

OTP Verification using Python



AMAN KHARWAL / ⌚ APRIL 14, 2021 / 📁 MACHINE LEARNING

OTP Verification System using Python

Do you know how you get a unique OTP every time you go through the payment process in an online transaction? Each company has its ways of creating an OTP for verification, but most of the companies have their systems programmed to generate a 6-digit random number. In this article, I will walk you through the task of OTP verification using Python. By the end of this article, you will be able to send a unique OTP to any email id for the OTP verification task.



Steps to Create an OTP Verification System using Python

OTP Verification is the process of verifying a user by sending a unique password so that the user can be verified before completing a registration or payment process. Most of the time, we get an OTP when we make an online payment, or when we forget our password, or when creating an account on any online platform. Thus, the sole purpose of an OTP is to verify the identity of a user by sending a unique password.

We can easily create an application for the task of OTP verification using Python by following the steps mentioned below:

1. First, create a 6-digit random number
2. Then store the number in a variable
3. Then we need to write a program to send emails

4. When sending email, we need to use OTP as a message
5. Finally, we need to request two user inputs; first for the user's email and then for the OTP that the user has received.

So this is the complete process of creating an OTP verification application using Python. In the section below, I will take you through how to implement these steps using Python for the task of OTP verification.



OTP Verification using Python

I hope you now have understood what is an OTP and how we can create an application for the task of OTP verification. Now let's follow the steps mentioned above by using Python to create an application for the task of OTP verification. I will start by importing the necessary Python library that we need for this task:

```
1 import os
2 import math
```

```
3 import random
4 import smtplib
```

Now I will generate a random number and store it in a variable which I will be using while sending emails to the users:

```
1 digits="0123456789"
2 OTP=""
3 for i in range(6):
4     OTP+=digits[math.floor(random.random()*10)]
5 otp = OTP + " is your OTP"
6 msg= otp
```

OTP verification.py hosted with ❤️ by GitHub

[view raw](#)

Now, before we go ahead, you need to have your Google app password to be able to send emails using your Gmail account. For this task, you need to follow the steps mentioned [here](#). After you create your app password for your Gmail account you will get a key. Copy that key and paste in the code below to send emails for OTP verification using Python:

```
1 s = smtplib.SMTP('smtp.gmail.com', 587)
2 s.starttls()
3 s.login("Your Gmail Account", "You app password")
4 emailid = input("Enter your email: ")
5 s.sendmail('&&&&&&&&&',emailid,msg)
6 a = input("Enter Your OTP >>: ")
7 if a == OTP:
8     print("Verified")
9 else:
10     print("Please Check your OTP again")
```

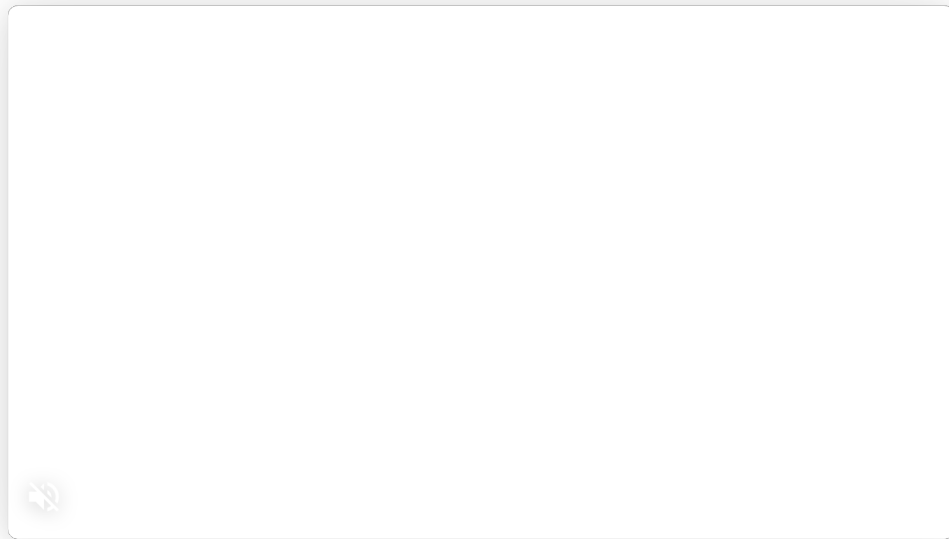
OTP verification.py hosted with ❤️ by GitHub

[view raw](#)

```
Enter your email: aman@thecleverprogrammer.com
Enter Your OTP >>: 478348
```

Verified

Once you run this code you enter an email where you want to send an OTP and then enter the OTP that you have received in the email. You can get the complete code used in this article for the task of OTP verification from below.



```
1  import os
2  import math
3  import random
4  import smtplib
5
6  digits="0123456789"
7  OTP=""
8  for i in range(6):
9      OTP+=digits[math.floor(random.random()*10)]
10 otp = OTP + " is your OTP"
11 msg= otp
12 s = smtplib.SMTP('smtp.gmail.com', 587)
13 s.starttls()
14 s.login("Your Gmail Account", "Your app password")
15 emailid = input("Enter your email: ")
16 s.sendmail('&&&&&&&&',emailid,msg)
17 a = input("Enter Your OTP >>: ")
18 if a == OTP:
19     print("Verified")
```

```
20     else:  
21         print("Please Check your OTP again")
```

OTP verification.py hosted with ❤ by GitHub

[view raw](#)

[Also, Read – Python Projects with Source Code.](#)

Summary

So this is how you can create an application for your task of OTP verification. The next thing that you can do is use this logic to create the same application in a form of a user interface. I hope you liked this article on OTP verification using Python. Feel free to ask your valuable questions in the comments section below.



Aman Kharwal

Data Strategist at Statso. My aim is to decode data science for the real world in the most simple words.

[ARTICLES: 1961](#)



PREVIOUS POST

Game Development Projects
using Python

NEXT POST

Best Datasets for Data science
Beginners



Recommended For You



Python One Liners for Data Cleaning

September 20, 2025



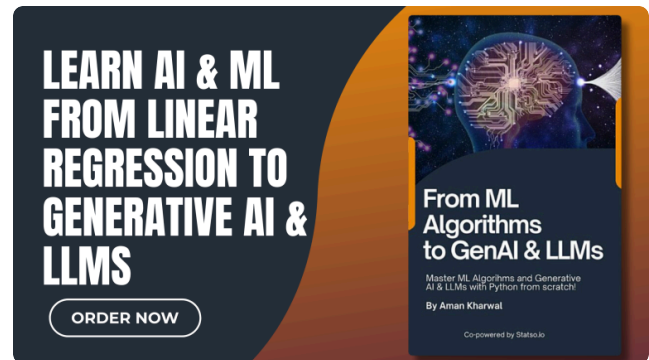
Pandas Tricks for Data Scientists

August 22, 2025



5 Python One-Liners Every ML Engineer Should Know

August 15, 2025



From ML Algorithms to GenAI & LLMs: Book Overview

October 26, 2024

Leave a Reply

 FACEBOOK  INSTAGRAM  MEDIUM  LINKEDIN

Copyright © AmanXai.com 2025