

Python Development

Zip, Moduli, Fajlovi



POWERED BY



COMTRADE

Recap

- Slice objekti
 - Dodela uz pomoć slajsovanja
 - Višedimenzioni nizovi
 - Inicijalizacija ugnježdenih listi
 - Argumenti komandne linije
-
- Karakter u njegovu unicode vrednost? `ord()` funkcija, inverz ove funkcije je `chr()` kojom se od unicode vrednosti dobije karakter

FizzBuzz

- <http://wiki.c2.com/?FizzBuzzTest>
- “The "Fizz-Buzz test" is an interview question designed to help filter out the 99.5% of programming job candidates who can't seem to program their way out of a wet paper bag.”
- <https://www.tomdalling.com/blog/software-design/fizzbuzz-in-too-much-detail/>



POWERED BY



Argumenti funkcije - Česta greška

```
>>> def funkcija(lista=[]):
```

```
...     lista.append(100)
```

```
...     return lista
```

```
...
```

```
>>> fja()
```

```
[100]
```

```
>>> fja()
```

```
[100, 100]
```

```
>>> fja()
```

```
[100, 100, 100]
```

Zip iteracija

- Koristi se za dve ili više listi istih dužina
- Iteracija kroz više listi istovremeno
- Problem: Imamo dve liste brojevi_1 i brojevi_2 istih dužina, napraviti treću listu koja predstavlja njihov zbir

```
>>> brojevi_1 = [1,2,3]
>>> brojevi_2 = [4,5,6]
>>> zbir_brojeva = []
>>> for br1, br2 in zip(brojevi_1, brojevi_2):
...     zbir_brojeva.append(br1 + br2)
...
>>> zbir_brojeva
[5, 7, 9]
```

Moduli

- Modul (*module*) - Python fajl koji sadrži kod
- Ugrađeni moduli (*built-in modules*) - moduli koji se “dolaze sa” Python-om, što znači da ih možemo koristiti na svakom kompjuteru koji ima instaliran Python
- Obično pravimo razliku između modula i programa:
 - Program je Python fajl koji izvršavamo
 - Modul je Python fajl koji *import*-ujemo



POWERED BY



COMTRADE

Moduli

- Module moramo uvesti (import) da bismo mogli da ih koristimo
- **import** *matematika*
- Pozivamo f-ju *sabiranje* iz modula *matematika*:
matematika.sabiranje(broj1, broj2)



POWERED BY



COMTRADE

Moduli

- Module moramo uvesti (import) da bismo mogli da ih koristimo
- Možemo uvesti i samo delove modula koje želimo da koristimo
- **from *matematika* import *sabiranje***
- Pozivamo f-ju *sabiranje* iz modula *matematika*: *sabiranje(broj1, broj2)*



POWERED BY



COMTRADE

Datoteke (fajlovi):

- `f = open('workfile', 'r')`
- Prvi argument predstavlja fajl koji želimo da otvorimo, npr. *ulaz.txt*, drugi argument predstavlja mode otvaranja, odnosno označava šta želimo da radimo sa fajlom, modovi su:
 - `r` - otvara fajl za čitanje
 - `w` - otvara fajl za pisanje
 - `a` - otvara fajl za dopisivanje tj. dodavanje na kraj (append-ovanje)
 - `r+` - otvara fajl za čitanje i za pisanje
- Sadržaj fajla čitamo sa: `f.read(size)` – gde `size` argument predstavlja količinu pročitanih karaktera, 1 čita jedan karakter, 2 dva karaktera, ukoliko ne navedemo `size` kao argument - čita se ceo fajl što nije preporučljivo ukoliko se radi sa velikim fajlovima
- Kada smo završili rad sa fajlom, isti zatvaramo korišćenjem: `f.close()`

Datoteke (fajlovi):

- `f = open('workfile', 'w')`
- `f.write(word)` – Korišćenjem `.write()` metoda možemo upisujemo string u fajl otvoren za pisanje

Kako možemo proveriti da li fajl postoji?

- Za pomenutu proveru možemo koristiti `os` modul koji dolazi preinstaliran sa svakim Python-om, tačnije `os.path` podmodul, još tačnije `os.path.isfile()` funkciju.

Napomena

Sve zadatke od sada pa do kraja kursa radimo korišćenjem

```
if __name__ == "__main__"
```

konstrukta.

