Pregled struktura podataka u Pythonu

AI centar Lipik

Liste

- Opis: Uređena zbirka elemenata. To znači da elementi imaju svoj specifičan redoslijed i taj redoslijed je bitan. Liste mogu sadržavati elemente različitih tipova (cijeli brojevi, decimalni brojevi, stringovi, druge liste itd.).
- Mijenjanje (Mutable): Liste se mogu mijenjati nakon što su stvorene. Možete dodavati, uklanjati ili mijenjati elemente.
- Duplikati: Liste dopuštaju duplikate (isti element se može pojaviti više puta).
- Indeksiranje: Elementima liste pristupa se pomoću indeksa (počevši od 0 za prvi element).

Liste - primjer

```
# Primjer liste
moja_lista = [10, "hello", 3.14, True, [1, 2]]
print(moja_lista[0])   # Ispisuje: 10
print(moja_lista[1])   # Ispisuje: hello

# Mijenjanje liste
moja_lista[0] = 20
moja_lista.append("world")
moja_lista.remove(3.14)
print(moja_lista)      # Ispisuje: [20, 'hello', True, [1, 2], 'world']
```

Liste – kada se koriste

- Kada je bitan redoslijed elemenata.
- Kada trebate zbirku elemenata kojoj se može mijenjati veličina i sadržaj tijekom izvođenja programa.
- Kada je potrebno pohraniti niz elemenata koji mogu imati duplikate.

- Primjer: red čekanja u banci, playlista na mobitelu, to do lista, jednostavna košarica na web shopu

Ditionary

- Opis: Neuređena zbirka parova ključ-vrijednost. Svaki element u rječniku ima jedinstveni ključ i pridruženu vrijednost. Rječnici su optimizirani za brzo dohvaćanje vrijednosti pomoću ključa.
- Mijenjanje (Mutable): Rječnici se mogu mijenjati nakon što su stvoreni. Možete dodavati, uklanjati ili mijenjati parove ključ-vrijednost.
- Duplikati ključeva: Ključevi moraju biti jedinstveni unutar jednog rječnika. Vrijednosti se mogu ponavljati.
- Pristup: Vrijednostima se pristupa pomoću njihovih ključeva (ne indeksa).

Dictionary - primjer

```
# Primjer rječnika
moj_rjecnik = {"ime": "Pero", "godine": 30, "grad": "Zagreb"}
print(moj_rjecnik["ime"])     # Ispisuje: Pero
print(moj_rjecnik["godine"])  # Ispisuje: 30

# Mijenjanje rječnika
moj_rjecnik["godine"] = 31
moj_rjecnik["spol"] = "Muški"
del moj_rjecnik["grad"]
print(moj_rjecnik)        # Ispisuje: {'ime': 'Pero', 'godine': 31, 'spol': 'Muški'}
```

Dictionary – kada se koriste

- Kada trebate pohraniti podatke u obliku parova ključ-vrijednost.
- Kada je važno brzo dohvaćati vrijednosti pomoću jedinstvenih ključeva.
- Kada redoslijed elemenata nije bitan.

- Primjer: košarica na web shopu (više komada istog proizvoda)

```
kosarica = {
    "proizvod_123": 2,  # Dvije košulje s ID-om "proizvod_123"
    "proizvod_456": 1,  # Jedan par cipela s ID-om "proizvod_456"
    "proizvod_789": 3   # Tri knjige s ID-om "proizvod_789"
}
```

Tuple

- Opis: Uređena zbirka elemenata, slično listama. Mogu sadržavati elemente različitih tipova.
- Nemijenjanje (Immutable): Tuples se ne mogu mijenjati nakon što su stvoreni. Ne možete dodavati, uklanjati ili mijenjati elemente.
- Duplikati: Tuples dopuštaju duplikate.
- Indeksiranje: Elementima tuplea pristupa se pomoću indeksa.

Tuple - primjer

- # Primjer tuplea
- moj_tuple = (10, "hello", 3.14, True)
- print(moj_tuple[0]) # Ispisuje: 10
- print(moj_tuple[1]) # Ispisuje: hello

Tuple – kada se koristi

- Za pohranu zbirki elemenata koji se ne bi trebali mijenjati tijekom izvođenja programa (npr., konstante, koordinate).
- Kao ključevi u rječnicima (liste ne mogu biti ključevi jer su promjenjive).
- Kada je performansa bitna za nepromjenjive sekvence (tuples su često malo brži od lista za iteraciju).

- Primjer: zemljopisne koordinate, rgb boje, konstante, ključevi u rječnicima

Set

- Opis: Neuređena zbirka jedinstvenih elemenata. Skupovi automatski uklanjaju duplikate.
- Mijenjanje (Mutable): Skupovi se mogu mijenjati nakon što su stvoreni. Možete dodavati ili uklanjati elemente.
- Duplikati: Skupovi ne dopuštaju duplikate. Ako pokušate dodati duplikat, on se neće pohraniti.
- Indeksiranje: Elementima skupa se ne može pristupiti pomoću indeksa jer su neuređeni. Koriste se metode za provjeru prisutnosti, dodavanje, uklanjanje i skupovne operacije (unija, presjek, razlika).

Set - primjer

```
# Primjer skupa
moj_skup = {1, 2, 3, 2, 1, 4}
print(moj_skup)      # Ispisuje: {1, 2, 3, 4} (duplikati su automatski uklonjeni)

# Mijenjanje skupa
moj_skup.add(5)
moj_skup.remove(2)
print(moj_skup)      # Ispisuje: {1, 3, 4, 5}
```

Set – kad se koristi

- Za pohranu jedinstvenih elemenata.
- Za brzo provjeravanje prisutnosti elementa u zbirci.
- Za izvođenje matematičkih skupovnih operacija (unija, presjek, razlika).

- Primjer: broj jedinstvenih posjetitlja web stranice

```
posjetitelji_danas = set()

# Simulacija posjeta
posjetitelji_danas.add("192.168.1.100")
posjetitelji_danas.add("10.0.0.5")
posjetitelji_danas.add("192.168.1.100")  # Duplikat se neće dodati
posjetitelji_danas.add("203.0.113.45")
posjetitelji_danas.add("10.0.0.5")      # Duplikat se neće dodati
```