AES Encryption

Click enter to start

ENTER

TABLE OF CONTENTS

01

02

03

04

INTRODUCTION

programming language

Coding

Output

01.

INTRODUCTION

The Advanced Encryption Standard (AES)

The Advanced Encryption Standard (AES)

01
02
03
04

In our presentation, we talking about the AES algorithm (encryption)

01
02
03
04

What is AES?

AES is a block cipher that encrypts a 128-bit block (original message) to a 128-bit block (encrypted message) or decrypts a 128-bit block (encrypted message) to a 128-bit block (original message).

01
02
03
04

AES Info

AES Cipher - Rijndael

Block size	128 bit data
key sizes	128/192/256 bit keys
Rounds	9/11/13 rounds

01
02
03
04

02.
programming language

We use python

01
02
03
04

What python provides for Cryptography?

- **Crypto.Cipher:** It package contains algorithms for encrypt data… and one of this algorithms is AES.

- **Crypto import Random:** It return random byte string…

01
02
03
04

We import the libraries that we will use in this program

- **Import os:** This package for interact with the operating system.
- **import os.path:** To read, write and, access to different files & path name manipulation.

We import the libraries that we will use in this program

- **from os.path import isfile, join:** It used to know if the file exists or not.
- **import time:** It handle time-related tasks.

01
02
03
04

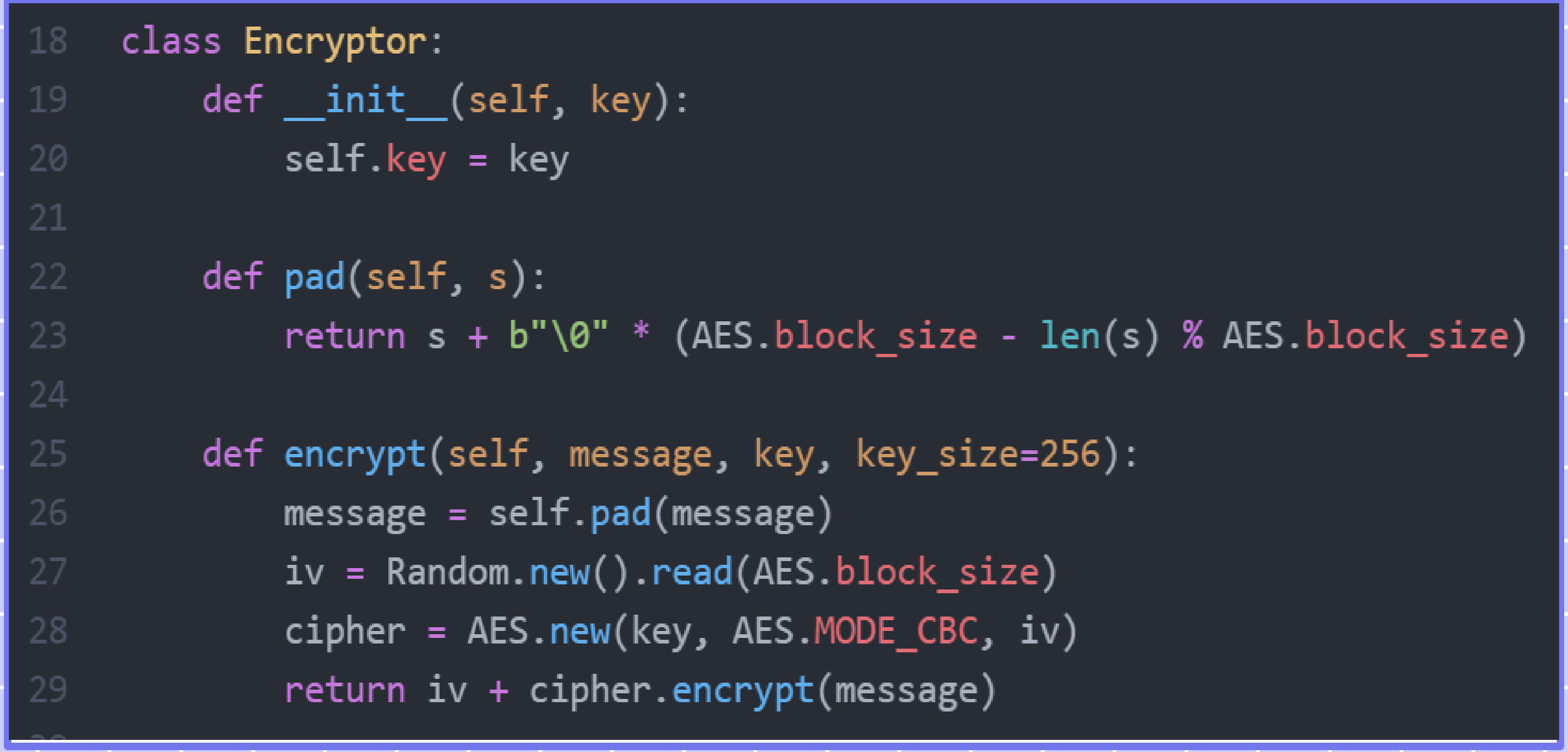
03.

