

Introductory Programming Using Python

Day 2

By Tan Kok Cheng and Seow Khee Wei Republic Polytechnic

Before we start...

- Mute your microphone when not speaking
- Unmute when you are answering questions / or asking questions in class
- Give me feedback as I need to know how you are doing so that I can adjust my pace or explain any concepts again.

Download course material at https://bit.ly/py-11nov

download day2.zip file, we will need the source code, excel and image files for the exercise later on



source



Introduction of trainer



Name Seow Khee Wei Telegram @kwseow

Email seow_khee_wei@rp.edu.sg



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 Creating Chart Generating PDF



Outline for the day

Time	Agenda
9.00am	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	
4.30pm – 5.00pm	Quiz, Q&A



File Paths

Absolute file paths are notated by a **leading forward slash or drive label**.

For example,

/home/example_user/examp
le_directory

or

C:/system32/cmd.exe

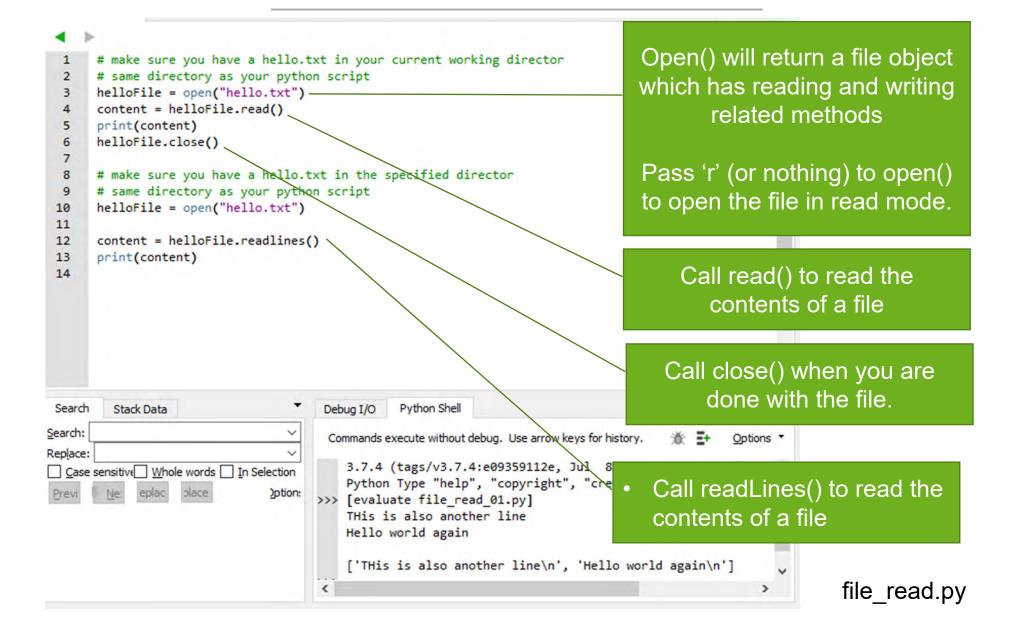
An absolute file path describes how to access a given file or directory, starting from the root of the file system. Relative file paths are notated by a lack of a leading forward slash.

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..

