
EMAIL
AUTOMATION

Recap

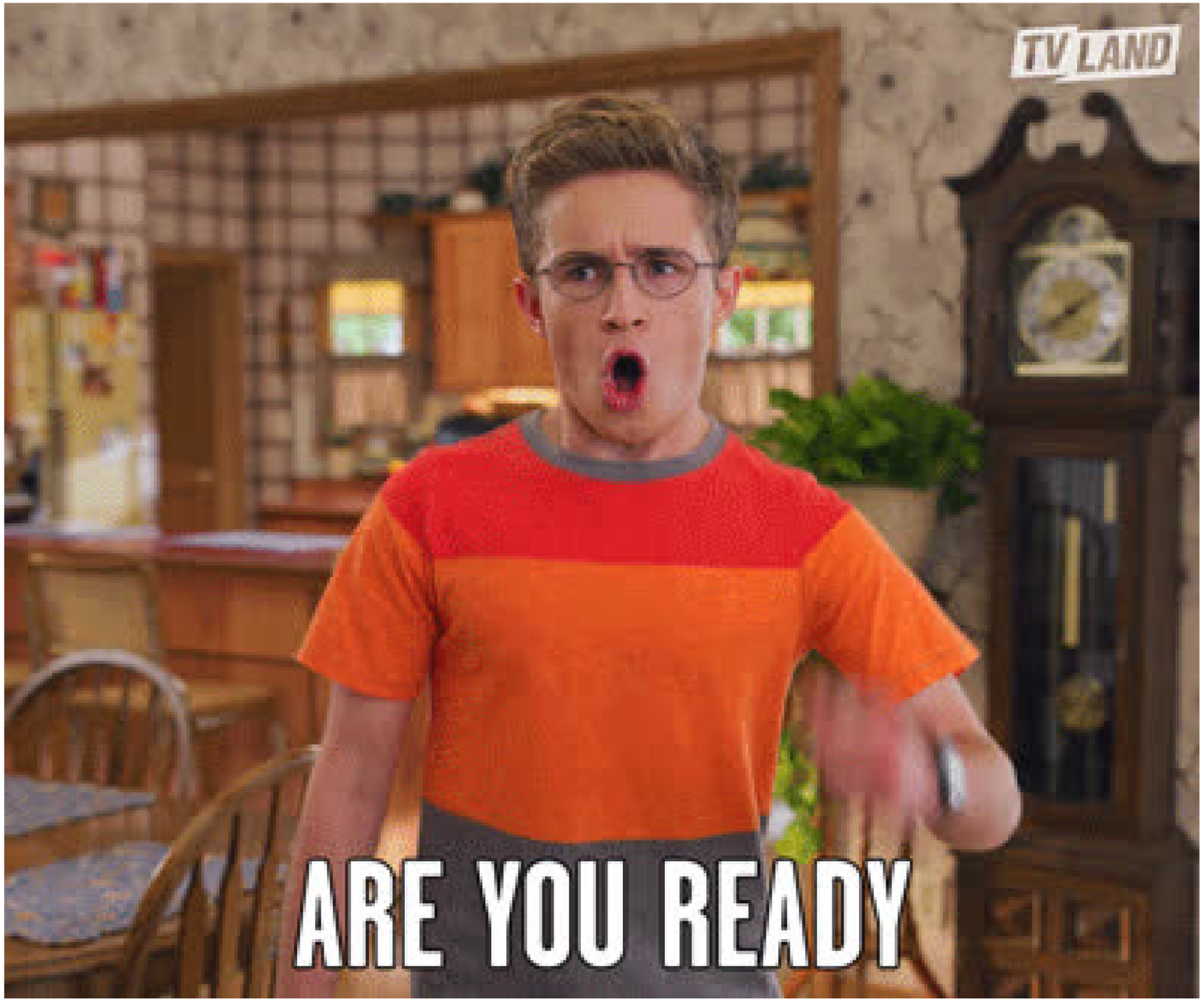
- Introduction to Python and Automation
- Why and What to automate
- Example Automation Project
- GUI Automation with pyautogui

