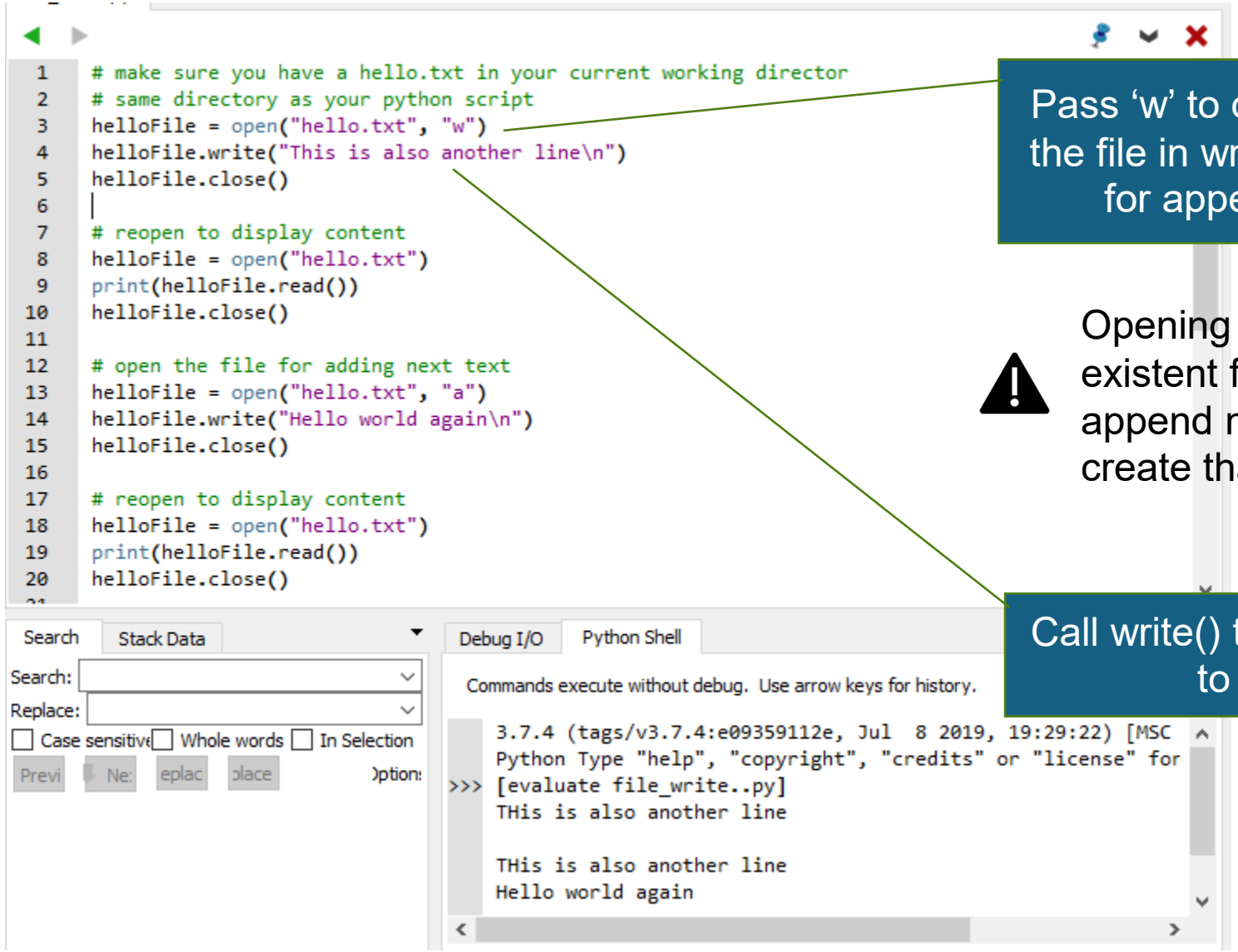
Call read() to read the contents of a file

Call close() when you are done with the file.

- Call readLines() to read the contents of a file

file_read.py

Write files



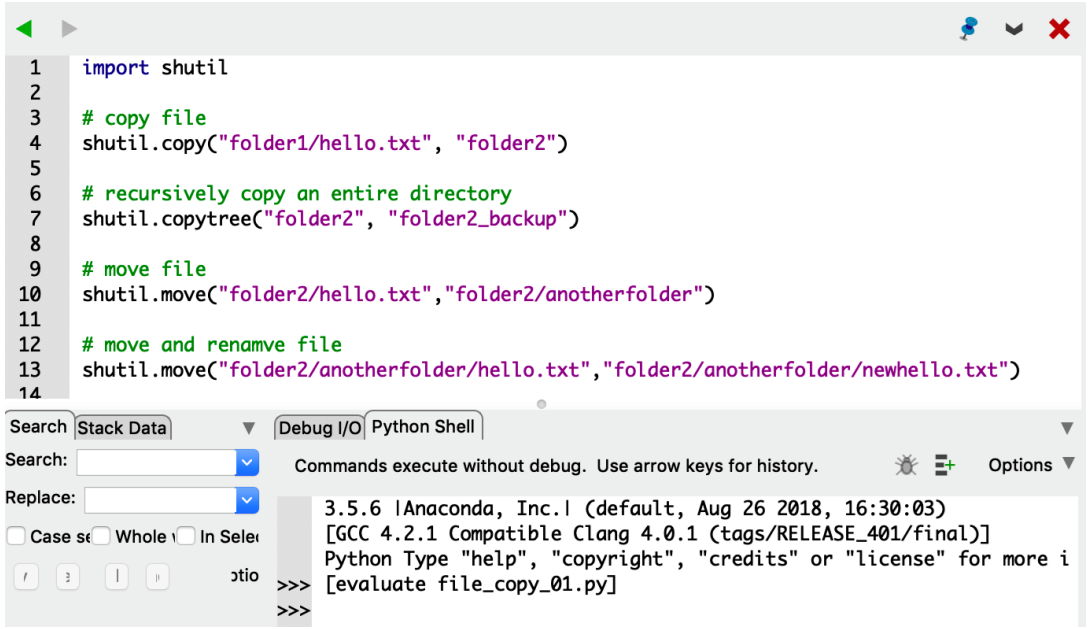
Pass ‘w’ to open() to open the file in write mode or ‘a’ for append mode.

Opening a non-existent file in write or append mode will create that file

Call write() to write a string to a file.

file_write.py

Copy and moving files



- shutil.copy(src, dst) – Copy the file *src* to the file or directory *dst*
- shutil.copytree(src, dst) - Recursively copy an entire directory tree rooted at *src*.
- shutil.move(src, dst) - Recursively move a file or directory (*src*) to another location (*dst*).

https://docs.python.org/3/library/shutil.html

file_copy.py

Deleting files


```python
import os

# error if file do not exist
os.unlink("hello.txt")

# get current working directory
print(os.getcwd())

# delete directory (can only delete empty folder)
os.rmdir("folder3")

import shutil
# delete directory (with content)
# error if folder is not found
shutil.rmtree("folder3")
```

- os.unlink() will delete a file
- os.rmdir() will delete a folder (but folder must be empty)
- shutil.rmtree() will delete a folder and all its contents

e.g. To delete all .docx file in the current folder

```python
import os

for filename in os.listdir():
    if filename.endswith(".docx"):
        print(filename)
        os.unlink(filename)
```