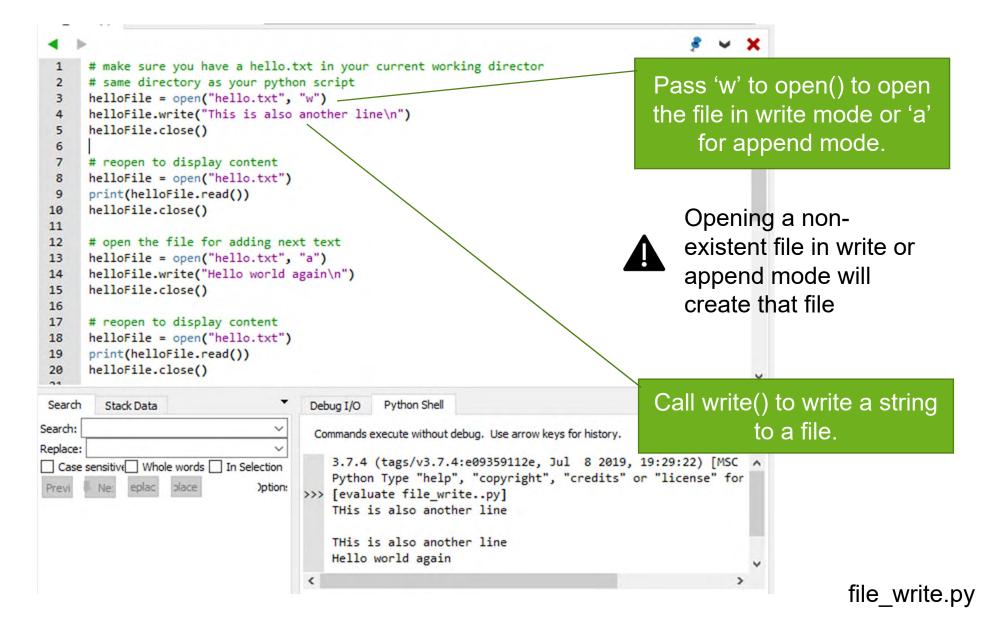


Read files



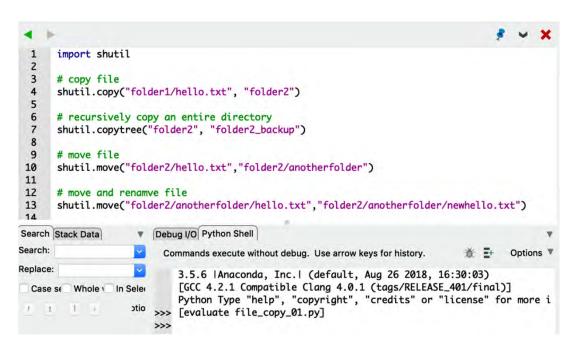


Write files





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



Deleting files

```
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

```
import os

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

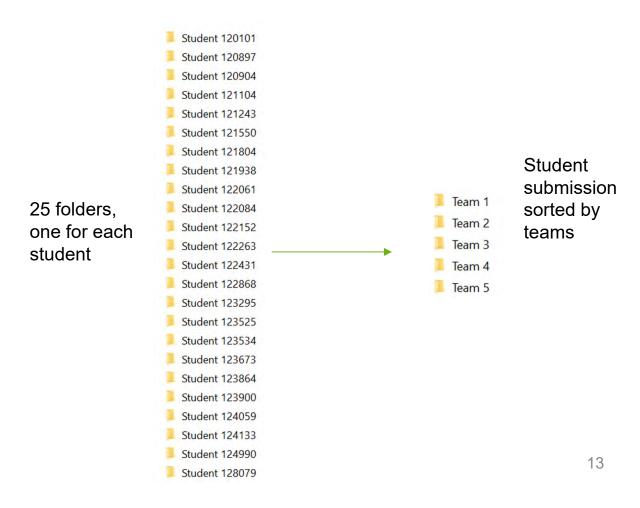


Deleting can be dangerous, so do a dry run first



Use Case Sharing

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





Other Use Cases

- System administrators can use these commands to
 - Copy and backup files to other hard-disks
 - Delete folders/ files at fixed schedules
 - End of financial year?
 - End of semester?
- Others use
 - Check timestamp of files, and delete all files created before a certain date



Exercise

Write code to achieve the following:

- 1. Create a file named: "myfile.txt".
- 2. Write the following line of text into the file:
 - Programming is fun!
- 3. Close the file
- 4. Create a folder called "myfolder"
 - Use os.mkdir() command
- 5. Copy myfile.txt to myfolder





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:\python38\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>

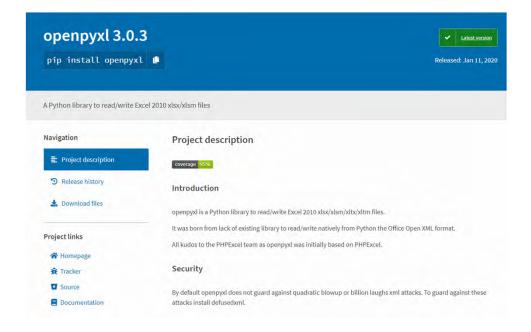
Double-Dash

Excel Spreadsheet Manipulation with Python



Working with Excel

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





Typical Workflow for Excel Automation

You are given some data in a spreadsheet

You want to do some or all of the following

- Analyze the data
- Manipulate the data

Output the processed data in another spreadsheet



Reading Excel file



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



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

```
1) Get the number of rows
import openpyxl
                                                          and columns
workbook = openpyxl.load_workbook("bmi.xlsx")
sheet=workbook["Sheet1"]
max_row = sheet.max_row # get number of rows
                                                               2) Use For loop to go through
#loop through every row
                                                               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
                                                                         3) Extract the status at
    print("name:%s \tweight: %d \theight: %f " % (name, weight, height))
                                                                         Column C to check for
                                                                         attendance
```



Update Excel file

```
import openpyxl
workbook = openpyxl.load_workbook("bmi.xlsx")
sheet=workbook["Sheet1"]
                                                     2) Load file into memory & get
max_row = sheet.max_row # get number of rows
                                                     the sheet
# 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
                                                         1) Perform calculation with
   name = sheet.cell(row=i, column=1).value
   weight = sheet.cell(row=i, column=2).value
                                                         values taken from the excel
   height = sheet.cell(row=i, column=3).value
                                                        files
   bmi = weight / (height * height)
   sheet.cell(row=i, column=4).value = bmi <
                                                          2) Add comments to cell
   print("name:%s \tBMI: %f" % (name, bmi))
#save the file
workbook.save("bmi update.xlsx")
                                                5) Save the spreadsheet
```



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 = [
                                                              1) Some data in nested list
   [225.7, 'Gone with the Wind', 'Victor Fleming'],
   [194.4, 'Star Wars', 'George Lucas'],
   [161.0, 'ET: The Extraterrestrial', 'Steven Spielberg']
                                      2) Using for loop to add each row
for row in data:
   sheet.append(row)
                                      of data into the excel sheet
#save the spreadsheet
workbook.save("movies.xlsx")
                                         3) Save the spreadsheet
```



Format Excel

```
import openpyxl
from openpyxl.styles import Font, PatternFill, Border, Side <
                                                                                1) Import necessary
workbook = openpyxl.load_workbook("bmi.xlsx")
                                                                                functions
sheet=workbook["Sheet1"]
#define the colors to use for styling
BLUE = "0033CC"
LIGHT BLUE = "E6ECFF"
WHITE = "FFFFFF"
                                                                              2) Setup colors and styles
#define styling
header_font = Font(name="Tahoma", size=14, color=WHITE)
header fill = PatternFill("solid", fgColor=BLUE)
# format header
                                                                        3) Loop through cell and set
for row in sheet["A1:c1"]:
 for cell in row:
                                                                        properties
   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")
```



Typical Workflow for Excel Automation

You are given some data in a spreadsheet

You want to do some or all of the following

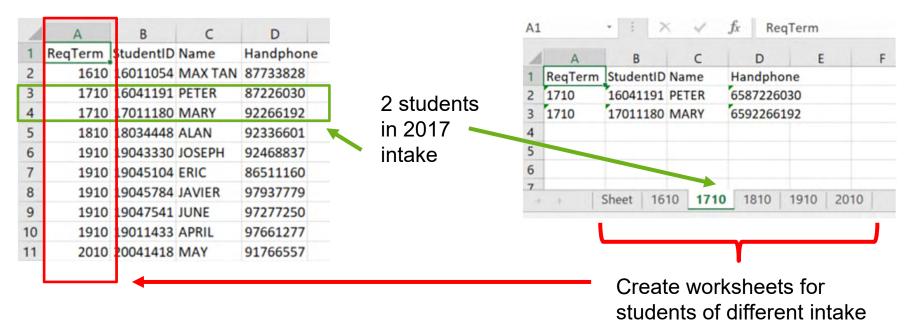
- Analyze the data
- Manipulate the data
- Create visualization (Charts, Pivot Table etc)

Output the processed data in another spreadsheet



Use Case Sharing

- I use RPA (Robotic Process Automation) to send WhatsApp messages to students individually
- Write script to split my contact list for students based on intake year
 - Added country code: 65 to all the numbers as well



^{*} Data shared here are randomly generated

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/ha ndbook/overview.html



Image Processing

As a start we need to import it:

import Image

