



PRAKTIKUM IZ PROGRAMIRANJA 3

VEŽBE 4

Lazar Vasović, Luka Marković,
Andreja Živić, Jovan Janićijević



Biblioteka **shutil** se koristi za kopiranje, premeštanje, preimenovanje i brisanje fajlova iz Python programa.

Primer:

```
import shutil, os
```

```
os.chdir('C:\\')
```

#Linija ispod kopira fajl spam.txt u folder spamFolder uz zadržavanje naziva fajla

```
Shutil.copy('C:\\spam.txt', 'C:\\spamFolder')
```

#Linija ispod kopira fajl spam.txt u folder spamFoldern ali se kopirani fajl zove spam2.txt

```
Shutil.copy('C:\\spam.txt', 'C:\\spamFolder\\spam2.txt')
```



Kopiranje čitavog sadržaja direktorijuma

```
import shutil, os
```

```
os.chdir('C:\\')
```

```
#Linija ispod kreira folder bacon_backup i kopira ceo sadrzaj foldera  
bacon u novokreirani folder
```

```
shutil.copytree('C:\\bacon', 'C:\\bacon_backup')
```



Premštanje i promena imena fajla

```
import shutil, os
```

```
#Ukoliko folder C:\\eggs postoji, linija ispod ce premestiti fajl  
bacon.txt u njega, u suprotnom se samo menja naziv fajla iz bacon.txt  
u eggs
```

```
shutil.move('C:\\bacon.txt', 'C:\\eggs')
```

```
#Ukoliko folder C:\\eggs postoji, linija ispod premesta fajl  
C:\\bacon.txt u folder C:\\eggs i menja mu ime u new_bacon.txt
```

```
shutil.move('C:\\bacon.txt', 'C:\\eggs\\new_bacon.txt')
```



Trajno brisanje fajlova i foldera

```
import os, shutil
```

```
#Brisanje fajla sa putanjom path  
os.unlink(path)
```

```
#Brisanje PRAZNOG foldera na putanji path  
os.rmdir(path)
```

```
#Brisanje foldera na putanji path i njegovog sadrzaja  
shutil.rmtree(path)
```



Napisati program koji briše sve **.txt** fajlove iz trenutnog radnog direktorijuma.

Rešenje:

```
import os

for filename in os.listdir():
    if filename.endswith('.txt'):
        os.unlink(filename)
```

Napomena: Pošto se radi o trajnom brisanju, dobra praksa je da se pre brisanja samo odštampaju imena fajlova koja bi trebala da se obrišu



Napisati program koji privremeno briše (premešta u korpu) sve **.txt** fajlove iz trenutnog radnog direktorijuma.

Rešenje:

```
pip3 install send2trash
```

```
import os, shutil, send2trash
```

```
#Kreiranje txt fajla da bi imalo sta da se obrise
```

```
f = open('a.txt','w')
```

```
f.close()
```

```
for filename in os.listdir():
```

```
    if filename.endswith('.txt'):
```

```
        send2trash.send2trash(filename)
```



Napisati program koji prolazi kroz sve podfoldere (po svim dubinama) počevši od foldera **C:\\Users\\student** i ispisuje njihova imena, kao i imena fajlova koji se nalaze u njima.

Rešenje:

```
import os

for folderName, subfolders, filenames in
os.walk('C:\\Users\\student'):
    print('The current folder is ' + folderName)

    for subfolder in subfolders:
        print('SUBFOLDER OF ' + folderName + ': ' + subfolder)

    for filename in filenames:
        print('FILE INSIDE ' + folderName + ': ' + filename)
```



Napisati program koji čita sadržaj fajla **zipFile.zip** i ispisuje informaciju o veličini svakog fajla koji se nalazi u njemu.

Rešenje:

```
import zipfile, os

exampleZip = zipfile.ZipFile('zipFile.zip')
fileList = exampleZip.namelist()

for file in fileList:
    fileInfo = exampleZip.getinfo(file)
    print('File: ', file, ' file size: ', fileInfo.file_size, ' zip
size: ', fileInfo.compress_size)

exampleZip.close()
```



Napisati program koji extract-uje fajl **zipFajl.zip** u folder **extractFolder**. Ukoliko taj folder ne postoji, potrebno je kreirati ga.

Rešenje:

```
import zipfile, os
```

```
exampleZip = zipfile.ZipFile('zipFile.zip')  
exampleZip.extractall('extractFolder')
```

```
exampleZip.close()
```

#Ukoliko zelite da extract-ujete pojedinačne fajlove, to se radi na sledeći način

```
exampleZip.extract('spam.txt')
```

#Drugi argument je opcioni. Označava putanju do foldera gde se extract-uje

```
exampleZip.extract('spam.txt', 'C:\\some\\new\\folders')
```



Napisati program koji kreira zip fajl **new.zip** i u njega upisuje datoteku **spam.txt**.

Rešenje:

```
import zipfile

newZip = zipfile.ZipFile('new.zip', 'w')
newZip.write('spam.txt', compress_type=zipfile.ZIP_DEFLATED)

newZip.close()
```

Napomena: Mod 'w' će obrisati sav sadržaj fajla. Ukoliko je potrebno dopuniti već postojeći fajl novim vrednostima, koristi se mod 'a'.



Napisati program koji kopira sve fajlove iz direktorijuma **dir1** u direktorijum **dir2** čiji naziv ima sledeće osobine:

1. počinje velikim slovom
2. u nazivu ne pojavljuje '_'
3. ekstenzija ima 3 mala slova.

Nakon kopiranja, kreirati zip fajl koji sadrži sve fajlove iz direktorijuma **dir2**. Pošto su sada ti fajlovi sačuvani unutar zip fajla, originalni fajlovi mogu da se obrišu. Fajlove veće od 100KB obrisati trajno, a sve ostale fajlove premestiti u korpu za otpatke.



Rešenje:

```
import os, shutil, zipfile, re, send2trash
```

```
os.mkdir('dir2')
```

```
reg = re.compile(r'^[A-Z][^_]*\.[a-z]{3}$')
```

```
for folderName, subfolders, filenames in os.walk('dir1'):
```

```
    for filename in filenames:
```

```
        lista = reg.findall(filename)
```

```
        if len(lista)>0:
```

```
            filePath = os.path.join(folderName, filename)
```

```
            shutil.copy(filePath, 'dir2')
```



Nastavak:

```
zipFile = zipfile.ZipFile('newZip.zip', 'w')
for file in os.listdir('dir2'):
    zipFile.write(file)
zipFile.close()

for file in os.listdir('dir2'):
    fileName = os.path.join('dir2', file)
    if(os.path.getsize(fileName) > 100*1024):
        os.unlink(fileName)
    else:
        send2trash.send2trash(fileName)
```