Coding

Class Encryptor

01
02
03
04

The methods inside Encryptor class



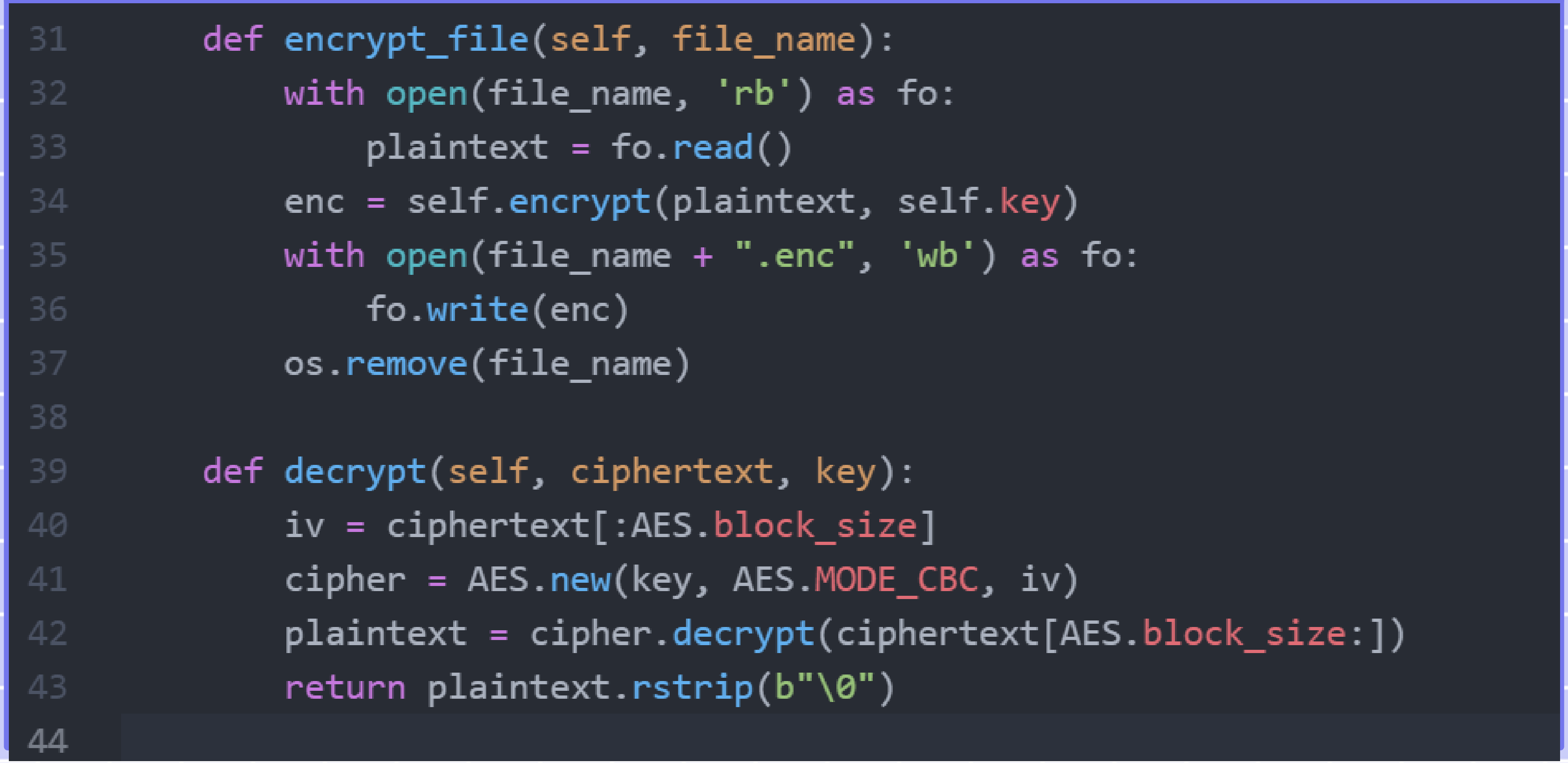
```
class Encryptor:
    def __init__(self, key):
        self.key = key

    def pad(self, s):
        return s + b"\0" * (AES.block_size - len(s) % AES.block_size)

    def encrypt(self, message, key, key_size=256):
        message = self.pad(message)
        iv = Random.new().read(AES.block_size)
        cipher = AES.new(key, AES.MODE_CBC, iv)
        return iv + cipher.encrypt(message)
```

01
02
03
04

The methods inside Encryptor class

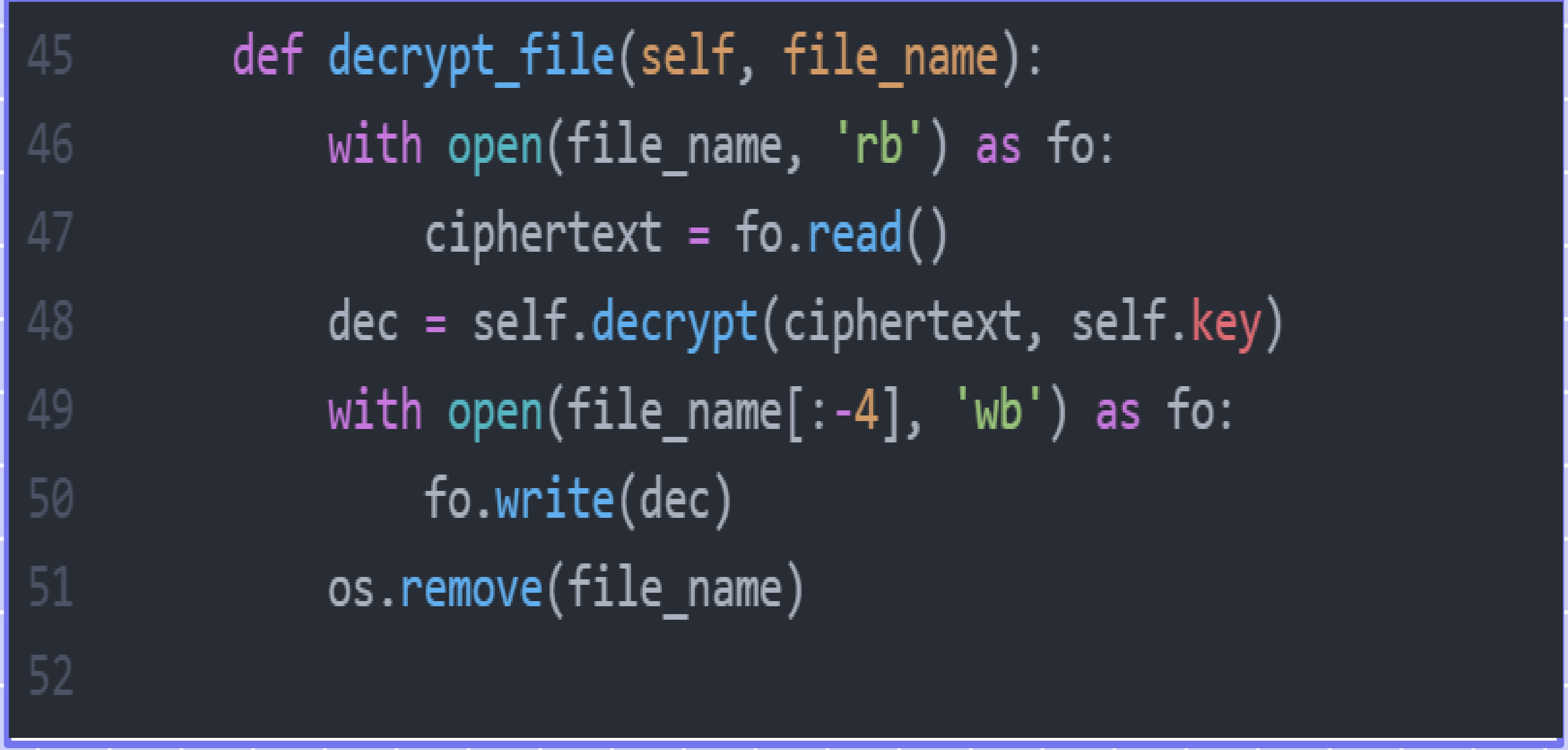


```
    def encrypt_file(self, file_name):
        with open(file_name, 'rb') as fo:
            plaintext = fo.read()
        enc = self.encrypt(plaintext, self.key)
        with open(file_name + ".enc", 'wb') as fo:
            fo.write(enc)
        os.remove(file_name)

    def decrypt(self, ciphertext, key):
        iv = ciphertext[:AES.block_size]
        cipher = AES.new(key, AES.MODE_CBC, iv)
        plaintext = cipher.decrypt(ciphertext[AES.block_size:])
        return plaintext.rstrip(b"\0")
```

