



Password Manager Werkstück A

Leon F. Fischer

Matrikelnummer: 1258246

lfischer@stud.fra-uas.de

Inhaltsverzeichnis

- Menüführung
- Erstellen eines neuen Passwortes
- Generierungsfunktion
- Speicher Funktion
- Timer Funktion
- Verschlüsselung
- Löschen eines Passwortes

Menüführung

```
Welcome to Passwordmanager.  
What would you like to do?  
  
Choose an option via arrow keys:  
Recall    <-  
New  
Options  
Support Us  
<-Exit  
  
Press Enter to continue...
```

Hauptmenü des Programmes

- Optionen in logischer Reihenfolge
- Verwendung mit Pfeiltasten und Eingabetaste

```
menu=[ 'Recall\t', 'New\t', 'Options\t', 'Support Us',  
'\x1b[6;31m'+ '<-Exit\t'+ '\x1b[0m' ]
```

```
def show_menu(): #displays main menu
```

```
...
```

```
print("\n" * 2)
```

```
print('\x1b[6;32m'+ 'Choose an option via arrow keys:' + '\x1b[0m')
```

```
for i in range(1, length):
```

```
print(menu[i-1], "{1}".format(i, "\x1b[6;32m" + "<-  
" + "\x1b[0m" if selected == i else " "))
```

```
print("\nPress Enter to continue...")
```

- String der Optionen

Menüfunktion

- Drucken der Erklärung
- Drucken des Menüs und Einfügen des Cursor mit .format

```
keyboard.add_hotkey('up', up)
keyboard.add_hotkey('down', down)
```

```
def down(): # moves cursor down
global selected
global move
if(move==True):
if selected == length-1:
return
selected += 1
clear()
```

```
if(Where=='menu'):
show_menu()
elif(Where=='options'):
show_options()
elif(Where=='newpass'):
show_newpass()
elif(Where=='delete'):
show_delete()
elif(Where=='alpha'):
show_alphabet()
```

- Ruft up() und down() Funktion auf, wenn Pfeiltasten gedrückt werden.
- ‚Move‘ erlaubt Bewegung des Cursors.
- ‚Selected‘ bestimmt Position des Cursors.
- ‚Where‘ Variable bestimmt aufgerufenes Menü nach Änderung.
 - Wichtig um zum richtigen Menü zurückzukehren.

Neues Passwort Erstellen

```
New <-
```

```
New password Title: Amazon
```

- ‚New‘ im Menü wählen, welches eine neue Funktion ausführt.

- Titel eingeben
 - Wird zum wiederfinden verwendet
 - Kann nur einmal existieren

```
Choose an option via arrow keys:  
Generate(Recomended) <-  
Custom  
<-Back  
  
Press Enter to continue...
```

- Entweder Passwort generieren lassen, selber eingeben oder abbrechen
 - Abbrechen führt zurück ins Menü
 - Generieren lassen wird empfohlen

```
Username: Billy34
Input Password length
20+ is recommended!
Password length (in numbers):34
```

- Benutzername gefolgt von Passwortlänge eingeben

```
Choose an option via arrow keys:
Lowercase=True <-
Uppercase=True
Digits=True
Punctuation=True
Confirm?

Press Enter to change selected option...
Press Enter on 'Confirm?' to confirm...
```

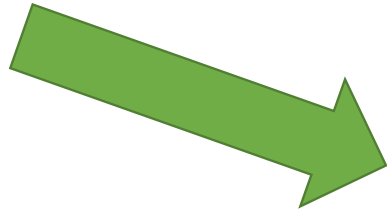
- Bestandteile des Passwortes an oder ausschalten
- Auswahl bestätigen

```
def getalphabet():
    clear()
    global lcase,ucase,dcase,pcase
    alphabet=''
```

```
show_alphabet()
input()
```

```
if(selected==1): #lower
if(lcase==True):
lcase=False
elif(lcase==False):
lcase=True
getalphabet()
```

```
if(selected==2):
if(ucase==True):
ucase=False
elif(ucase==False):
ucase=True
getalphabet()
if(selected==3):
if(dcase==True):
dcase=False
elif(dcase==False):
dcase=True
getalphabet()
if(selected==4):
if(pcase==True):
pcase=False
elif(pcase==False):
pcase=True
getalphabet()
```



```
Lowercase=True <-
```

```
if(selected==1): #lower
if(lcase==True):
lcase=False
elif(lcase==False):
lcase=True
getalphabet()
```

```
Lowercase=False <-
```

