

Бесконачне петље

Осим у специјалним случајевима, петље морају имати унутар њих начин да се саме заврше, а то значи да унутар тела петље мора постојати нешто што ће променити вредност услова на False.

Ако петља нема начин да сама себе заустави, она се назива **бесконачна петља**.

Бесконачна петља наставља са радом све док се програм некако не прекине и генерално треба избегавати њено креирање.

Коришћење циљне промењиве унутар петље

У for петљи, смисао циљне промењиве је да упућује на сваки елемент низа док петља итерира.

Често се користи циљна промењива у рачунању или другим задацима унутар тела петље.

Пример 1: Приказ целих бројева од 1 до 4 и њихових квадрата

```
print('Broj\tKvadrat broja')
print('-----')
for broj in range(1, 5):
    kvadrat = broj ** 2
    print(broj, '\t', kvadrat)
```

Broj	Kvadrat broja
1	1
2	4
3	9
4	16

У телу петље се користи циљна промењива broj за рачунање вредности квадрата броја.

Пошто циљна промењива мења своју вредност сваком итерацијом (од 1 до 4) онда ће се сваком итерацијом мењати и вредност промењиве kvadrat (биће 1, 4, 9 и 16).

Корисничко управљање итерацијама

Често, програмер зна тачан број итерација које петља мора да изведе.

Али, понекад програмер мора да допусти кориснику да преузме контролу над бројем итерација у петљи.

Пример 2: Корисничка контрола броја итерација

```
print('Program prikazuje listu brojeva')
print('pocevsi od 1 i njihovih kvadrata.')
kraj = int(input('Do kojeg broja da idem? '))
print('Broj\tKvadrat broja')
print('-----')
for broj in range(1, kraj + 1):
    kvadrat = broj ** 2
    print(broj, '\t', kvadrat)
```

```
Program prikazuje listu brojeva
pocevsi od 1 i njihovih kvadrata.
Do kojeg broja da idem? 5
Broj    Kvadrat broja
```

Broj	Kvadrat broja
1	1
2	4
3	9
4	16
5	25

Постављање аргумената range функције

До сада range функција се користила за прављење низа са бројевима који су поређани од најмањег до највећег. Функција range се може користити за генерисање секвенце бројева који су поређани од највећег до најмањег.

Пример 3: Рад са функцијом range

```
print('Broj\tKvadrat broja')
print('-----')
for broj in range(4, 0, -1):
    kvadrat = broj ** 2
    print(broj, '\t', kvadrat)
```

```
Broj      Kvadrat broja
```

```
-----
```

```
4         16
3         9
2         4
1         1
```

Као први аргумент, 4 означава прву вредност за циљну промењиву, други аргумент 0 ограничава до које вредности се може ићи а трећи аргумент -1 каже да се иде од највећег ка мањим бројевима у кораку по 1.

Задатак 044: Бесконачна петља

1. Креирати бесконачну петљу која испишује дату бројчану вредност на екрану преко while петље.
2. Креирати бесконачну петљу која испишује дату стринг вредност на екрану преко for петље.
3. Петљу из задатка 1. модификовати тако да више не буде бесконачна петља.

Задатак 045: Циљна промењива унутар петље

1. Коришћењем циљне промењиве x унутар for петље, направити програм који приказује половину квадрата сваког броја у датом опсегу целих бројева.
2. Коришћењем циљне промењиве a унутар for петље, направити програм који из датог опсега целих бројева приказује све бројеве од броја a до 0 (ако је a позитивно) или од 0 до a (ако је a негативно).

Задатак 046: Корисничко управљање итерацијама

1. Без обзира коју целобројну вредност корисник унесе, програм користи опсег вредности од 10 до 20. Написати код који омогућава да и поред унете корисникове вредности код ради у задатом опсегу приказујући на екрану бројеве из тог опсега. Да ли је ово пример корисничког управљања итерацијама?
2. Написати код који за унету позитивну целобројну вредност, сабира све бројеве између 0 и те вредности. Да ли је ово пример корисничког управљања итерацијама?

Прожебати следеће задатке:

1. Написати програм који сабира све позитивне целе бројеве које корисник уноси све док корисник не унесе негативну вредност. Приказати укупан збир унетих бројева.
2. Ако се доток ваздуха у авионским моторима представи квадратима растућих бројева у опсегу од 1 до 10, написати програм који приказује губитак дотока ваздуха ако се проценат губитака повећава као опсег целих бројева од 1 до 10 (за доток вредности 25 губитак је 5% од те вредности, за доток вредности 64 губитак је 8% од те вредности).
3. За бројеве у опсегу од 1 до 10 прво приказати парне а затим и непарне бројеве (2, 4,...10, 1, 3,...).