
Tema 11

RAD SA FAJLOVIMA

- Funkcije
 - open()
 - close()
 - read(), readline(), readlines()
 - write(), writeline()
 - seek()
- Obratiti pažnju na deo sa:
 - Prosleđivana informacija datim funkcijama mora biti str, kao i povratna vrednost
 - Tipovi otvaranja fajla
- Šifrovana i dešifrovana poruka

PRIMER 1 - OVO JE BITNO DA SE URADI

```
# čitane iz fajla
```

Fajl.txt

```
Danas programiram u Pythonu.  
Volim da programiram.  
Algoritmika je moja omiljena škola programiranja.
```

Primer1.py

```
# čitanje mod  
fajl = open("Fajl.txt", "r")  
tekst = fajl.read()  
fajl.close()  
print(tekst)
```

OBJASNJENJE

- isprobati opcije sa read(), readline() i readlines(), objasniti razlike (šta vraćaju kao rezultat, koji deo pročitaju, kog tipa je povratna vrednost), probati kroz ovaj primer sve te opcije
- Fajl sa strane možemo da tretiramo kao neku mini bazu podataka u našem slučaju (navesti primere baze podataka - šta se to čuva - u igricama)
- U igricama u ovom levelu (kviz i vešalo) txt fajlovi će nam biti kao neke mini baze podataka za čitanje svih pitanja za kviz, i za uzimanje random reči za pogađanje u igri vešalo

- Fajl txt i fajl py moraju biti u istom folderu, inace je neophodno da se navede putanja od py fajla do txt fajla
- Open može da ima još jedan bitan parametar, encoding, koji ćemo koristiti u igricama u ovom kursu. To je parametar koji ima podešenu defaultnu vrednost, pa ga zato ne navodimo uvek. On će nam ograničiti čitanje karaktera iz fajla koji otvarama (po tome ascii tabela karaktera je čitljiva)
- Objasniti ascii i UTF-8 istoriju i razlike ako učenik ne zna
- Podešavanjem encoding="UTF-8" kao treci parametar funkcije open(), možemo da čitamo kukice na slovima, tj naša specifična slova, kao i sva kineska.

PRIMER 2 - OVO JE BITNO DA SE URADI

```
# upis u fajl - nadovezivanje
```

Fajl.txt

Danas programiram u Pythonu.
Volim da programiram.
Algoritmika je moja omiljena škola programiranja.

Primer2.py

```
fajl = open("Fajl.txt", "a")
fajl.write("Kad porastem bicu programer!")
fajl.close()
fajl = open("Fajl.txt", "r")
tekst = fajl.read()
fajl.close()
print(tekst)
```

OBJASNJENJE

- "a" povezati sa append reči iz listi

PRIMER 3 - OVO JE BITNO DA SE URADI

```
# upis u fajl uz brisanje prethodnog sadržaja
```

Fajl.txt

Danas programiram u Pythonu.
Volim da programiram.
Algoritmika je moja omiljena škola programiranja.

Primer3.py

```
fajl = open("Fajl.txt", "w")
fajl.write("Kad porastem bicu programer!")
fajl.close()
fajl = open("Fajl.txt", "r")
tekst = fajl.read()
fajl.close()
print(tekst)
```

OBJASNJENJE

- "w" povezati sa write reči
- Napomenuti da je normalan tok da se nakon upisa onda ponovo otvara isti fajl ali u modu za citanje, ne može da se iskoristi to što je prethodno taj fajl već otvoren

PRIMER 4 - OVO JE BITNO DA SE URADI

```
# dekodiranje poruke iz fajla
# Unosi se poruka i kod za koliko ce se pomeriti svaki karakter poruke
# program treba da upise kodiranu poruku u fajl
# Nakon toga se unosi kod za dekodiranje i ispisuje početna poruka,
čitanjem iz fajla i dekodiranjem te poruke
```

```
Fajl.txt #prazan
```

```
Primer6.py
```

```
#ucitavanje poruke
print("Unesi poruku")
poruka = input()
print("Unesi tajni kod")
kod = int(input())
kodiranaPoruka = ""
for element in poruka:
    kodiranaPoruka += chr(ord(element)+kod)
fajl = open("Fajl.txt", "w")
fajl.write(kodiranaPoruka)
fajl.close()
print("Poruka je kodirana i upisana u fajl")
print("Unesi kod da je desifrujes")
kod = int(input())
fajl = open("Fajl.txt", "r")
tekst = fajl.read()
dekodiraniTekst = ""
for element in tekst:
    dekodiraniTekst += chr(ord(element) - kod)
fajl.close()
```

```
print(dekodiraniTekst)
```

OBJASNJENJE

- Podsetiti se funkcija `chr()` i `ord()` i šta one rade (spominjane su u stringovima u lvl2 - trebalo bi)

PRIMER 5 - SAMO AKO OSTANE VREMENA

```
# korišćenje seek i tell funkcija
```

```
Fajl.txt
```

```
"Code is like humor. When you have to explain it, it's bad."
```

```
Primer4.py
```

```
f = open("Fajl.txt", "r")
# Drugi parametar seek fje je 0, znaci da setujemo 20poz od starta fajla
f.seek(20)
print(f.tell())
print(f.readline())
f.close()
```

Rezultat

```
20
```

```
When you have to explain it, it's bad.
```

OBJASNJENJE

- Objasniti da `seek()` funkcijom zapravo pomeramo kursor u fajlu odakle treba da čitamo ili dopisujemo.
- Kursor je po default podešen da je na vrhu fajla
- **U nekim verzijama neće raditi `seek()` funkcija sa `open` modom "r", već u binary modu "rb"**
- Bitno je da učenik vidi da može da se kreće kroz fajl
- Može da se ofsetuje pozicija i od početka i sa kraja fajla koji se čita

PRIMER 6 - SAMO AKO OSTANE VREMENA

```
# korišćenje seek funkcije sa brojanjem pozicije unazad
```

```
Fajl.txt
```

```
"Code is like humor. When you have to explain it, its bad."
```

Primer5.py

```
f = open("Fajl.txt", "r")
# Pozicioniramo kursor na deseto mesto sa desne strane
f.seek(-10, 2)
print(f.tell())
print(f.readline())
f.close()
```

Rezultat

47
, its bad.