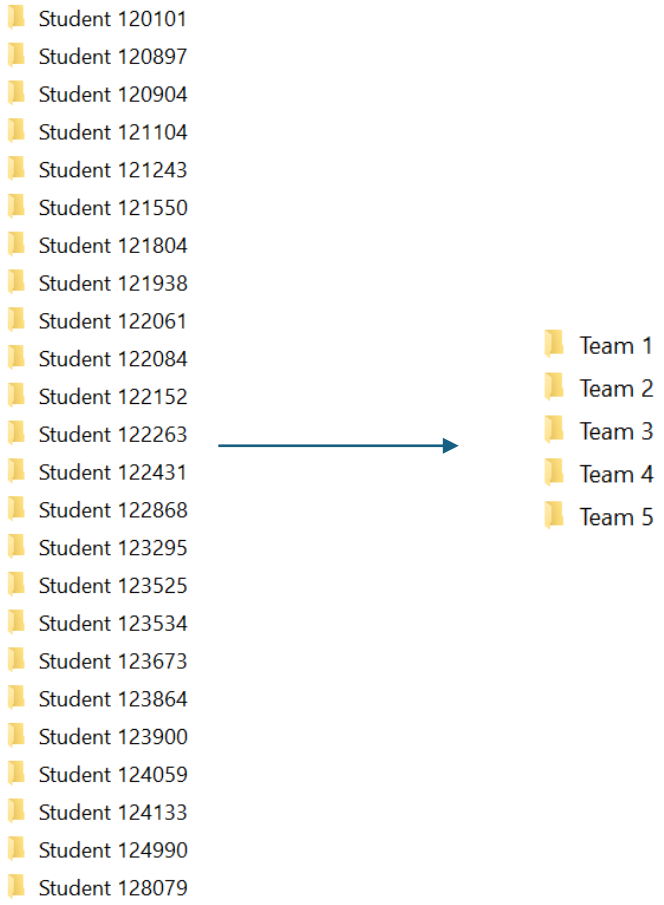
Deleting can be dangerous, so do a dry run first

file_delete.py

Use Case Sharing

- Organizing students' submissions into separate folder
  - Class of 25 students

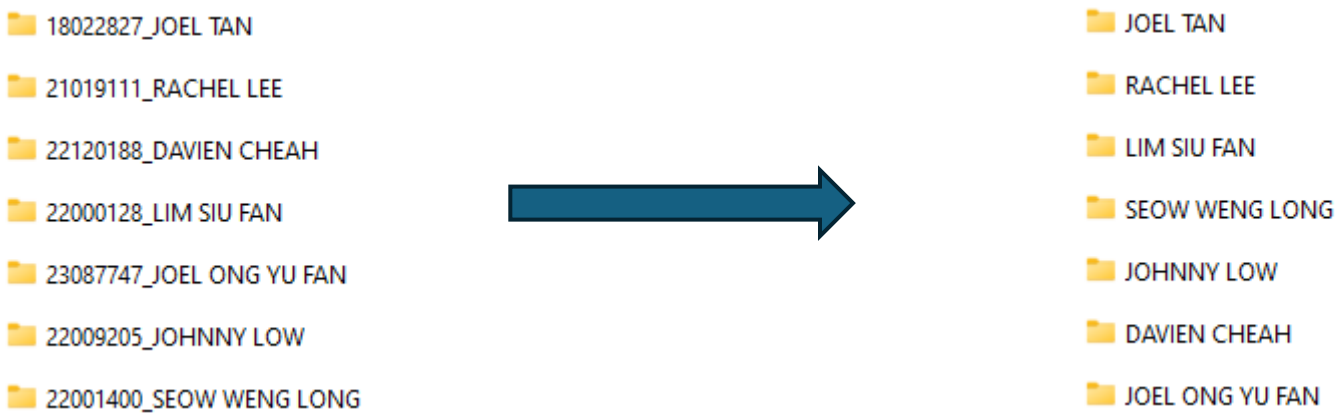
25 folders, one for each student



Student submission sorted by teams

Exercise: Coding with ChatGPT

Use ChatGPT to generate the code to rename all the folders based on the requirement shown below



Python Package Index

- https://pypi.org/
- A repository of software for the Python Programming Language
- Python Installation provides the core libraries needed for the common tasks
  - Additional packages can be found at the website and installed as extension
    - E.g. send2trash, openpyxl, pillow etc

- Installation is easy done with the following command
  - **pip install <software_package>**

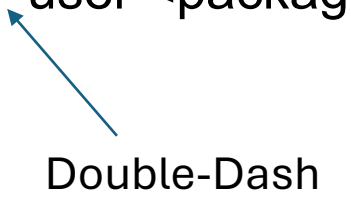
- Installed packages can be found at:
  - C:\<python312>\Lib\site-packages

Using pip install

- For all windows users by default
  - Open command prompt
  - pip install <package_name>

- For Mac User
  - Open terminal
  - **pip3** install <package_name>

- For staff using company issued laptop with no Admin rights
  - Open command prompt
  - pip install --user <package_name>



Using pip install

REPUBLIC POLYTECHNIC

- Install 3rd party library using python code
  - This is the preferred approach if there is multiple installation of python in your laptop, and you are not sure which version of python you are currently working on.

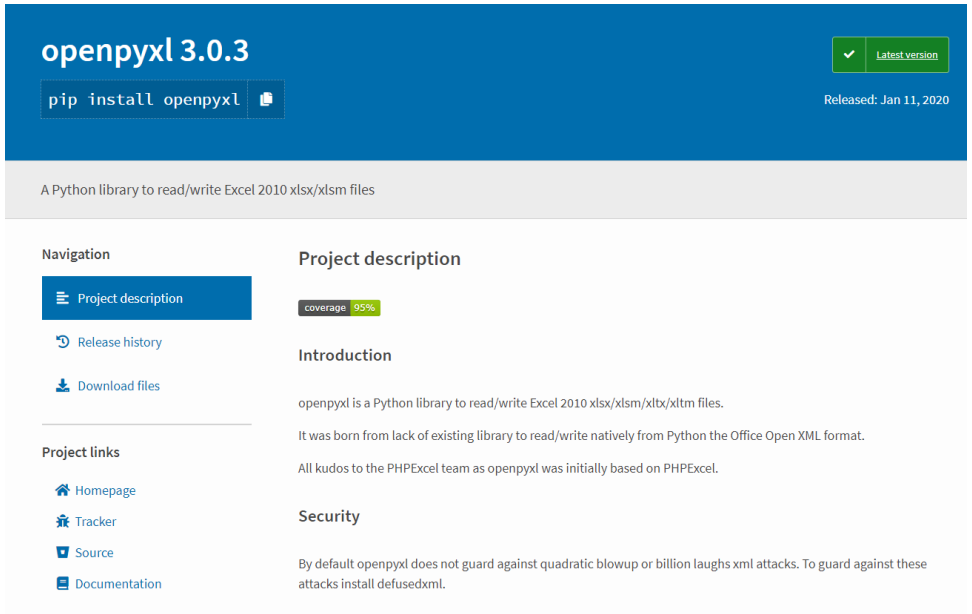
```python
import subprocess
import sys

# replace "fpdf" with the library you plan to install
subprocess.check_call([sys.executable, "-m", "pip", "install", "fpdf"])
```

install.py

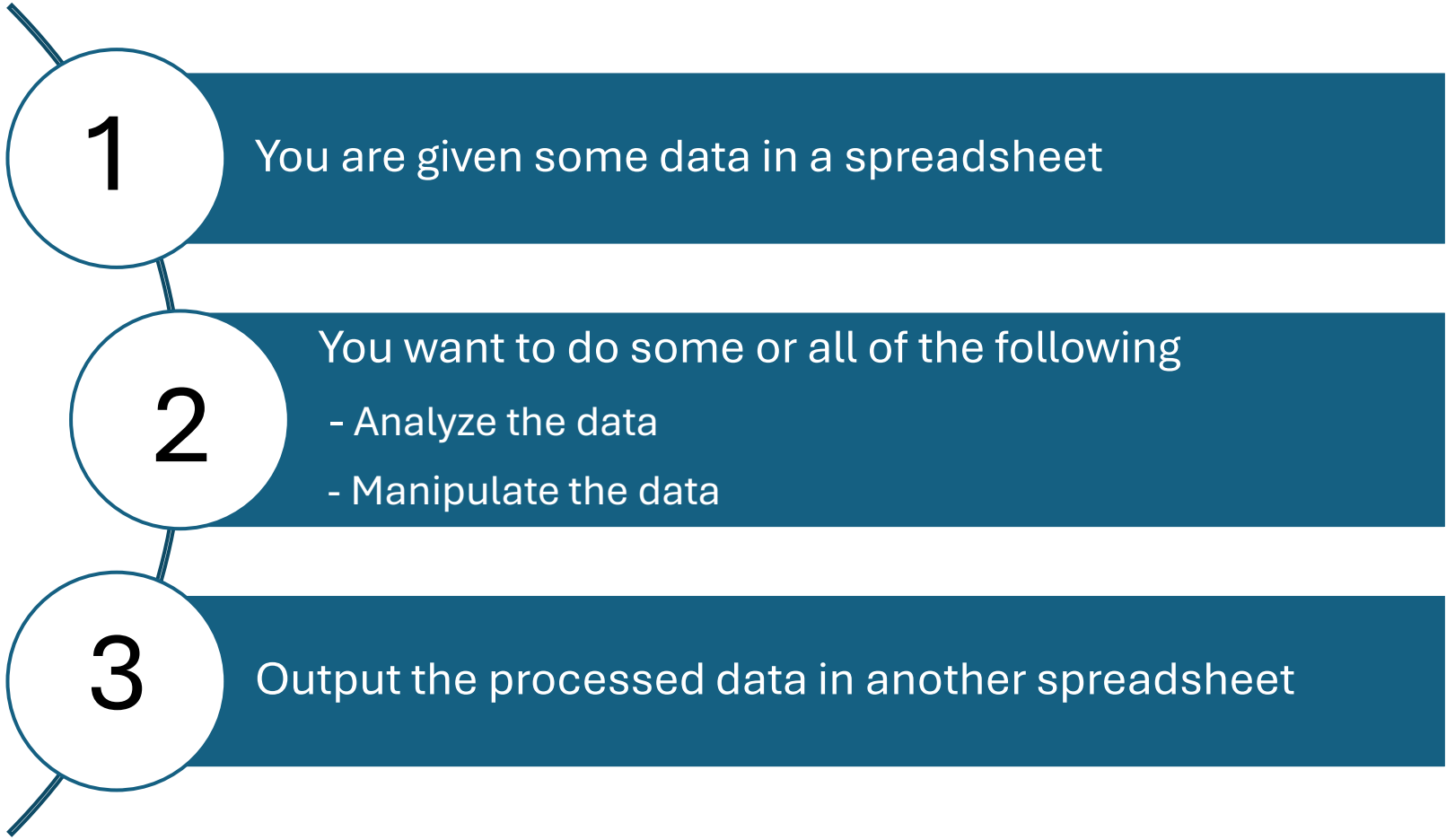
Excel Spreadsheet Manipulation with Python

