

Python Development

Rad sa fajlovima



POWERED BY



COMTRADE

Recap

- Koja je uloga zip-a u petljama?
- Kako pravimo module?
- Kako importujemo funkcije iz modula?
- Koja je uloga `__name__` dundera?
- Otvaranje fajla vrši se uz pomoć koje funkcije?
- Da li se zatvaranje fajla vrši uz pomoć funkcije ili metoda i kako se ta/taj funkcija/metod zove?
- Zašto je potrebno zatvarati fajlove?
- Kojom funkcijom se dobija unicode vrednost nekog karaktera?
- Kojom funkcijom se od unicode vrednosti dobija sam karakter?
- Kojom funkcijom proveravamo da li fajl postoji na fajl sistemu?

Kontekst menadžer - With

- *Pythonic way* za otvaranje fajlova nije korišćenjem klasične *open()* funkcije i *close()* metoda već korišćenjem *with* kontekst menadžera.

-

```
with open('some_file', 'w') as opened_file:
```

```
    opened_file.write('Hola!')
```

- Fajl “some_file” je otvoren samo gde doseže with, čim se with blok završi, fajl se automatski sam zatvara
- Na ovaj način ne postoji opasnost da se fajl ne zatvori

Standardni izlaz za greške

- Do sada smo ispisivali poruke na standardni izlaz korišćenjem *print* funkcije ili u fajl korišćenjem `.write()` metoda.
- Nekada je potrebno ispisivati stvari na standardni izlaz za greške kako ne bi mešali obične poruke (logove) i same greške ili upozorenja

```
print("fatal error", file=sys.stderr)
```

- O standardnom izlazu za greške možemo razmišljati kao o “posebnom” fajlu
- <https://www.jstorimer.com/blogs/workingwithcode/7766119-when-to-use-stderr-instead-of-stdout>

Čitanje manjih fajlova

- Ukoliko čitamo manje fajlove, okej je koristiti konstrukt `f.read()`
- Tako npr. ako želimo pročitati zasebno sve reči fajla možemo koristiti:

```
reci = [rec for rec in f.read().split()]
```

- Problem: Da li će `split` deliti po tabu, spejsu, znaku za nov red?
- <https://docs.python.org/2/library/stdtypes.html#str.split>

Čitanje fajlova uz pomoć linija

- Ukoliko želimo da pročitamo jednu liniju fajla, možemo koristiti `f.readline()`
- Kao i u slučaju `f.read()`, ukoliko sledeća linija ne postoji, vraća se `None`

Čitanje fajlova uz pomoć linija

- Ukoliko želimo da iteriramo kroz linije to možemo uraditi uz pomoć:

```
for line in f.readlines():  
    print(line)
```

Gde smo stali sa čitanjem fajla?

- Nekad se javlja potreba da znamo do koje pozicije u samom fajlu smo stigli sa čitanjem, ta pozicija se zove *offset*
- Za dobijanje trenutnog offseta u fajlu koristi se metod `f.tell()`
- Za skok na neki specifičan offset koristi se metod `f.seek(offset, from_what)` gde argument `from_what` koji ima vrednosti: 0 (od početka fajla), 1 (počevši od trenutne pozicije)

Zadaci za vežbanje (11):

Napisati program koji učitava ime datoteke kao argument komandne linije, a potom ispisuje broj reči u datoteci. U slučaju greške na standardni izlaz za greške ispisati odgovarajuću poruku.

Zadaci za vežbanje (12):

Napisati program koji iz datoteke razno.txt u datoteku palindromi.txt prepisuje sve palindrome. Reč je palindrom ako se isto čita sa leve i sa desne strane bez obzira na veličinu slova. U slučaju greške, na standardni izlaz za greške ispisati odgovarajuću poruku.

```
RAZNO.TXT  
ovde nema palindroma
```

```
PALINDROMI.TXT:
```

```
RAZNO.TXT  
Ana i melem su  
primeri palindroma.
```

```
PALINDROMI.TXT:  
Ana i melem
```

Zadaci za vežbanje (13):

U datoteci čije se ime zadaje kao prvi argument komandne linije nalazi se ceo pozitivan broj n , a zatim i n celih brojeva. Napisati program koji na standardni izlaz ispisuje koliko k -tocifrenih brojeva postoji u datoteci, pri čemu se pozitivan ceo broj k zadaje kao drugi argument komandne linije. U slučaju greške, na standardni izlaz za greške ispisati odgovarajuću poruku.

```
ULAZ.TXT
```

```
6
```

```
15 193 -27 9790 35 1
```

```
IZLAZ:
```

```
Broj 2-cifrenih brojeva: 3
```

```
ULAZ.TXT
```

```
4
```

```
15 193 -27 9790
```

```
IZLAZ:
```

```
Broj 5-cifrenih brojeva: 0
```

Zadaci za vežbanje (14):

Napisati program koji na standardni izlaz ispisuje maksimum brojeva iz datoteke *brojevi.txt*. U slučaju greške, na standardni izlaz za greške ispisati odgovarajuću poruku.

```
BROJEVI.TXT
2.36 -16.11 5.96 8.88
-265.31 54.96 38.4

IZLAZ:
Najveci broj je: 54.96
```

```
BROJEVI.TXT
10.5 183.111 -90.2 3.167

IZLAZ:
Najveci broj je: 183.111
```

```
BROJEVI.TXT
-62.7 -190.2 -2.3 -1000
-198.25 -8

IZLAZ:
Najveci broj je: -2.3
```