Setting Up

- Cloning the repository
- Creating **APP_PASSWORD**

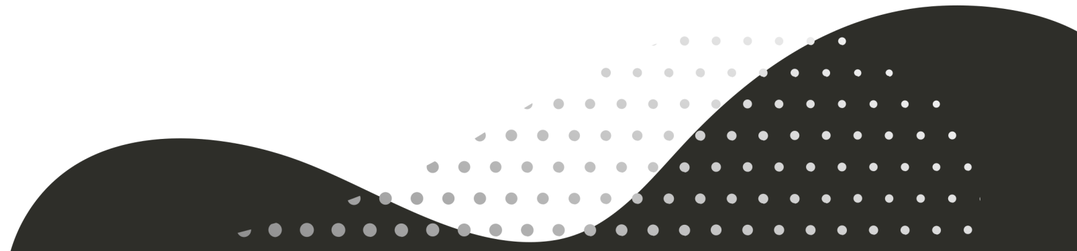
Objectives

- Learn to Auto-Send Multiple Emails
- Auto-Send Emails with Attachments
- Dynamic Email sending from CSV Data
- Pratical Implementation of Email Automation


TV LAND
ARE YOU READY

Sending Basic Mail

$ cd ***01_basic_email/start***



How does it work?

- Simple Mail Transfer Protocol(**SMTP**)
- Internet Message Access Protocol(**IMAP**)
- Post Office Protocol(**POP3**)

How does SMTP work?