The methods inside Encryptor class

01
02
03
04

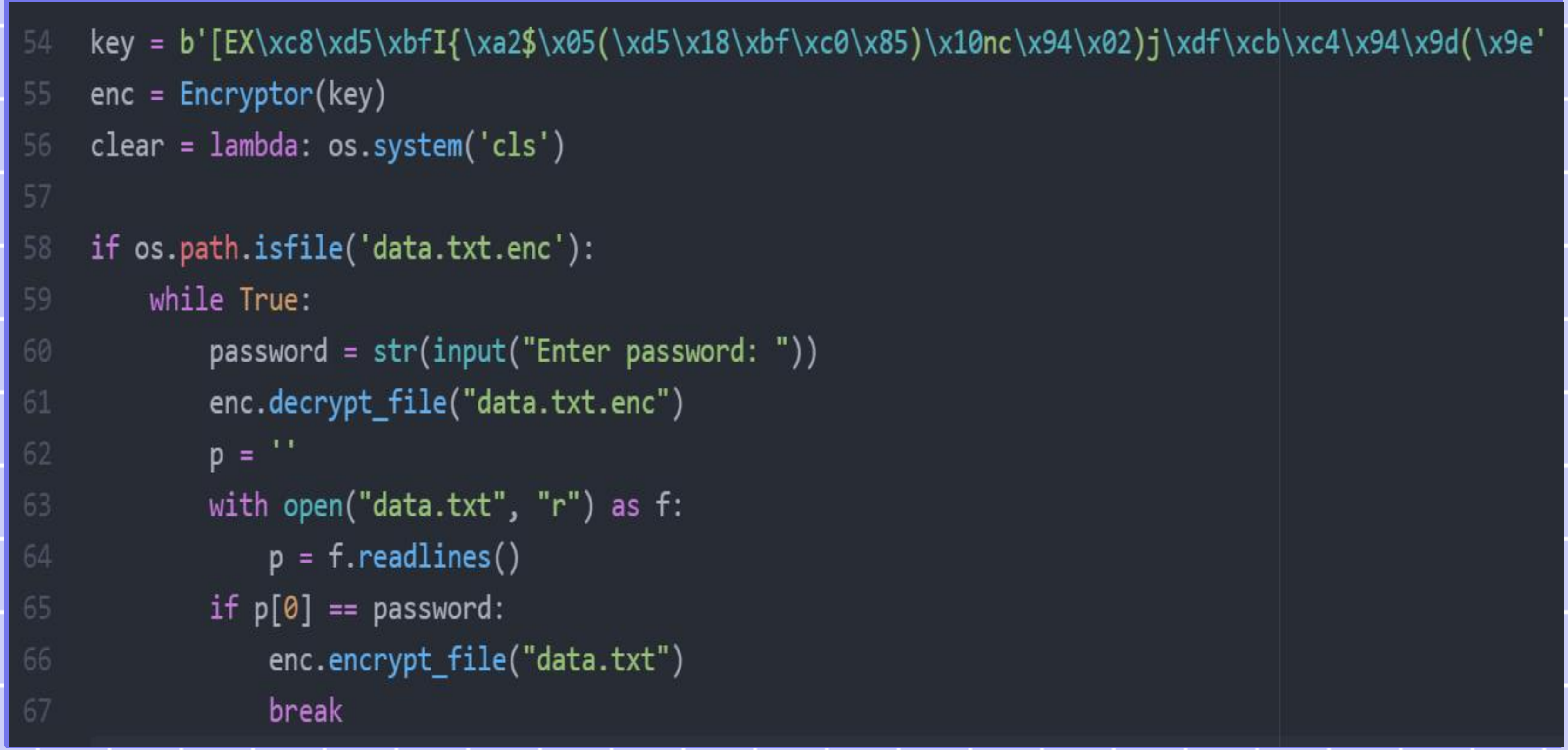


```python
    def decrypt_file(self, file_name):
        with open(file_name, 'rb') as fo:
            ciphertext = fo.read()
        dec = self.decrypt(ciphertext, self.key)
        with open(file_name[:-4], 'wb') as fo:
            fo.write(dec)
        os.remove(file_name)

```

01
02
03
04

Key formatting and password

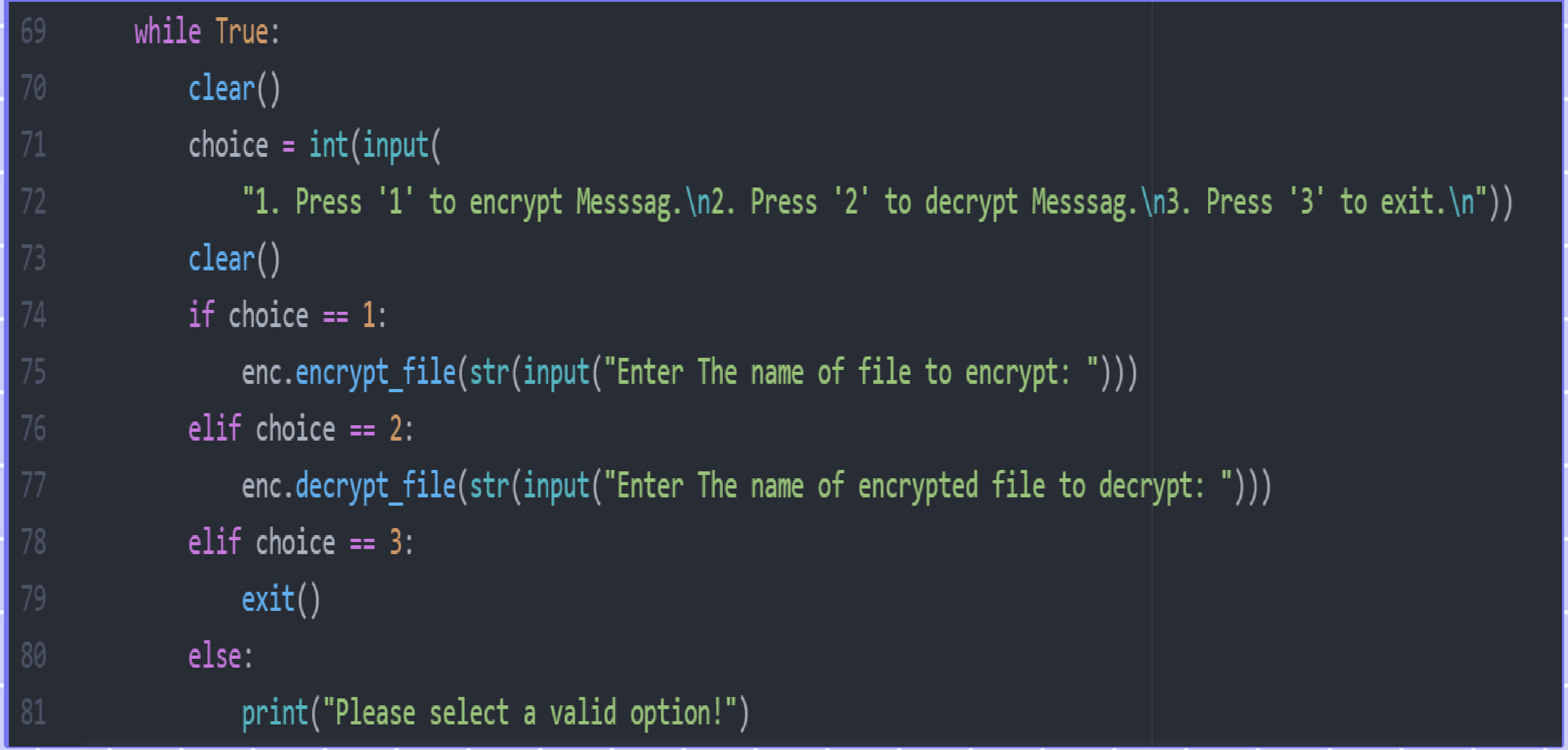


```python
key = b'[EX\xc8\xd5\xbfI{\xa2$\x05(\xd5\x18\xbf\xc0\x85)\x10nc\x94\x02)j\xdf\xcb\xc4\x94\x9d(\x9e'
enc = Encryptor(key)
clear = lambda: os.system('cls')

if os.path.isfile('data.txt.enc'):
    while True:
        password = str(input("Enter password: "))
        enc.decrypt_file("data.txt.enc")
        p = ''
        with open("data.txt", "r") as f:
            p = f.readlines()
        if p[0] == password:
            enc.encrypt_file("data.txt")
            break
```