```
import os
                                                We can open images with
from PIL import Image
                                                im = Image.open(fullname)
filename = "img/clungup.jpg"
                                                Then we can display the image using
im = Image.open(filename)
                                                im.show()
print ("%s - %s" % (im.size, im.mode))
                                                Print some info about the image using
# show the image
                                                im.size and im.mode
im.show()
# close the file
                                      Debug I/O
im.close()
                                                  Python Shell
                                                                           OS Commands
                                                               Messages
                                     Debug I/O (stdin, stdout, stderr) appears below
                                    (800, 600) - RGB
                                          Size: 800 x 600, Mode: RGB
```



Image Processing

```
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!

3

Image processing - filters



image = ImageOps.grayscale(image)



image = image.filter(ImageFilter.FIND EDGES)



image = image.filter(ImageFilter.CONTOUR)



image = ImageOps.solarize(image)



* 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 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 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 processing – Converting

```
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)

# split the filename and the extension
f, e = os.path.splitext(filename)

# add the gif extension to the filename
fname2 = f + ".gif"

outFilename = out_folder + fname2 # out/clungup.gif
out.save(outFilename)
```

os.path.splitext(file) returns a list.
We are only interested in f which is
the first item in the list.



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.





Image processing – Watermark

```
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:





Exercise: Batch Resize

- 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



Web Automation with Python



requests – download files and web pages from the Web

pip install requests

import requests

Get the required information from the given URL

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



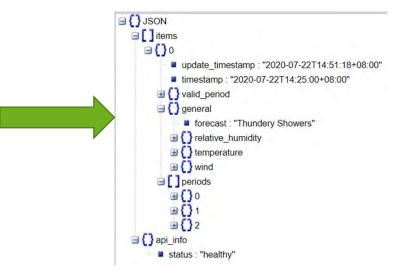


- Data is in JSON format
- Use a JSON formatter tool to present the data in a nicer form

http://jsonviewer.stack.hu/

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

```
"items":[{"update_timestamp":"2020-07-22T14:51:18+08:00","timesta":"2020-07-22T14:25:00+08:00","valid_period":{"start":"2020-07-22T1:"end":"2020-07-23T12:00:00+08:00"},"general":{"forecast":"Thunderg"relative_humidity":{"low":70,"high":95},"temperature":{"low":22,'"wind":{"speed":{"low":10,"high":20},"direction":"ESE"}},"periods'"start":"2020-07-22T12:00:00+08:00","end":"2020-07-22T18:00:00+08:{"west":"Moderate Rain","east":"Moderate Rain","central":"Light I"Light Rain","north":"Light Rain"}},{"time":{"start":"2020-07-22T1,"end":"2020-07-23T06:00:00+08:00"},"regions":{"west":"Partly Cloudy (Night)","east":"Partly Cloudy (Night)","Partly Cloudy (Night)","north":"Partly Cloudy (Night)","refine"""2020-07-23T06:00:00+08:00","end":"2020-07-23T12:00:00+08:00"},"refine"""2020-07-23T06:00:00+08:00","end":"2020-07-23T12:00:00+08:00"},"refine"""2020-07-23T06:00:00+08:00","end":"2020-07-23T12:00:00+08:00"},"refine"""2020-07-23T06:00:00+08:00","end":"2020-07-23T12:00:00+08:00"},"refine"""2020-07-23T06:00:00+08:00","end":"2020-07-23T12:00:00+08:00"},"refine"""2020-07-23T12:00:00+08:00","refine"""2020-07-23T12:00:00+08:00"},"refine"""2020-07-23T12:00:00+08:00"},"refine"""2020-07-23T12:00:00+08:00"},"refine"""2020-07-23T12:00:00+08:00"},"refine"""2020-07-23T12:00:00+08:00"},"refine"""2020-07-23T12:00:00+08:00"},"refine"""2020-07-23T12:00:00+08:00"},"refine"""2020-07-23T12:00:00+08:00"},"refine"""2020-07-23T12:00:00+08:00"},"refine"""2020-07-23T12:00:00+08:00"},"refine"""2020-07-23T12:00:00+08:00"},"refine"""2020-07-23T12:00:00+08:00"},"refine"""2020-07-23T12:00:00+08:00"},"refine""2020-07-23T12:00:00+08:00"},"refine"""2020-07-23T12:00:00+08:00"},"refine"""2020-07-23T12:00:00+08:00"},"refine""2020-07-23T12:00:00+08:00"},"refine""2020-07-23T12:00:00+08:00"},"refine""2020-07-23T12:00:00+08:00"},"refine""2020-07-23T12:00:00+08:00"}
```





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



Connecting to the Web (Backup)

requests – download files and web pages from the Web

pip install requests

import json
import requests

Get the required information from the given URL

url="https://api.data.gov.sg/v1/environment/air-temperature"
req=requests.get(url)





Connecting to the Web (Backup)

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

http://jsonviewer.stack.hu/

```
import json
                                                                                                  ■{}JSON
     import requests

■ stations

     url="https://api.data.gov.sg/v1/environment/air-temperature"
     req=requests.get(url)
                                                                                                          reading type: "DBT 1M F"
                                                                                                          ■ reading unit: "deg C"
                                                                                                     items
                                                                                                       ∃{}0
{"metadata":{"stations":[{"id":"S50","device id":"S50","name":"Clementi |
                                                                                                             timestamp: "2021-10-04T15:35:00+08:00"
{"latitude":1.3337,"longitude":103.7768}},{"id":"S44","device id":"S44",