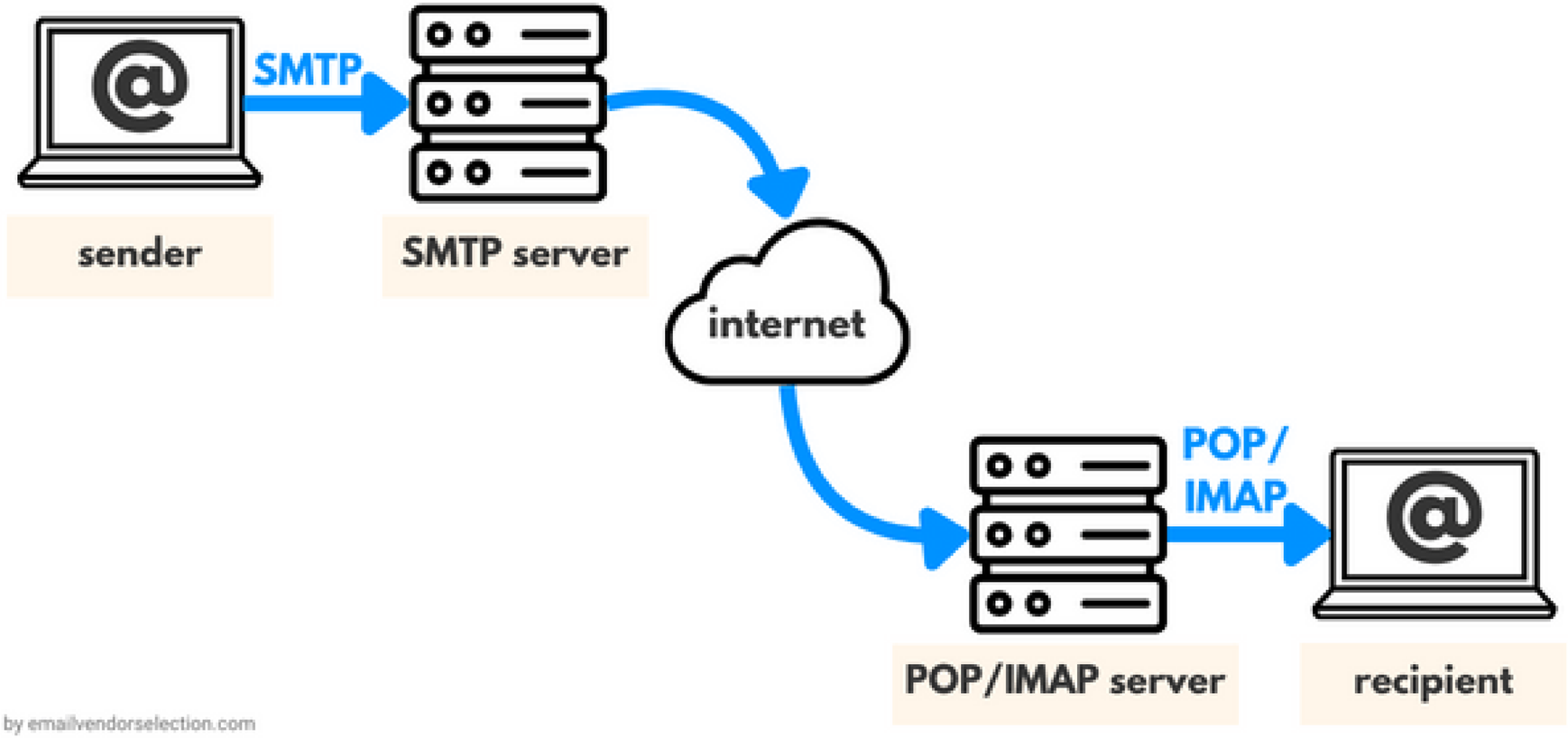


image by emailvendorselection.com

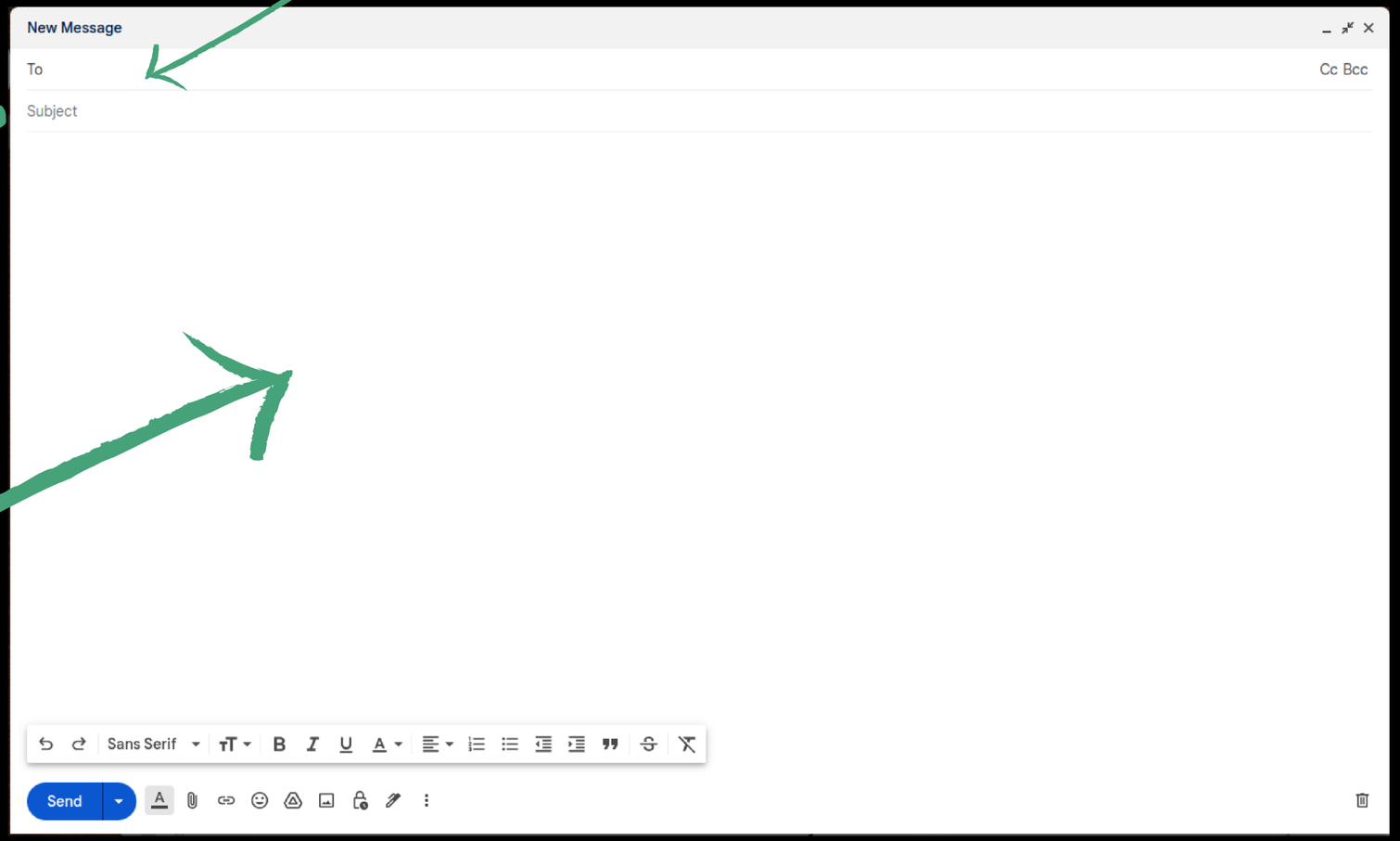
SMTP Server List

SERVER TYPE	OUTGOING SMTP SERVER NAME	SMTP SERVER PORT
Googlemail/Gmail SMTP	smtp.gmail.com	SSL Port 465 StartTLS Port 587
Outlook.com SMTP	smtp.live.com	StartTLS Port 587
Yahoo Mail SMTP	smtp.mail.yahoo.com	SSL Port 465
Yahoo Mail Plus SMTP	plus.smtp.mail.yahoo.com	SSL Port 465
AT&T SMTP	smtp.att.yahoo.com	SSL Port 465
Hotmail SMTP	smtp.live.com	StartTLS Port 587
Comcast SMTP	smtp.comcast.net	Port 587
Verizon SMTP	outgoing.verizon.net	SSL Port 465
Verizon SMTP POP3 Server	outgoing.yahoo.verizon.net	SSL Port 465

