

Uvod v Python

Funkcije in globalne
spremenljivke

Funkcije v Pythonu

- Funkcija je blok organizirane kode za večkratno uporabo, ki je namenjena za enotno povezano dejanje.
- Ta nam omogoča boljšo zgradbo našega programa in visoko stopnjo ponovne uporabe dela programa.

Primer 1.1:

Preprosta funkcija z dvema spremenljivkama

```
def ime_funkcije(): #Definiranje funkcije
    print("Začetek funkcije.")
    z = 3 + 5
    print(z)
    print("Konec funkcije.")
ime_funkcije() #Ukaz za zagon funkcije
```

Primer 1.2:

Parametri v funkciji

```
# -*- coding: utf-8 -*-  
def funkcija(a,b): #Definiranje funkcije in določanje  
    # vhodnih parametrov (a in b)  
  
    z = a + b  
    print("Rezultat je: ", z)  
    #print("Rezultat je: ", a + b)  
funkcija(3, 10) #Zagon funkcije in definiranje  
    # vhodnih parametrov  
funkcija(a = 3, b = 10) #Bolj pregleden, daljši zapis  
    # ukaza zagona funkcije
```

Primer 1.3:

Privzeti (default) parametri v funkciji

```
def f_privzeti(pp1 = 10, pp2 = 5):  
    #Ko definiramo funkcijo lahko parametrom določimo  
    #privzeto vrednost parametrov  
    print("Rezultat je: ", pp1**pp2)  
f_privzeti() #Uporabljeni bodo privzeti parametri  
f_privzeti(5, 2) #Določimo svoje vrednosti parametrov
```

Primer 1.4:

```
# -*- coding: utf-8 -*-
def račun(cena1 , cena2 , valuta = "€"):
    print("Pepsi: ", cena1, valuta)
    print("Jufka: ", cena2, valuta)
    print("Skupaj: ", cena1+cena2, valuta)
račun(2, 4)
print("-----")
račun(2, 4, "$")
print("Thank you, come again!")
print("-----")
```

Lokalne in globalne spremenljivke

- Pogoji globalne in lokalne spremenljivke ustrezajo določenemu dosegu spremenljivke v skripti ali programu
- Globalna spremenljivka je tista, do katere je mogoče dostopati kjerkoli v programu
- Lokalna spremenljivka je ravno nasprotno, do nje je v programu mogoče dostopati le v določenem okviru

Primer 2.1:

Definiranje globalne in lokalne spremenljivke in poskus izpisa lokalne spremenljivke zunaj svojega obsega

```
x = 6 #Definiranje vrednosti globalne spremenljivke
def example():
    print("Vrednost globalne spremenljivke x: ", x)
    z = 5 #Definiranje vrednosti lokalne spremenljivke
    print("Vrednost lokalne spremenljivke z: ", z)
example()
print(z) #Poskus izpisa lokalne spremenljivke zunaj
        # definicije funkcije -> NameError: name 'z' is
        # not defined
```


Primer 2.2:

Napačen način spremembe globalne spremenljivke znotraj funkcije

```
x = 6 #Globalna spremenljivka
def example():
    print("Vrednost globalne spremenljivke x: ", x)
    print(x+5)
    x += 6 #Znotraj funkcije smo želeli spremeniti
          # (overwrite) globalno spremenljivko. To na
          # tak način ni možno -> ERROR
example()
```

Primer 2.3:

Pravilen način spremembe globalne spremenljivke znotraj funkcije

```
x = 6
def example():
    global x #Definiramo "x" kot globalno spremenljivko
    print("Vrednost globalne spremenljivke x: ", x)
    print(x+5)
    x += 6 #Sedaj lahko spreminjamo globalno
          # spremenljivko "x"
example()
```

Primer 2.4:

Lokalna spremenljivka z referenco na globalno spremenljivko

```
x = 6
def example():
    glob_x = x #Nova lokalna spremenljivka z referenco
               # na globalno spremenljivko "x"
    glob_x += 8
    print(glob_x)
example()
print(x) #Naša globalna spremenljivka "x" ostane
        # nespremenjena
```

Primer 2.5:

Funkcija za spremembo globalne spremenljivke

```
x = 6
def example(x):
    print(x)
    x += 5
    print(x)
    return(x)
x = example(x) #S pomočjo funkcije spremenimo vrednost
               # globalne spremenljivke

print(x)
#Zgornja koda je za namene poglobljanja znanja. Globalno
spremenljivko lahko brez problema spremenimo v
"globalnem" področju kode:

y = 6
y += 6
print(y)
```

Naloga: Definicija

Napiši Python funkcijo, ki bo izpisala naslednje:

1. Oktober 2015
2. Oktober 2015
3. Oktober 2015
- ...
31. Oktober 2015

Uporabi:

-Funkcijo v Pythonu

(Hint: uporabi `for` zanko skupaj z `in range`)

Če Python v `print()` sam meče nepotrebne presledke si pomagaj s `sep=''`, ki izključi avtomatsko dodajanje presledkov med output-i (dodaj na skrajnem koncu `print()` stavka!)

Primer: `print("Indeks: ", i, sep='')`

Naloga: Rešitev

```
x = int(input("Vstavi range: "))
def generiran_seznam(x_range):
    for i in range(x_range):
        if(i+1 <= 9):
            print(i+1, ". Oktober 2015", sep='')
        else:
            print(i+1, ". Oktober 2015", sep='')
generiran_seznam(x)
```