Os მოდული

Os მოდულს ჭირდება იმპორტი

დირექტორიის შექმნა

Os.mkdir() - შექმნის დირექტორიას

import os  
os.mkdir('test')

შექმნის დირექტორიას იმ ფოლდერში სადაც შესულივართ

ხოლო ხელმეორედ გამოძახებით გავა ერორზე რადგან უკვე არსებობს ასეთი დირექტორია

ერორზე რომ არ გავიდეს შევამოწმებთ თუ არ არსებობს დირექტორია მაშინ შექმნის და თუ არსებობს მაშინ დაპრინტავს რომ არსებობს

import os  
if not os.path.exists("test"):  
 os.mkdir("test")  
else:  
 print("dir is exists")

ფოლდერების და დირექტორიების გაგება

Os.listdir() - დააბრუნებს ყველაფერს რაც დირექტორიაში იქნება

import os  
print(os.listdir())

path\_ის გაგება

os.getcwd() - სადაც ვიმყოფებით იმის აბსოლუტურ მისამართს გამოიტანს

import os  
print(os.getcwd())

დირექტორიის წაშლა

Os.rmdir () - წაშლის დირექტორიას თუ არსებობს და თუ არა მაშინ დაპრინტავს რომ ერორზე არ გავიდეს

import os  
if os.path.exists("test"):  
 os.rmdir("test")  
else:  
 print("No files found")

ჯეერ ვიგებთ არის თუ არა და თუ არის ვშლით

თუმცა თუ დირექტორიაში რაიმე მონაცემია მაშინ ერორზე გავა და ვე წაშლის

ამისთვის უნდა დავაიმპორტოთ shutil მეთოდი და rmtreee() წაშლის დირექტორიას

import os  
import shutil  
if os.path.exists("direct"):  
 shutil.rmtree("direct")  
else:  
 print("file not found")

თუ დირექტორიაში ვართ შესული მას ვერ წავშლით რადგან იქიდან გამოსვლაა საჭირო

უკან დაბრუნებით მივწვდებით დირექტორიას

import os  
import shutil  
if os.path.exists("../direct"):  
 shutil.rmtree("../direct")  
else:  
 print("file not found")

ფაილის წაშლა

თუ გაფართოება არ იქნება მითითებული პითონი აღიქვმს როგორც დირექტორია ხოლო თუ გაფართოებას მივუთითებთ მაშინ როგორც ფაილს ისე აღიქვმს

Os.remove () - წაშლის ფაილს

import os  
if os.path.exists("person.csv"):  
 os.remove("person.csv")  
else:  
 print("person.csv doesn't exist")

ბინარული სისტემა

Encode () - გადაიყვანს ბინარულში ცვლადს

name = "vano"  
binar = name.encode()

ფაილში ბინარული ტექსტის შეტანა ხდება wb\_რეჟიმით

name = "vano"  
binar = name.encode()  
with open("persons.txt", "wb") as f:  
 f.write(binar)

ბინარული ფაილიდან წაკითხვა

Rb - რეჟიმით უნდა გაეშვას რომ წაკითხვადი იყოს ბინარულზე

with open("binar.bin", "rb") as f:  
 dara = f.read()  
 print(dara)

**CSV**

Csv – comma separated values ( მძიმით გამოყოფილი ველიუები )

Csv ფაილი წარმოსდგება როგორც ექსელის ფაილში ჩაწერილი ინფორმაცია და მას აქვს ჰედერი რომლის მიხედვითაც იქმნება ველიუები

Csv ფაილში მონაცემი ინახება ცხრილური სახით და ერთმანეთისგან მძიმით გამოიყოფა

მასში ინახება მარტივი სტუქტურიზებული მონაცემები

გამოიყენება მონაცემთა მიმოცვლისთვის

ბევრი აპლიკაცია იყენებს მონაცემების იმპორტის და ექსპორტისთვის

Csv ფაილებთან მუშაობა

იმისთვის რომ csv ფაილებს ვემუშაოთ მისი იმპორტია საჭირო

import csv

წაკითხვა ფაილიდან

Csv.reader () - კითხულობს რიგებს ფაილიდან

import csv  
with open("new\_csv.csv") as csvfile:  
 reader = csv.reader(csvfile)  
 for row in reader:  
 print(row)

reader მოგვცემს მისამართს ხოლო ციკლით row სათითაოდ გამოვიტანთ მონაცემებს ლისტში ჩაყრილს სტრინგის ტიპით