Requirements for a mail

subject

receiver_email_address



body

Gmail Interface

Requirements for mail

Python script

- SMTP_SERVER
- SMTP_SERVER_PORT
- SENDER_EMAIL_ADDRESS
- SENDER_EMAIL_PASSWORD
- RECEIVER_EMAIL_ADDRESS
- SUBJECT
- BODY

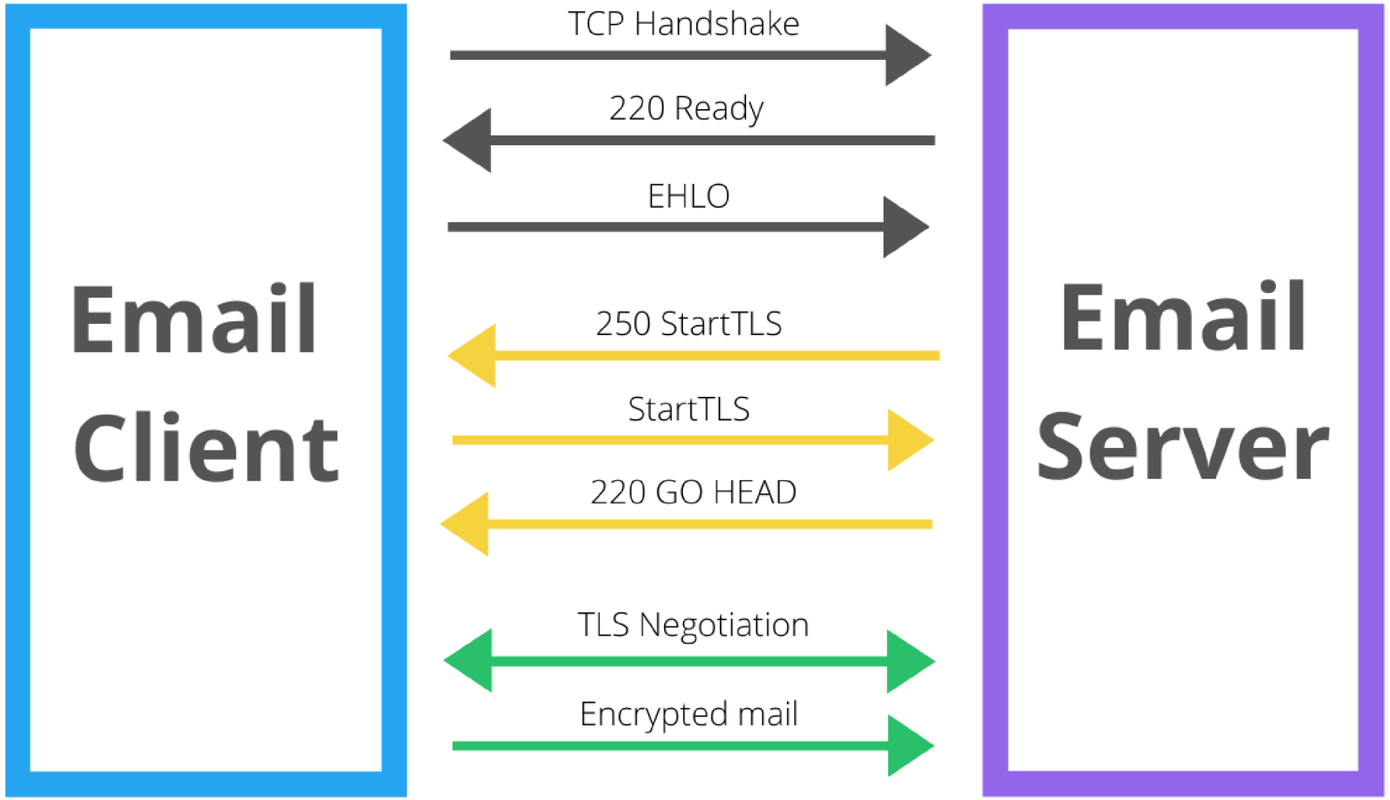
Defining the value

```
SMTP_SERVER = “smtp.gmail.com”
SMTP_SERVER_PORT = 587
SENDER_EMAIL_ADDRESS = “susheelthapa90@gmail.com”
SENDER_EMAIL_PASSSWORD=“abcdefghijkl”
RECEIVER_EMAIL_ADDRESS=”077bct090.susheel@pcampus.edu.np”
```

