Automation and Scripting with Python

By: Eng. Shereen Ebrahim

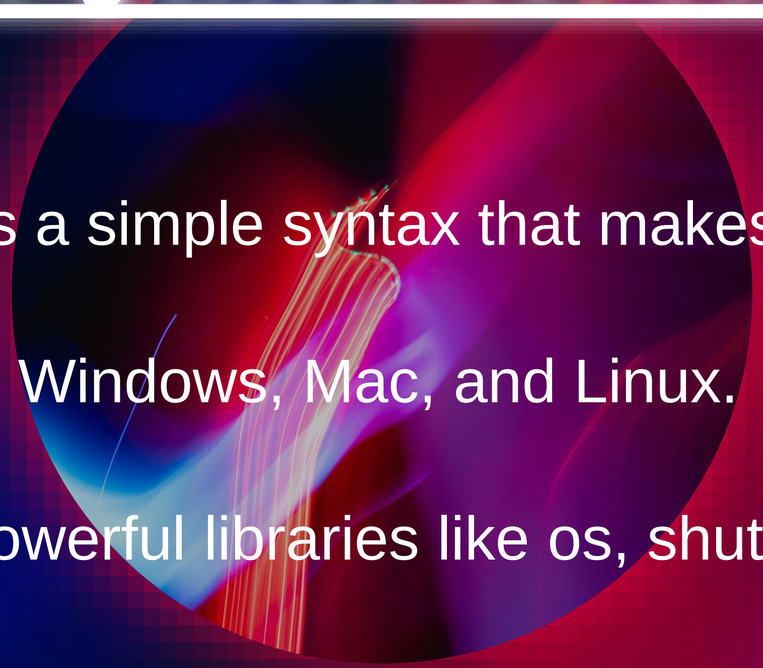
INCEPTUM
EGYPT

INTRODUCTION

Automation and Scripting

Automation and scripting are essential concepts in the modern world. They help in reducing repetitive tasks, minimizing human errors, and increasing efficiency in various fields like IT, business, and personal productivity.

Why Use Python for Automation?



- Easy to Learn: Python has a simple syntax that makes it beginner-friendly.
- Cross-Platform: Works on Windows, Mac, and Linux.
- Rich Libraries: Includes powerful libraries like os, shutil, smtplib, and pandas.
- Wide Applications: Used in web automation, data analysis, task scheduling, and more.

- Automation with Python allows users to handle tasks such as:
- File management (renaming, moving, deleting files automatically).
- Email automation (sending reports, alerts, and notifications).
- Web scraping (extracting useful information from websites).
- Task scheduling (running scripts at specific times).

2_Practical Examples with Python Code

- **1_ Automatically Renaming Multiple Files**
- Benefit: Saves time when managing large sets of files.
- ✅ **Code:**

```
import os
folder_path = "./files"  # Specify the folder path
for count, filename in enumerate(os.listdir(folder_path)):
    new_name = f"file_{count}.txt"  # Renaming files systematically
    old_path = os.path.join(folder_path, filename)
    new_path = os.path.join(folder_path, new_name)
    os.rename(old_path, new_path)
print("All files have been renamed successfully!")
```

2_This script helps organize images in a specific directory by moving them into a separate folder named "Images".

```
import os
import shutil
# Get the current directory
current_dir = os.path.dirname(os.path.realpath(__file__))
# Loop through all files in the directory
for filename in os.listdir(current_dir):
    ## Organize images into the "Images" folder
    if filename.endswith((".png", ".jpg")):    # Check if the file is an image
        if not os.path.exists("Images"):    # Check if the "Images" folder exists
            os.mkdir("Images")       # Create "Images" folder if it doesn't exist
        shutil.copy(filename, "Images")     # Copy image to the "Images" folder
        os.remove(filename)        # Delete the original file
        print("Done")
```

➢When Can This Code Be Used?

- ➢When working with a directory full of images and you want to organize them automatically.
- ➢ When sorting different types of files into separate folders (you can modify this script to sort PDFs, videos, etc.).
- ➢ To save time and effort instead of manually organizing files.

To automate sending a WhatsApp message every morning using Python, follow these steps:

- **1. Install Required Library**

➢ You can use the pywhatkit library to send WhatsApp messages. Install it using:

- pip install pywhatkit

2. Write the Python Script

➢ **This script will send a message to a specific number at 8:00 AM every day.**

```
import pywhatkit
# Define recipient number (include country code) and message
phone_number = "+1234567890"  # Replace with your recipient's number
message = "Good morning! Have a great day! ☀️"

# Schedule message at 8:00 AM
pywhatkit.sendwhatmsg(phone_number, message, 8, 0)
print("Message scheduled successfully!")
```

3. Automate the Script to Run Daily

➢ **Windows (Task Scheduler)**

- ❖ **Open Task Scheduler.**
- ❖ **Click on Create Basic Task.**
- ❖ **Set the trigger to Daily at 8:00 AM.**
- ❖ **Under Action, select Start a Program.**
- ❖ **In the Program/Script field, enter the path to Python (e.g., C:\Users\YourName\AppData\Local\Programs\Python\Python39\python.exe).**
- ❖ **In the Add arguments field, enter the path to your script (e.g., "C:\Users\YourName\Desktop\whatsapp_script.py").**
- ❖ **Click Finish.**

Find Your Python Path on Your Device

Open Command Prompt (cmd) or Terminal and type:

```
where python  # For Windows
which python  # For macOS
```

To automate sending a weekly email report using Python, follow these steps:

- 1. Install Required Library

- Use the smtplib and email modules (built-in in Python). If you want to format your email in HTML, install email.message:

- pip install email

2. Write the Python Script
This script will send an email every Monday at 9:00 AM.

```
import yagmail
import schedule
import time
# إنشاء حساب الإرسال
yag = yagmail.SMTP("your_email@gmail.com")
def send_weekly_report():
    subject = "Weekly Work Report"
    body = ""  Hello
    This is your automated weekly report.
      ✅ Task 1 completed
      ⏳ Task 2 in progress
      🔜 Upcoming tasks for next week
```

```
    Regards,
    Your Name
    """
    yag.send("receiver_email@gmail.com", subject, body)
    print("📧 Email sent successfully!")

# جدولة الإرسال كل يوم اثنين الساعة 9 صباحًا
schedule.every().monday.at("09:00").do(send_weekly_report)

print("📌 Email scheduler is running...")

# تشغيل الجدولة إلى الأبد
while True:
    schedule.run_pending()
    time.sleep(60
```

3. Automate the Script to Run Weekly

➢ **Windows (Task Scheduler)**

- Open Task Scheduler.
- Click on Create Basic Task → Weekly → Monday at 9:00 AM.
- Under Action, select Start a Program.
- In Program/Script, enter the path to Python (e.g., C:\Python39\python.exe).
- In Add arguments, enter the path to your script (e.g., "C:\Users\YourName\weekly_email.py").
- Click Finish.

Auto Shutdown Your PC at a Specific Time

- **Use Case: Automatically shut down your computer after work hours**

➢ ✅ **Python Code**

```
import os


shutdown_time = "22:30"  # Set shutdown time (24-hour format)


os.system(f"shutdown /s /t 0")
print("✅ PC will shut down at", shutdown_time)
```

See you in the next session

INCEPTUM
EGYPT