    □ readings

"latitude":1.34583,"longitude":103.68166}},{"id":"S106","device id":"S10
                                                                                                             ⊟{}0
{"latitude":1.4168, "longitude":103.9673}}, {"id": "S116", "device id": "S116"
{"latitude":1.281, "longitude":103.754}}, {"id": "S100", "device_id": "S100", "
                                                                                                                  station id: "S50"
"latitude":1.4172,"longitude":103.74855}}],"reading type":"DBT 1M F","re
                                                                                                                  ■ value: 32.1
[{"timestamp":"2021-10-04T15:45:00+08:00","readings":[{"station id":"S50
                                                                                                             ⊟{}1
{"station_id":"S44","value":32.6},{"station_id":"S106","value":32.3},{"station_id":"S106","value":32.3},
                                                                                                                  station_id: "S44"
{"station id": "S100", "value": 33.1}]}], "api info": {"status": "healthy"}}
                                                                                                                 ■ value: 32
                                                                                                             H{}2
                                                                                                             ⊞{}4

→ { } api info
```



Connecting to the Web (Backup)

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

```
import json
import requests
url="https://api.data.gov.sg/v1/environment/air-temperature"
req=requests.get(url)
data = json.loads(req.text)
first id = data["metadata"]["stations"][0]["id"]
                                                                     id: 550
# print the first item's id
                                                                     air temp: 31.5
print("id: " + first id)
air_temp = data["items"][0]["readings"][0]["value"]
# print the first item's temperature
print("air temp: " + str(air_temp))
```



Exercise

Car Park Availability Data:

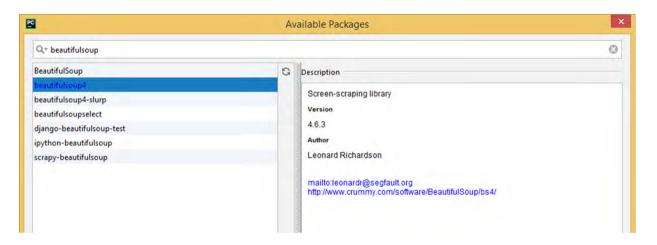
- 1. url: https://api.data.gov.sg/v1/transport/carpark-availability
- 2. Write the code to get the timestamp and the Carpark Number for the first set of carpark data.
- 3. Print out the result as shown.



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

Web Scraping – download and process Web content

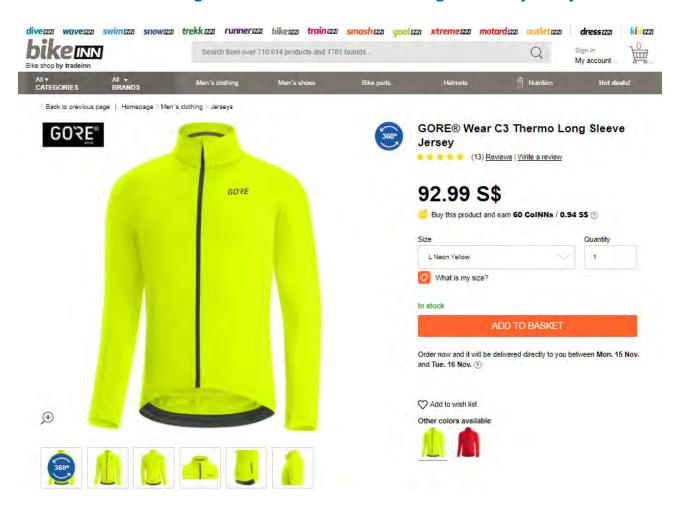
Install Beautiful Soup 4 - pip install beautifulsoup4





What's the URL?

https://www.bikeinn.com/bike/gore--wear-c3-thermo-long-sleeve-jersey/137533545/p

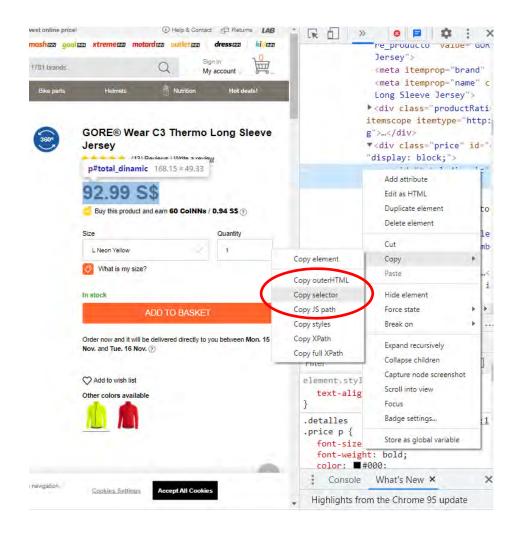




· Get the url

https://www.bikeinn.com/bike/g ore--wear-c3-thermo-longsleeve-jersey/137533545/p

- Select the element to extract, right-click "Inspect"