['name', 'numeracia']

['vano', '1']

['gio', '2']

['dato', '3']

['nika', '4']

['aleko', '5']

Csv.DictReader() – პირველ ხაზს ფაილში გახდის ჰედერს დასათითაოდ მიანიჭებს შემდეგ ველიუებს , პირველ ხაზზე რომ გვქონდესდ დაწერილი name, numeracia დაწერდა ასე

with open("new\_csv.csv") as csvfile:  
 file = csv.DictReader(csvfile)  
 for row in file:  
 print(row)

{'name': 'vano', 'numeracia': '1'}

{'name': 'gio', 'numeracia': '2'}

{'name': 'dato', 'numeracia': '3'}

{'name': 'nika', 'numeracia': '4'}

{'name': 'aleko', 'numeracia': '5'}

ფაილში დაწერა

Cvs.writer() - შეიტანს მონაცემებს ფაილში ოღონდ ლისტში უნდა ჩავწეროთ სტრინგი

with open("new\_csv.csv", "w") as csvfile:  
 input = csv.writer(csvfile)  
 input.writerow(["vano", "khokhobashvili"])

writerow - შეიტანს ერთ ხაზს ანუ ერთი რიგს ჩაწერს

ანუ თუ გვინდა ჯერ ჰედერების ჩაწერა გამოვიყენებთ writerow\_ს ხოლო მონაცემების ჩასაწერად შემდეგ writerows\_ს

writerows() - ვწერთ თუ გვინდა ბევრ მონაცემიანი ლისტი შევიტანოთ

listt = [  
 ["vano", "1"],  
 ["gio", "2"],  
 ["dato", "3"],  
 ["nika", "4"],  
 ["aleko", "5"],  
]  
with open("new\_csv.csv", "w", newline="") as csvfile:  
 inputt = csv.writer(csvfile)  
 inputt.writerow(["name", "numeracia"])  
 inputt.writerows(listt)

ჯერ მიიღო ჰედერი ანუ name numeracia და შემდეგ შეიტანა ლისტის მონაცემები სათითაო სათითაო ხაზზე

Newline =”” რახამ გადავეცით ახალ ხაზს არ გამოიტანს და ერთმანეთის ქვემოთ დაწერს მონაცემებს

csv.Dictwriter() - დაწერს ფაილში ბევრ მონაცემს დიქტის სახით და გადაეცემა ფაილი და fieldnames = ანუ ჰედერები რა უნდა ქონდეს ქეი იგივე

diction = [  
 {"name": "vano", "age":50},  
 {"name": "vano", "age": 50},  
 {"name": "vano", "age": 50},  
 {"name": "vano", "age": 50}  
]  
with open("new\_csv.csv", "w", newline="") as csvfile:  
 writes= csv.DictWriter(csvfile, fieldnames=diction[0].keys())  
 writes.writeheader()  
 writes.writerows(diction)

ფაილს გავატანე ჰედერები რა უნდა ქონოდა რადგან დიქტის ინდექსზე name:vano age:50\_ია მისი ქიები არის name და age და მათ მიანიჭებდა

და დავბეჭდე ჰედერები writeheader() რადგან პირველი ჰედერები ყოფილიყო

ხოლო მეორედ writerows გადავცემთ თითო ხაზზე თითო დიქტის მონაცემს

თუ მეტი ჰედერ იქნება ვიდრე ველიუ, პრობლემა არაა მაგრამ თუ ნაკლები ჰედერი იქნება ვიდრე ველიუ მაშინ ერორზე გავა

Fake მოდული

ინსტალაცია ისეთი მოდულის რომლის დაიმპორტება დადაინსტალირებაა საჭირო ტერმინალში იწერება ჯერ

Pip install და მოდულის სახელი ამ შემთხვევაში faker

import faker  
fake = faker.Faker()  
print(fake.name())

faker\_ს აქვს წინასწარ შეყვანილი რენდომ მონაცემები რაც შეგვიძლია ადვილად გამოვიყენოთ რომ არ ვწეროთ ხელით