Zadaci za vežbanje (15):

Kao argumenti komandne linije zadaju se ime datoteke i ceo broj k. Napisati program koji na standardni izlaz ispisuje sve linije zadate datoteke čija je dužina veća od k. U slučaju greške, na standardni izlaz za greške ispisati odgovarajuću poruku.

```
TEST.TXT
Teme koje su obradjivane:
Petlje
Funkcije
Nizovi
Strukture

IZLAZ:
Teme koje su obradjivane:
Funkcije
Strukture
```

Zadaci za vežbanje (16):

Napisati program koji u datoteci čije se ime navodi kao argument komandne linije određuje liniju maksimalne dužine i ispisuje je na standardni izlaz. Ukoliko ima više takvih linija, ispisati onu koja je leksikografski prva. U slučaju greške, na standardni izlaz za greške ispisati odgovarajuću poruku.

```
TEST.TXT
```

```
Danas je veoma hladno decembarsko  
popodne. Ne pada sneg, kazu mozda  
ce sutra.
```

```
IZLAZ:
```

```
Danas je veoma hladno decembarsko
```

```
IN.TXT NE POSTOJI
```

```
IZLAZ ZA GREŠKE:
```

```
Greska: neuspesno otvaranje  
ulazne datoteke.
```

Zadaci za vežbanje (17):

U datoteci čije se ime navodi kao prvi argument komandne linije navedena je reč *r* i niz linija. Napisati program koji u datoteku čije se ime navodi kao drugi argument komandne linije upisuje sve linije prve datoteke u kojima se reč *r* pojavljuje bar *n* puta gde je *n* pozitivan ceo broj koji se unosi sa standardnog ulaza. Prilikom prebrojavanja, računaju se i samostalna pojavljivanja reči *r* i pojavljivanja u okviru neke druge reči. Ispis treba da bude u formatu broj_pojavljivanja: linija. U slučaju greške na standardni izlaz za greške ispisati odgovarajuću poruku.

```
INTERAKCIJA SA PROGRAMOM:
```

```
Unesite broj n: 2
```

```
INPUT.TXT
```

```
sto  
stolica lampa  
postotak Stopiranje stopa  
presto Ostoja stotina prostorija
```

```
OUTPUT.TXT
```

```
2: postotak Stopiranje stopa  
4: presto Ostoja stotina prostorija
```

```
INTERAKCIJA SA PROGRAMOM:
```

```
Unesite broj n: 3
```

```
INPUT.TXT
```

```
red  
redar za ovu nedelju  
redosled ured  
odrediti raspored
```

```
OUTPUT.TXT
```

```
IN.TXT NE POSTOJI
```

```
IZLAZ ZA GREŠKE:
```

```
Greska: neuspesno otvaranje  
ulazne datoteke.
```

Zadaci za vežbanje (18):

Napisati program koji prebrojava koliko se puta linija datoteke ulaz.txt završava niskom s koja se učitava sa standardnog ulaza. U slučaju greške na standardni izlaz za greške ispisati odgovarajuću poruku.

```
ULAZ.TXT
```

```
/var/log/apache2/error.log  
/var/log/dpkg.log  
moj_log.log  
/home/korisnik.login  
/home/korisnik.log.txt
```

```
INTERAKCIJA SA PROGRAMOM:
```

```
Unesite nisku s: .log  
Broj linija: 3
```

```
ULAZ.TXT
```

```
/home/korisnik/imena.txt  
/home/korisnik/a.out  
/home/cv.pdf  
/home/korisnik/ulaz.txt  
/home/rezultati.xlsx  
/var/log/apache2/error.log
```

```
INTERAKCIJA SA PROGRAMOM:
```

```
Unesite nisku s: .txt  
Broj linija: 2
```


Zadaci za vežbanje (19):

Napisati program koji poredi dve datoteke i ispisuje redni broj linija u kojima se datoteke razlikuju. Imena datoteka se zadaju kao argumenti komandne linije. U slučaju greške, na standardni izlaz za greške ispisati odgovarajuću poruku.

```
ULAZ.TXT
  danas vezbamo
  programiranje
  ovo je primer kad su
  datoteke iste
```

```
IZLAZ.TXT:
  danas vezbamo
  programiranje
  ovo je primer kad su
  datoteke iste
```

```
IZLAZ:
```

```
U1.DAT
  danas vezbamo
  analizu
  ovo je primer kad
  su datoteke razlicite
```

```
U2.DAT
  danas vezbamo
  programiranje
  ovo je primer kad su
  datoteke razlicite
```

```
IZLAZ:
  2 3 4
```

```
PRVA.DAT
  ovo je primer
  kada su
  datoteke
  razlicite duzine
```

```
DRUGA.DAT
  ovo je primer kada
  su
  datoteke
  razlicite
  duzine
  i kada treba ispisati broj
  tih redova
```

```
IZLAZ:
  1 2 4 5 6 7
```

* Zadaci za vežbanje (20):

Napisati program koji kao argument komandne linije dobija ime datoteke koja sadrži samo brojeve, pročitati datoteku i sve brojeve učitati u listu. Napisati funkciju koja izbacuje duplikate iz učitane liste, upisati novu listu (bez duplikata) u datoteku koja se zadaje kao drugi argument komandne linije.