



PRAKTIKUM IZ PROGRAMIRANJA 3

VEŽBE 3

Lazar Vasović, Luka Marković,
Andreja Živić, Jovan Janićijević



Napisati funkciju koja za prosleđeni string proverava da li on predstavlja broj telefona. String predstavlja broj telefona ukoliko ima 11 karaktera pri čemu važi da su prva 3 karaktera cifre, nakon kojih sledi „/“, zatim još 3 cifre, pa „-“ i na kraju još 3 cifre.

Primer:

Ulaz	Izlaz
034/322-737	True
034/322 737	False



```
def daLiJeBrojTelefona(br):  
    if len(br) != 11:  
        return False  
    for i in range(0, 3):  
        if not br[i].isdecimal():  
            return False  
    if br[3] != '/':  
        return False  
    for i in range(4, 7):  
        if not br[i].isdecimal():  
            return False  
    if br[7] != '-':  
        return False  
    for i in range(8, 11):  
        if not br[i].isdecimal():  
            return False  
    return True
```



```
import re
```

```
#Definisanje Regex promenljive
```

```
#Oznaka \d oznacava da je u pitanju cifra
```

```
brTelRegex = re.compile(r'\d\d\d/\d\d\d-\d\d\d')
```

```
#Trazenje paterna u stringu
```

```
brTel = brTelRegex.search("Ovo je moj broj: 034/111-222")
```

```
#Ispisivanje nadjenog broja (ukoliko postoji)
```

```
if(brTel):
```

```
    print(brTel.group(0))
```



Napisati regularni izraz koji za prosleđeni string proverava da li se u njemu nalazi podstring **“Batman”** ili **“Batmobile”**.

Rešenje:

```
import re

#Zagrade su specijalni karakteri
#Znak | (ili) oznacava da se u stringu moze pojaviti ili
njenov levi ili desni deo
betmenRegex = re.compile(r'Bet(men|mobil)')

bet = betmenRegex.search("Betmobil se pokvario.")

if(bet):
    print(bet.group(0))
```



Napisati regularni izraz koji za prosleđeni string proverava da li se u njemu nalazi podstring **“Batman”** ili **“Batwomen”**.

Rešenje:

```
import re
```

```
#Znak ? oznacava da se string u zagradama moze, ali i ne mora  
pojaviti
```

```
betRegex = re.compile(r'Bet(wo)?man')
```

```
bet = betRegex.search("The adventures of Betwoman.")
```

```
if(bet):
```

```
    print(bet.group(0))
```

Još neki primeri specijalnih karaktera u regularnim izrazima:

- Znak „*“ označava da se ono što se nalazi levo od tog znaka može pojaviti 0 ili više puta.
- Znak „+“ označava da se ono što se nalazi levo od tog znaka može pojaviti 1 ili više puta.

Primeri:

RegEx	String	Rezultat
Bat(wo)*man	The Adventures of Batman	Batman
Bat(wo)*man	The Adventures of Batwowowoman	Batwowowoman
Bat(wo)+man	The Adventures of Batwoman	Batwoman
Bat(wo)+man	The Adventures of Batwowoman	Batwowoman



Napisati regularni izraz koji za prosleđeni string proverava da li se u njemu pojavljuje podstring „Ha“:

1. Tačno 3 puta
2. Minimalno 4 puta
3. Maksimalno 7 puta
4. Minimalno 3, a maksimalno 5 puta

Rešenje:

```
import re
```

```
regex1 = re.compile(r'(Ha){3}')
```

```
regex2 = re.compile(r'(Ha){4,}')
```

```
regex3 = re.compile(r'(Ha){,7}')
```

```
regex4 = re.compile(r'(Ha){3,5}')
```




Napisati regularni izraz koji za prosleđeni string pronalazi sve brojeve telefona koji se nalaze u njemu.

Rešenje:

```
import re
```

```
regex = re.compile(r'\d{3}/\d{3}-\d{3}')
```

```
lista = regex.findall("Kuca: 034/123-456, Posao: 021/111-222")
```

```
print(lista)
```

Izlaz:

```
['034/123-456', '021/111-222']
```



Napisati regularni izraz koji za prosleđeni string proverava da li on predstavlja nečije ime i prezime. U stringu se ne sme naći ništa sem toga.

Rešenje:

```
import re
```

```
regex = re.compile(r'^[A-Z][a-z]{1,}\ [A-Z][a-z]{1,}$')  
re = regex.search('Aleksandar Vulin')
```

```
print(re.group(0))
```

Izlaz:

Aleksandar Vulin



Objašnjenje:

- Znak $\wedge$ označava da string mora početi znakom koji se nalazi posle njega, a znak $\$$ označava da se string mora završiti znakom koji se nalazi pre njega.
- Unutar zagrada $[]$ se mogu navoditi karakteri koji se poklapaju, npr. $[aeiou]$ će poklopiti sve samoglasnike. Ukoliko se dva karaktera navedu sa znakom – između, to znači da će se poklopiti svaki karakter koji ima ASCII vrednost veću ili jednaku od prvog, a manju ili jednaku od drugog navedenog karaktera. Npr. $[0-9]$ će poklopiti svaku pojavu cifre.