Zadaci za vežbanje (1):

Napisati program koji prepisuje sadržaj datoteke čije se ime prosleđuje kao argument komandne linije u datoteku čije se ime takođe prosleđuje kao argument komandne linije karakter po karakter. U slučaju greške, na standardni izlaz za greške ispisati odgovarajuću poruku.

```
ULAZ.TXT
```

```
Danas je 21. mart.  
To je prvi dan proleca.
```

```
IZLAZ.TXT
```

```
Danas je 21. mart.  
To je prvi dan proleca.
```

```
ULAZ.TXT
```

```
Ispit iz Programiranja 1 je  
zakazan za 10. jun.
```

```
IZLAZ.TXT
```

```
Ispit iz Programiranja 1 je  
zakazan za 10. jun.
```

```
ULAZ.TXT NE POSTOJI
```

```
IZLAZ ZA GREŠKE:
```

```
Greska: neuspesno otvaranje  
datoteke ulaz.txt
```

Zadaci za vežbanje (2):

Napisati program koji prepisuje svaki treći karakter fajla u drugi fajl, imena fajlova se učitavaju kao argumenti komandne linije. U slučaju greške, na standardni izlaz ispisati odgovarajuću poruku.

ULAZ.TXT

Ova datoteka
sadrzi tekst
u vise
linija.

IZLAZ.TXT

O te
diet sli.

ULAZ.TXT

U Beogradu ce biti
suncan i lep
dan.

IZLAZ.TXT

Ueruei
nn pa

ULAZ.TXT

1234567890

IZLAZ.TXT

1470

Zadaci za vežbanje (3):

Napisati program koji šifruje sadržaj fajla podaci.txt tako što svako slovo ciklično zamenjuje njegovim prethodnikom suprotne veličine i upisuje u fajl sifra.txt. Na primer, slovo b se zamenjuje slovom A, slovo B slovom a, slovo a slovom ` , slovo A slovom z itd. Ostali karakteri ostaju nepromenjeni. U slučaju greške na standardni izlaz za greške ispisati odgovarajuću poruku.

```
podaci.txt:
```

```
a=x+y
```

```
x=B+5
```

```
sifra.txt:
```

```
`<W*X
```

```
W<a*4
```

```
PODACI.TXT NE POSTOJI
```

```
IZLAZ ZA GREŠKE:
```

```
Greska: neuspesno otvaranje  
datoteke podaci.txt.
```

Zadaci za vežbanje (4):

Napisati program koji za dve datoteke čija se imena unose sa standardnog ulaza radi sledeće:

- Za svaku cifru u prvoj datoteci, u drugu upisuje 0
- Za svako slovo u prvoj datoteci, u drugu upisuje 1
- Za sve ostale karaktere u prvoj datoteci, u drugu upisuje 2

```
INTERAKCIJA SA PROGRAMOM:  
Unesite ime prve datoteke:  
kilometraze.txt  
Unesite ime druge datoteke:  
sifra.txt
```

```
KILOMETRAZE.TXT  
Beograd - Nis 230km  
Uzice - Cacak 56.3km  
Subotica - Ruma 139km
```

```
SIFRA.TXT  
111111122211120001121111122  
21111120020112111111122211  
11200011
```

```
INTERAKCIJA SA PROGRAMOM:  
Unesite ime prve datoteke:  
ulaz.txt  
Unesite ime druge datoteke:  
izlaz.txt
```

```
ULAZ.TXT  
18. februar 2019.
```

```
IZLAZ.TXT  
00221111111200002
```

```
INTERAKCIJA SA PROGRAMOM:  
Unesite ime prve datoteke:  
in.dat  
Unesite ime druge datoteke:  
out.dat
```

```
IN.DAT NE POSTOJI
```

```
IZLAZ ZA GREŠKE:  
Greska: neuspesno otvaranje  
prve datoteke.
```


Zadaci za vežbanje (5):

Sa standardnog ulaza učitavaju se imena dveju datoteka i jedan karakter koji označava opciju. Napisati program koji prepisuje sadržaj prve datoteke u drugu tako što u slučaju da je navedena opcija “U”, sva mala slova zamenjuje velikim, a u slučaju da je navedena opcija “L” sva velika slova zamenjuje malim.

