

Passwort Manager

Betriebssysteme und Rechnernetze

Petar Milutinovic, Zahra Mechershavi, Zehra Yildiz

Gliederung

- Import
- Masterpasswort
- Zwischenablage
- Liste strukturieren
- Passwort generieren
- Sichtbarkeit der Passwörter
- Passwort löschen
- Passwort ändern
- Programm beenden

Import

```
from datetime import datetime
import getopt
import secrets
import string
import time
from threading import Thread
import pyperclip
import sys
import os
```

Masterpassword

- Benutzer erstellt sein Masterpassword

```
mp = stdiomask.getpass("Bitte geben Sie ein Masterpassword für Ihren Passwort Manager ein: ")
global masterPassword
masterPassword=mp
EingabeMp = stdiomask.getpass("Bitte geben Sie nun Ihr Masterpassword ein: ")

while mp != EingabeMp:
    if mp != EingabeMp:
        print(("Falsches Passwort! "))
        break
if mp == EingabeMp:
    print("Die Eingabe war korrekt." + '\n')
    print("Bitte wählen Sie eines der folgenden Möglichkeiten aus: ")
```

Masterpassword Timer

- Masterpassword wird nach 10 min abgefragt

```
masterPassword=""
startTime=datetime.now()

def checkIfMasterIsneeded():
    if (datetime.now()-startTime).total_seconds() > 600:
        askMaster=input("Please write MP: ")
        if askMaster==masterPassword:
            print("Richtig")
        else:
            print("Falsch")
            sys.exit(0)
```

Zwischenablage

- Passwort wird für 30 Sekunden in der Zwischenablage kopiert

```
def startDeleteTimer():  
    def deletePassword():  
        time.sleep(30)  
        pyperclip.copy("")  
        print("Zwischenablage geleert")  
    thread = Thread(target=deletePassword)  
    thread.start()
```

```
pyperclip.copy(pw)
```

Liste strukturieren

- Liste in der File wird geordnet
- Append: neue Daten werden hinzugefügt
- Split: Erstellung Liste aus den Strings

```
file = open('password.txt', 'r')
lines = file.readlines()

for index, line in enumerate(lines):
    userdata = line.split(';')
    websites.append(userdata[0].split(" ")[2])
    usernames.append(userdata[1].split(" ")[2])
    passwords.append(userdata[2].split(" ")[2].replace('\n', ''))
```

Passwort generieren

- Überprüfen, ob die Länge des Passwortes mit den Eingaben übereinstimmen
- Passwort wird zusammen gewürfelt mit den Eingaben
- Zur Veranschaulichung, müssen wir den Code öffnen

Auswahlmöglichkeiten

```
print("Auswahl 1: Sichtbarkeit Passwörter")  
print("Auswahl 2: Passwort generieren")  
print("Auswahl 3: Passwort löschen")  
print("Auswahl 4: Psswort ändern")  
print("Auswahl 5: Beenden")  
auswahl = int(input("Ihre gewünschte Auswahl : "))  
moeglichkeit = auswahl
```

Anschau Passwörter

- Die File wird geöffnet und die ganzen Passwörter werden angezeigt

```
if moeglichkeit == 1:
    checkIfMasterIsneeded()
    with open('password.txt', 'r') as f:
        f_contents = f.read()
        print(f_contents + '\n')
```

Passwort generieren

- Benutzer kann sich das Passwort selbst aussuchen
- Oder er bekommt nach seinen Angaben eins zugestellt
- Wird in der File gespeichert

```
elif moeglichkeit == 2:
    checkIfMasterIsneeded()
    ws = input("Bitte geben Sie den Namen der Webseite an: ").lower()
    bn = input("Bitte geben Sie Ihren Benutzernamen ein: ")
    frage = input("Möchten Sie ein neues Passwort generieren lassen oder Ihr eingenes eingeben? [g/e]")
    if frage == "g":
        generiertes_passwort = passGenerieren()
        fe(ws, bn, generiertes_passwort)
    elif frage == "e":
        eigenes_passwort = input("Bitte geben Sie Ihr Passwort ein: ")
        pyperclip.copy(eigenes_passwort)
        print("Dein erstelltes Passwort wurde im Clipboard gespeichert. ")
        fe(ws, bn, eigenes_passwort)
    startDeleteTimer();
```

Passwort löschen

- Benutzer hat die Möglichkeit sein Passwort zu löschen
- Er muss dafür die dazugehörige Webseite angeben

```
elif moeglichkeit == 3:
    checkIfMasterIsneeded()
    print("Passwort loeschen:")
    websiteZuLoeschen = input("Website?:")
    auswahlSicher = input("Sicher? J/N")
    if auswahlSicher.lower() == "j":
        elementAnStelle = -1
        for i in range(len(websites)):
            if websites[i].lower() == websiteZuLoeschen.lower():
                websites.pop(i)
                usernames.pop(i)
                passwords.pop(i)
                rewriteFile()
                elementAnStelle = i
                break
        if elementAnStelle == -1:
            print("Eintrag nicht gefunden")
        else:
            print("Eintrag geloescht")
```

```
def rewriteFile():
    os.remove('password.txt')
    outfile = open('password.txt', 'a')
    for i in range(len(websites)):
        outfile.write(f" Webseite: {websites[i]} ; Benutzername: {usernames[i]} ; Passwort: {str(passwords[i])}\n")
    readPasswords()
```

Passwort ändern

- Benutzer hat die Option Passwort ändern zu können
- Webseite angeben

```
elif moeglichkeit == 4:
    checkIfMasterIsneeded()
    print("Passwort Aendern:")
    passwrdaenderung = input("Website?:")
    elementAnStelle = -1
    for i in range(len(websites)):
        if websites[i].lower() == passwrdaenderung.lower():
            elementAnStelle = i
            break
    if elementAnStelle == -1:
        print("Eintrag nicht gefunden")
    else:
        neuesPasswort = input("Bitte geben Sie das neue Passwort ein: ")
        passwords[elementAnStelle] = neuesPasswort
        print("Passwort geandert")
        rewriteFile()
```

```
def rewriteFile():
    os.remove('password.txt')
    outfile = open('password.txt', 'a')
    for i in range(len(websites)):
        outfile.write(f" Webseite: {websites[i]} ; Benutzername: {usernames[i]} ; Passwort: {str(passwords[i])}\n")
    readPasswords()
```

Programm beenden

- Programm wird beendet

```
elif moeglichkeit == 5:  
    print("Auf Wiedersehen!")  
    exit()
```