




PROGRAMIRANJE
Predavanje 05

Ishod 2

1

RJEČNICI

Strana • 3



3

Rječnici

- Rječnik (engl. *dictionary, associative array, map, symbol table*) je kontejner koji sadrži kolekciju parova
 - Svaki par se sastoji od ključa i vrijednosti
- Rječnik programerima minimalno omogućuje:
 - Umetanje novog para zadavanjem ključa i vrijednosti
 - Uklanjanja para prema ključu
 - Izmjenu vrijednosti postojećeg para prema ključu (ali ne i ključa)
 - Dohvat vrijednosti prema ključu (naglasak)
- U Pythonu su rječnici napravljeni pomoću *hash* tablica

Strana • 4



4

Rječnik vs lista

- Rječnici i liste se koriste za različite svrhe:
 - Lista:
 - Kolekcija elemenata kojima želimo pristupiti pomoću indeksa
 - Za svaki element se zna tko mu je prethodnik, a tko sljedbenik
 - Svaka vrijednost se može promijeniti
 - Najčešće se koristi kad želimo proći (iterirati) po (skoro) svim elementima
 - Rječnik:
 - Koristi se kad želimo brzo pronaći vrijednost prema njenom ključu
 - Ključevi se ne mogu mijenjati
 - Nije prikladan za prolazak (iteriranje) po svim elementima

Strana • 5



5

Izrada rječnika

- Rječnik kreiramo pomoću vitičastih zagrada
- U vitičastim zagradama navodimo parove ključ i vrijednost
 - Ključ i vrijednost mogu biti bilo kojeg ugrađenog tipa
 - Između ključa i vrijednosti pišemo dvotočku
 - Parove odvajamo zarezima
- Izrada praznog rječnika


```
drzave = {}
```
- Izrada rječnika s jednim elementom


```
drzave = { "HR": "Hrvatska" }
```
- Izrada rječnika s dva elementa


```
drzave = { "HR": "Hrvatska", "FI": "Finska" }
```

Strana • 6



6

Pristup elementima rječnika

- Za dohvaćanje vrijednost iz rječnika, koristimo uglate zagrade
 - Kod liste smo u iste zagrade pisali indeks
 - Kod rječnika u zagrade pišemo ključ


```
print(drzave["FI"])
```
- Ako ključ ne postoji, dobit ćemo pogrešku
 - Ako želimo provjeriti postoji li ključ, koristimo operator `in`

```
if "FI" in drzave:
    print(drzave["FI"])
```
 - Ako želimo dobiti listu svih ključeva, možemo koristiti ugrađenu funkciju `list()`:


```
print(list(drzave))
```

Strana • 7



7

Izmjena, dodavanje i uklanjanje elemenata

- Vrijednost pod nekim ključem mijenjamo ovako:
`drzave["RU"] = "Rusija"`
- Vrijedi sljedeće:
 - Ako ključ postoji, vrijednost uz njega se mijenja
 - Ako ključ ne postoji, dodaje se ključ i vrijednost
- Ključ i vezanu vrijednost uklanjamo operatorom `del`:
`del drzave["FI"]`

Strana • 8



8

Prolazak kroz sve elemente rječnika

- Petlja `for` nam pomaže u prolasku kroz sve elemente
- Lakša varijanta prolazi kroz sve ključeve:

```
for k in drzave:
    print(k)
```
- Kompleksnija varijanta:

```
for k, v in drzave.items():
    print(k, v)
```

Strana • 9



9

Primjeri

1. Napravite rječnik čiji ključ će biti ime i prezime osobe, a vrijednost broj telefona te osobe.
2. Napišite program koji će od korisnika učitati broj n te zatim u rječnik dodati sve brojevi od 1 do n kao ključeve, a njihove kvadrate kao vrijednosti.
3. Napišite program i zadajte listu. Kreirajte zatim rječnik na način da indeksi liste budu ključevi, a elementi liste budu vrijednosti u rječniku.
4. Napišite program i zadajte rječnik. Kreirajte zatim jednu listu od svih ključeva rječnika, a drugi od svih vrijednosti rječnika.