```
# https://www.bikeinn.com/bike/gore--wear-c3-thermo-long-sleeve-jersey/137533545/p
1
2
3
       from urllib.request import Request, urlopen
4
       from bs4 import BeautifulSoup
5
6
       site= "https://www.bikeinn.com/bike/gore--wear-c3-thermo-long-sleeve-jersey/137533545/p"
       hdr = {'User-Agent': 'Mozilla/5.0'}
7
       req = Request(site,headers=hdr)
8
9
       page = urlopen(req)
       soup = BeautifulSoup(page, 'html.parser')
10
11
       elements = soup.select("#total dinamic") # 92.99 S$
12
       print(elements)
13
       price = elements[0].text
14
       print("Current Price: " + elements[0].text)
15
```



Exercise

Table Tennis Bat Price:

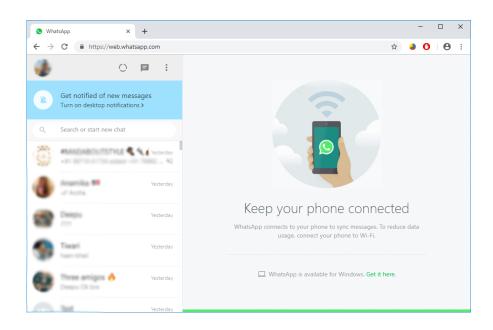
- 1. url: https://www.tabletennis11.com/other-eng/butterfly-viscaria
- 2. Write the code to get the price of the table tennis bat
- 3. Print out the result as shown.

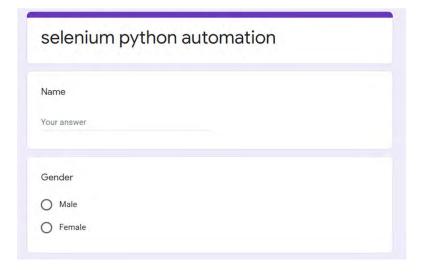




Sharing other Use Cases

- Using another library: selenium
 - Filling up google form
 - Sending WhatsApp message
 - Scrape job sites.

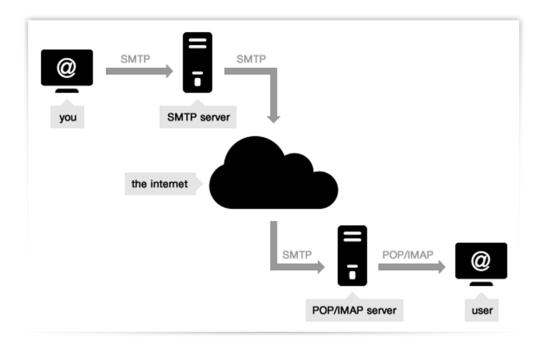




Email Automation with Python



Send Email



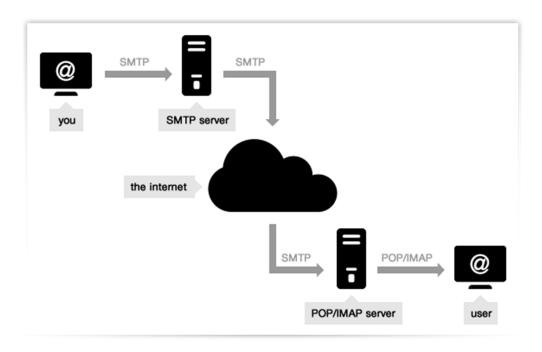
- SMTP (Simple Mail Transfer Protocol) is used for sending and delivering from a client to a server via port 25, 465 or 587: it's the outgoing server.
- IMAP and POP are two methods to access email. IMAP is the recommended method when you need to check your emails from several different devices, such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



Incoming Mail (IMAP) Server	imap.gmail.com Requires SSL: Yes Port: 993
Outgoing Mail (SMTP) Server	smtp.gmail.com Requires SSL: Yes Requires TLS: Yes (if available) Requires Authentication: Yes Port for SSL: 465 Port for TLS/STARTTLS: 587
Full Name or Display Name	Your name
Account Name, User name, or Email address	Your full email address
Password	Your Gmail password

Note: If you are using your office network, most port numbers, including 587, may be blocked.



- Import smtplib module
- Specify Gmail email & password, receiver's email address, email title & content
- Connect to SMTP server using Port 587
- Call starttls() to enable encryption for your connection
- Login using email and password
- Call sendmail()
- Call quit() to disconnect from the SMTP server

```
import smtplib

sender_email_address = "your_email_address@gmail.com"

sender_email_password = "xxxxxxxxxxxxxxx"

receiver_email_address = "another_email_address@gmail.com"

email_title_content = "Subject: Sending Email Using Python\nThis is a test email."

email_title_content = "Subject: Sending Email Using Python\nThis is a test email."
```

➤ The start of the email body must begin with "Subject: " for the subject line. The "\n" newline character separates the subject line from the main body content.

```
print("Trying to connect to Gmail SMTP server")
smtpObj = smtplib.SMTP("smtp.gmail.com", 587)
smtpObj.starttls()

print("Connected. Logging in...")
smtpObj.login(sender_email_address, sender_email_password)

smtpObj.sendmail(sender_email_address, receiver_email_address, email_title_content)
print("Email sent successfully...")

smtpObj.quit()
```



 Google may block attempted sign-in from unknown devices that don't meet their security standards!