The selection of which operation to do

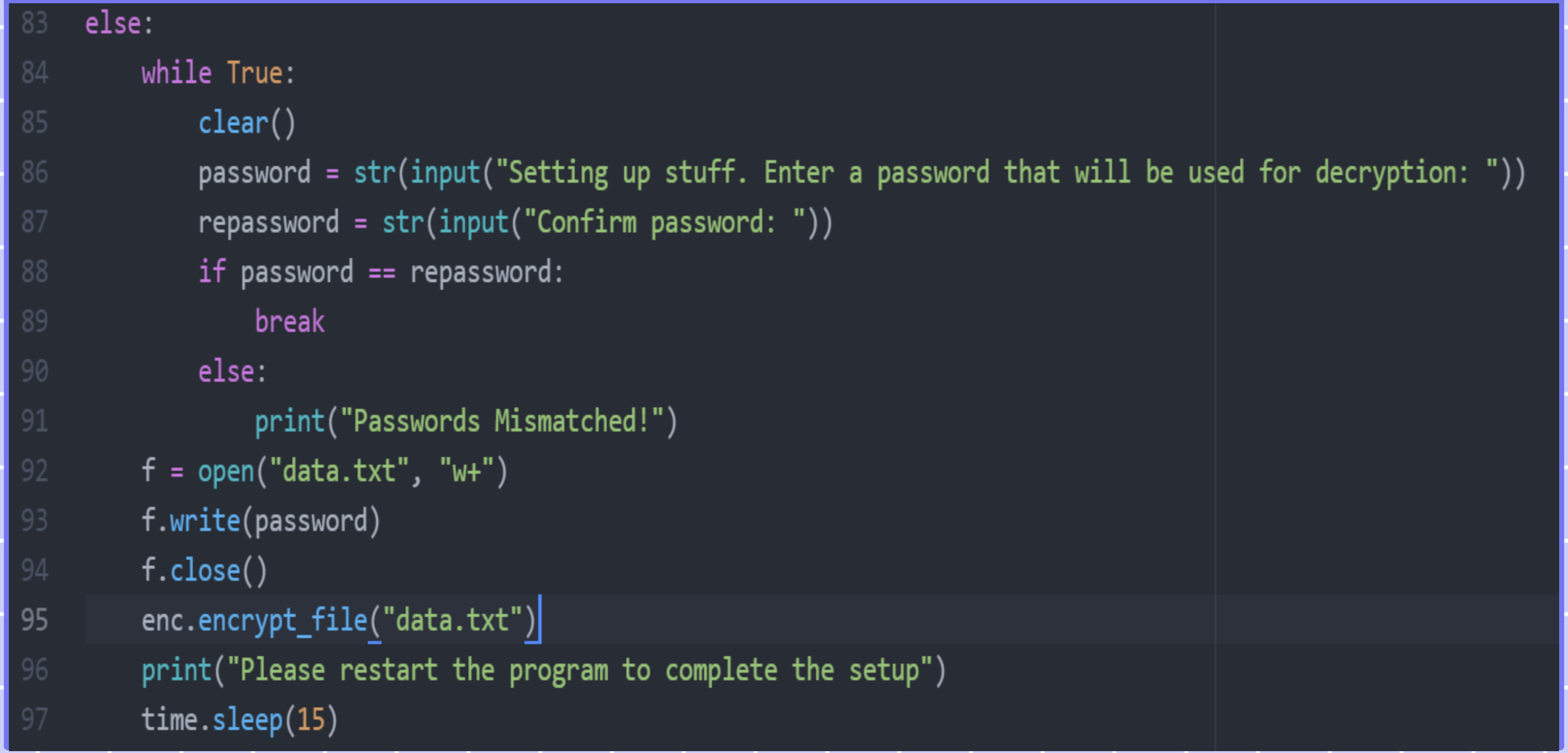
01
02
03
04



```python
    while True:
        clear()
        choice = int(input(
            "1. Press '1' to encrypt Messsag.\n2. Press '2' to decrypt Messsag.\n3. Press '3' to exit.\n"))
        clear()
        if choice == 1:
            enc.encrypt_file(str(input("Enter The name of file to encrypt: ")))
        elif choice == 2:
            enc.decrypt_file(str(input("Enter The name of encrypted file to decrypt: ")))
        elif choice == 3:
            exit()
        else:
            print("Please select a valid option!")
```

First time to use

01
02
03
04



```python
else:
    while True:
        clear()
        password = str(input("Setting up stuff. Enter a password that will be used for decryption: "))
        repassword = str(input("Confirm password: "))
        if password == repassword:
            break
        else:
            print("Passwords Mismatched!")
    f = open("data.txt", "w+")
    f.write(password)
    f.close()
    enc.encrypt_file("data.txt")
    print("Please restart the program to complete the setup")
    time.sleep(15)
```