Defining the value

```
subject = “Meeting Reminder: Project Update”

body = “““
Dear Students,

This is a reminder about our upcoming project update meeting scheduled for tomorrow at 10:00 AM.

Please ensure that you have your progress reports ready and be prepared to discuss any blockers you are facing.
Looking forward to your participation.

Best regards,
Susheel Thapa
”””
```

Sending the mail



Sending the mail

```
message = f'Subject: {subject}\n\n{body}'
```

```
smtp = smtplib.SMTP(SMTP_SERVER, SMTP_SERVER_PORT)
```

```
smtp.starttls()
```

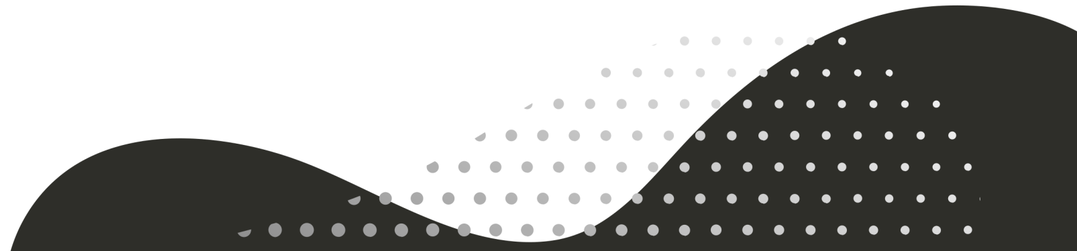
```
smtp.login(SENDER_EMAIL_ADDRESS, SENDER_EMAIL_PASSWORD)
```

```
smtp.sendmail(
 SENDER_EMAIL_ADDRESS,
 RECEIVER_EMAIL_ADDRESS,
 message
)
```

```
smtp.quit()
```

Email with .env

$ cd ***02_env_file/start***



Issue in Previous Approach

The code directly includes **sensitive information** (**such as email credentials**) within the script, which can lead to **security risks** *if the script is shared or stored insecurely*.

Solution

Create .env file

```
SMTP_SERVER = “smtp.gmail.com”
SMTP_SERVER_PORT = 587
SENDER_EMAIL_ADDRESS = “susheelthapa90@gmail.com”
SENDER_EMAIL_PASSSWORD=“abcdefghijkl”
RECEIVER_EMAIL_ADDRESS=”077bct090.susheel@pcampus.edu.np”
```

*Note: Rename **.env.example** to **.env** and update it.*

Update code

```
from dotenv import load_dotenv

import os


load_dotenv()
```

Update code

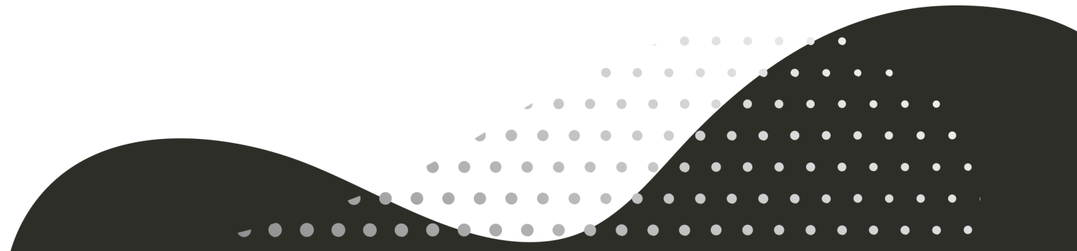
```
SMTP_SERVER = os.getenv("SMTP_SERVER")
SMTP_SERVER_PORT = os.getenv("SMTP_SERVER_PORT")
SENDER_EMAIL_ADDRESS = os.getenv("SENDER_EMAIL_ADDRESS")
SENDER_EMAIL_PASSWORD = os.getenv("SENDER_EMAIL_PASSWORD")
RECEIVER_EMAIL_ADDRESS = os.getenv("RECEIVER_EMAIL_ADDRESS")
```

