

PROGRAMIRANJE 2

Tuples (n-torke)

Robert Rozić, asis.

dr. sc. Marko Rosić, prof.

Creative Commons

- **Slobodno smijete:**
 - **dijeliti** - umnožavati, distribuirati i javnosti priopćavati djelo
 - **remiksirati** - prerađivati djelo
- **pod sljedećim uvjetima:**
 - **Imenovanje.** Morate priznati i označiti autorstvo djela na način kako je specificirao autor ili davatelj licence (ali ne način koji bi sugerirao da Vi ili Vaše korištenje njegova djela imate njegovu izravnu podršku).
 - **Nekomercijalno.** Ovo djelo ne smijete koristiti u komercijalne svrhe.
 - **Dijeli pod istim uvjetima.** Ako ovo djelo izmijenite, preoblikujete ili stvarate koristeći ga, prerađu možete distribuirati samo pod licencom koja je ista ili slična ovoj.

U slučaju daljnjeg korištenja ili distribuiranja morate drugima jasno dati do znanja licencne uvjete ovog djela. Najbolji način da to učinite je linkom na ovu internetsku stranicu. Od svakog od gornjih uvjeta moguće je odstupiti, ako dobijete dopuštenje nositelja autorskog prava.

Ništa u ovoj licenci ne narušava ili ograničava autorova moralna prava.

Tekst licence preuzet je s <https://creativecommons.org/>.



N-Torke (Tuples)

N-torke (tuples) tj. uređeni parovi imaju široku primjenu u raznim sustavima poput npr. baza podataka.

Tuple (n-torka) je jedan od 4 ugrađena tipa podataka u Pythonu koji se koriste za pohranu skupa podataka.

Koja su ostala 3?

Tuples

Često nam podaci trebaju biti u određenom poretku kako bi bili korisni.

U Pythonu podatke u određenom redoslijedu možemo spremati u:

- **listama**
- **tuples** (n-torkama, uređenim parovima)

Tuples su brži i zauzimaju manje memorije.

- **rječnici** od verzije python-a 3.7

List vs. Tuples

```
# Definiranje Liste
```

```
a = [1, 1, 2, 3, 5, 8, 13]
```

```
# Definiranje Tuple-a (ntorke)
```

```
b = (1, 1, 2, 3, 5, 8, 13)
```

```
# Određivanje duljine
```

```
len(a)
```

```
len(b)
```

```
# Iteriranje i pristup elementima
```

```
for broj in a:
```

```
    print(broj)
```

```
print(a[2])
```

```
for broj in b:
```

```
    print(broj)
```

```
print(b[2])
```

List vs. Tuples

```
# Za usporedbu ćemo koristiti
# ugrađenu funkciju dir
for dl in dir(a):
    if dl[0] != '_':
        print (dl)
print ('-----')
for dl in dir(b):
    if dl[0] != '_':
        print (dl)
```

```
# Provjera veličine liste i ntorke
sys.getsizeof(a)
sys.getsizeof(b)
```

List vs. Tuples

Tuple se za razliku od liste **ne može mijenjati** - *immutable*

Tuple se zbog toga brže kreira.

```
import timeit
```

```
lista = timeit.timeit(stmt="[1,2,3,4,5,6]", number=1000000)
```

```
ntorka = timeit.timeit(stmt="(1,2,3,4,5,6)", number=1000000)
```


List vs. Tuples

```
# Izmjena liste  
lista = [3, 8, 17]  
lista[2] = 5  
print(lista)
```

```
# Izmjena ntorke?  
ntorka = (3, 8, 17)  
# ntorka[2] = 5  
# Greska
```


Tuples (n-torke)

```
prazna_ntorka = ()
```

```
a = (1)
```

```
b = (1, 2)
```

```
c = (1, 2, 3)
```

```
print(prazna_ntorka)
```

```
print(a)
```

```
print(b)
```

```
print(c)
```

```
# Drugi način definiranja
```

```
a = 1,
```

```
b = 1, 2
```

```
c = 1, 2, 3
```

```
print(type(a))
```

```
print(type(b))
```

```
print(type(c))
```

Tuples (n-torke) - Dodjeljivanje vrijednosti

```
# (ime, prezime, godine)
osoba = tuple(("Ivan", "Ivic", 20))

ime = osoba[0]
prezime = osoba[1]
godine = osoba[2]
```

```
# tuple way
ime, prezime, godine = osoba

print("Ime:", ime)
print("Prezime:", prezime)
print("Godine:", godine)
```

Tuples (n-torke) - Dodjeljivanje vrijednosti

```
a, b, c = (1, 2, 3, 4)
```

```
a, b, c = (1, 2)
```

zip - uparivanje

```
imena = ["Ivan", "Ana", "Marko"]  
godine = [20, 19, 21]  
  
studenti = list(zip(imena, godine))  
  
print(studenti)
```

Zadatak 1

Podatke iz datoteke rezultati.csv učitati kao listu ntorki oblika (ime, prezime, bodovi).

Iterirati kroz sve studente i ispisati samo one koji su položili ispit (br. bodova veci od 49)

<https://thispointer.com/python-read-csv-into-a-list-of-lists-or-tuples-or-dictionaries-import-csv-to-list/>

Zadaća

Iz podataka učitanih u prethodnom primjeru sortirati listu po prezimenima.

Napraviti novi rječnik gdje će se po bodovnom rangu upisivati broj ostvarenih ocjena.

Nedovoljan
0-49%

Dovoljan
50-65%

Dobar
65-80%

Vrlodobar
80-90%

Izvrstan
90-100%



sumarum.sum.ba



marko.rosic@pmfst.hr

robert.rozic@fpmoz.sum.ba



HVALA NA POZORNOSTI !

FPMOZ