U slučaju greške, na standardni izlaz za greške ispisati odgovarajuću poruku.

```
INTERAKCIJA SA PROGRAMOM:  
Unesite imena datoteka i opciju:  
ulaz.txt izlaz.txt u
```

```
ULAZ.TXT  
danas je lep dan  
i Ja zelim  
da postanem programer
```

```
IZLAZ.TXT  
DANAS JE LEP DAN  
I JA ZELIM  
DA POSTANEM PROGRAMER
```

```
INTERAKCIJA SA PROGRAMOM:  
Unesite imena datoteka i opciju:  
prva.dat druga.dat l
```

```
PRVA.DAT  
Cena soka je 30  
Cena vina je 150  
Cena limunade je 200  
Cena sendvica je 120
```

```
DRUGA.DAT  
cena soka je 30  
cena vina je 150  
cena limunade je 200  
cena sendvica je 120
```

Zadaci za vežbanje (6):

Napisati program koji prebrojava mala slova u datoteci "podaci.txt" i dobijeni rezultat ispisuje na standardni izlaz. U slučaju greške na standardni izlaz za greške ispisati odgovarajuću poruku.

```
PODACI.TXT
Matematicki fakultet
Studentski trg 16
Beograd

IZLAZ:
Broj malih slova je: 36
```

```
PODACI.TXT
PrograMiranje

IZLAZ:
Broj malih slova je: 11
```

```
PODACI.TXT
MATEMATIKA
12+34=46

IZLAZ:
Broj malih slova je: 0
```

Zadaci za vežbanje (7):

Napisati program koji u datoteci čije se ime unosi sa standardnog ulaza prebrojava koliko se puta pojavljuje svaka od cifara i na standardni izlaz ispisuje cifru sa najvećim brojem pojavljivanja. Ukoliko ima više takvih cifara ispisati sve. Ukoliko datoteka ne sadrži nijednu cifru, ispisati odgovarajuću poruku. U slučaju greške, na standardni izlaz ispisati odgovarajuću poruku o grešci.

INTERAKCIJA SA PROGRAMOM:

Unesite ime datoteke:

ulaz.txt

ULAZ.TXT

dan
as je lep dan
i ja zelim
da postanem programer

IZLAZ:

Datoteka ne sadrzi cifre.

INTERAKCIJA SA PROGRAMOM:

Unesite ime datoteke:

prva.dat

PRVA.DAT

Cena soka je 30
Cena vina je 150
Cena limunade je 200
Cena sendvica je 120

IZLAZ:

Najcesce cifre: 0

INTERAKCIJA SA PROGRAMOM:

Unesite ime datoteke:

primer.c

PRIMER.C

1 22 333.444

IZLAZ:

Najcesce cifre: 3 4

Zadaci za vežbanje (8):

Napisati program koji u datoteci čije je ime dato kao argument komandne linije proverava da li su zagrade pravilno uparene. U slučaju greške na standardni izlaz za greške ispisati odgovarajuću poruku.

```
ZAGRADE.TXT
```

```
ab( cd) ..  
((3+4)*5+1)*9
```

```
IZLAZ:
```

```
Zagrade jesu uparene.
```

```
PRIMER2.DAT
```

```
(7+8  
nisu(  
uparene
```

```
IZLAZ:
```

```
Zagrade nisu uparene.
```

```
PRIMER3.DAT
```

```
)) 7 + 6 ((
```

```
IZLAZ:
```

```
Zagrade nisu uparene.
```

Zadaci za vežbanje (9)

Napisati program koji prebrojava slova i cifre u datoteci:

- a. Napisati funkciju `ucitaj_karaktere(fajl)` koja vraća listu karaktera procitanih iz fajla. Dozvoljeni karakteri za učitavanje su mala i velika slova engleske abecede kao i cifre. Učitavanje se prekida kada se naiđe na znak za novi red ili nedozvoljeni karakter. Funkcija vraća broj elemenata niza uspešno učitanih karaktera.
- b. Napisati funkciju `prebroj(karakteri)` koja vraća broj slova, broj cifara.

Napisati program koji koristeći prethodne funkcije prebrojava cifre i slova u datoteci čije se ime zadaje kao argument komandne linije, a zatim ispisuje dobijene vrednosti na standardni izlaz.

Zadaci za vežbanje (10):

Napisati program koji sa standardnog ulaza učitava reč *s* i u datoteku *rotacije.txt* upisuje sve njene rotacije. Pretpostaviti da je maksimalna dužina reči 20 karaktera. U slučaju greške, na standardni izlaz za greške ispisati odgovarajuću poruku.

```
INTERAKCIJA SA PROGRAMOM:  
Unesite rec: abcde
```

```
ROTACIJE.TXT  
abcde  
bcdea  
cdeab  
deabc  
eabcd
```

```
INTERAKCIJA SA PROGRAMOM:  
Unesite rec: 1234
```

```
ROTACIJE.TXT  
1234  
2341  
3412  
4123
```

```
INTERAKCIJA SA PROGRAMOM:  
Unesite rec: a=3*x+5;
```

```
ROTACIJE.TXT  
a=3*x+5;  
=3*x+5;a  
3*x+5;a=  
*x+5;a=3  
x+5;a=3*  
+5;a=3*x  
5;a=3*x+  
;a=3*x+5
```