Working with Excel

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



Typical Workflow for Excel Automation



Reading Excel file

1) Import openpyxl

```
import openpyxl

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

2) Load Excel content into “workbook“ object by specifying the file name

3) Get the worksheet named "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
```

4) Get the value of each cell by specifying the row and column

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

5) Get the value of each cell by specifying the row and column

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) Extract the status at Column C to check for attendance

excel_read_bmi.py

Update Excel file

REPUBLIC POLYTECHNIC

```
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")
```

2) Load file into memory & get the sheet

1) Perform calculation with values taken from the excel files

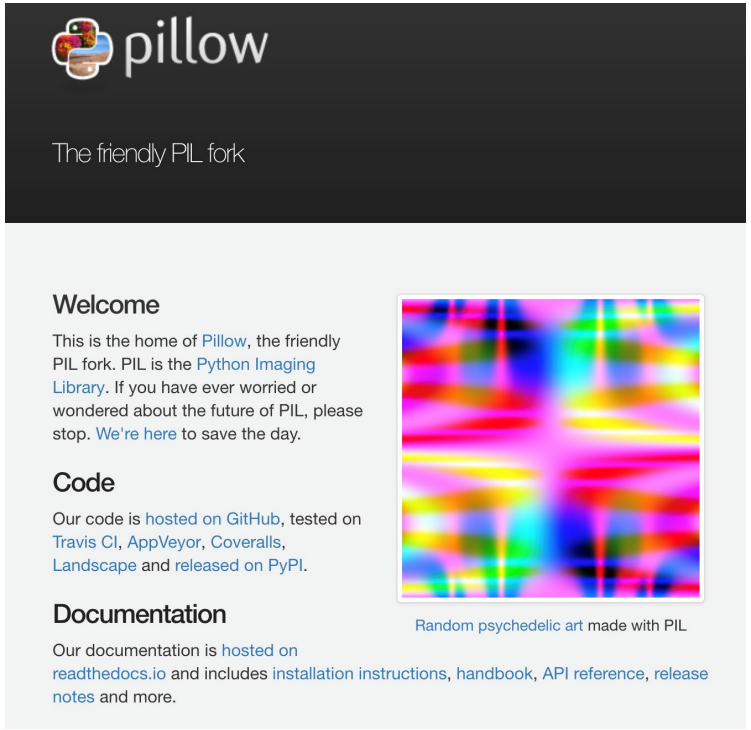
2) Add comments to cell

5) Save the spreadsheet

excel_update_bmi.py

Image Processing with Python

Image Processing



For the next section we are going to use the Python Image Library, or in short Pillow.

Install using the following command:
pip install pillow

The documentation is at:
https://pillow.readthedocs.io/en/stable/handbook/overview.html

Image Processing

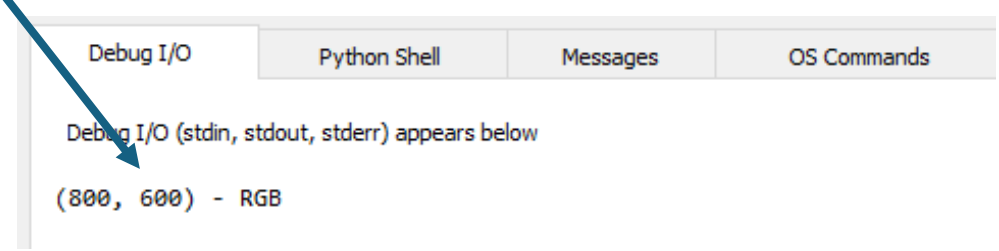
```
import os
from PIL import Image

filename = "img/clungup.jpg"

im = Image.open(filename)
print ("%s - %s" % (im.size, im.mode))

# show the image
im.show()

# close the file
im.close()
```

As a start we need to import it:
import Image

We can open images with
im = Image.open(fullname)

Then we can display the image using
im.show()

Print some info about the image using
im.size and im.mode



Size: 800 x 600, Mode: RGB

image_show_info.py

Image Processing

REPUBLIC POLYTECHNIC

```
import os
from PIL import Image, ImageFilter

filename = "img/clungup.jpg"

im = Image.open(filename)

out = im.filter(ImageFilter.BLUR)

im.show()
out.show()
```

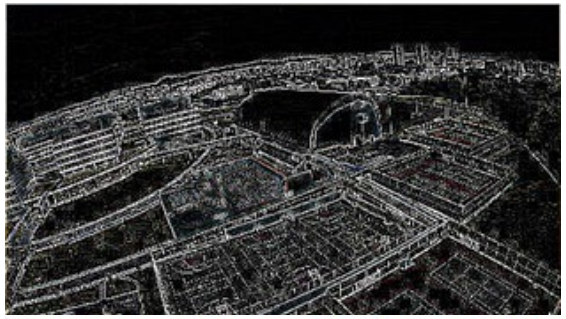
Pillow has many conversion and filters, to use filters we need to extend our import:
from PIL import Image, ImageFilter

The way you can apply filters is :
out = im.filter(ImageFilter.BLUR)

Try other different filters!

image_blur_filter.py

Image Processing - Filters



```
image = image.filter(ImageFilter.FIND_EDGES)
```

```
image = ImageOps.grayscale(image)
```