Napisati regularni izraz koji proverava da li je string šifra. String je šifra ukoliko počinje sa tačno 3 cifre, nakon kojih se nalaze minimalno 2, a maksimalno 4 bilo koja karaktera i završava se sa 2 velika slova.

Rešenje:

```
import re
#Znak . oznacava bilo koji karakter, osim karaktera za novi
red
regex = re.compile(r'^[0-9]{3}.{2,4}[A-Z]{2}$')
re = regex.search('123+-c7AB')

print(re.group(0))
```

Izlaz:

123+-c7AB

Uvod

- Svaki fajl ima dve glavne osobine, naziv i putanju.
- Na Windows-u se putanje pišu sa „\“, a na Mac-u i Linux-u sa „/“.
- Kod koji se piše mora da radi na svakom operativnom sistemu.
- Za kreiranje naziva fajla nezavisno od tipa OS-a, može se koristiti sledeće:

```
import os  
os.path.join('usr', 'bin', 'spam')
```

Izlaz:

'usr\\bin\\spam'

Trenutni radni direktorijum

- `os.getcwd()` – vraća trenutni radni direktorijum, odnosno direktorijum u kojem se nalazi fajl koji se pokreće
- `os.chdir('C:\\Windows\\System32')` – služi da promeni trenutni radni direktorijum

Apsolutne i relativne putanje

- Apsolutna putanja je ona putanja koja počinje od root direktorijuma (neka od particija diska)
- Relativna putanja je putanja relativna u odnosu na trenutni radni direktorijum

Više o putanjama i folderima

- Postoje i folderi koji se označavaju sa `(.)` i sa `(..)`. To nisu pravi folderi već specijalna imena koja se mogu koristiti u putanjama. `(.)` označava trenutni direktorijum, a `(..)` označava roditeljski direktorijum.
- Oznaka `.\` na početku relativne putanje je opcionalna. Npr. `.\spam.txt` i `spam.txt` se odnose na isti fajl.
- `os.makedirs('C:\\folder\\podfolder\\podpodfolder')` – ova funkcija će napraviti sve neophodne foldere kako bi se osigurala da navedena putanja postoji.
- `os.path.abspath('.')` – vraća apsolutnu putanju trenutnog radnog direktorijuma
- `os.path.relpath(path, start)` – vraća relativnu putanju od `start` do `path` putanje

Veličina fajla i sadržaj foldera

- `os.path.getsize(path)` – vraća veličinu fajla prosleđenog kao argument
- `os.listdir(path)` – vraća listu sa nazivima fajlova koji se nalaze u **path** direktorijumu
- `os.path.isfile(path)` – vraća True ukoliko je prosleđeni argument fajl, u suprotnom vraća False
- `os.path.isdir(path)` – vraća True ukoliko je prosleđeni argument direktorijum, u suprotnom vraća False



Napisati program koji računa ukupnu veličinu svih fajlova u trenutnom direktorijumu.

Rešenje:

```
import os

totalSize = 0
for filename in os.listdir('.'):
    totalSize = totalSize + os.path.getsize(os.path.join('.',
filename))

print(totalSize)
```

Čitanje i pisanje fajlova - koraci

1. Otvaranje fajla
2. Čitanje/Pisanje fajla
3. Zatvaranje fajla

Otvoravanje fajla se radi funkcijom *open(path, mode)*. **path** predstavlja putanju do fajla, a **mode** definiše u koju svrhu se fajl otvara. **mode** može imati vrednosti „r“ (čitanje), „w“ (pisanje), „a“ (dodavanje u fajl).

Čitanje fajlova

1. `read()` – ova funkcija čita ceo fajl kao string
2. `readlines()` – ova funkcija čita ceo fajl i od njega pravi listu stringova gde je svaki element liste jedan red u fajlu

Primer:

```
import os
file = open('fajl.txt', 'r')
sadrzaj = file.read()
print(sadrzaj)
file.close()
```

Pisanje u fajl

- Koristi se funkcija **write()**
- Ukoliko je fajl koji već postoji otvoren u modu za pisanje, sav sadržaj tog fajla biće obrisan. Ukoliko fajl nije postojao, biće kreiran.
- Ukoliko je fajl otvoren u modu za dodavanje, novi sadržaj se dodaje na kraj fajla prilikom upisa u njega

Primer:

```
import os  
file = open('fajl.txt', 'w')  
file.write('abcd') #sada fajl sadrzi samo "abcd"  
file.close()
```

Napisati program koji kreira novi direktorijum pod nazivom „kopije“ u tekućem direktorijumu. Nakon toga program treba da napravi kopije svakog fajla iz trenutnog direktorijuma i sačuva ih u novokreiranom folderu. Kopije pravi tako što čita sadržaj svakog fajla i to što pročita upiše u novi fajl pod istim imenom.

Rešenje:

```
import os
os.mkdir('kopije')
for filename in os.listdir('.'):
    if(os.path.isfile(filename)):
        print(filename)
        file1 = open(filename, 'r')
        sadrzaj = file1.read()
        file1.close()

        file2 = open(os.path.join('.\kopije', filename), 'w')
        file2.write(sadrzaj)
        file2.close()
```