01

02

03

04

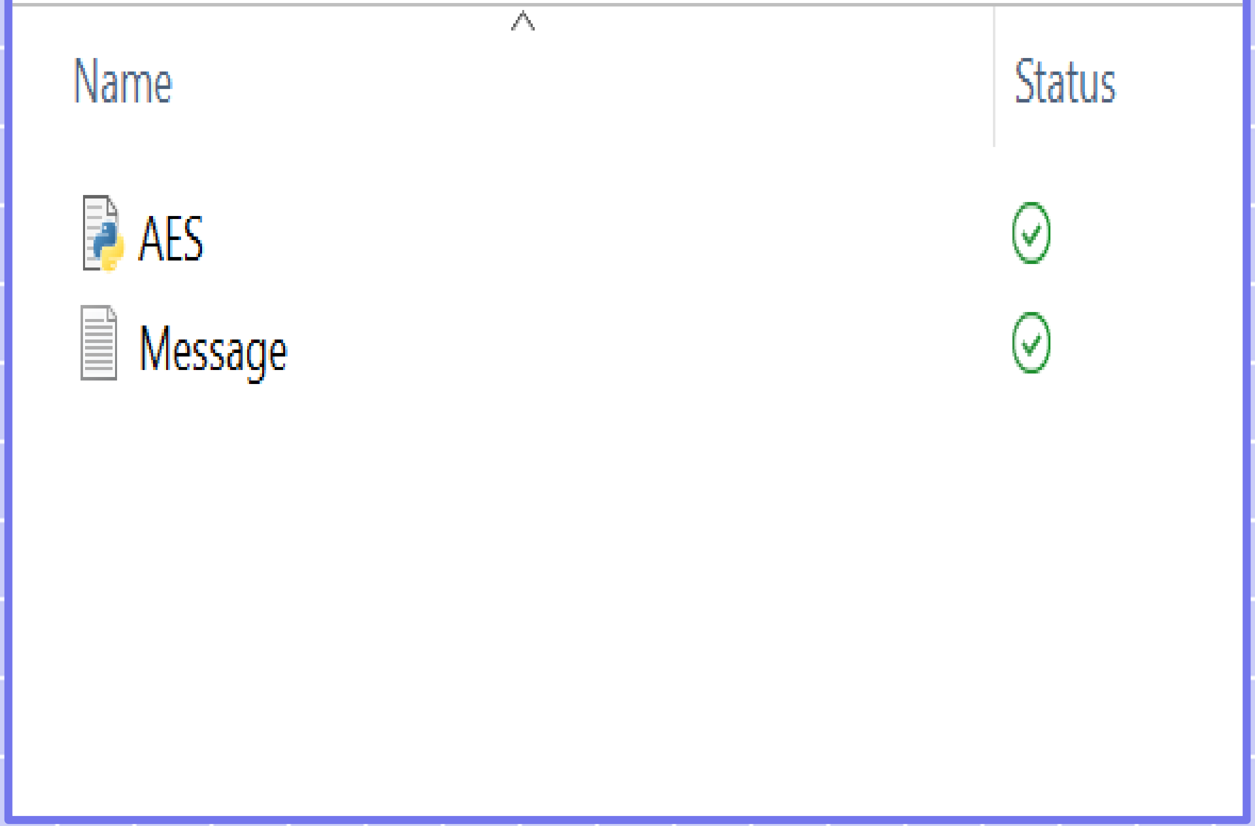
04.