String

```
smtp = smtplib.SMTP(SMTP_SERVER, int(SMTP_SERVER_PORT))
```

Email with Attachment

$ cd ***03_single_attachment/start***

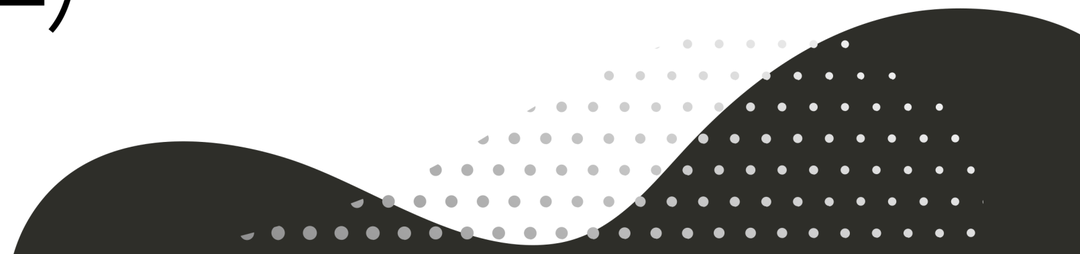


Limitation of Plain Text mail

- No support for rich text (HTML).
- Cannot include attachments or multimedia.
- Limited to simple text messages.


SO WHAT NOW?

Multipurpose Internet Mail Extensions (MIME)



MIME

- Allows for multiple parts in an email (*plain text*, *HTML*, *attachments*).
- Supports different content types (*text*, *images*, *files*).
- Essential for professional and feature-rich emails.

Shifting to MIME

```
subject = "PDSC Python for Automation Workshop - Certificate of Participation"

body = """Dear Participant,
Thank you for participating in the PDSC Python for Automation workshop!
Attached to this email, you will find your certificate of participation.
We hope you found the workshop informative and valuable.
Should you have any questions or feedback, feel free to reach out to us.
Best regards,
PDSC Team
"""
```

Shifting to MIME

```
from email.mime.multipart import MIMEMultipart
from email.mime.text import MIMEText
from email.mime.base import MIMEBase
from email import encoders
```

```
message = MIMEMultipart()
message["Subject"] = subject
message["From"] = SENDER_EMAIL_ADDRESS
message["To"] = RECEIVER_EMAIL_ADDRESS
message.attach(MIMEText(body, "plain"))
```

Shifting to MIME

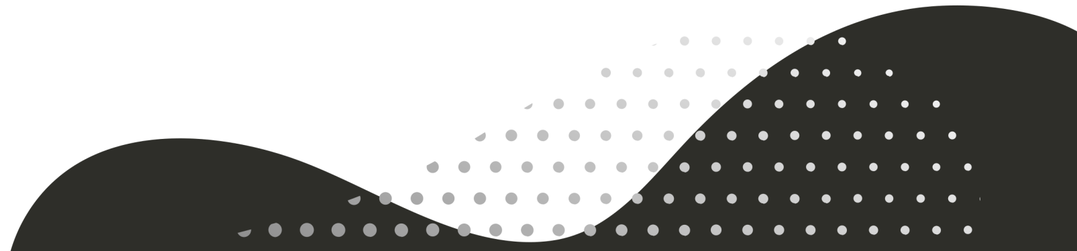
```
file_path = "schedule.pdf"
attachment = open(file_path, "rb")
part = MIMEBase("application", "octet-stream")
part.set_payload(attachment.read())
encoders.encode_base64(part)
part.add_header(
    "Content-Disposition",
    f"attachment; filename={os.path.basename(file_path)}"
)
message.attach(part)
```

```
smtp.sendmail(SENDER_EMAIL_ADDRESS, RECEIVER_EMAIL_ADDRESS, message.as_string())
```

Note: ***.env*** *file is present in the same folder from where you are executing the scripts*

HTML Email

$ cd ***04_html_content/start***



Which one??

Simple Text mail

PDSC Python for Automation Workshop - Certificate of Participation External

susheelthapa90@gmail.com
to me

Dear Participant,

Thank you for participating in the PDSC Python for Automation workshop!

Attached to this email, you will find your certificate of participation.

We hope you found the workshop informative and valuable. Should you have any questions or feedback, feel free to reach out to us.