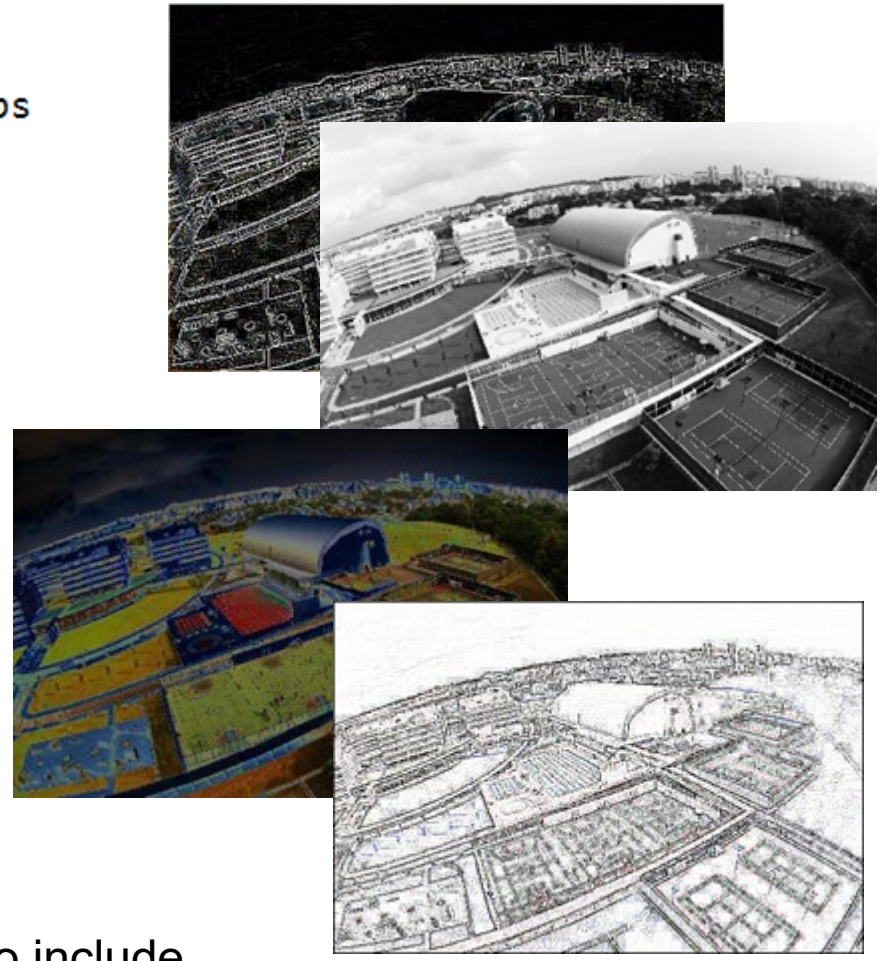
```
image = ImageOps.solarize(image)
```

```
image = image.filter(ImageFilter.CONTOUR)
```

⚠ * Remember to include ImageOps in your import statement

Image Processing - Filters

```
import os
from PIL import Image, ImageFilter, ImageOps

filename = "img/clungup.jpg"

im = Image.open(filename)

# Filter
#out = im.filter(ImageFilter.BLUR)
#out = im.filter(ImageFilter.FIND_EDGES)
#out = im.filter(ImageFilter.CONTOUR)

# ImageOps
out = ImageOps.grayscale(im)
#out = ImageOps.solarize(im)

im.show()
out.show()
```



⚠ * Remember to include ImageOps in your import statement

image_other_filters.py

Image Processing - Rotating

Flipping the image horizontally or vertically
out = im.transpose(Image.FLIP_LEFT_RIGHT)
out = im.transpose(Image.FLIP_TOP_BOTTOM)

Flip images

Rotating the image
out = im.transpose(Image.ROTATE_90)
out = im.transpose(Image.ROTATE_180)
out = im.transpose(Image.ROTATE_270)

Rotate images

Contrast
First add ImageEnhance to our imports:
from PIL import Image, ImageFilter, ImageEnhance

Then:

enh = ImageEnhance.Contrast(im)
out = enh.enhance(1.3)

make image brighter by changing the contrast

image_rotate.py

Image Processing - Writing

```
import os
from PIL import Image, ImageFilter, ImageOps

filename = "clungup.jpg"

src_folder = "img/"
out_folder = "out/"

im = Image.open(src_folder + filename) # img/clungup.jpg
out = im.filter(ImageFilter.BLUR)

outFilename = out_folder + filename # out/clungup.jpg

out.save(outFilename)
```

You can see the image, but it's not being saved !

All you need to do to save the images in the "out" folder is: out.save(the name of the output file)

image_save.py

Image processing – Watermark

Create the mark image
You can reduce the size to 100,100

```
mark = Image.open("img\\watermark.png")
mark = mark.resize((100,100))
```

Create a new function called

```
def watermark(im, mark, position):
        ....
```

It takes the original image, the watermark image and the desired position that we want the watermark to appear. The function will return the result.

We can use this function like:

```
watermark(im, mark, (0, 50)).show()
```

or

```
imOut = watermark(im, mark, (0,50))
imOut.save(fileOut)
```

Maybe you want to leave a small footprint on your images, called watermark.

In this case we can use the \\img\\watermark.png and place it in each image on the bottom right.

Copyright @RP

Image processing – Watermark

```
from PIL import Image

def watermark(im, mark, position):
    layer = Image.new("RGBA", im.size, (0,0,0,0))
    layer.paste(mark, position)
    return Image.composite(layer, im, layer)

im = Image.open("img\\clungup.jpg")
mark = Image.open("img\\watermark.png")
mark = mark.resize((100,100))
mark.putalpha(128)

out = watermark(im, mark, (0,0))
out.show()
```

First we need to create a new layer with the size of the original image.

Then we paste the watermark image at the desired position and we return the composite.

Finally we merge the image and the layer together and return the result.

Then you can use it like this:

image_watermark.py

Use Case I: Batch Resize

REPUBLIC POLYTECHNIC

1. Find all the files in “img” folder with “.jpg” extension
2. Resize all the file to 60 x 90.
3. Save all the files to the resized folder

```
import os
from PIL import Image, ImageFilter, ImageOps

files = os.listdir('img')
size = 60, 90

for file in files:
    if file.lower().endswith(".jpg"):
        im = Image.open("img/" + file)
        im.thumbnail(size, Image.ANTIALIAS)
        im.save("resized/" + file, "JPEG")
```

image_batch_resize.py

Web Automation with Python

Connecting to the Web

REPUBLIC POLYTECHNIC

- requests – download files and web pages from the Web

pip install requests

Get the required information from the given URL

```
import requests

url="https://api.data.gov.sg/v1/environment/24-hour-weather-forecast"
req=requests.get(url)
print(req.text)
```

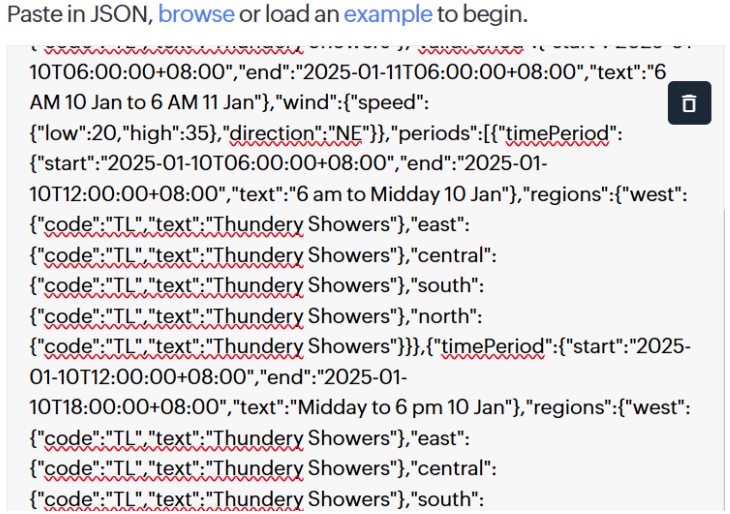
web_request.py

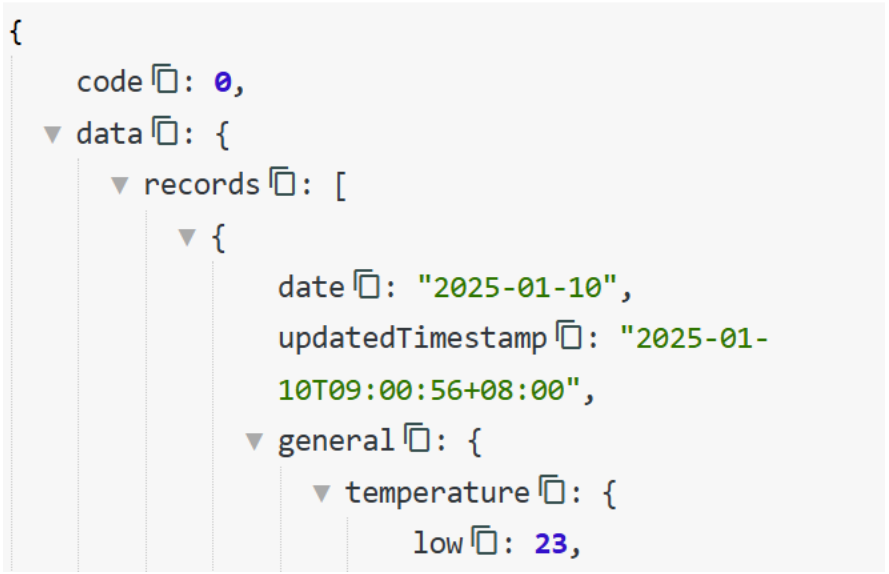
Connecting to the Web

- Data is in JSON format
- Use a JSON formatter tool to present the data in a nicer form

https://www.site24x7.com/tools/jsonpath-finder-validator.html

```
import json
import requests

url="https://api-open.data.gov.sg/v2/real-time/api/twenty-four-hr-forecast"

req=requests.get(url)

data = json.loads(req.text)
```