Output

Result

01
02
03
04

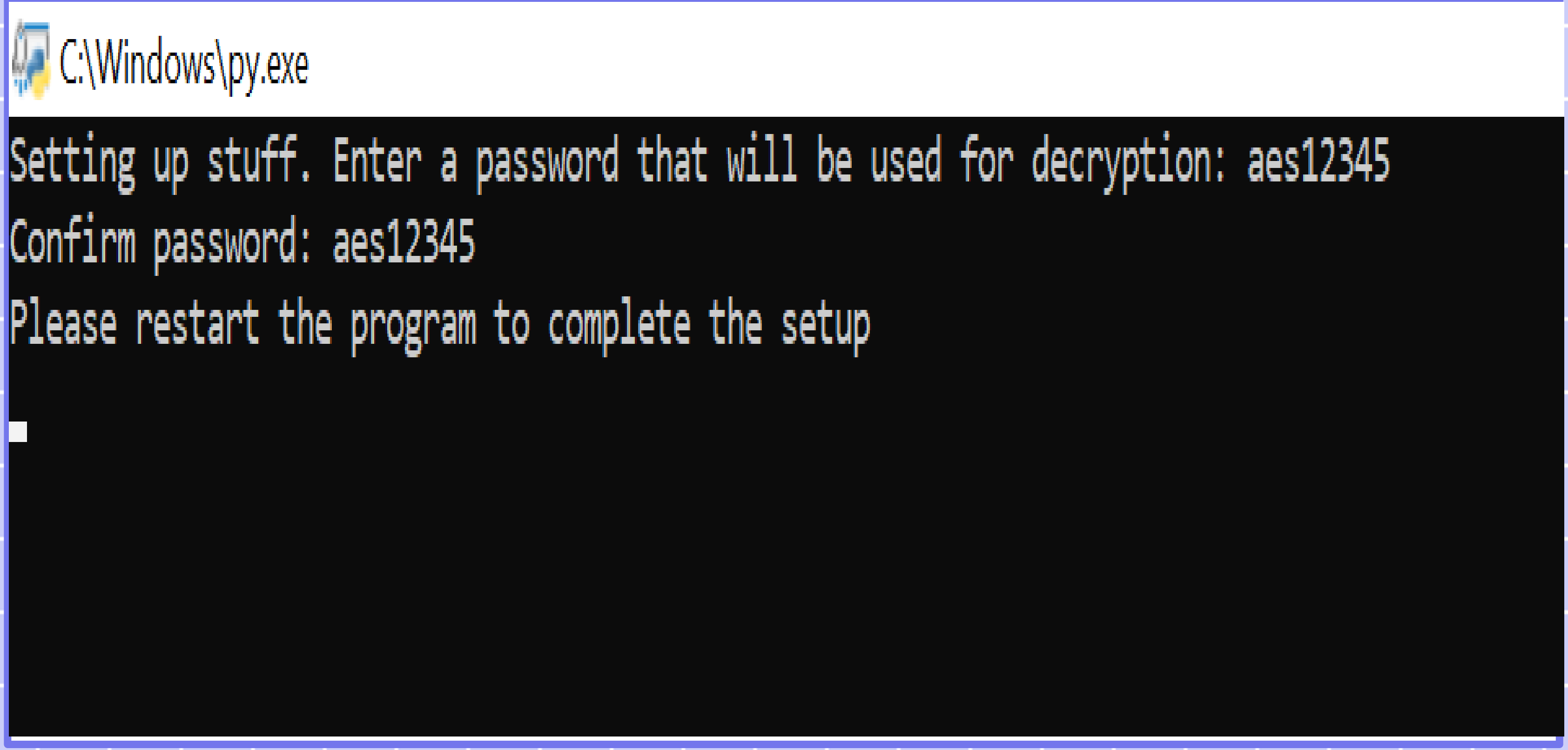
Folder content



Name	Status
AES	✓
Message	✓

First time use

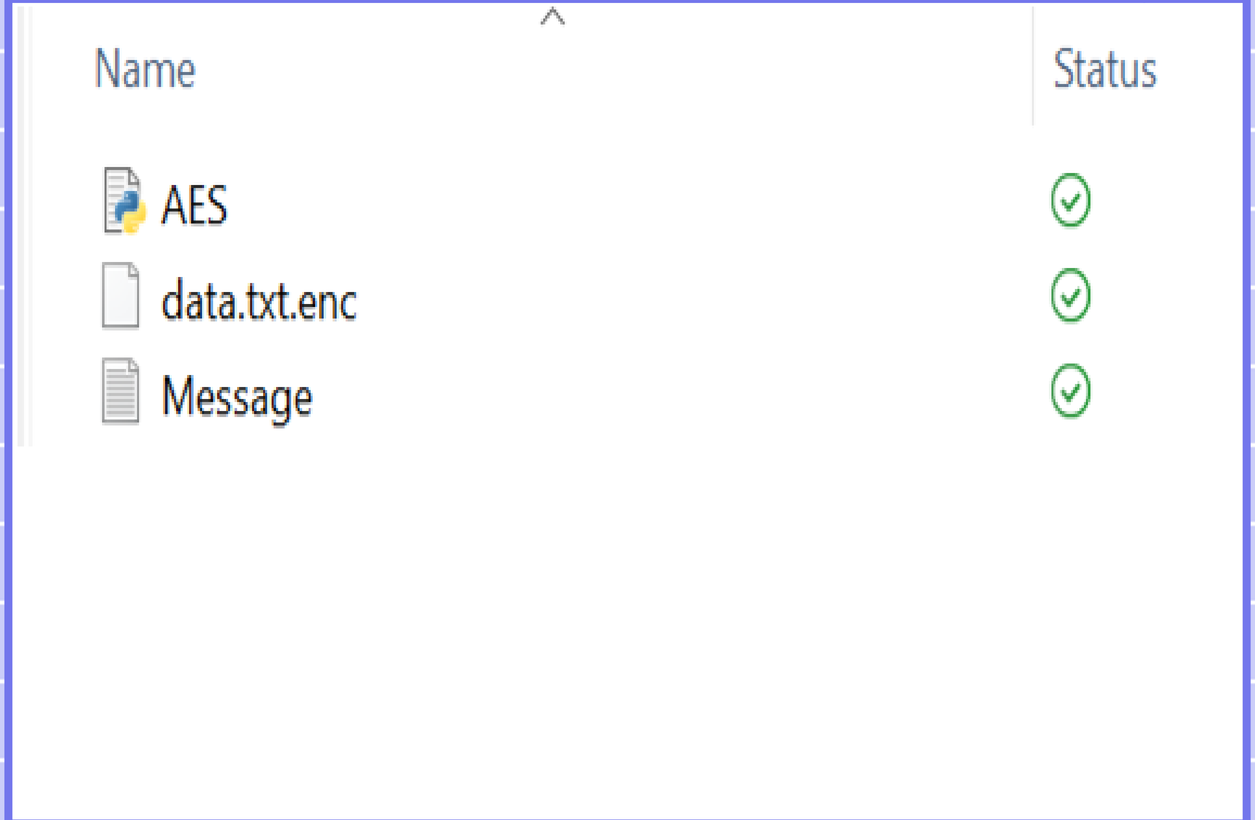
01
02
03
04



```
Setting up stuff. Enter a password that will be used for decryption: aes12345
Confirm password: aes12345
Please restart the program to complete the setup
```

01
02
03
04

New file add (data)

