

For петље

Као и while петља, петља for је петља са излазом на врху, али основна разлика је у томе што је ово петља контролисана бројачем.

For петља се извршава унапред задатим бројем пута.

Општи формат for петље:

```
for promenjiva in [vrednost1, vrednost2,...]:
    iskaz
```

овде је promenjiva име промењиве а у правоуглој загради је низ вредности. У Пајтону низ вредности у загради одвојени зарезом се назива листа. Исказ чини тело петље.

Прво се промењивој додељује прва вредност из листе а затим се извршава исказ у телу петље.

Затим се промењивој додељује друга вредност из листе и поново се изврши исказ из тела петље итд.

Петља се завршава када је извршен исказ из тела петље са последњом вредности у листи.

Пример 01: Приказ бројева из уређене листе

```
for broj in [1, 2, 3]:
    print(broj)
```

Приказаће се:

```
1
2
3
```

Пример 02: Приказ бројева из неуређене листе

```
for broj in [10, 5, -3]:
    print(broj)
```

Приказаће се:

```
10
5
-3
```

У првој итерацији, промењива broj има вредност 1 и извршава се функција print која штампа ту вредност.

Затим се промењивој broj додели следећа вредност из листе (2) коју функција print штампа, као и за 3.

Каже се да је петља итерирала 3 пута.

У Пајтону се промењива у петљи for назива и циљна промењива (target variable) пошто је она циљ доделе на почетку сваке итерације.

Пример 3: for петља са стринговима у листи

```
for ime in ['Ana', 'Jovana', 'Milanka']:
    print(ime)
```

```
Ana
Jovana
Milanka
```

Употреба функције range са for петљом

Пајтон има уграђену функцију range која поједностављује процес писања for петље.

Функција range креира тип објекта познат као итерабла (iterable).

Итерабла је објекат који садржи низ вредности које итерирају преко нечега налик на петљу.

Пример 4: употреба функције range

```
for broj in range(3):
    print(broj)
```

Уместо листе вредности, преко функције range се даје број 3 као аргумент.

То значи да ће се генерисати итерабилна секвенца целих бројева у опсегу од 0 (укључујући и 0) до 3 (не укључујући и 3).

Пример 4 се може написати и као:

```
for broj in [0, 1, 2]:
    print(broj)
```

Пошто се листа састоји од три вредности, петља ће итерирати три пута.

Пример 5: Исписивање стринга одређени број пута

```
for x in range(3):
    print('Zdravo svete')
```

Zdravo svete

Zdravo svete

Zdravo svete

Када постоји један аргумент унутар range функције, он се користи за лимитирање низа бројева.

Ако постоје два аргумента у range функцији, први аргумент се користи као почетна вредност секвенце а други аргумент као лимитер секвенце вредности.

Пример 6: употреба два аргумента у функцији range

```
for broj in range(1, 5):
    print(broj)
```

1

2

3

4

Пример 7: употреба три аргумента у функцији range

```
for broj in range(1, 8, 2):
    print(broj)
```

1

3

5

7

Када се користи и трећи аргумент (он се назива вредност корака (step value)), сваки следећи број у секвенци ће се повећати за вредност корака.

Задатак 043: Једноставна for петља

1. Помоћу for петље написати код који штампа листу [-100, -200, 0, 100, 200] 5 пута.
2. Помоћу for петље написати код који штампа појединачне елементе листе [-100, -200, 0, 100, 200].
3. Следећи код изменити тако да се уместо листе користи функција range:


```
for x in [0, 1, 2, 3]:
    print("Pajton!!")
```
4. Колико циклуса се извршава у следећем коду:


```
for x in range[-2, 5, 2]:
    pass
```
5. Шта ће се приказати на екрану после ове петље:


```
for x in range[10, 5, -1]:
    print(x)
```
6. Написати код са for петљом који омогућава да се на екрану виде вредности за 1 веће од -10, 20 и 45.
7. Написати код са for петљом који омогућава да корисник 5 пута унесе реч коју он жели.

Прожебати следеће задатке:

1. Направити код који испишује име корисника жељени број пута у две варијанте употребом обе врсте петљи.
2. Написати код који омогућава да се четири пута испише име корисника, али обезбедити да корисник може да у свакој итерацији промени име које жели да се испише.
3. Написати програм који је модификација задатка 2, тако што код обезбеђује да се на екрану не испишују иста имена иако можда корисник то жели.