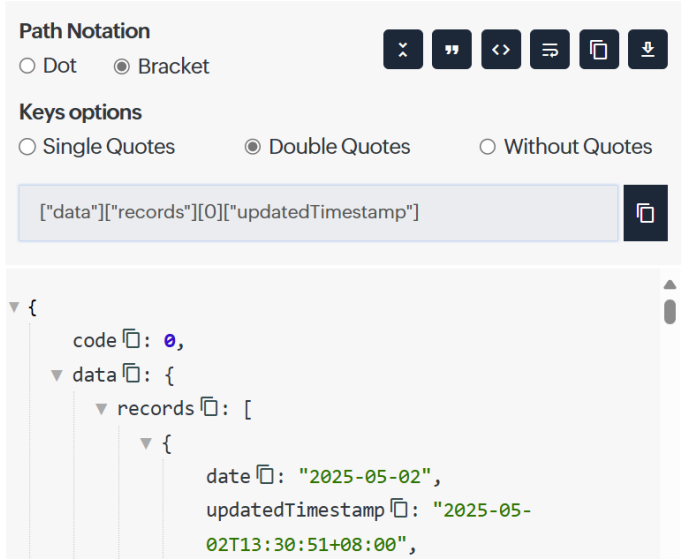
Connecting to the Web

REPUBLIC POLYTECHNIC

- To work with JSON data, import json first
- Use json.loads() to load the data in JSON format
- Extract and retrieve the required data

```
Debug I/O | Exceptions | Python Shell | Messages | OS Commands

Debug I/O (stdin, stdout, stderr) appears below

Update time: 2025-01-10T09:00:56+08:00
Forecast: Thundery Showers
```

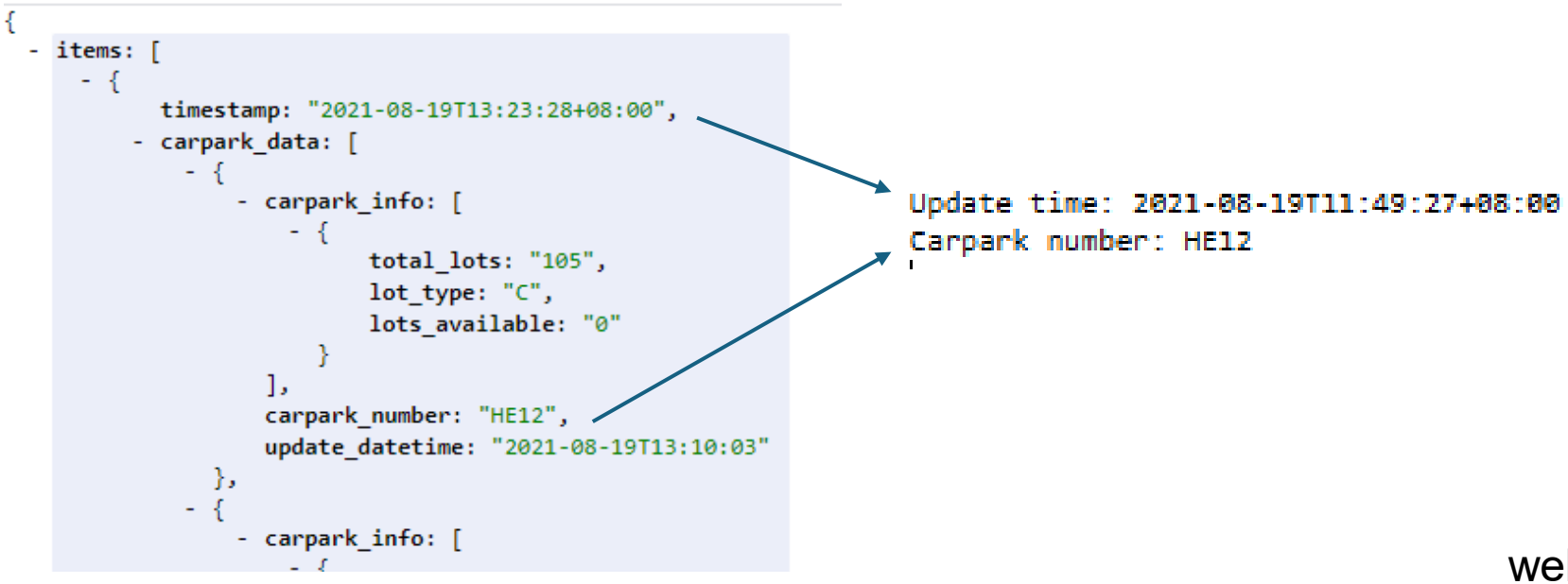
```python
import json
import requests

#url="https://api.data.gov.sg/v1/environment/24-hour-weather-forecast"
url="https://api-open.data.gov.sg/v2/real-time/api/twenty-four-hr-forecast"

req=requests.get(url)

data = json.loads(req.text)

# print update timestamp
update_time = data["data"]["records"][0]["updatedTimestamp"]
print("Update time: " + update_time)

# print forecast
forecast = data["data"]["records"][0]["general"]["forecast"]["text"]
print("Forecast: " + forecast)
```



web_json.py

Exercise

- Car Park Availability Data:
  1. url: https://api.data.gov.sg/v1/transport/carpark-availability
  2. Write the code to get the timestamp and the Carpark Number for the first set of carpark data.
  3. Print out the result as shown.



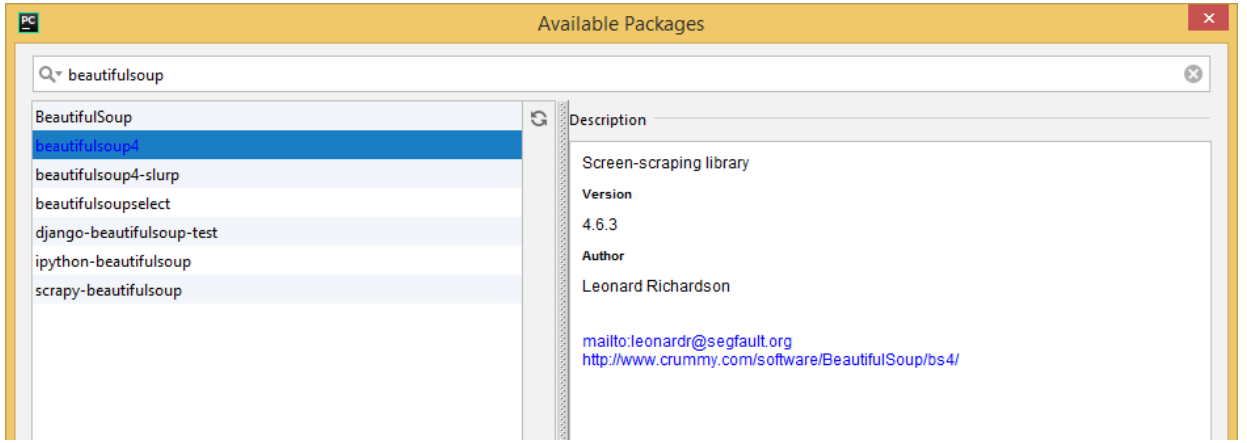
web_json_carpark.py

Connecting to the Web

- Beautiful Soup – a third party module that parses HTML (web pages)