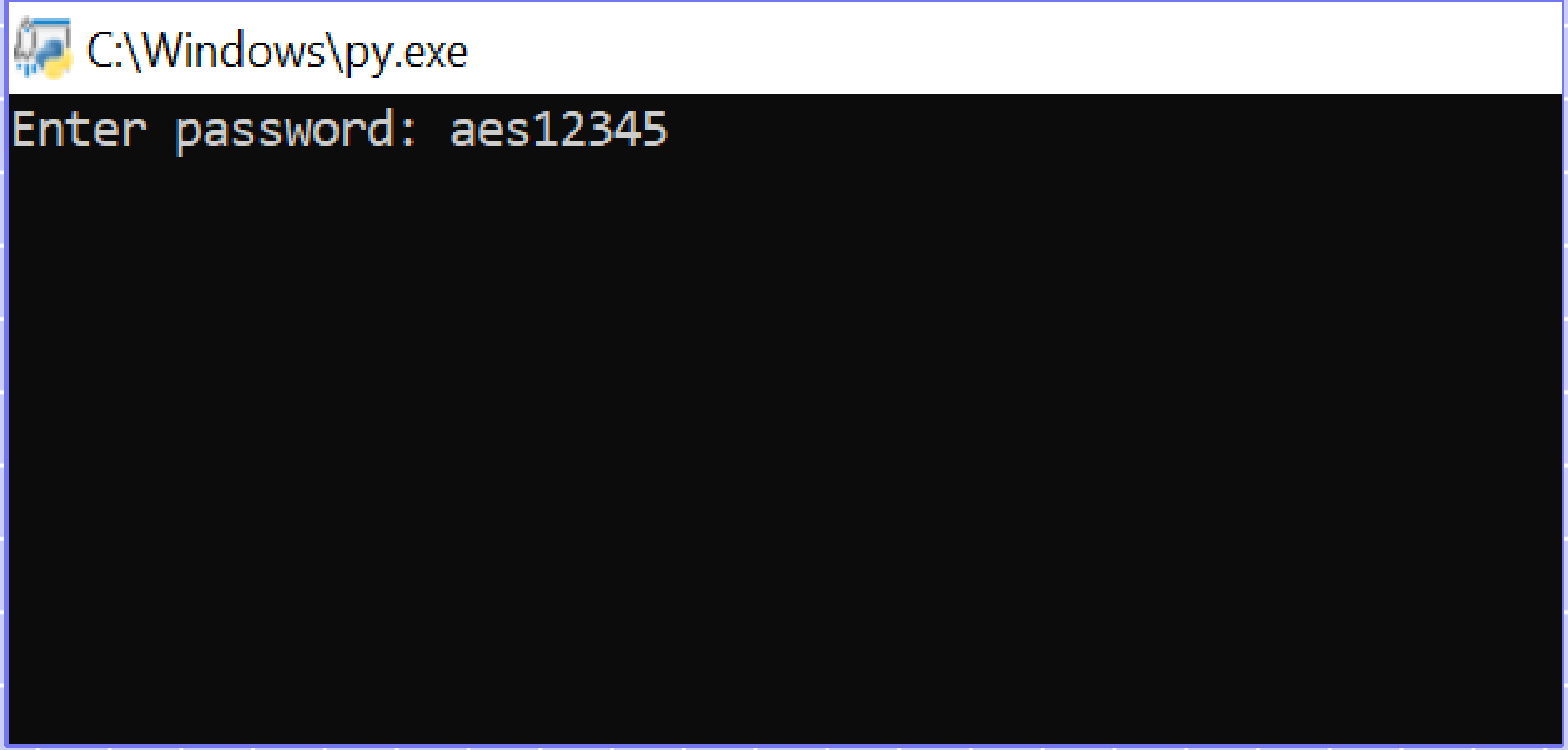


Name	Status
AES	✓
data.txt.enc	✓
Message	✓

☰
❮
01
02
03
04
❯
⬆

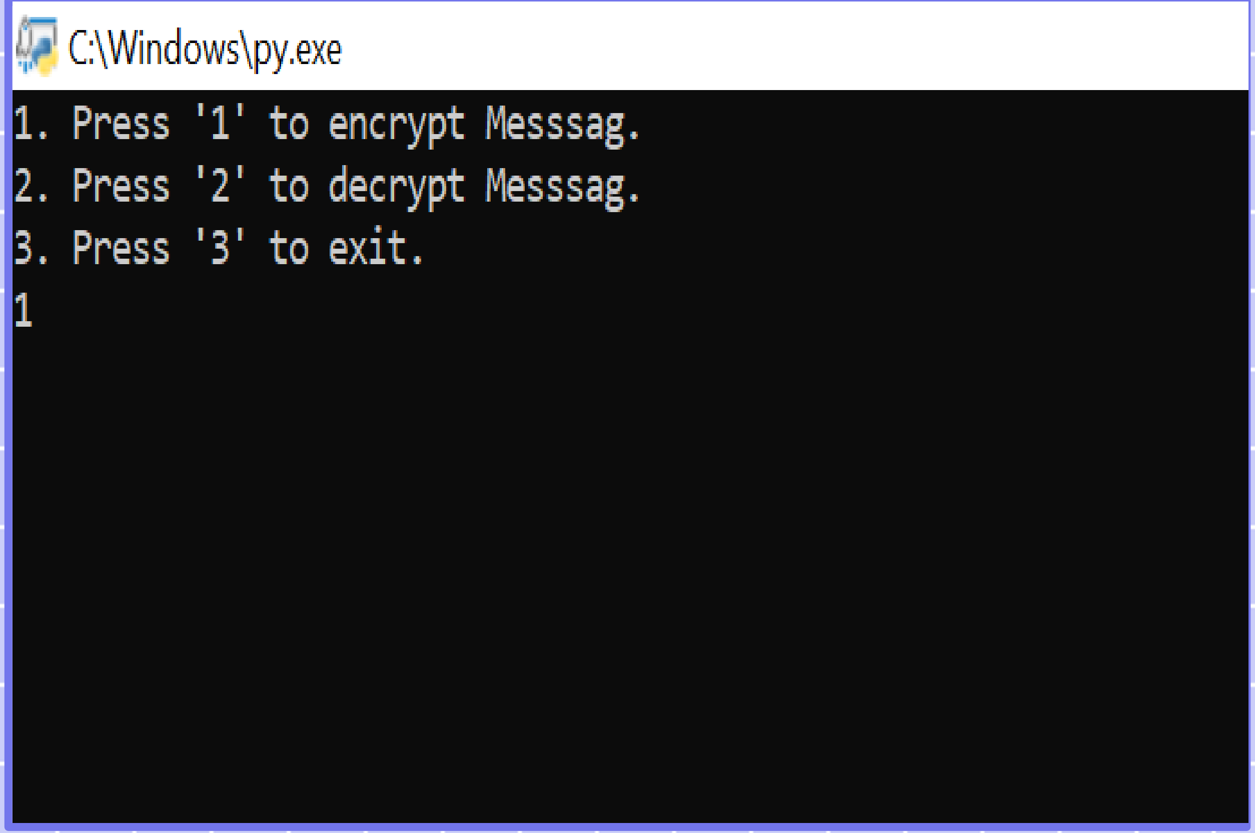
Second time use

✕



```
Enter password: aes12345
```

The selection of which operation to do

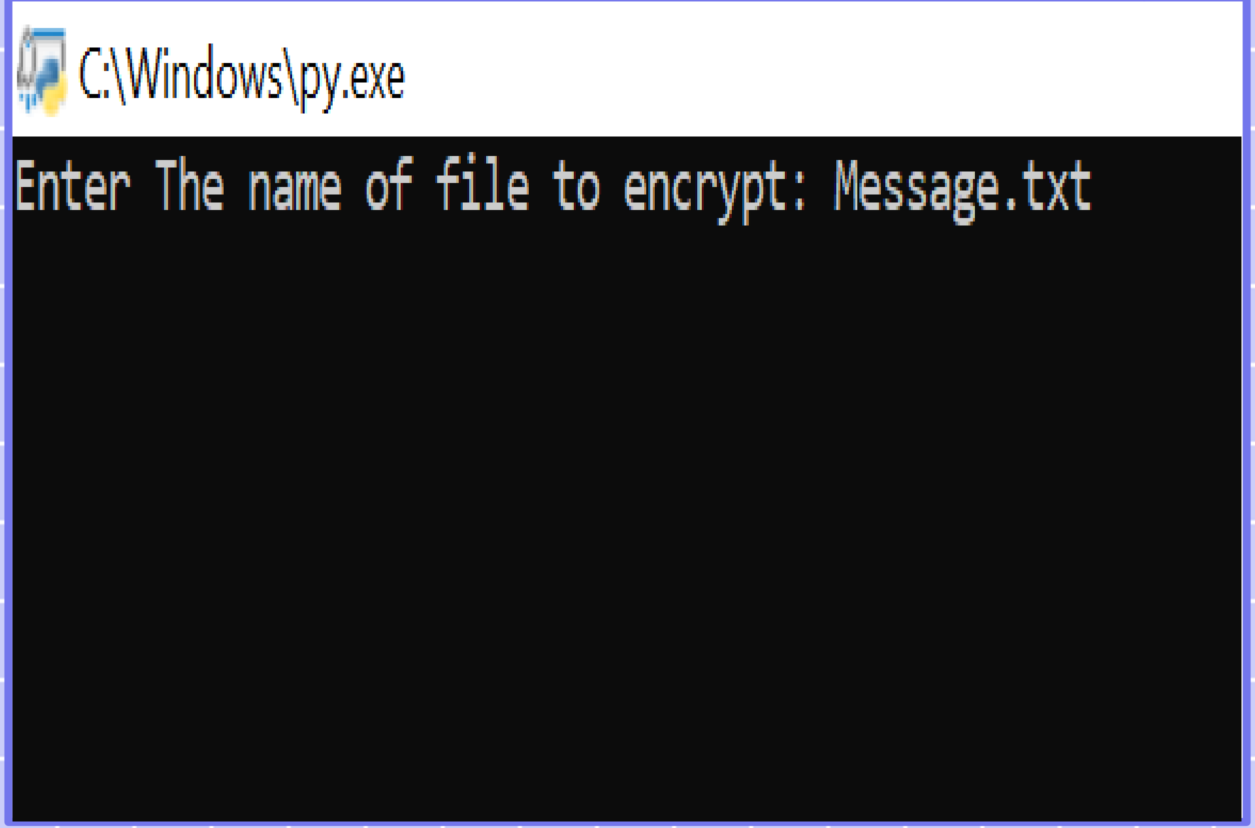


C:\Windows\py.exe

```
1. Press '1' to encrypt Messsag.
2. Press '2' to decrypt Messsag.
3. Press '3' to exit.
1
```

