

Potprogrami

NASTAVLJAMO UČITI O POTPROGRAMIMA

Što smo naučili prošli put?

Prošli put smo naučili da složeni problem možemo rastaviti na više manjih (jednostavnijih) problema. Rješavanjem tih manjih problema možemo u konačnici riješiti složeni problem.

Naučili smo također da složeni program u Pythonu možemo rastaviti na potprograme koje potom možemo pozivati unutar glavnog programa.

Što su potprogrami?

Potprogrami su dijelovi programa koji služe za rješavanje nekih jednostavnijih problema. Oni uvijek počinju riječju `def`, nakon koje slijedi naziv potprograma (možemo upisati koji god naziv hoćemo) i ulazne vrijednosti (parametri) potprograma. Na kraju ide dvotočka!

Promotrite sljedeći primjer potprograma za crtanje trokuta:

Naziv potprograma

Ulazna vrijednost ili parametar (može ih biti i više)

```
def trokut(a):  
    for k in range(3):  
        fd(a)  
        rt(120)
```

Nacrtaj ravnu crtu duljine a

Okreni se u desno za 120°

Kako potprograme smještamo u glavnom programu?

Potprograme uvijek pišemo pri vrhu programa. Evo primjera programa koji u sebi sadrži potprogram za crtanje trokuta:

```
from turtle import *
```

```
def trokut(a):  
    for k in range(3):  
        fd(a)  
        rt(120)
```

```
a = 80  
trokut(a)  
print('nacrtao sam trokut')
```

Ovo je potprogram za crtanje trokuta. **Potprograme uvijek pišemo na vrhu programa. Ako imamo *import* naredbe, onda potprograme pišemo ispod njih.**

Ovo su dijelovi glavnog programa koji sadrži četiri naredbe

Koji je redoslijed izvođenja naredbi?

Promotrite ponovno program koji je prikazan na prethodnoj i ovoj stranici. Prva naredba koja se izvršava je *from turtle import **. Što nakon nje? Sigurno mislite da će Python krenuti izvršavati naredbe potprograma kada izvrši naredbu *from turtle import **. Međutim, neće. **Potprogrami se izvršavaju samo onda kada ih pozovemo!**

```
from turtle import *  
  
def trokut(a):  
    for k in range(3):  
        fd(a)  
        rt(120)  
  
a = 80  
trokut(a)  
print('nacrtao sam trokut')
```

Ovo je prva naredba koja će se izvršiti kada pokrenemo program

Ovo je druga naredba koja će se izvršiti kada pokrenemo program! Python „preskače” potprogram

Pozivanje potprograma

Naredbe unutar potprograma se kreću izvršavati tek kada potprogram pozovemo! Potprogram pozivamo tako da napišemo njegovo ime i unutar zagrada stavimo parametre ili ulazne vrijednosti.

```
from turtle import *

def trokut(a):
    for k in range(3):
        fd(a)
        rt(120)

a = 80
trokut(a)
print('nacrtao sam trokut')
```

Ovo je naredba koja poziva potprogram. Naveli smo njegovo ime i u zagradi stavili varijablu *a* kao parametar. Kada pozovemo potprogram, Python će „skočiti” u potprogram i izvršavat će njegove naredbe. Tek kada izvrši sve naredbe potprograma, nastavlja s izvođenjem naredbe koja je ispod poziva potprograma. **Dakle naredba *print('nacrtao sam trokut')* će se izvršiti tek kada se izvrše sve naredbe potprograma!**

Parametri ili ulazne vrijednosti potprograma

Primijetite na prethodnoj stranici da smo prije poziva potprograma stavili vrijednost 80 u varijablu a . Kada pozovemo potprogram i prosljedimo mu tu vrijednost, potprogram će nacrtati trokut duljine 80.

U tom primjeru potprogram ima samo jednu ulaznu vrijednost ili samo jedan parametar. No, potprogram može imati i više parametara (ili čak nijedan). Evo i primjera:

```
from turtle import *  
  
def trokut (a) (b) :  
    for k in range(3) :  
        fd(a)  
        rt(b)  
    ...
```

Prvi parametar

Drugi parametar

Korištenje prvog parametra

Korištenje drugog parametra

Čemu potprogrami?

Zamislamo da trebamo napisati program koji će računati razliku umnoška i količnika dvaju brojeva, i to raditi 20 puta i pritom svaki put koristiti različite brojeve. Evo kako bi to izgledalo:

```
a = 8
b = 7
rezultat = (a*b) - (a/b)
print('Rezultat je', rezultat)
```

```
a = 3
b = 2
rezultat = (a*b) - (a/b)
print('Rezultat je', rezultat)
```

...

```
a = 9
b = 3
rezultat = (a*b) - (a/b)
print('Rezultat je', rezultat)
```

I tako 20
puta

Čemu potprogrami?

Očito je da će program na prethodnoj stranici biti poprilično dug. Primijetimo da se određene stvari ponavljaju, odnosno da problem možemo rastaviti na jednostavnije probleme. Jedan od tih jednostavnijih problema je računanje razlike umnoška i količnika dvaju brojeva. Za taj manji problem možemo napisati potprogram i onda ga pozivati u glavnom programu. Evo kako bi to izgledalo:

```
def izracun(a, b):  
    rezultat = (a*b) - (a/b)  
    print('Rezultat je', rezultat)  
  
izracun(8, 7)  
izracun(3, 2)  
...  
izracun(9, 3)
```

Ovaj program će biti manji i pregledniji u odnosu na program s prethodne stranice

Vaši zadaci

Vaš zadatak je da odgovorite na pitanja koja se nalaze na sljedećim stranicama. Odgovore pošaljite u obliku privatnih poruka na Yammeru!

1. Pitanje

Koji je naziv ovog potprograma?

```
def operacija(a):  
    rezultat = a * a  
    print('Rezultat je', rezultat)
```

2. Pitanje

Koliko parametara ima ovaj potprogram?

```
def potprogram(c, b, a):  
    rezultat = a + b + c  
    print('Rezultat je', rezultat)
```

3. Pitanje

U ovom primjeru svaka linija programa ima svoj redni broj. Koji je redni broj naredbe pomoću koje se poziva potprogram?

```
1. def izracun(a, b):  
2.     rezultat = (a*b) - (a/b)  
3.     print('Rezultat je', rezultat)  
4.  
5.     a = 8  
6.     b = 1  
7.     izracun(a, b)  
8.     print('Ovdje pozivam potprogram')
```

4. Pitanje

Koja je prva naredba koja će se izvršiti kada pokrenemo ovaj program?

```
def potprogram():  
    print('Ja sam potprogram')  
  
print('Informatika')  
potprogram()
```

5. Pitanje

Napiši program koji će u sebi sadržavati potprogram. Potprogram bi trebao imati dva parametra. Parametre možete nazvati kako hoćete. Potprogram treba pomnožiti vrijednosti tih dvaju parametara te tom umnošku treba nadodati broj 2. Potom treba ispisati rezultat. U glavnom dijelu programa definirajte dvije varijable i pridružite im neke brojčane vrijednosti. Varijable možete nazvati kako hoćete. Pozovite potprogram i kod poziva iskoristite te varijable kao parametre potprograma.

Program ne morate pisati u IDLE-u, možete ga napisati kao tekst. Pazite samo na to da uvučete odgovarajuće naredbe udesno.