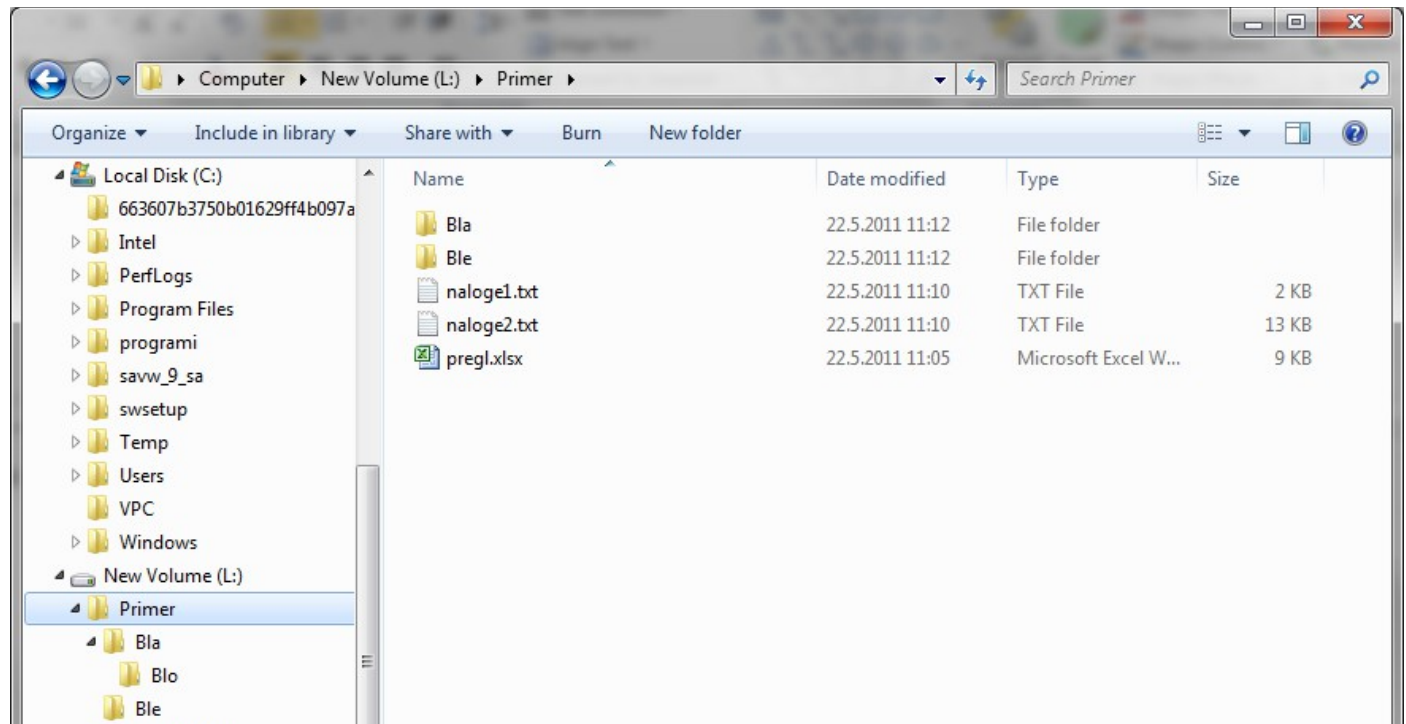


Datoteke, rekurzija,  
operacijski sistem

# Problem

- Ugotoviti moramo, koliko prostora v določenem imeniku zavzemajo datoteke



# Kaj potrebujemo

- Seznam datotek v določenem imeniku

```
>>> import os
>>> os.listdir('L:\\Primer')
['Bla', 'Ble', 'naloge1.txt', 'naloge2.txt', 'preg1.xlsx']
>>>
```

- Njihovo velikost

```
>>> os.path.getsize('L:\\Primer\\preg1.xlsx')
8829
>>>
```

# Opombi

- Velikost je v bytih (ne v kB, kot je izpis v WinExplorer)
- Pozor pri navajanju imen

```
>>> os.path.getsize('L:\Primer\naloge1.txt')
```

```
Traceback (most recent call last):
```

```
File "<pyshell#3>", line 1, in <module>
```

```
os.path.getsize('L:\Primer\naloge1.txt')
```

```
File "C:\programi\Python31\lib\genericpath.py", line 49, in getsize
```

```
return os.stat(filename).st_size
```

```
WindowsError: [Error 123] The filename, directory name, or volume  
label syntax is incorrect: 'L:\\Primer\naloge1.txt'
```

- Pravilno podano ime

```
>>> os.path.getsize('L:\\Primer\\naloge1.txt')
```

```
1204
```

# Ideja

- Ustvarimo seznam vseh datotek v imeniku
- Gremo preko vseh datotek v seznamu
- Seštevamo njihovo velikost

