

Python

Automation

with

Python

Class IX

Lab 15

**Lab Objectives:**

* Sending an Email using python Script

You have learned many syntax and theoretical things regarding python. But now you will do an automating stuff with python. We will create a simple python mailing script. By which you can send email to anyone from your python code without having visited to Gmail or Yahoo.



Checking and replying to email is a huge time sink. Of course, you can’t just write a program to handle all your email for you, since each message requires its own response. But you can still automate plenty of email-related tasks once you know how to write programs that can send and receive email.



For example, you are the owner of an Online Banking System. Maybe you have a spreadsheet full of customer records and want to send each customer a different form letter depending on their age and location details. Commercial software might not be able to do this for you; fortunately, you can write your own program to send these emails, saving yourself a lot of time copying and pasting form emails.



You can also write programs to send emails and SMS texts to notify you of things even while you’re away from your computer. If you’re automating a task that takes a couple of hours to do, you don’t want to go back to your computer every few minutes to check on the program’s status. Instead, the program can just text your phone when it’s done—freeing you to focus on more important things while you’re away from your computer.

What is SMTP?

Much like HTTP is the protocol used by computers to send web pages across the Internet, Simple Mail Transfer Protocol (SMTP) is the protocol used for sending email.

What is IMAP?

In computing, the “Internet Message Access Protocol” is an Internet standard protocol used by email clients to retrieve email messages from a mail server over a TCP/IP connection.

Sending Email

You may be familiar with sending emails from Outlook or Thunderbird or through a website such as Gmail or Yahoo! Mail. Unfortunately, Python doesn’t offer you a nice graphical user interface like those services. Instead, you call functions to perform each major step of SMTP, as shown in the following. Here we have used sublime text for the text editor. You can use any text editor, if you have not you can also write code in python interactive shell.

We have created a folder named “Email” and created “test\_email.py” inside this folder. You can use any of your desired word for naming the folder and python file except “mail.py”. Because “mail.py” is a built in python package included in Python 3. So it won’t work.

Python has a library “smtplib” to manage the email. You just need to import this.

|  |
| --- |
| test\_email.py /  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/   1. import smtplib 2. from\_address = "deepto69@gmail.com" 3. to\_address = "mahmud750@diu.edu.bd" 4. message = "Hi! I am sending this email using python script" 5. password = "type\_your\_pssword" 6. server = smtplib.SMTP('smtp.gmail.com:587') 7. server.starttls() 8. server.login(from\_address, password) 9. server.sendmail(from\_address, to\_address, message) 10. print("email sent successfully!")   11.server.quit() |

To send an email you need your own email address and another email address where you will send your mail.

In line 2 and 3 we have taken two variable and declare two mail address. We will send mail from [deepto69@gmail.com](mailto:deepto69@gmail.com) to [mahmud750@diu.edu.bd](mailto:mahmud750@diu.edu.bd) .

In line 4, we have taken a variable “message” and declare our text which we want to send.

In line 5, you have to write your password or your personal email address.

server = smtplib.SMTP('smtp.gmail.com:587')

here in server variable we have initiated smtp library. smtplib.SMTP means we called SMTP port from smtplib library. We have created a server setup with gmail. 587 is the port. Actually there are two types of email port SSL and TLS. 587 is the TLS port number. Port number 465 is used for SSL.

In line 7, we have called TLS

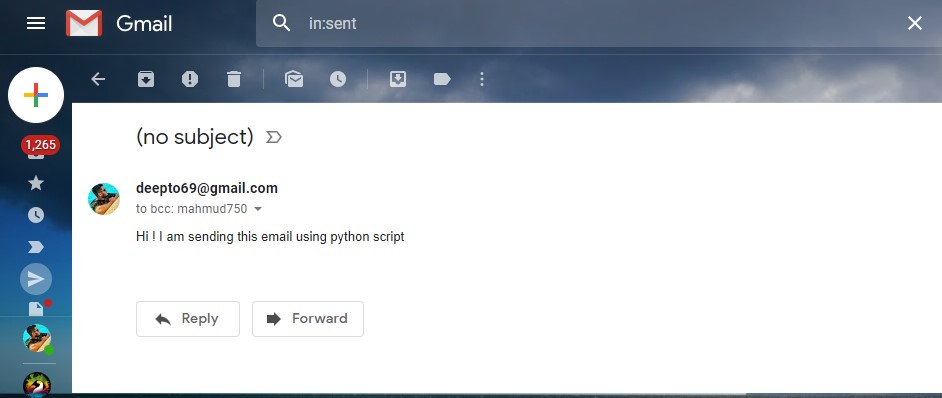
In line 8, we have logged in to our email by parsing the parameter email id and password.

In line 9, we have sent the mail using from address, to address and message.

In line 10, we print a successful message

Finally, we quit the server.

Run your test\_mail.py file. If you see the success message, then mail is sent. Check your mail sent items.



Here are the sent items. We can see we have successfully sent our mail.

You can perform thousands of automating stuffs using python. If you want to know about that, keep searching on Google. You will get plenty of resource.