❮
01
02
03
04
❯

We type file name

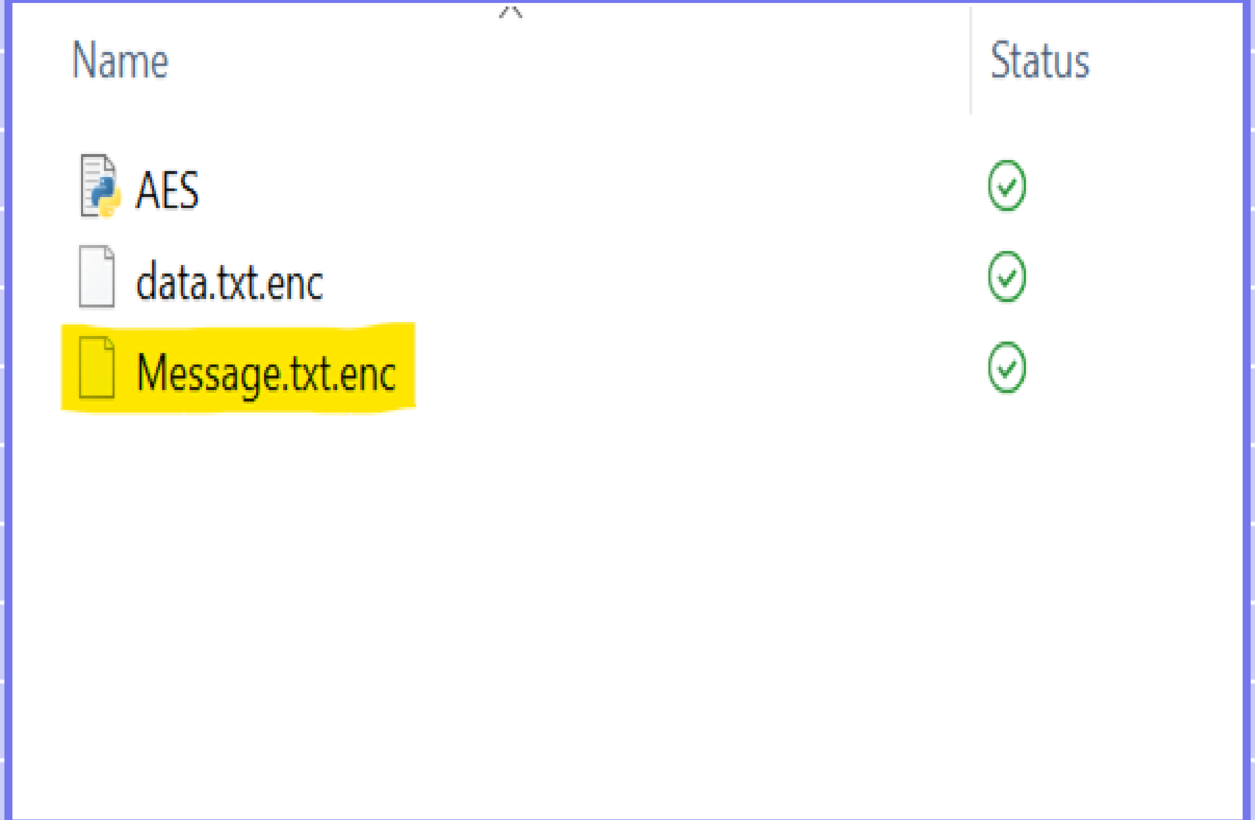


C:\Windows\py.exe

```
Enter The name of file to encrypt: Message.txt
```

01
02
03
04

The extension of the file changed



Name	Status
AES	✓
data.txt.enc	✓
Message.txt.enc	✓

01
02
03
04

References:

[1] Javapocalypse, “Python AES Encryption/Decryption using PyCrypto Tutorial,” 2018. https://www.youtube.com/watch?v=UB2VX4vNUa0 (accessed Apr. 01, 2021).

[2] Developer Tool-kit, “AES Encryption Decryption Tool.” https://aesencryptiondecryption.tool-kit.dev/ (accessed Mar. 30, 2021).

[3] PyCryptodome, “AES -- PyCryptodome3.9.9 docimentation.” https://pycryptodome.readthedocs.io/en/latest/src/cipher/aes.html (accessed Apr. 02, 2021).

[4] PyCryptodome, “API documentation -- PyCryptodome.3.9.9 documentation.” https://pycryptodome.readthedocs.io/en/latest/src/api.html (accessed Apr. 02, 2021).

01
02
03
04

References

[5] Python, “os.path — Common pathname manipulations.” https://docs.python.org/3/library/os.path.html (accessed Apr. 03, 2021).

[6] Tutorialspoint, “SDLC - Waterfall Model - Tutorialspoint.” https://www.tutorialspoint.com/sdlc/sdlc_waterfall_model.htm#:~:text=It is very simple to,was used for software development. (accessed Oct. 12, 2020).

[7] GeeksforGeeks, “Python | os.path.isfile() method - GeeksforGeeks.” https://www.geeksforgeeks.org/python-os-path-isfile-method/ (accessed Apr. 05, 2021).

[8] programiz, “Python time Module (with Examples).” https://www.programiz.com/python-programming/time (accessed Apr. 07, 2021).

01
02
03
04

References

[9] Muhammad Hashir, "Padding Strings in Python." https://stackabuse.com/padding-strings-in-python/ (accessed Apr. 06, 2021).

[10] Edureka, "Init In Python | Python Init Class | What is Inti Function |Edureka," 2019. https://www.edureka.co/blog/init-in-python/#:~:text=The self in keyword in,is known as a constructor. (accessed Apr. 10, 2021).

01

02

03

04

Created by:

Fatima Mohammed Hour

Wafa Mohammed Bahubail

Marwah Saleh Allawi

Seham Naif AL qauod

01

02

03

04

THANKS!

Do you have any questions?

addyouremail@freepik.com
+91 620 421 838
yourcompany.com

CREDITS: This presentation template was created by **Slidesgo**, including icons by **Flaticon**, and infographics & images by **Freepik**

Please keep this slide for attribution