> Web Scraping – download and process Web content

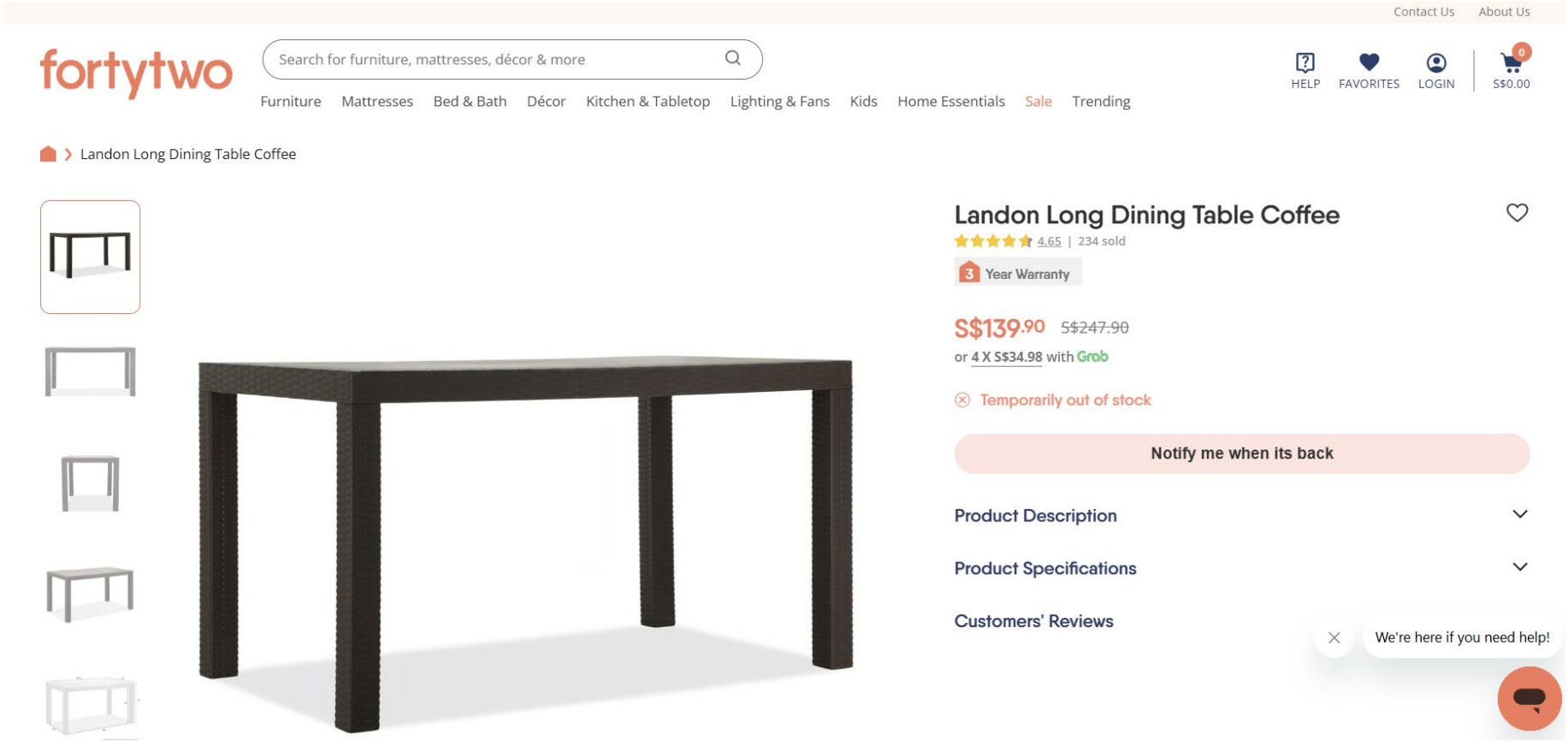
- Install Beautiful Soup 4 - **pip install beautifulsoup4**



Connecting to the Web

- What's the URL?

https://www.fortytwo.sg/landon-long-dining-table-coffee.html

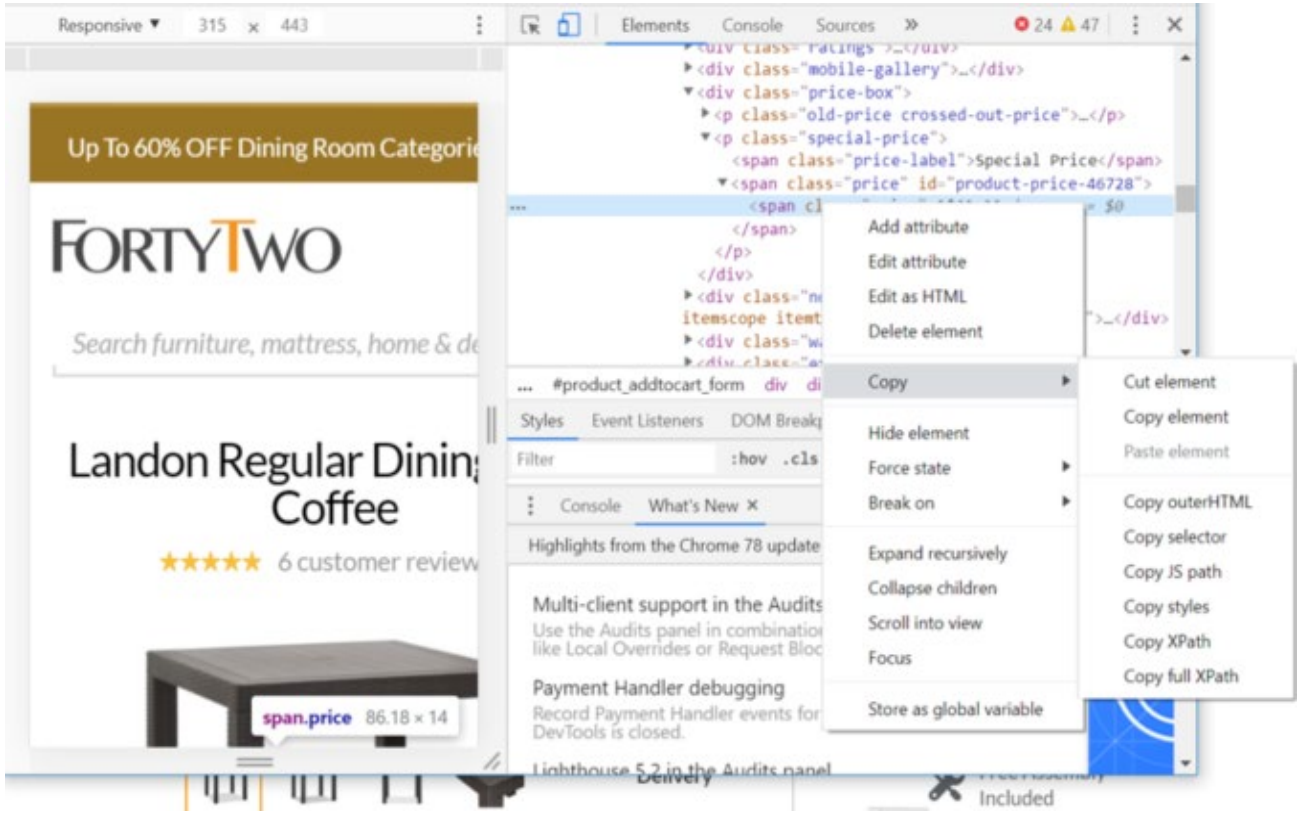


Connecting to the Web

- Get the url

https://www.fortytwo.sg/landon-long-dining-table-coffee.html

- Select the element to extract, right-click "Inspect"
- Right-click "Copy" □ "Copy selector



Connecting to the Web

REPUBLIC POLYTECHNIC

```
from urllib.request import Request, urlopen
from bs4 import BeautifulSoup

site = "https://www.fortytwo.sg/landon-long-dining-table-coffee.html"

hdr = {'User-Agent': 'Mozilla/5.0'}
req = Request(site,headers=hdr)
page = urlopen(req)
soup = BeautifulSoup(page, 'html.parser')

#new-price
new_price_element = soup.find(class_="new-price")

print("Current Price: " + new_price_element.text)


#old-price
old_price_element = soup.find(class_="old-price")

print("\nOld Price: " + old_price_element.text)
```

