

Python Development

Struktur u
Python-u



POWERED BY



COMTRADE

Mapiranje

`map(naziv_funkcije , *kolekcija)`

- Map poziva funkciju za sve elemente koji se nalaze u kolekciji i kao rezultat vraća objekat tipa map. Da bismo taj rezultat konvertovali u listu, potrebno je da pozovemo metodu `list()`.
- Šta označava zvezdica(*) u argumentu?



POWERED BY



COMTRADE

Map može imati više od jednog argumenta

U slučaju da funkcija koja se mapira prima više od jednog argumenta, nakon funkcije nam je potreban srazmeran broj kolekcija koje će se koristiti kao argumenti.



POWERED BY



COMTRADE

Filtriranje

`filter(naziv_funkcije , kolekcija)`

- Funkcija `filter` zahteva da funkcija koja je prvi argument mora da vraća bool tip. `True` ili `False`, u slučaju da funkcija vraća neku drugačiju vrednost rezultat je samo cela kolekcija.
- Drugi argument za razliku od mape ne sadrži `*` , tako da može se filtrirati samo 1 kolekcija.



POWERED BY



COMTRADE

List comprehension

Kraći način zapisivanja generisanja liste

```
nova_lista = [ izraz_sa_vrednost for vrednost in lista if USLOV]
```



POWERED BY



COMTRADE

Mutable vs Immutable

- Mutable tip podataka - može mu se pristupiti i promeniti vrednost (lista)
- Immutable tip podatka – može mu se pristupiti, ali ne može se menjati vrednost (string)



POWERED BY



COMTRADE

Kolekcije

- List (lista)
- Tuple (torka)
- Set (skup)
- Dictionary (rečnik)



POWERED BY



COMTRADE

Tuple (Torka)

- Immutable tip podatka.
- Može sadržati različite tipove podataka kao i liste
- Deklaracija:

`tup = (1,2,3)` # inicijalizuje se sa malim zagradama

`tup = tuple([1,2,3])` # ili sa pozivom metode torke

`tup[0]` – pristup elementu torke



POWERED BY



COMTRADE

List vs Tuple

Prednosti Tuple-a:

- Koriste manje memorije od listi
- Mogu se koristiti kao ključ u rečnicima za razliku od listi
- Možemo pristupiti elementu na isti način kao i u listi, ali brže.

Mane:

- Ne možemo da dodamo element torke
- Ne možemo da sortiramo elemente torke
- Ne možemo da izbacimo element iz torke
- Ne možemo da promenimo vrednosti elementa u torki



POWERED BY



Metode torki

- naziv.**count**(vrednost) – vraća koliko puta se neka vrednost pojavjuje u torki
- naziv.**index**(vrednost) – vraća poziciju prvog pojavljivanja vrednosti
- **Any**(naziv) – vraća podatak o tome da li je makar jedan element True u torki
- **All**(naziv) – vraća podatak da li su svi elementi torke True
- **sum**(naziv) – vraća sumu svih elemenata torke
- **min/max**(naziv) – vraća min i max
- **sorted**(naziv) – vraća sortiranu LISTU od elemenata torke
- **reversed**(naziv) – vraća opadajući sortiranu listu



POWERED BY



Set

- Setovi predstavljaju skup jedinstvenih elemenata, ne može im se pojedinačno pristupiti niti menjati vrednosti.
- Setovi se iniciraju na sledeće načine
- `skup = { 1, 2, 3 }` # skupovi se iniciraju sa vitičastim zagradama
- `skup = set((1,2,3))` # sa **dvostrukim** zagradama preko funkcije `set`



POWERED BY



COMTRADE

Metode setova

- naziv.**remove**(vrednost) - briše vrednost iz seta ako postoji u setu,u suprotnom izbacuje grešku
- naziv.**discard**(vrednost) – briše vrednost iz seta,bez mogućnosti vraćanja greške.
- naziv.**add**(vrednost) - dodaje vrednost u set
- naziv.**pop**() - ne prima argumente,izbacuje element sa pozicije 0 u setu.
- naziv.**clear**() - izbacuje sve elemente iz seta



POWERED BY



- naziv.**union**(drugi_set) - unija skupova(operator |)
- naziv.**intersection**(drugi_set) - presek skupova(operator &)
- naziv.**difference**(drugi_set) - razlika skupova (operator -)
- naziv.**symmetric_difference**(drugi_set) - vraća sve elemente oba skupa, osim preseka. (operator ^)



POWERED BY



- naziv.**update**(drugi_set) - pravi uniju i apdejuje skup "naziv"
- naziv.**intersection_update**(drugi_set)
- naziv.**difference_update**(drugi_set)
- naziv.**symmetric_difference_update**(drugi_set)
- naziv.**isdisjoint**(drugi_set) - vraća da li je presek prazan skup
- naziv.**issubset**(drugi_set) - vraća da li je drugi_set podskup seta naziv



POWERED BY



Dictionary (rečnik)

Rečnik predstavlja niz kombinacija **key:value** vrednosti.

Key predstavljaju naziv polja u rečniku.

Value predstavlja vrednost tog polja.

```
osoba = {"ime":"Dusan","prezime":"Sijacic"}
```

```
osoba = dict( {"ime":'Dusan', "prezime":'Sijacic'})
```

Pristup elementu i dodavanje novog ključa

```
osoba.get("ime")    ->    Dusan
```

```
osoba["datumRodjenja"] = "1. Februar"
```



POWERED BY



COMTRADE

Metode rečnika

recnik.**popitem**() – izbacuje poslednji ključ iz rečnika

recnik.**clear**() – briše sve elemente iz rečnika

recnik.**values**() – lista vrednosti

recnik.**keys**() – lista ključeva

recnik.**items**() – lista parova ključ,vrednost



POWERED BY



COMTRADE