```
C:\Users\denise_quek\AppData\Local\Programs\Python\Python37\python.exe D:/CET_Python/Denise/TestEmail.py
Trying to connect to Gmail SMTP server
Connected. Logging in...
Traceback (most recent call last):
   File "D:/CET Python/Denise/TestEmail.py", line 13, in <module>
        smtpObj.login(sender_email_address, sender_email_password)
   File "C:\Users\denise quek\AppData\Local\Programs\Python\Python37\lib\smtplib.py", line 730, in login
        raise last_exception
   File "C:\Users\denise quek\AppData\Local\Programs\Python\Python37\lib\smtplib.py", line 721, in login
        initial_response_ok=initial_response_ok)
   File "C:\Users\denise quek\AppData\Local\Programs\Python\Python37\lib\smtplib.py", line 642, in auth
        raise SMTPAuthenticationError(code, resp)
smtplib.SMTPAuthenticationError: (534, b'5.7.9 Application-specific password required. Learn more at\n5.7.9
```

Process finished with exit code 1

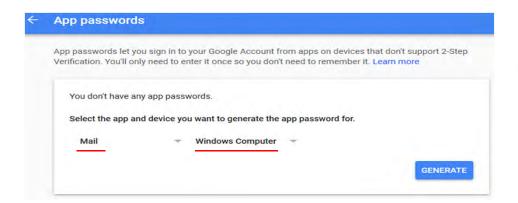


Steps To Create Google App Password

Step 1: Login to Gmail. Go to Account [[Signing in to Google

Step 2: Make sure that 2-Step Verification is on

Step 3: Create an App password



Generated app password

********	•
assword	
securesally@gmail.com	
mail address	
inter the information below to connect to you	r Google account.
Add your Google account	

Your app password for Windows Computer

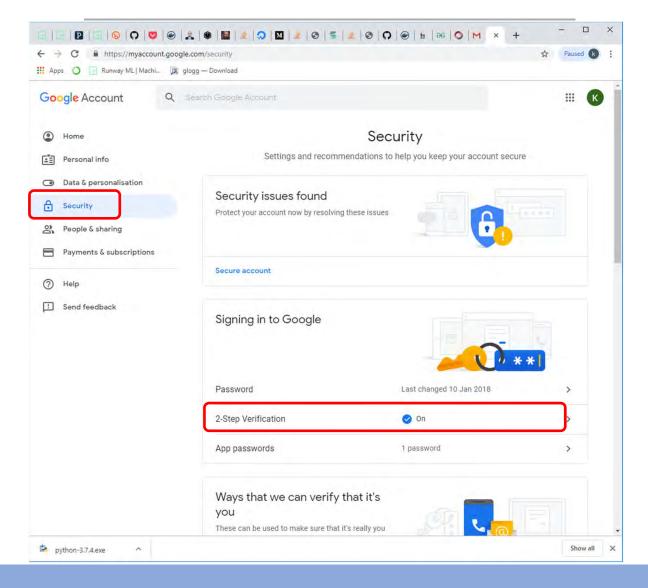
How to use it

- 1. Open the "Mail" app.
- 2. Open the "Settings" menu.
- Select "Accounts" and then select your Google Account.
- Replace your password with the 16-character password shown above.

Just like your normal password, this app password grants complete access to your Google Account. You won't need to remember it, so don't write it down or share it with anyone. Learn more

DONE







Replace your actual password with the App password

Run your email program

```
C:\Users\denise_quek\AppData\Local\Programs\Python\Python37\python.exe D:/CET_Python/Denise/TestEmail.py
Trying to connect to Gmail SMTP server
Connected. Logging in...
Email sent successfully...

Process finished with exit code 0
```



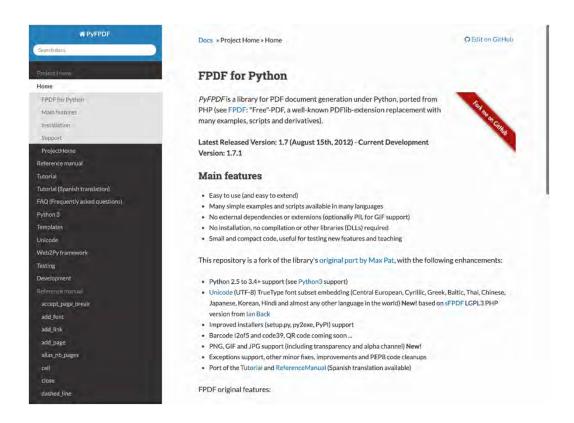
Sharing other Use Cases

- Sending Emails using Outlook
- Create Appointment using Outlook

Generate PDF Report with Python







- Install fpdf
 - · pip install fpdf

https://pyfpdf.readthedocs.io/en/latest/Tutorial/index.html



PDF - Basic document

```
import fpdf
#create a new pdf
document = fpdf.FPDF()
#define font and color for title and add the first page
document.set_font("Times", "B", 14)
document.set text color(19,83,173)
document.add page()
#write the title of the document
document.cell(0,5,"PDF Test Document")
document.ln()
#write a long paragraph
document.set_font("Times", "", 11)
document.set text color(0)
document.multi_cell(0,10, "This is an example of a long paragraph. \n" * 10)
document.ln()
#save the document
document.output("pdf_report.pdf")
```

- Import fpdf
- Create a new pdf document
- Add page
- Add text
- Save file

PDF Test Document

This is an example of a long paragraph. This is an example of a long paragraph. This is an example of a long paragraph This is an example of a long paragraph. This is an example of a long paragraph.



PDF – Basic document

```
import fpdf
#create a new pdf
document = fpdf.FPDF()
#define font and color for title and add the first page
document.set_font("Times", "B", 14)
document.set text color(19,83,173)
document.add page()
#write the title of the document
document.cell(0,5,"PDF Test Document")
document.ln()
#write a long paragraph
document.set_font("Times", "", 11)
document.set text color(0)
document.multi cell(0,10, "This is an example of a long paragraph. n * 10)
document.ln()
#save the document
document.output("pdf_report.pdf")
    fpdf.cell(w, h = 0, txt = '', border = 0, ln = 0,
               align = '', fill = False, link = '')
```

- Import fpdf
- Create a new pdf document
- Add page
- Add text
- Save file

```
PDF Test Document

This is an example of a long paragraph.