- Lowercase wird verändert
- Boolean wird verändert
- Funktion wird erneut aufgerufen
- Menü wird neu gedruckt
- Menü zeigt Veränderung


```

if(selected==5):
if lcase+ucase+dcase+pcase == False: #Checks if
at least one option is selected
print('At lest one option needs to be active!')
input()
getalphabet()

else:
if lcase == True: #Lowercase
alphabet=alphabet+string.ascii_lowercase

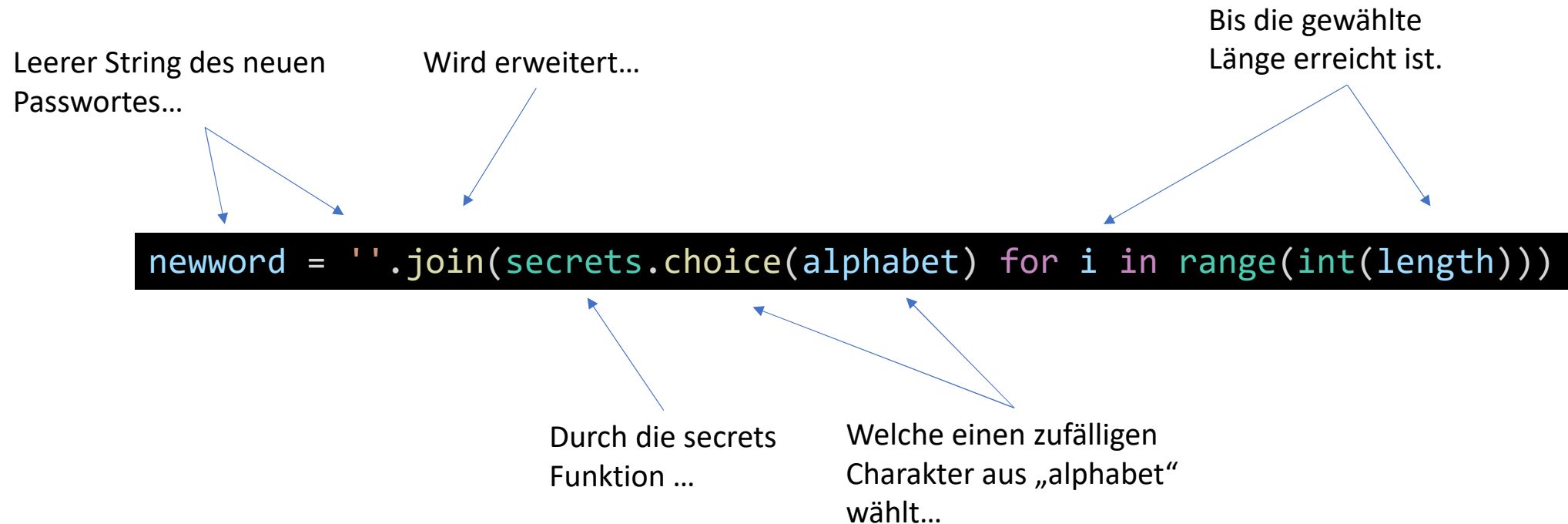
if ucase == True:
alphabet=alphabet+string.ascii_uppercase
if dcase == True:
alphabet=alphabet+string.digits
if pcase == True:
alphabet=alphabet+string.punctuation

return alphabet

```

- Nach Bestätigung, also Option 5:
 - Wenn alle Optionen ausgeschaltet sind wird das Menü wieder aufgerufen.
 - Falls mindestens eine Option aktiv ist, wird ein neuer String „alphabet“ mit den gewählten Charakteren gefüllt.
 - In diesem Beispiel werden dem leeren String *ABCDEFGHIJKLMNOPQRSTUVWXYZ & 0123456789 & !"#\$%&'()*+,-./:;<=>?@[\\]^_`{|}~.* angehängt.
- Danach wird „alphabet“ an die Funktion zurückgegeben.