# 1. poskus

```
import os

def velikostDatotek(imenik):
    ''' skupna velikost vseh datotek v imeniku '''
    skupnaVelikost = 0
    seznamDatotek = os.listdir(imenik)
    for dat in seznamDatotek:
        velikost = os.path.getsize(dat)
        skupnaVelikost = skupnaVelikost + velikost
    return skupnaVelikost
```

# In rezultat

```
>>> velikostDatotek('L:\\Primer')
Traceback (most recent call last):
  File "<pyshell#14>", line 1, in <module>
    velikostDatotek('L:\\Primer')
  File "L:/velikostDatotek.py", line 6, in velikostDatotek
    velikost = os.path.getsize(dat)
  File "C:\programi\Python31\lib\genericpath.py", line 49, in getsize
    return os.stat(filename).st_size
WindowsError: [Error 2] The system cannot find the file specified: 'Bla'
>>>
```

- Zakaj

- Kaj vrne `listdir`?

```
>>> os.listdir('L:\\Primer')
['Bla', 'Ble', 'naloge1.txt', 'naloge2.txt', 'pregl.xlsx']
```

- In potem poskušamo

- `os.path.getsize('Bla')` namesto

- `os.path.getsize('L:\\Primer\\Bla')`

## 2. poskus

```
import os

def velikostDatotek1(imenik):
    ''' skupna velikost vseh datotek v imeniku '''
    skupnaVelikost = 0
    seznamDatotek = os.listdir(imenik)
    for dat in seznamDatotek:
        # pozor na pravo ime!
        velikost = os.path.getsize(imenik + '\\\\' + dat)
        skupnaVelikost = skupnaVelikost + velikost
    return skupnaVelikost
```



# In rezultat

- `>>> velikostDatotek1('L:\\Primer')`

- 23277

- Ker

- `>>> 23277 / 1024`

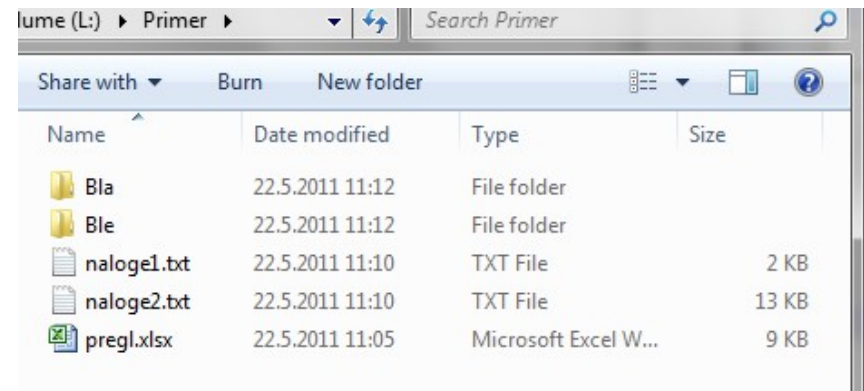
- 22.7314453125

- Smo očitno res sešteli le datoteke v tem imeniku (in ne v podimenikih)

- Samo v seznamu so tudi imeniki:

```
>>> os.listdir('L:\\Primer')
```

```
['Bla', 'Ble', 'naloge1.txt', 'naloge2.txt', 'pregl.xlsx']
```



# Velikost imenika

- kot datoteke je 0!

```
>>> os.listdir('c:\\Primer')
```

```
['Bla', 'Ble', 'datoteke4.pptx', 'naloge1.txt',  
'naloge2.txt', 'pregl.xlsx', 'ukaziDat.txt',  
'velikostDatotek.py', 'velikostDatotek1.py',  
'~$datoteke4.pptx']
```

