

LAB 4. Fișiere, prelucrarea fișierelor.

Să cunoască:

- particularitățile tipului de date: fișiere

Să fie capabil:

- să creeze și să efectueze operații de bază cu fișierele (scriere, citire, extragere, modificare de informații).

Gestionarea fișierelor este o parte importantă a oricărei aplicații web.

Python are mai multe funcții pentru crearea, citirea, actualizarea și ștergerea fișierelor.

Manipularea fișierelor

Funcția cheie pentru lucrul cu fișiere în Python este `open()`funcția.

Funcția `open()`ia doi parametri; *numele fișierului și modul*.

Există patru metode (moduri) diferite pentru deschiderea unui fișier:

"r" - Citire - Valoare implicită. Deschide un fișier pentru citire, eroare dacă fișierul nu există

"a" - Adăugați - Deschide un fișier pentru atașare, creează fișierul dacă nu există

"w" - Scriere - Deschide un fișier pentru scriere, creează fișierul dacă nu există

"x" - Create - Creează fișierul specificat, returnează o eroare dacă fișierul există

În plus, puteți specifica dacă fișierul trebuie tratat ca mod binar sau text

"t" - Text - Valoare implicită. Modul text

"b"- Binar - Modul binar (de ex. imagini)

Sintaxă

Pentru a deschide un fișier pentru citire, este suficient să specificați numele fișierului:

```
f = open("demofile.txt")
```

Codul de mai sus este același cu:

```
f = open("demofile.txt", "rt")
```

Deoarece "r" pentru citire și "t" pentru text sunt valorile implicate, nu trebuie să le specificați.

Notă: Asigurați-vă că fișierul există, altfel veți primi o eroare.

Deschideți un fișier pe server

Să presupunem că avem următorul fișier, situat în același folder ca Python:

demofile.txt

```
Hello! Welcome to demofile.txt  
This file is for testing purposes.  
Good Luck!
```

Pentru a deschide fișierul, utilizați funcția încorporată `open()`.

Funcția `open()` returnează un obiect fișier, care are o `read()` metodă de citire a conținutului fișierului:

```
f = open("demofile.txt", "r")
print(f.read())
```

Dacă fișierul se află într-o locație diferită, va trebui să specificați calea fișierului, astfel:

```
f = open("D:\\myfiles\\welcome.txt", "r")
print(f.read())
```

Citiți numai părți ale fișierului

În mod implicit, `read()` metoda returnează întregul text, dar puteți specifica și câte caractere dorîți să returnați:

Exemplu

```
f = open("demofile.txt", "r")
print(f.read(5))
```

Citiți rânduri

Puteți returna o linie folosind `readline()` metoda:

```
f = open("demofile.txt", "r")
print(f.readline())
```

Sunând `readline()` de două ori, puteți citi primele două rânduri:

```
f = open("demofile.txt", "r")
print(f.readline())
print(f.readline())
```

Parcugând în buclă liniile fișierului, puteți citi întregul fișier, rând cu linie:

```
f = open("demofile.txt", "r")
for x in f:
    print(x)
```

Închideți fișierele

Este o practică bună să închideți întotdeauna fișierul când ați terminat cu el.

```
f = open("demofile.txt", "r")
print(f.readline())
f.close()
```

Scrieți într-un fișier existent

Pentru a scrie într-un fișier existent, trebuie să adăugați un parametru la `open()` funcție:

"`a`" - Adăugați - se va adăuga la sfârșitul fișierului

"`w`" - Scriere - va suprascrie orice conținut existent

Deschideți fișierul „demofile2.txt” și adăugați conținut la fișier:

```
f = open("demofile2.txt", "a")
f.write("Now the file has more content!")
f.close()
```

#open and read the file after the appending:

```
f = open("demofile2.txt", "r")
print(f.read())
```

Deschideți fișierul „demofile3.txt” și suprascrieți conținutul:

```
f = open("demofile3.txt", "w")
f.write("Woops! I have deleted the content!")
f.close()

#open and read the file after the overwriting:
f = open("demofile3.txt", "r")
print(f.read())
```

Notă: metoda „w” va suprascrie întregul fișier.

Creați un fișier nou

Pentru a crea un fișier nou în Python, utilizați `open()` metoda, cu unul dintre următorii parametri:

“x”- Create - va crea un fișier, returnează o eroare dacă fișierul există

“a”- Adăugați - va crea un fișier dacă fișierul specificat nu există

“w”- Write - va crea un fișier dacă fișierul specificat nu există

Creați un fișier numit „myfile.txt”:

```
f = open("myfile.txt", "x")
```

Rezultat: este creat un nou fișier gol!

```
f = open("myfile.txt", "w")
```

Stergeți un fișier

Pentru a șterge un fișier, trebuie să importați modulul OS și să rulați `os.remove()` funcția acestuia:

Eliminați fișierul „demofile.txt”:

```
import os  
os.remove("demofile.txt")
```

Verificați dacă fișierul există:

Pentru a evita apariția unei erori, este posibil să doriți să verificați dacă fișierul există înainte de a încerca să-l ștergeți:

```
import os  
if os.path.exists("demofile.txt"):  
    os.remove("demofile.txt")  
else:  
    print("The file does not exist")
```

Ștergeți dosarul

Pentru a șterge un întreg folder, utilizați `os.rmdir()` metoda:

Exemplu

Eliminați folderul „myfolder”:

```
import os  
os.rmdir("myfolder")
```

Notă: puteți elimina doar dosarele *goale*.