Debug I/O | Exceptions | Python Shell | Messages

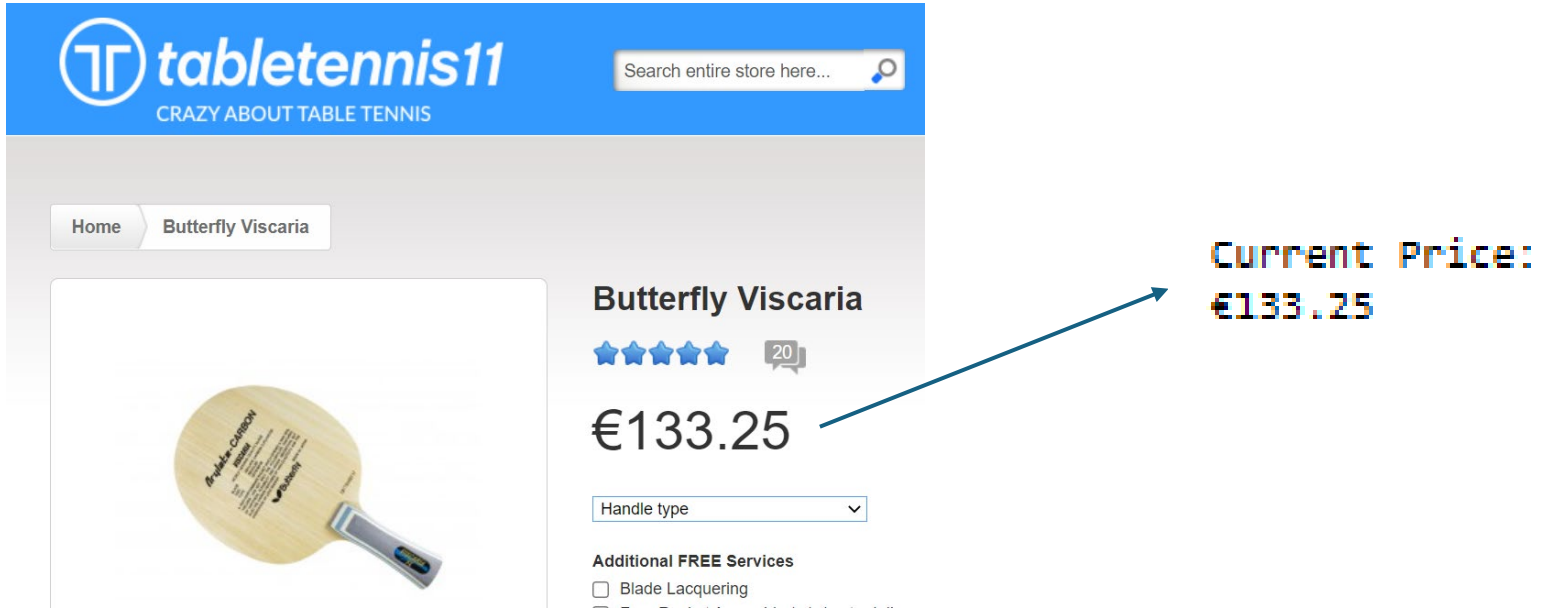
Debug I/O (stdin, stdout, stderr) appears below

```
Current Price: SGD139.90

Old Price: SGD247.90
```

web_scrap.py

Exercise

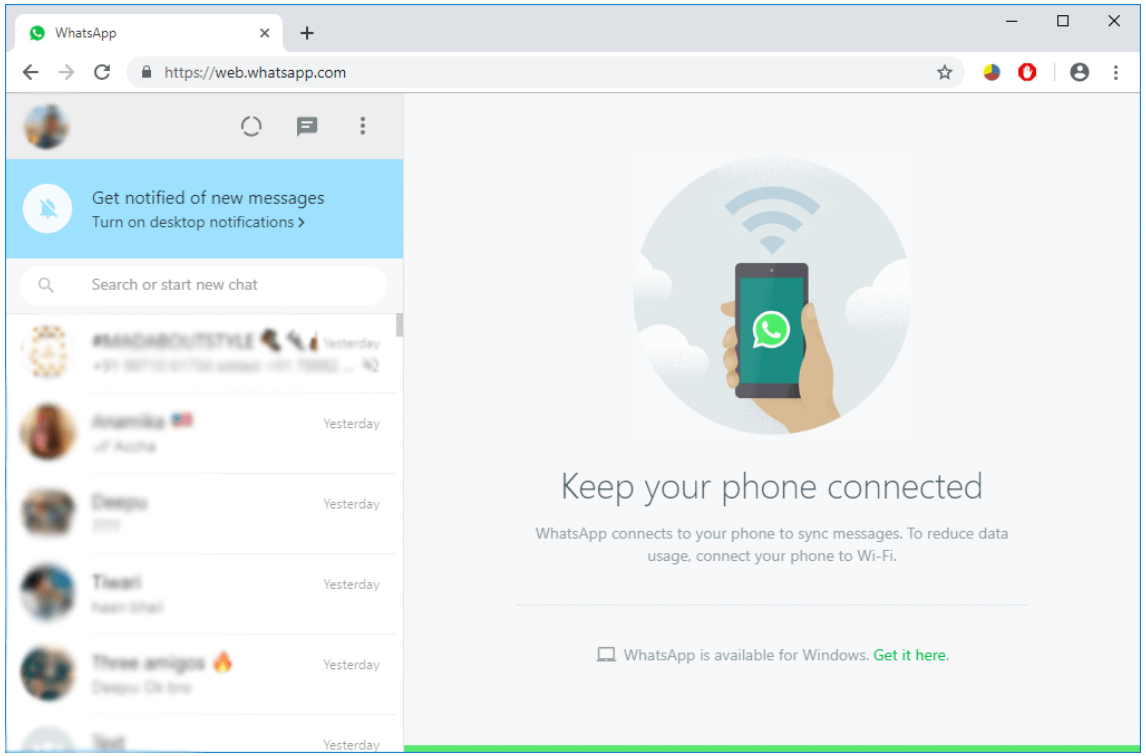
- Table Tennis Bat Price:
  1. url: https://www.tabletennis11.com/other_eng/butterfly-viscaria
  2. Write the code to get the price of the table tennis bat
  3. Print out the result as shown.

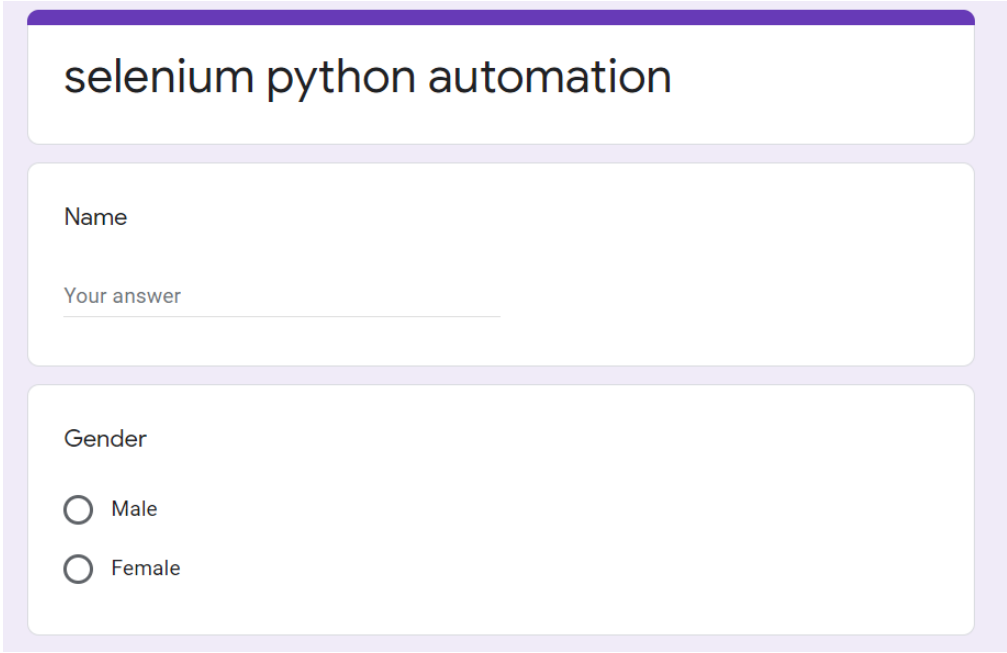


web_scrap_tabletennis.py

Sharing other Use Cases

- Using another library: selenium
  - Filling up google form
  - Sending WhatsApp message





Other public APIs

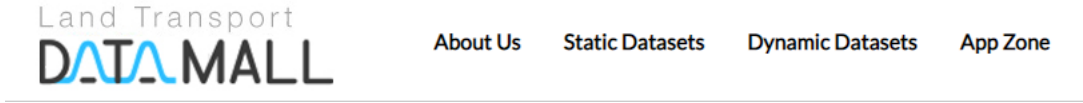
REPUBLIC POLYTECHNIC

- Open Weather API
  - openweathermap.org/api
- TMDB – Movie API
  - themoviedb.org/documentation/api
- News API
  - newsapi.org
- Jokes API
  - sv443.net/jokeapi/v2
- Quotes API
  - api.quotable.io