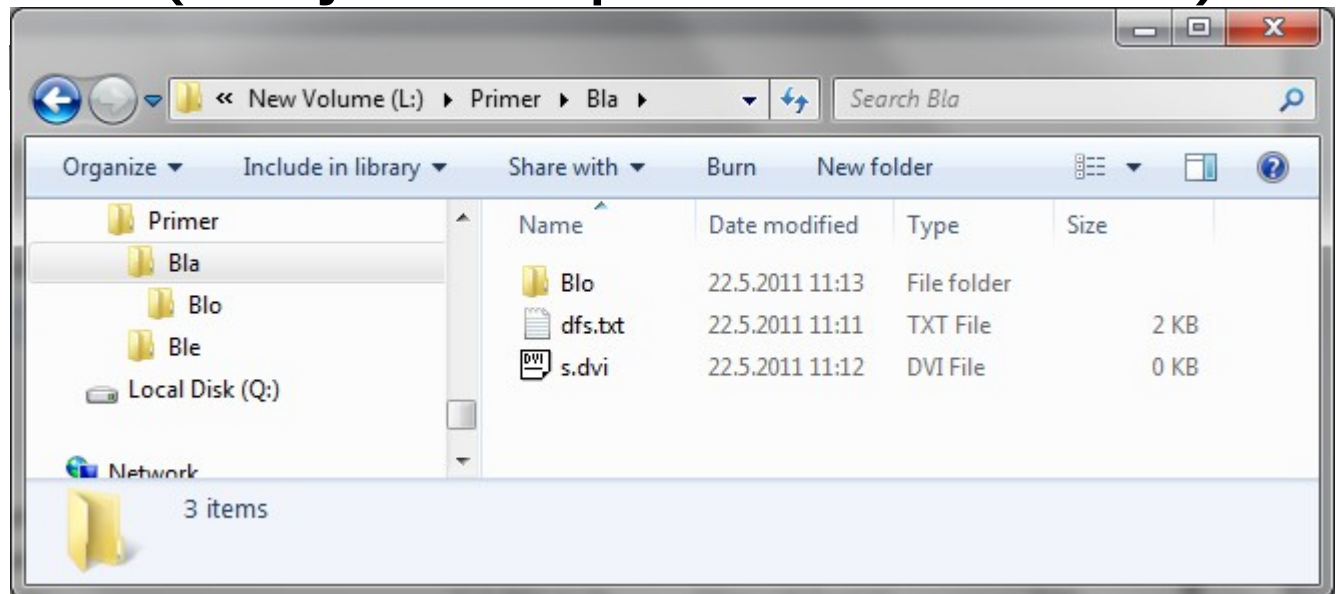
```
>>> os.path.getsize('c:\\Primer\\Bla')
```

```
0
```

- Torej smo sešteli le "vsebinske" datoteke na gornjem nivoju

# Kaj pa podimeniki?

- Sedaj: upoštevamo le datoteke neposredno v tem imeniku
- Nov problem: upoštevamo tudi datoteke v podimenikih (in njihovih podimeniki in ...) tega ime



# Ideja

- Spet najprej naredimo seznam datotek
- Če je to običajna datoteka
  - Metoda `os.path.isfile(ime datoteke)` → `True`
  - Kot do sedaj
- Če je imenik
  - Potrebujemo velikost vseh datotek v tem podimeniku
  - Rekurzivni klic!

# 3. poskus

```
import os

def velikostDatotek2(imenik):
    skupnaVelikost = 0
    seznamDatotek = os.listdir(imenik)
    for dat in seznamDatotek:
        polnoIme = imenik + '\\\ ' + dat
        if os.path.isfile(polnoIme) : # običajna datoteka
            velikost = os.path.getsize(polnoIme)
        else : # gre za imenik - rekurzija
            velikost = velikostDatotek2(polnoIme)
        skupnaVelikost = skupnaVelikost + velikost
    return skupnaVelikost
```

# Še različica z izpisovanjem po podimenikih

```
import os

def velikostDatotek3(imenik, izpis=False):
    skupnaVelikost = 0
    seznamDatotek = os.listdir(imenik)
    for dat in seznamDatotek:
        polnoIme = imenik + '\\' + dat
        if os.path.isfile(polnoIme) : # običajna datoteka
            velikost = os.path.getsize(polnoIme)
        else : # gre za imenik - rekurzija
            velikost = velikostDatotek3(polnoIme, izpis)
        skupnaVelikost = skupnaVelikost + velikost
    if izpis :
        print(imenik, ' : ', skupnaVelikost)
    return skupnaVelikost
```

# Določene težave

```
>>> velikostDatotek2('E:/')
Traceback (most recent call last):
  File "<pyshell#9>", line 1, in <module>
    velikostDatotek2('E:/')
  File "D:/Desktop/Datoteke_rek_P3.py", line 10, in velikostDatotek2
    velikost = velikostDatotek2(polnoIme)
  File "D:/Desktop/Datoteke_rek_P3.py", line 10, in velikostDatotek2
    velikost = velikostDatotek2(polnoIme)
  File "D:/Desktop/Datoteke_rek_P3.py", line 4, in velikostDatotek2
    seznamDatotek = os.listdir(imenik)
WindowsError: [Error 5] Access is denied: 'E://$RECYCLE.BIN\\S-1-5-21-382538178
7-1106546968-3042506532-1108\\*.*'
>>> |
```

Ln: 69 Col: 4

Sistemske datoteke, zaščitene datoteke ...

# In "lepotni dodatki"

- Kaj če niz ne predstavlja nekega imenika
  - Vrzimo izjemo
  - `os.path.isdir(pot)`: ali pot predstavlja obstoječ imenik
- Združiti obe različici v eno (parameter pove, ali gremo v globino ali ne – denimo, privzeto da)
- Parameter pove, ali računamo velikost v zlogih ali v kZlogih
  - Ni dovolj preračun le na koncu, ampak sproti!



# Različice

- Določi število datotek (in ne njihovo velikost) in pri tem ne šteje podimenikov
- Upoštevaj le datoteke določenega tipa (z določeno končnico - npr. jpg)
  - Naj bo možnih končnic več (seznam)
- Izpiši vse datoteke /vključno s podimeniki/(z zamikanjem)
- ...