This is an example of a long paragraph.
```



PDF – adding images

```
import fpdf
#create a new pdf
document = fpdf.FPDF()
#define font and color for title and add the first page
document.set font("Times", "B", 14)
document.set_text_color(19,83,173)
document.add page()
#add a image
document.image("rp_logo.png", x=10, y=5, w=23)
document.set_y(40);
#write the title of the document
document.cell(0,5,"PDF Test Document")
document.ln()
#write a long paragraph
document.set_font("Times", "", 11)
document.set_text_color(0)
document.multi_cell(0,5, "This is an example of a long paragraph. " * 10)
document.ln()
#save the document
document.output("pdf_report.pdf")
```

- Import fpdf
- Create a new pdf document
- Add page
- Add text, logo
- Save file



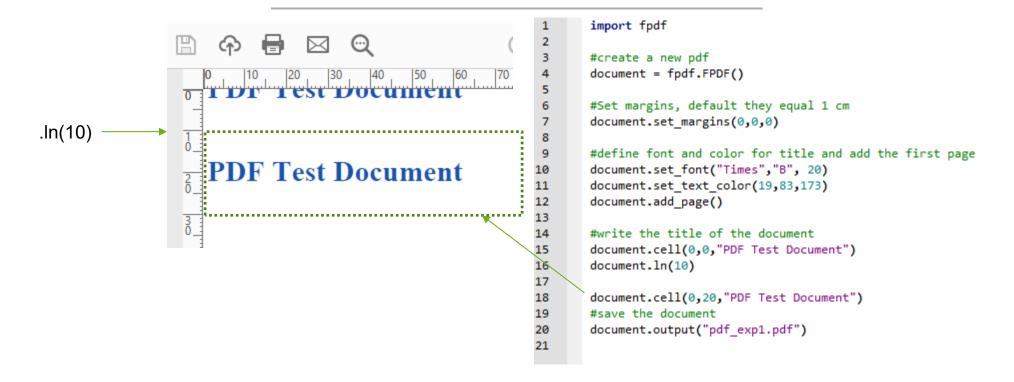
PDF Test Documen

This is an example of a long paragraph. This is an example of a long paragraph.

https://pyfpdf.readthedocs.io/en/latest/reference/image/index.html



More on cell...





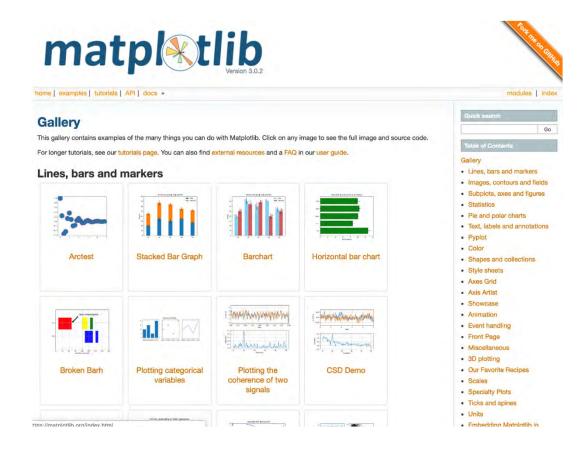
Use Cases

- Automation:
 - Generation of reports with data from spreadsheet or database
 - Generation of Course Certificates in PDF format

Charting/Visualisation with Python



Charting



pip install matplotlib

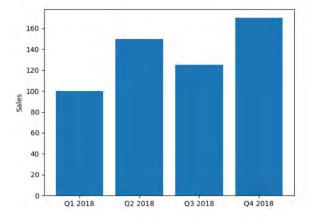
Full documentation: https://matplotlib.org/



Charting

```
import matplotlib.pyplot as plt
 1
2
     #set up values
 3
     VALUES = [100, 150, 125, 170]
 5
     POS = [0,1,2,3]
     LABELS = ['Q1 2018','Q2 2018','Q3 2018','Q4 2018']
 6
 7
 8
     #set up the chart
9
     plt.bar(POS, VALUES)
     plt.xticks(POS, LABELS)
10
     plt.ylabel('Sales')
11
12
13
     #to display the chart
14
     plt.show()
```

- Install matplotlib
- Prepare data
- Create bar graph
- Display the chart



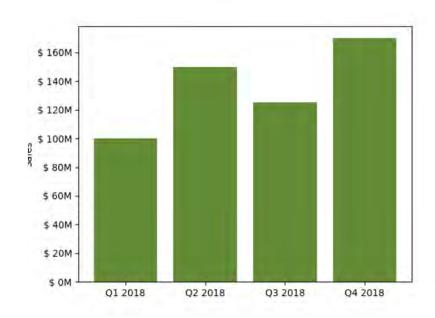
https://matplotlib.org/api/_as_gen/matplotlib.pyplot.bar.html



Charting - Formatting

```
import matplotlib.pyplot as plt
 1
 2
      from matplotlib.ticker import FuncFormatter
 4
      def value_format(value, position):
 5
              return '$ {}M'.format(int(value))
 6
 7
      # set up values
 8
     VALUES = [100, 150, 125, 170]
 9
     POS = [0,1,2,3]
     LABELS = ['Q1 2018','Q2 2018','Q3 2018','Q4 2018']
10
11
12
     # set up the chart
     # Colors can be specified in multiple formats, as
13
     # described in https://matplotlib.org/api/colors_api.html
14
15
      # https://xkcd.com/color/rgb/
16
      plt.bar(POS, VALUES, color='xkcd:moss green')
17
     plt.xticks(POS, LABELS)
18
      plt.ylabel('Sales')
19
20
     # retreive the current axes and apply formatter
21
      axes = plt.gca()
22
      axes.yaxis.set_major_formatter(FuncFormatter(value_format))
23
24
      # to display the chart
25
     plt.show()
```

- Install matplotlib
- Prepare data
- Customise graph options
- Create bar graph
- Display the chart



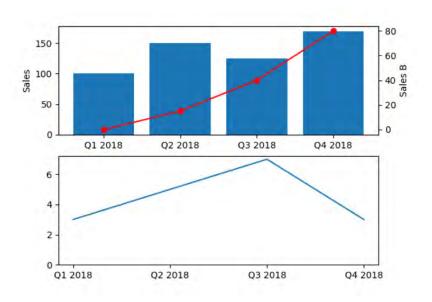


Charting - Subplots

```
import matplotlib.pyplot as plt
 1
 2
 3
      #set up values
 4
     VALUESA = [100, 150, 125, 170]
 5
     VALUESB = [0, 15, 40, 80]
 6
     VALUESC = [3,5,7,3]
 7
      POS = [0,1,2,3]
 8
     LABELS = ['Q1 2018','Q2 2018','Q3 2018','Q4 2018']
9
10
      # Create the first plot
11
     plt.subplot(2,1,1)
12
13
      #creata a bar graph with informaton about VALUESA
14
      plt.bar(POS, VALUESA)
15
      plt.ylabel('Sales')
16
17
      #create a different Y axis, and add information
18
     #about VALUESB as a line plot
19
      plt.twinx()
20
     plt.plot(POS, VALUESB, 'o-', color='red')
21
      plt.xticks(POS, LABELS)
22
      plt.ylabel('Sales B')
23
     plt.xticks(POS, LABELS)
24
25
      #create another subplot and fill it iwth VALUESC
26
     plt.subplot(2,1,2)
27
     plt.plot(POS, VALUESC)
28
      plt.gca().set_ylim(bottom=0)
29
      plt.xticks(POS, LABELS)
30
31
      plt.show()
```

Multiple charts

2 rows, 1 column, index starting from 1)



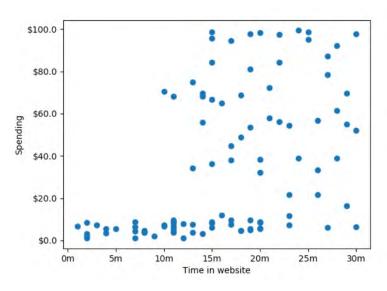
https://matplotlib.org/api/_as_gen/matplotlib.pyplot.subplot.html



Charting – Scatter Plot