Best regards,
PDSC Team

One attachment • Scanned by Gmail

schedule.pdf

HTML mail

PDSC
Plan | Design | Solve | Create

Dear Participant,

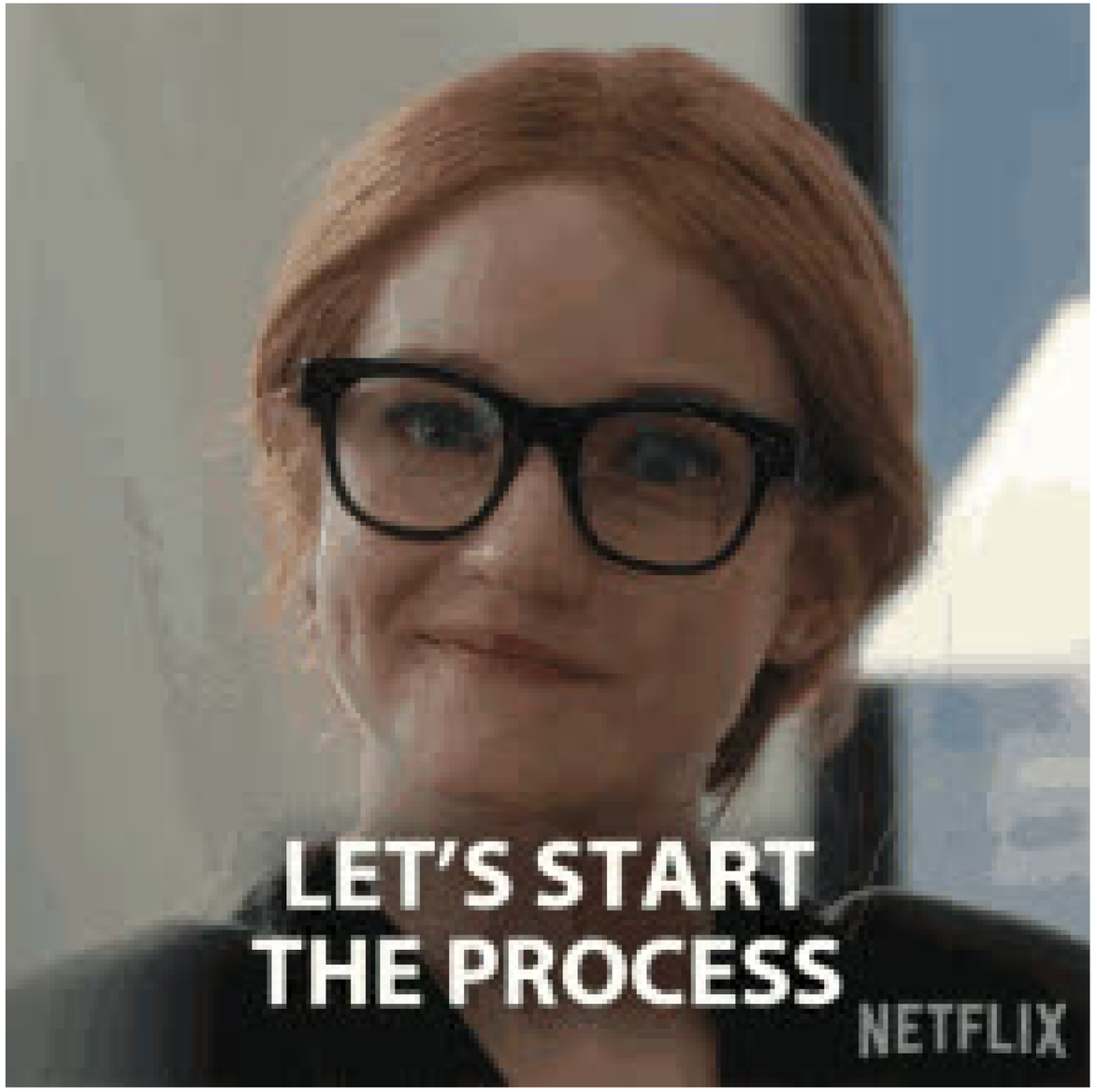
Congratulations on successfully completing the *PDSC: Python for Automation* workshop!

We are pleased to present you with a **certificate of participation**, which is attached to this email.

We hope you found the workshop informative and engaging. Thank you for your participation.

