Napredna svojstva funkcija u Pythonu

Tvoji zadaci

Pročitaj tekst koji se nalazi na sljedećim stranicama ovog dokumenta.

Riješi zadatke koji se nalaze na zadnje dvije stranice ovog dokumenta.

Rješenja pošalji u obliku privatne poruke na Yammeru.

Rješenja možeš poslati do sljedećeg petka.

Lokalne varijable

Lokalne varijable su one varijable koje se definiraju unutar potprograma (funkcija).

One se mogu koristiti i mijenjati jedino unutar potprograma, ne i u glavnom programu.

Primjer:

```
def funkcija():
    a = 2


funkcija()
print(a)
```

Izvršavanjem ovog programa dobit ćemo grešku. Varijabla `a` se ne može koristiti ni mijenjati u glavnom programu, nego samo u potprogramu koji se zove `funkcija`.

Globalne varijable

Globalne varijable se definiraju u glavnom programu. One se mogu upotrebljavati u svim dijelovima programa (uključujući i potprograme), ali ako ih želimo mijenjati unutar potprograma, moramo upotrijebiti naredbu `global` prije mijenjanja varijable u potprogramu:

Primjeri:

Primjer **upotrebe** globalne varijable `a` u potprogramu:

```
def funkcija():
    print(a)

a = 2
funkcija()
```

Ispisat će se broj 2.

Primjer **mijenjanja** globalne varijable `a` u potprogramu:

```
def funkcija():
    global a
    a = 3

a = 2
funkcija()
print(a)
```

Ispisat će se broj 3.

Izborni parametri u funkcijama

Funkcija (potprogram) može primati 0, 1 ili više parametara. Međutim, što ako neki parametar želimo proslijediti funkciji samo u određenim slučajevima, ne i pri svakom pozivu te funkcije?

U tom slučaju taj parametar možemo učiniti **izbornim** u definiciji funkcije. Jednostavno napišemo ime parametra i dodijelimo mu standardnu (*default*) vrijednost. Ako ga pri pozivu funkcije ne proslijedimo, koristit će se standardna (*default*) vrijednost unutar funkcije. Ako ga proslijedimo, zanemarit će se standardna vrijednost.

Primjer izbornog parametra (označen podebljanim slovima) kojem ne prosljeđujemo vrijednost kod poziva:

```
def funkcija(a, b = 10):
    print(a + b)

funkcija(5)
```

Ispisat će se broj 15.

Primjer izbornog parametra (označen podebljanim slovima) kojem prosljeđujemo vrijednost kod poziva:

```
def funkcija(a, b = 10):
    print(a + b)

funkcija(5, 20)
```

Ispisat će se broj 25.

*args

Što ako ne znamo unaprijed koliko ćemo parametara prosljeđivati nekoj funkciji pri njezinom pozivu? U tom slučaju možemo koristiti naprednu definiciju parametara.

Ako kao parametar u definiciji funkcije stavimo *args, moći ćemo tom parametru prosljeđivati koliko god vrijednosti hoćemo. U tom slučaju, taj parametar postaje n-torka (engl. *tuple*).

Primjer parametra *args (označen podebljanim slovima):

```
def funkcija(*args):
    print(args)

funkcija(7, 8, 9)
```

Ispisat će se brojevi 7, 8, 9 u obliku n-torke.

Primjer kombinacije parametra *args (označen podebljanim slovima) i obaveznog parametra:

```
def funkcija(a, *args):
    print(a)
    print(args)

funkcija(1, 7, 8, 9)
```

Prvo će se ispisati broj 1. Ispod tog broja će se ispisati n-torka s brojevima 7, 8 i 9.

**kwargs

**kwargs također pripada naprednoj definiciji parametara i vrlo je sličan parametru *args. Razlika u odnosu na *args je u tome što funkcija parametar **kwargs tumači kao rječnik. To znači da kod pozivanja funkcije ne možemo napisati samo vrijednosti koje želimo proslijediti tom parametru, nego moramo definirati i imena kojima ćemo pridružiti te vrijednosti. Primjer:

Primjer parametra **kwargs (označen podebljanim slovima) i poziva funkcije s tim parametrom:

```
def funkcija(**kwargs):
    print(kwargs)

funkcija(a = 5, b = 7, c = 9)
```

Ispisat će se sljedeći rječnik:

```
{,a': 5, ,b': 7, ,c':9}
```

Napomena

Imena *args* i *kwargs* nisu obavezna. Mogu se koristiti i druga imena, ali većina programera u Pythonu trudi se držati ovih naziva.

Zadaci

1. Što će se ispisati izvođenjem sljedećeg programa? Je li varijabla `a` globalna ili lokalna u funkciji `ispisi`?

```
def ispisi():
    a = 3

a = 4
ispisi()
print(a)
```

2. Što će se ispisati izvođenjem sljedećeg programa?

```
def funkcija(a, b = 5):
    print(a * b)

funkcija(7, 1)
```

Zadaci

3. Napiši funkciju (potprogram) koji ima jedan obavezan parametar i parametar **args*. Funkcija treba ispisati vrijednosti obaju parametara. U glavnom programu pozovi funkciju s nekim vrijednostima.

4. Napiši funkciju (potprogram) koji ima jedan obavezan parametar i parametar ***kwargs*. Funkcija treba ispisati vrijednosti obaju parametara. U glavnom programu pozovi funkciju s nekim vrijednostima.