```
1
      import csv
 2
      import matplotlib.pyplot as plt
      from matplotlib.ticker import FuncFormatter
 3
 4
 5
      def format minutes(value, pos):
 6
              return '{}m'.format(int(value))
 7
 8
      def format dollars(value, pos):
 9
              return '${}'.format(value)
10
      # read data from csv
11
      fp = open("scatter.csv", "r", newline='')
12
      reader = csv.reader(fp)
13
14
      data = list(reader)
15
16
      data x=[]
17
      data y=[]
      for x, y in data:
18
              data x.append(float(x))
19
20
              data y.append(float(y))
21
      plt.scatter(data x, data y)
22
23
      plt.gca().xaxis.set major formatter(FuncFormatter(format minutes))
24
      plt.xlabel('Time in website')
25
      plt.gca().yaxis.set major formatter(FuncFormatter(format dollars))
26
27
      plt.ylabel('Spending')
28
29
      plt.show()
```

- To save a plot: plt.savefig(filename)
- Save the plot before you display

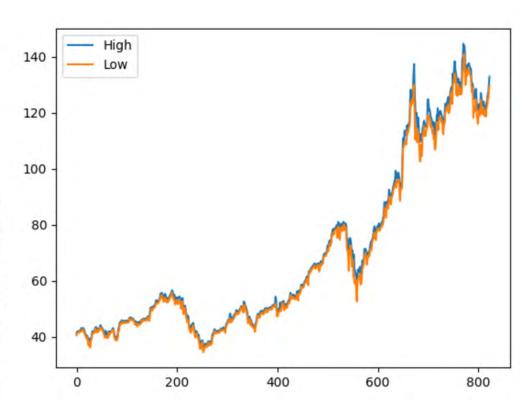




Use Case: Stock Chart

- Use for algorithmic Trading
 - Plotting of Stock Chart

A	А	В	C	D	E	F
1	Date	Open	High	Low	Close	Volume
2	1/2/2018	40.80647	41.31966	40.59063	41.31007	1.02E+08
3	1/3/2018	41.37482	41.85925	41.23813	41.30288	1.18E+08
4	1/4/2018	41.37723	41.60025	41.26691	41.49474	89738400
5	1/5/2018	41.59306	42.05589	41.49953	41.96716	94640000
6	1/8/2018	41.81128	42.11345	41.71056	41.81128	82271200
7	1/9/2018	41.85925	41.98156	41.58587	41.8065	86336000
8	########	41.52591	41.79929	41.48754	41.79689	95839600
9	########	41.86884	42.08467	41.84486	42.03431	74670800
10	########	42.25013	42.53312	42.12303	42.46836	1.02E+08





Other Python Libraries

- Play music using winsound
- Generate QR code using qrcode
- Face detection using opency / face_recognition
- News Summarisation using huggingface





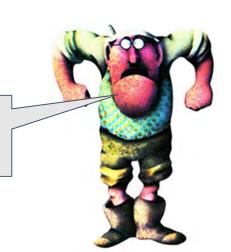


End of Day 2

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

Thank you!

QUESTIONS?



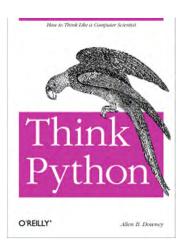
Quiz





Where to go from here?





Think Python is an introduction to Python programming for beginners. It starts with basic concepts of programming, and is carefully designed to define all terms when they are first used and to develop each new concept in a logical progression. Larger pieces, like recursion and object-oriented programming are divided into a sequence of smaller steps and introduced over the course of several chapters.

Think Python is a Free Book. It is available under the <u>Creative</u> <u>Commons Attribution-NonCommercial 3.0 Unported License</u>, which means that you are free to copy, distribute, and modify it, as long as you attribute the work and don't use it for commercial purposes. http://greenteapress.com/thinkpython/thinkpython.pdf

Where to go from here?



Getting started step by step

http://www.python.org/about/gettingstarted/

Run through the python tutorials:

http://docs.python.org/tutorial/index.html

Keep the API doc under your pillow:

http://docs.python.org/library/index.html

Advanced examples:

http://www.diveintopython.org/toc/index.html

Where to go from here?



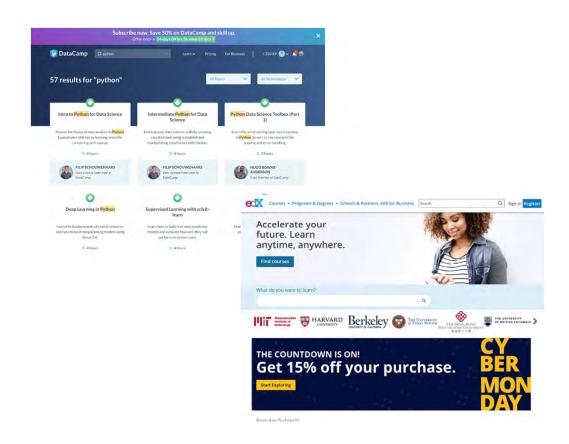
MOOC: DataCamp

https://www.datacamp.com/

Edx

https://www.edx.org/

Udemy (freemium course) https://t.me/freecourse





Thank you