Best regards,
Susheel Thapa

Connect with us on:


LET'S START
THE PROCESS
NETFLIX

Precaution

Update email.html

```
$ cp ../end/email.html email.html
```

Shifting to HTML mail

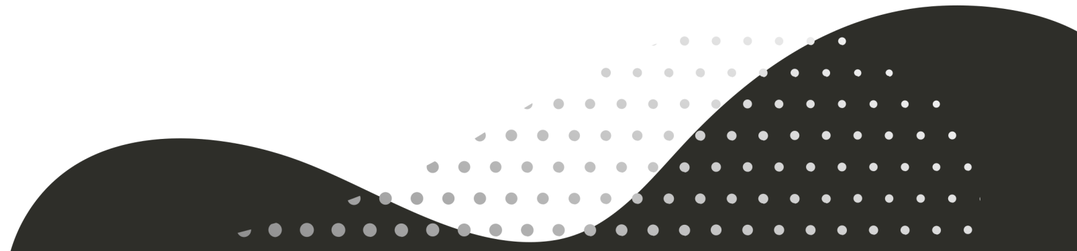
```
from email.mime.image import MIMEImage

subject = "Congratulations on Completing the Workshop!"

plain_text_body = """
Dear Participant,
Congratulations on successfully completing the PDSC: Python
for Automation workshop!
We are pleased to present you with a certificate of
participation, which is attached to this email.
We hope you found the workshop informative and engaging.
Thank you for your participation.
Best regards,
Susheel Thapa
"""
```

Shifting to HTML mail

```
with open("email.html", "r") as file:
    html_body = file.read()

message = MIMEMultipart("alternative")

part1 = MIMEText(plain_text_body, "plain")
message.attach(part1)

part2 = MIMEText(html_body, "html")
message.attach(part2)

with open("pdsc_logo.png", "rb") as image_file:
    image = MIMEImage(image_file.read())
    image.add_header('Content-ID', '<pdsc_logo>')
    message.attach(image)
```

Sending Multiple Email

$ cd ***05_multiple_emails/start***



Function in Python

A function is a block of code that performs a specific task.

```
def send_email(receiver_email):
    # Logic to send the single email
    pass
```

for loop in Python

In Python, we use a for loop to iterate over various sequences, such as **lists**, **tuples**, **sets**, **strings**, or **dictionaries**.

```
for email in email_list:
    send_mail(email)
```

send_mail()

Let’s encapsulate the logic to send single mail in a function name **send_mail()**

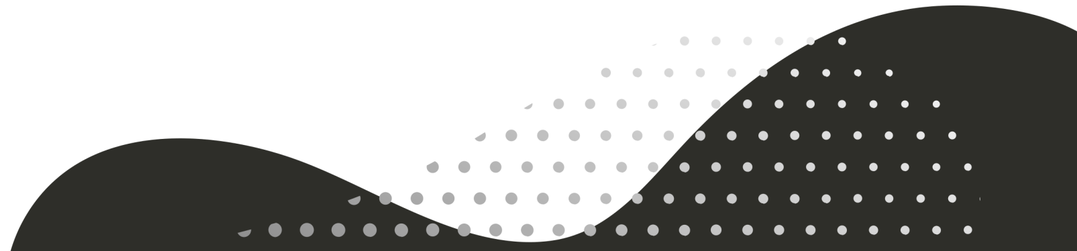
for_loop

```
email_list =["RECIPIENT_EMAIL_LIST"]
```

```
for email in email_list:
    send_email(email)
```

Dynamic Mail via CSV

$ cd ***06_dynamic_emails/start***



What is CSV file?

Comma-separated values (CSV) is a text file format that uses commas to separate values, and newlines to separate records.

Creating sample csv file

```
Name,Email
Susheel Thapa, 077bct090.susheel@pcampus.edu.np
```

Loading csv via Pandas

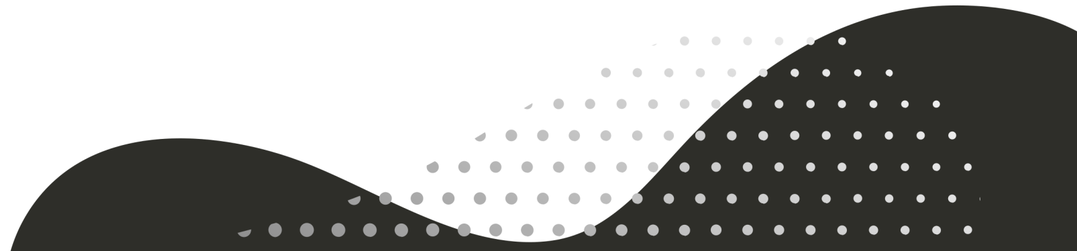
```
import pandas as pd

df = pd.read_csv("./sample.csv")

for index, row in df.iterrows():
    print(row["Name"], row["Email"])
```

Updating code

```
import pandas as pd

df = pd.read_csv("./sample.csv")

for index, row in df.iterrows():
    name = row["Name"]
    email = row["Email"]
    print(name, email)
    send_email(email, name)
```

Assignment

$ cd ***07_assignment/***



Any Question?

THANK YOU