

Дефиниција секвенци

Секвенце (низови)

Секвенца (низ) је објекат који садржи велики број ствари који су заправо подаци.

Ствари у низу се смештају један после другог.

Пајтон омогућава велики број начина на које се изводе операције на стварима у низу.

Постоји неколико различитих типова објеката низова у Пајтону.

Два типа низова су основна: торке и листе, обе су секвенце које могу садржавати различите типове података.

Торке

Торке су врста секвенци (низова) и оне су непромењиве (immutable).

То значи да када се једном направе, не могу се променити.

Када се креира торка, њени елементи се затварају у заграда:

```
>>> moja_torka = (1, 2, 3, 4, 5)
>>> print(moja_torka)
(1, 2, 3, 4, 5)
>>>
```

У првој линији се креира торка додељивањем секвенце елемената промењивој `moja_torka`.

093 Итерација преко торке

```
imena = ('Miki', 'Kiki', 'Jovan')
for n in imena:
    print(n)
```

```
Miki
Kiki
Jovan
```

Торке подржавају индексирање, методе попут `index`, уграђене функције (`len`, `min`, `max`), изразе одсецања, оператор `in`, операторе `-` и `+`.

Али не подржавају методе попут `append`, `remove`, `insert`, `reverse`, `sort`.

Ако се жели направити торка само са једним елементом, мора се навести зарез после вредности елемента: `moja_torka = (1,)`

Торке имају смисла пошто је обрада торки бржа од обраде листа па се често користе за смештање великог броја података које се никада неће брисати или модификовати.

Рад са торкама

Оператор `*` се до сада користио као симбол за бинарну операцију множења.

Али, када је на левој страни `*` симбола секвенца (попут торке) а на десној страни је операнд у виду целог броја, овај симбол постаје оператор понављања.

Оператор понављања прави више копија торке а затим их удружује.

Општи формат оператора понављања: торка `*` `n`, где је `n` број копија торке.

094 Оператор понављања

```
>>> brojevi = (0, ) * 5
>>> print(brojevi)
(0, 0, 0, 0, 0)
>>>
```

Види се да израз $(0) * 5$ прави пет копија торке $(0,)$ а затим их удружује у једну торку, а крајња торка је додељена промењивој `brojevi`.

095 Оператор понављања код торки

```
>>> brojevi = (1, 2, 3) * 3
>>> print(brojevi)
(1, 2, 3, 1, 2, 3, 1, 2, 3)
>>>
```

096 Садржај постојеће торке доделити новој торци.

```
a = (1, 4, 8, 10)
b = a
print(b)
Даје: (1, 4, 8, 10)
```

097 Приказати поруку преко празне торке.

```
a = ()
if a == ():
    print("Torka je prazna")
else:
    print("Torka nije prazna")
```

098 Торка са различитим типовима података

```
a = (1, "tekst", 2.0965, "", (445, 78))
print(a)
```

099 Креирање торке преко функције `range()`.

```
a = tuple(range(6))
print(a)
```

0100 Креирање торке функцијом `range()` са опсегом.

```
a = tuple(range(5, 11))
print(a)
```

Питања и задаци за самосталан рад

Задаци

0204 Написати програм који приказује на екрану следећу секвенцу (1, 4, 8, 10) као торку и уз то и одговарајућу текстуалну поруку.

- 0205 Написати програм који елементе секвенце (1, 4, 8, 10) повећава за 1 и приказује добијену секвенцу као нову торку.
- 0206 Написати програм који уноси све позитивне целе бројеве од 0 до унетог целог броја у торку.
- 0207 Написати програм који за унети број мањи од 10, уноси све бројеве између 0 и тог броја у једну торку а остале до 10 у другу торку.
- 0208 Ако је унети број паран, креирати торку са парним бројевима од 0 до тог броја, у супротном то исто урадити са непарним бројевима.
- 0209 Написати програм који упоређује елементе две тортке (састављене од целих бројева). Резултат упоређивања може бити „vesi prvi“ ако је већи елемент у првој торци, „vesi drugi“ ако је већи елемент у другој торци, „isti su“ ако су оба елемента истих вредности. Приказати на екрану резултате упоређивања.
- 0210 Корисник уноси цео двоцифрени број са тастатуре. У прву торку сместити цифру јединица па цифру десетица а у другу торку исте цифре у обрнутом редоследу.
- 0211 Корисник уноси двоцифрени и троцифрени број. Цифре бројева сместити у различите торке. На екрану приказати поруке ако су највећи и најмањи бројеви у тим торкама исти.