1. Verifică dacă fișierul "nume_fisier" există.`if os.path.exists(nume_fisier):`

Aici introdu specificația operației pe care trebuie să o realizezi dacă fișierul există.

Exemplu: Deschide fișierul și citește conținutul.

2. Verifică dimensiunea fișierului "nume_fisier".

`if os.path.getsize(nume_fisier) > 1024:` # Aici introdu specificația operației pe care trebuie să o realizezi dacă fișierul are o anumită dimensiune.

Exemplu: Copiază fișierul într-un alt fișier.

3. Verifică dacă fișierul "nume_fisier" conține un anumit şablon sau cuvânt.`if "şablon" in open(nume_fisier).read():`

Aici introdu specificația operației pe care trebuie să o realizezi dacă fișierul conține şablonul sau cuvântul. # Exemplu: Înlocuiește şablonul sau cuvântul cu alt text.

4. Verifică extensia fișierului "nume_fisier".

`if nume_fisier.endswith('.extensie'):` # Aici introdu specificația operației pe care trebuie să o realizezi dacă fișierul are o anumită extensie.

Exemplu: Redenumește fișierul.

5. Verifică dacă fișierul "nume_fisier" are o dată de modificare recentă.

`if os.path.getmtime(nume_fisier) > time.time() - 7*24*60*60:`

Aici introdu specificația operației pe care trebuie să o realizezi dacă fișierul a fost modificat recent. # Exemplu: Afisează data ultimei modificări.

6. Verifică dacă fișierul "nume_fisier" conține un număr specific de linii.

`if len(open(nume_fisier).readlines()) > 50:` # Aici introdu specificația operației pe care trebuie să o realizezi dacă fișierul conține un număr specific de linii.

Exemplu: Creează un rezumat al fișierului.

7. Verifică dacă fișierul "nume_fisier" conține un anumit caracter.if "caracter" in open(nume_fisier).read():

Aici introdu specificația operației pe care trebuie să o realizezi dacă fișierul conține caracterul. # Exemplu: Numără aparițiile caracterului.

8. Verifică dacă fișierul "nume_fisier" are o anumită permisiune de citire.

if os.access(nume_fisier, os.R_OK): # Aici introdu specificația operației pe care trebuie să o realizezi dacă fișierul are permisiunea de citire.

Exemplu: Afisează conținutul fișierului.

9. Verifică dacă fișierul "nume_fisier" este vid.if os.path.getsize(nume_fisier) == 0:

Aici introdu specificația operației pe care trebuie să o realizezi dacă fișierul este vid. # Exemplu: Șterge fișierul.

10. Verifică dacă fișierul "nume_fisier" conține numai caractere alfanumerice.

if all(c.isalnum() or c.isspace() for c in open(nume_fisier).read()): # Aici introdu specificația operației pe care trebuie să o realizezi dacă fișierul conține numai caractere alfanumerice.

Exemplu: Converteste toate literele mici în majuscule.

11. Verifică dacă fișierul "nume_fisier" conține un număr specific de cuvinte.if len(open(nume_fisier).read().split()) == 100:

Aici introdu specificația operației pe care trebuie să o realizezi dacă fișierul conține un număr specific de cuvinte. # Exemplu: Creează un rezumat al fișierului.

12. Verifică dacă fișierul "nume_fisier" conține un anumit număr de linii.

if len(open(nume_fisier).readlines()) < 20: # Aici introdu specificația operației pe care trebuie să o realizezi dacă fișierul conține un anumit număr de linii.

Exemplu: Adaugă linii până la numărul specificat.

13. Verifică dacă fișierul "nume_fisier" conține caractere specifice de punctuație.`if any(c for c in open(nume_fisier).read() if c in string.punctuation):`

Aici introdu specificația operației pe care trebuie să o realizezi dacă fișierul conține caractere specifice de punctuație. # Exemplu: Elimină toate caracterele de punctuație.

14. Verifică dacă fișierul "nume_fisier" conține un anumit număr de caractere numerice.

`if sum(1 for c in open(nume_fisier).read() if c.isdigit()) > 30:` # Aici introdu specificația operației pe care trebuie să o realizezi dacă fișierul conține un anumit număr de caractere numerice.

Exemplu: Extrage toate numerele din fișier.

15. Verifică dacă fișierul "nume_fisier" are extensia ".

`txt" și conține cel puțin o linie.if nume_fisier.endswith('.txt') and len(open(nume_fisier).readlines()) > 0:`

Aici introdu specificația operației pe care trebuie să o realizezi dacă fișierul are extensia ".txt" și conține cel puțin o linie. # Exemplu: Salvează linia cu cel mai multe caractere în alt fișier.