Generierungsfunktion



ABCDEFGHIJKLMNOPQRSTUVWXYZ & 0123456789 & !"#%&'()*+,-./:;<=>?@[\\]^_`{|}~.



```
newword = ''.join(secrets.choice(alphabet) for i in range(int(length)))
```



```
newword = LBNL-~#\5}W'B]@{1A:Y4:30(7Q)%^|ZET
```

Speicher Funktion

Amazon

LBNL~#\5}W'B]@{1A:Y4:30(7Q)%^|ZET

Billy34

```
def save(title,passw,uname): # puts provided
strings into the file

saveline='->'+title+'=>'+passw+'_>'+uname
f=open('verysecure.txt','a+')
f.write(saveline+"\n")
f.Close

return True
```

- Daten werden an Speicherfunktion übergeben
- Sie werden zu einem String mit Trennzeichen zusammengesetzt
- Datei wird geöffnet und String wird angehängt

Zeile in Datei:

```
->Amazon=>LBNL~#\5}W'B]@{1A:Y4:30(7Q)%^|ZET_>Billy34
```

Password saved successfully and copied into clipboard
for 30 seconds, Press any key to return.

```
...  
pyperclip.copy(newword)  
inclip=True  
cliptimer()  
newword=''  
...
```

- Neues Passwort wird in die Zwischenablage kopiert.
- Variable zeigt an, dass sich ein Passwort dort befindet.
- Timer wird aufgerufen.
- Passwort wird aus der aktiven Variablen gelöscht.

Timer Funktionen

```
def cliptimer():  
    global inclip  
    if(inclip==True):  
        threading.Timer(30.0,cliptimer2).start()  
  
def cliptimer2():  
    pyperclip.copy('')  
    global inclip  
    inclip=False
```

- Neuer Timer der Threading Bibliothek läuft für 30 Sekunden und ruft danach ,cliptimer2' auf.
- Diese Funktion löscht den Inhalt der Zwischenablage mithilfe der Pyperclip Bibliothek und setzt Variable zurück.

Verschlüsselung

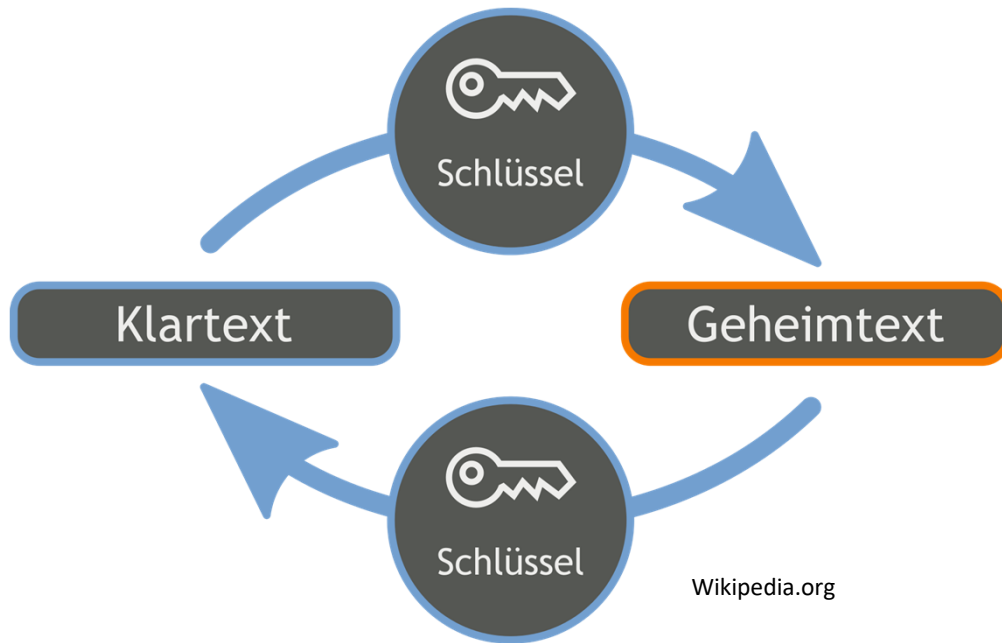
```
Masterpass=root
```

```
->facebook=>passwort123_>test@gmail.com  
->amazon=>chicken123_>test@gmail.com  
->Netflix=>dabbies_>test@gmail.com
```



```
gAAAAABg12FDSK4q9KdZXWBvIeGXQxfaY  
LbkAvmEGjx-jGbt25XDyBU8vTWJeylrSb  
HOIQsc7FMM5amFaIJZwNqRO2jUMXMfCUU  
-Zet0PdyMo7CoMXYWHxau_oiv2_EurA_I  
u0bgbM0BVTE5k-SSZ1G5GIDd3_NTJsxTQ  
ga8L2CarRdCHYzMDGH3bI3LKUGTiYNtfe  
_3I_d05ICiDDy6Ug0PEzCSv0I904Ljam9  
55_nBVuuIt7CrmRr9IYQN1DfJ4ZybrZn7  
Xr6qYX_oUWouM4ZQqVhOfImO9Q==
```