LTA DataMall

- LTA Datamall:
  - https://datamall.lta.gov.sg/content/datamall/en/dynamic-data.html

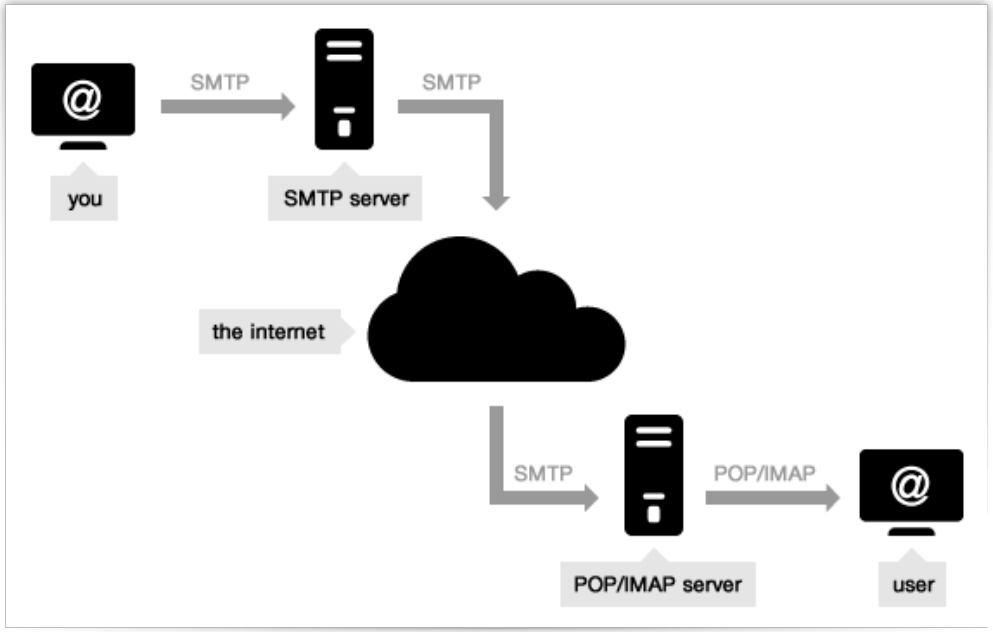
- Sample Dataset available
  - Bus Arrival
  - Bus Routes



- Get Bus Arrival Time
  - Get bus stop number:
    - https://www.transitlink.com.sg/eservice/eguide/service_route.php
  - Get buses and its arrival time

Email Automation with Python

Send Email

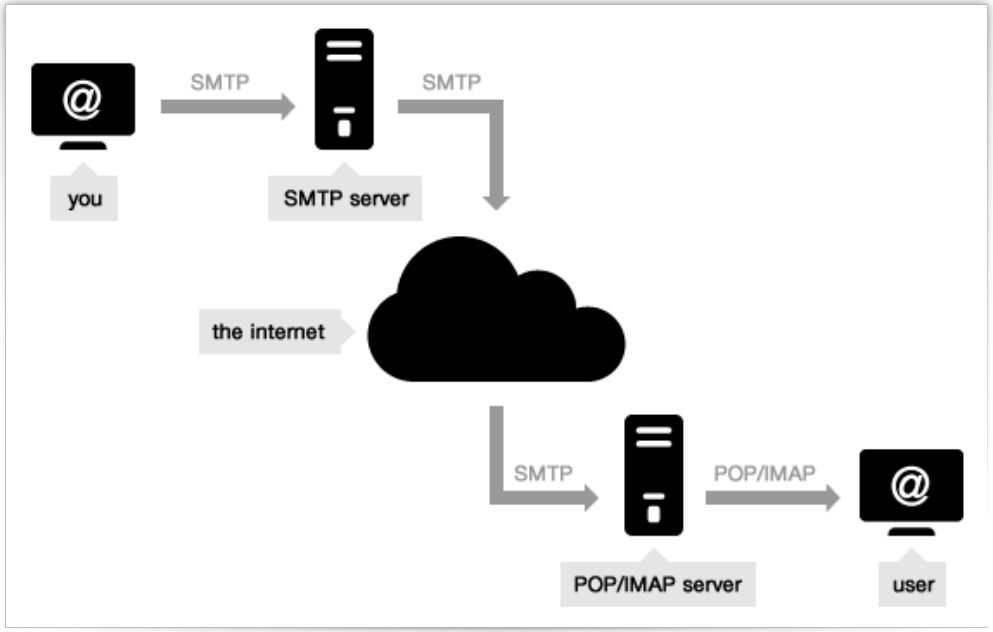


- SMTP (Simple Mail Transfer Protocol) is used for sending and delivering from a client to a server via port 25, 465 or 587: it's the **outgoing server**.
- IMAP and POP are two methods to access email. IMAP is the recommended method when you need to check your emails from several different devices, such as a phone, laptop, and tablet.

https://www.mailgun.com/blog/which-smtp-port-understanding-ports-25-465-587/

https://serversmtp.com/what-is-smtp-server/

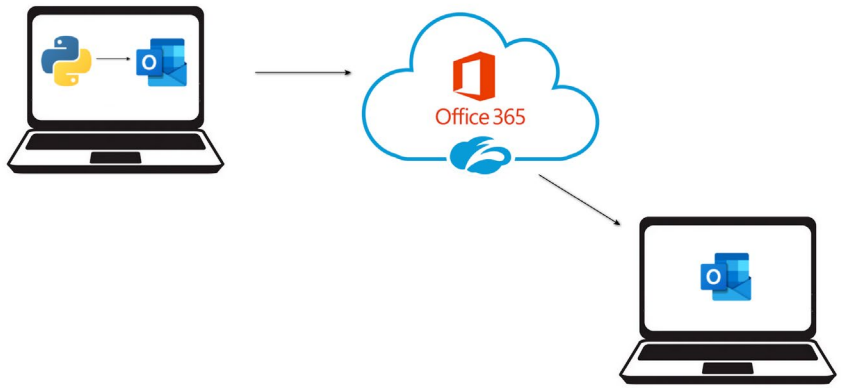
Send Email



- **Note**: The SMTP servers used when you send your emails- Hotmail, Gmail , Yahoo Mail – are **shared among users**
- Common providers establish some **strict limits** on the number of emails you can send (e.g. Yahoo's restriction is 100 emails per hour).
- If you plan to send a bulk email or set up an email campaign you should opt for a professional outgoing email server like turboSMTP,
- which guarantees a controlled IP and ensure that all your messages reach their destination.

pywin32 package

- This assumes that we are using MS Outlook on the machine the python code is run
- pywin32 package is used to allow Python to access MS Outlook
- Python is not accessing the Outlook server directly, but it is using existing account set up in the MS Outlook



Send Email using Outlook

REPUBLIC POLYTECHNIC

- The code above sends a simple HTML (meaning we can have tables, text formatting in it)
- No credentials are needed for the Python program as Python uses MS Outlook

```
## REMEMBER to perform a pip install pywin32 first.

import win32com.client

outlook = win32com.client.Dispatch('outlook.application')
mail = outlook.CreateItem(0)
mail.To = 'tan_kok_cheng@rp.edu.sg'
mail.Subject = 'Demo Email on 20 Nov 2024'
mail.HTMLBody = '<h3>This is an email sent from Outlook client on 20 Nov 2024...</h3>'
#mail.Body = "This is the normal body"
#mail.Attachments.Add('c:\\sample.xlsx')
#mail.Attachments.Add('c:\\sample2.xlsx')
#mail.CC = 'somebody@company.com'
mail.Send()
print("Sent")
```

Create appt using Outlook

- The code above creates a calendar appointment
- No credentials are needed for the Python program as Python uses MS Outlook

```
import win32com.client

outlook = win32com.client.Dispatch('outlook.application')
appt = outlook.CreateItem(1)
appt.Start = "2024-11-20 16:00"
appt.Subject = "Online Sales Starts!"
appt.Duration = 60
appt.Location = "Singapore"

#mail.Attachments.Add('c:\\sample.xlsx')
#mail.Attachments.Add('c:\\sample2.xlsx')
#mail.CC = 'somebody@company.com'
appt.MeetingStatus = 1

appt.Recipients.Add("tan_kok_cheng@rp.edu.sg")

appt.Save()
appt.Send()
print("Appt made")
```

End of Day 2

This concludes the Introduction to Python,
I hope you enjoyed it.

Thank you !