Strana • 10



10

Primjeri

5. Zadajte dva rječnika. Napišite program koji će napraviti treći rječnik sastavljen od prva dva.
6. Napravite rječnik koji će čuvati ime i prezime studenta pod ključem koji je njegov JMBAG. Pitajte korisnika da unese JMBAG, a vi mu ispišite ime i prezime studenta ili NEMA ako taj JMBAG ne postoji. Nakon svakog učitavanja pitate korisnika želi li dalje i ponavljate dok to korisnik želi.
7. Napravite prazni rječnik. Od korisnika učitajte u rječnik pet zaposlenika i njihove plaće. Nakon toga, omogućite korisniku da upiše zaposlenika, a vi ispišite njegovu plaću ili poruku da nema zaposlenika.

Strana • 11



11

Primjeri

8. Napravite rječnik čiji su ključevi registarske oznake, a vrijednosti vrsta vozila. Omogućite korisniku sljedeće operacije (neka ih može unositi dok god to želi):
 - Unos novog vozila
 - Pretraživanje vozila
 - Brisanje vozila
9. *Promijenite prethodni program tako da uz svaku registracijsku oznaku čuvate i godinu proizvodnje vozila.
10. *Promijenite prethodni program tako da uz svaku registracijsku oznaku čuvate i listu prethodnih tehničkih pregleda.

Strana * 12



12

N-TORKE

Strana * 13



13

N-torke

- N-torka je slična listi
 - Glavna razlika je što se n-torke ne mogu mijenjati
 - Ne možemo mijenjati postojeće elemente
 - Ne možemo dodavati nove elemente
 - Ne možemo brisati postojeće elemente
 - Dodatna razlika je što u n-torki puno češće miješamo vrijednosti različitih tipova

Strana • 14



14

Izrada n-torke

- Izrada n-torke se radi okruglim zagradama
- Izrada prazne n-torke:


```
podaci = ()
```
- Izrada n-torke s jednim elementom:


```
podaci = ("jperic",) # Bez zareza bismo kreirali string!
```
- Izrada n-torke s više elemenata:


```
podaci = ("jperic", "MojPa$$74", True)
```

Strana • 15



15

Pristup elementima n-torke

- Elementima pristupamo indeksiranjem, potpuno jednako kao i kod listi

```
print(podaci[0])
```

- Možemo koristiti i negativno indeksiranje
- Funkcija `len()` vraća broj elemenata u n-torki
- Petlja `for` se koristi jednako kao i kod listi

Strana * 16



16

Pakiranje i otpakiranje n-torki

- Pakiranje vrijednosti u n-torke znači smještanje proizvoljnog broja vrijednosti u novu n-torku

```
podaci = "jperic", "MojPa$$74", True
```

- Otpakiranje vrijednosti iz n-torke znači uzimanje vrijednosti iz n-torke u varijable

```
uid, pwd, remember_me = podaci
print(uid)
print(pwd)
print(remember_me)
```

- Ako nam neka od vrijednosti ne treba, otpakiramo je u `_`

```
podaci = "jperic", "MojPa$$74", True
_, pwd, _ = podaci
print(pwd)
```

Strana * 17



17

Primjeri

1. Učitajte od korisnika ime, prezime i godinu rođenja pa od toga napravite n-torku.
2. Učitajte od korisnika ime, prezime i godinu rođenja pa od toga napravite n-torku. Otpakirajte i ispišite godinu rođenja.
3. Zadajte listu s tri vrijednosti. Napravite od te liste n-torku.
4. Zadajte n-torku od tri vrijednosti. Smislite način kako ćete u n-torku dodati još jednu vrijednost.
5. Zadatak vam je od korisnika učitati naziv lokacije, zemljopisnu širinu i duljinu. Riješite zadatak koristeći najprikladniji kontejner.

Strana • 18



18

Primjeri

6. Zadatak vam je od korisnika učitati 5 brojeva pa ispisati njihov umnožak. Riješite zadatak koristeći najprikladniji kontejner.
7. Zadatak vam je čuvati ime, prezime i broj osobne korisnika. Riješite zadatak koristeći najprikladniji kontejner.
8. Riješite zadatke 5, 6 i 7 koristeći ostale kontejnere.

Strana • 19



19

Za sljedeće predavanje

- ✓ ▪ Ponoviti sve iz ovog predavanja
- ✓ ▪ Pročitati i isprobati primjere iz knjige Think Python: How to Think Like a Computer Scientist:
 - 11.1 A dictionary is a mapping
 - 11.2 Dictionary as a collection of counters
 - 11.3 Looping and dictionaries
 - 11.4 Reverse lookup
- ✓ ▪ Pročitati i isprobati primjere:
 - programiz.com/python-programming/dictionary
 - programiz.com/python-programming/tuple
 - programiz.com/python-programming/string

Strana • 20