- Einfacher Beispieltext in Speicherdatei.
- Inhalt der Speicherdatei wird bei Beendigung des Programmes verschlüsselt.



- Verschlüsselung basiert auf Schlüssel Datei: *key.key*
- Dies wird *Symmetrisches Kryptosystem* genannt.
- Datei wird bei Programmstart entschlüsselt und bei Beendigung neu verschlüsselt.

Die aktuelle key.key Datei enthält:

Y9w_2OY7SYuMO0OQZsovUckEaq4JOO29kKhFuTh5Sk=


Passwort Löschen

```
Choose an option via arrow keys:  
Delete Password <-  
Change Master-Password  
Change Password Interval  
Request Plaintext Table  
<-Back  
  
Press Enter to continue...
```

- Optionen im Hauptmenü wählen
- Löschfunktion im Optionsmenü wählen
- Nach zu löschendem Passwort suchen

```
Delete password for which service? :
```

```
Delete password for which service? :Amaz
```



```
You are about to delete:"AMAZON"  
Choose an option via arrow keys:  
Confirm <-  
<-Back  
Press Enter to continue...
```

```
Input Masterpassword to confirm:
```

```
Password delete successfully!  
Press Enter to continue...
```

- Programm zeigt gefundenen Titel an
- Das Löschen muss noch einmal bestätigt werden
- Mit „<-Back“ kann der Vorgang abgebrochen werden
- Masterpassword muss eingegeben werden
- Bei richtiger Eingabe wird das Passwort aus der Datei gelöscht

```
lookup1=input("Delete password for which service? :")

lookup2='->'+lookup1.lower()

with open('verysecure.txt','rt') as file:
    for line in file:
        if lookup2 in line.lower():

            nline=line.split("=>")

            if(lookup2 in nline[0].lower()):
                nline=nline[0].split("->")
```

```
You are about to delete:"AMAZON"
Choose an option via arrow keys:
Confirm <-
<-Back
```

```
Press Enter to continue...
```

Eingabe von „Amaz“ ist lookup1

lookup2 ist „->amaz“


Text Datei wird in file eingelesen

Jede Zeile wird auf „->amaz“ untersucht:

->amazon=>lbnl-~#\x05}w'b]@{1a:y4:30(7q)%^|zet_>billy34

Falls gefunden wird die Zeile an „=>“, also nach dem Titel, zerschnitten: nline = [->amazon],[lbnl-~#\]

Falls „->amaz“ sich in „->amazon“ befindet, wird nline an „->“ weiter zerschnitten. nline = [''],[amazon].



todel wird zu „AMAZON“ und delete wird zu der gesamten Zeile!
Die Datei wird geschlossen und die Schleife beendet.

```
with open('verysecure.txt', 'r') as file:
    data = file.read().replace(delete, ',')
    file.close()

file = open('verysecure.txt', 'w')
file.write(data)
file.close()
data= ''
print('Password delete successfully!')
print("\nPress Enter to continue...")
input()
```

- Datei wird in `file` geöffnet und in `data` eingelesen.
- Zu löschende Zeile wird durch nichts ersetzt.
- Dateiinhalt wird überschreiben und Datei geschlossen.
- Der `data` String wird durch nichts ersetzt.