Files

ნებისმიერი სახის ინფორმაცია, რომელიც კომპიეტერშია შენახული წარმოდგენილია ფაილების სახით. თითოეულ მათგანს გააჩნია თავისი ადგილი სისტემაში.

სქემა რის მიხედვითაც შესაძლერბელია ფაილების შენახვა, მათო მოძებნა და გახსნა ეწოდები ფაილების სისტემა., რომელიც მოიცავს ორ ტიპს დირექტორიებს და ფაილებს.

პითონში შეგვიძლია ფაილის გახსნა , წაკითხვა, ჩაწერა, დახურვა და წაშლა,

ასევე დირექტორიის გახსნაც შეგვიძლია,

ფაილის გახსნისთვის -

Text\_file=open(“test.txt”, “r”) რ- ნიშნავს რიდ, w- დაწერა, x - ფაილის შექმნა

Txt- ნიშნავს გაფართოებას.

ფაილების მუშაობის მეთოდები

Read () ფაილის შიგთავსის წაკითხვა

Readline () ფაილში არსებული ერთი ხაზის წაკითხვა

Readlines() ფაილის ყველა ხაზის წაკითხვა , აბრუნებს ლისტს მნიშვნელობით

Write () ფაილის ჩაწერა

Close () ფაილის დახურვა

OS მოდული

Listdir() აბრუნებს დირექტორიაში მდებარე ელემენტებს

Remove() შლის ფაილს

Mkdir() ქმნის ახალ დირექტორიას

Test\_file=open (“text\_test.txt”, “x”) შექმნის მხოლოდ ფაილს თუ მივუთითებ სპეციფიკურ მისამართს მაშინ იქ შეიქმნება თუ არა და იმავე ფაილში, მიმდინარე დირექტორიაში

მისამართის დროს უნდა დაიწეროს რუთი სლეშებით

(“/home/irakli/text\_test.txt”, “x”)

test\_file=open (“text\_test.txt”, “r”) წაკითხვა

print (test\_file.read()) - ის წაიკითხავს მთლიან ფაილს და დაბეჭდავს

read(100) წაიკითხავს 100 ბაიტს, მითითების გარეშე სრულად კითხულობს

readline() - ეტაპობრივად თითო ხაზი იკითხება , მეორედ გამოძახზებაზე წაიკითხავს მეორე ხაზს.

თუ გვინდა კონკრეტული ადგილის წაკითხვა

Read\_file Test\_file.readlines()

For line in read\_file:

If “john” in line:

Print(line)

Test\_file.close() - რასაც ვუთითებთ გაღებას აუცილებლად ბოლოს უნდა დავხუროთ რომ ერორი არ იყოს

Writable\_file= open(“writable\_file”, “w”)

Writable\_file.write(“100\nhello from file”) წერის დროს უნდა იყოს სტრინგის ტიპის , ციფრიც ბრჭყალებში უნდა იყოს

Writable\_file.close()

Writable\_file= open(“text\_test.txt”, “w”) ჯერ შლის და მერე ამატებს რაც უნდა აისახოს

Writable)file.writelines(list) - აქ ჩაწერს ერთ ხაზად ქვემო ცვლადს ლისტს

List=[“line1”, “line2”, “line3”]

Writable\_file.writelines(list2) - აქ ჩაწერს როგორც სიას

List2=[“line1\n”, “line2\n”, “line3\n”]

Writable() and readable()- აბრუნებს თრუს და ფოლს.

პლიუს თუ დავუწერთ ნიშნავს რომ ორივე შეიძლება წაკითხვაც და წერაც / თუ გვჭირდება ორივე ერთდროულად ჯერ წავიკიnთხავთ და მერე ნებისმიერ რამე ჩავწეროთ

Append – a - რეჟიმი, არის ჩაწერადი , ისე წერს რომ ფაილში არსებულ მონაცემებს არ შლის და ახალს წერს

Writable\_file= open(“text\_test.txt”, “a”)

Writable \_file.write(‘\n New Line from Append Method”) - ამატებს ლაინებად

და არა ერთ ხაზად

კონტექსტ მენეჯერი- გამოიყენება ფაილებთან სამუშაოდ, რომელიც ავტომატურად ხურავს ფაილს მასთან მუშაობის დასრულების შემდეგ, ამისთვის გამოიყენება ქივორდი with, ის თვითონ ხურავს და ქლოუზი ცალკე არ სჭირდება, იგივე მეთოდები აქვს რაც ზემოთ

With open ( “test.txt”, “+r” ) as file:

Read\_file = file.readlines() გამოიტანს სიის სახით

with open (“writable\_file.txt”, “a”) as writable\_file:

for line in read\_file:

if “method” in line:

writable \_file.write(line+ “\n” )