5. Definicija funkcije

Uporabniška funkcija je eden ključnih programskih konstruktov. Preden se je lotimo, moramo razumeti razliko med interaktivnim delom v okolju IDLE ter programiranjem. Pri programiranju namreč ukaze pišemo v posebno datoteko in se ne izvajajo sproti. Tak način dela ima veliko prednosti pred sprotnim (interaktivnim) izvajanjem ukazov. To nam namreč omogoča preglednejše razmišljanje, izdelek ima "trajno" vrednost, ukvarjamo se lahko z bolj zahtevnimi problemi, ki jih rešujemo dalj časa. Pri takem načinu dela, ko se k izdelku vračamo večkrat in načrta na moremo več obdržati v celotu v glavi, je nujno, da si ključne točke zabeležimo. Če je nekaj razumljivo računalniku, še ni nujno, da je razumljivo človeku. Skoraj vsi programski jeziki in tudi Python poznajo mehanizem, ko lahko v program, zapišemo opombe, ki na potek programa nimajo nobenega vpliva in so namenjene le osebi, ki program piše ali bere. Takemu mehanizmu pravimo komentar.

5.1. Komentarji

V jeziku Python je # znak za komentar. Interpreter (tolmač) vse kar stoji za tem znakom v vrstici ignorira. (Izjema je le, če znak za komentar uporabimo v nizu)

S komentarji označujemo večje sklope programa, včasih pa le pojasnimo kakšen človeku težko umljiv stavek.

5.2. Definicija funkcije - stavek def

Spoznali smo že nekaj vgrajenih funkcij, ki je Python prepozna. Več jih dobimo iz knjižnic. Kasneje si bomo ogledali *Standardno kjnižnico*, ki je ni težko uporabljati. Lahko pa napišemo svojo funkcijo.

Svojo funkcijo definiramo z uporabo rezervirane besede `def`, ki ji sledi ime funkcije (ime naj bo takšno, da lahko iz njega sklepamo, kaj funkcija počne) in imena parametrov, zapisana v oklepajih. Če je parametrov več, jih ločimo z vejicami. Na koncu vrstice je obvezno dvopičje.
Sledijo stavki, ki se bodo izvršili vsakič, ko bomo funkcijo poklicali. Ti stavki morajo biti odmaknjeni od levega roba.

```
def f(x):
      A
      B
      return(rezultat)
```

5.3. Ukaz `return`

Z ukazom return prekinemo tek programa v funkciji in vrnemo rezultat funkcije. V zgornjem primeru je f ime funkcije, x parameter, končna vrednost, ki jo vrnemo pa je rezultat.
Če se funkcija konča brez ukaza return, je vrednost funkcije posebna konstanta `None`.

5.4. Parametri

S parametri praviloma določimo vhod v funkcijo. Ob definiciji govorimo o formalnih parametrih, pri klicu pa o dejanskih parametrih. Parametre ločimo z vejico. Funkcija ima lahko tudi prazen seznam parametrov.

5.5. Zgleda:
1. Napišite funkcijo, ki za dolžine stranic a,b,c vrne polovični obseg ustreznega trikotnika.

```
def s(a,b,c):
      return (a+b+c)/2
```

2. Napišite funkcijo, ki za dolžine stranic a,b,c vrne ploščino ustreznega trikotnika po Heronovem obrazcu.

```
def Heron(a,b,c):
      s = (a+b+c)/2 # polovicni obseg
      p2 = s*(s-a)*(s-b)*(s-c)
      return p2**(1/2)
```

5.6. Dokumentacijski niz

Takoj za vrstico def običajno zapišemo dokumentacijski niz

```
def f(x):
""" opis funkcije """
...
```

Kasneje ga lahko prikličemo z ukazom `help(f)`, če smo pozabili pomen funkcije in njenih parametrov. Dokumentacijski niz se lahko razteza prek več vrstic.

Naprimer, tole bi bilo bolje:

```
def Heron(a,b,c):
""" Heron(a,b,c) izračuna ploščino trikotnika s
stranicami a, b in c po Heronovem obrazcu."""
      s = (a+b+c)/2 # polovicni obseg
      p2 = s*(s-a)*(s-b)*(s-c)
      return p2**(1/2)
```

Poskusite poklicati `help(Heron)` .

5.7. Lokalne spremenljivke.

Spremenljivke definirane znoraj funkcije so lokalne. To pomeni, da jih okolje, v katerem kličemofunkcijo ne pozna.

5.8. Klic funkcije.
Funkcijo pokličemo tako, da na mesta formalnih parametrov postvimo izraze, ki se izračunajo v vrednosti dejanskih parametrov.

5.9. Stranski učinki(*)

Pri programiranju funkcij, ki udejanjajo paradigmo strukturiranega programiranja, se izogibamo stranskim učinkom. Običajno je pri klicu funkcije, npr. y = f(x) edina sprememba v okolju nova vrednost spremenljivke y. Stranski učinek je na primer dosežen, če ob računanju vrednosti y popravimo tudi vrednost x. Včasih pa se stranskim učinkom ne moremo ali ne želimo odpovedati. Funkcije, ki se končajo brez stavka `return` so najbrž napačno sprogramirane ali pa imajo stranske učinke. Zato moramo biti nanje še posebej pozorni. Pogosta napaka, ki jo delajo začetniki pri funkcijah, je nerazumevanje razlike med stavkom `return` in funkcijo `print`. Na to razliko bomo večkrat opozarjali.

5.10. Naloge (Naloge od 5-11 smo že srečali).

1. Sestavite funkcijo, ki vrne desetice celega števila (števka, ki je na predzadnjem mestu).
2. Sestavite funkcijo, ki vrne vrednost 1, če je argument med 0 in 1, sicer pa naj bo vrednost 0.
3. Sestavite funkcijo, ki vrne obrat tromestnega celega števila.
4. Sestavite funkcijo, ki dano dolžino v yardih pretvori v metre, decimetre in centimetre. Ta funkcija naj rezultate izpiše.
5. Sestavite funkcijo, ki bo izračunala vrednost periodične realne funkcije, ki je na [0,10) definirana takole:
    1. če je $0 <= x < 2$, potem je $f(x) = x + x^2$
    2. če je $2 <= x <= 4$, potem je $f(x) = |x-3|$
    3. če je $4 < x < 7$, potem je $f(x) = (x - 5)^2$
    4. če je $7 <= x < 10$, potem je $f(x) = 12 - 5x$
6. Pravokotnik v ravnini, ki ima stranice vzporedne s koordinatnima osema, predstavimo s koordinatami dveh nasprotnih si oglišč. Sestavite funkcijo, ki bo za dani pravokotnik in dano točko v ravnini preverila, kje leži točka: znotraj, zunaj ali na robu pravokotnika.
7. Sestavite funkcijo, ki bo za dani premici v ravnini preverila, ali sta enaki, vzporedni, ali pa se sečeta v natanko eni točki (to točko naj tudi izračuna). Vsaka od premic je podana s smernim koeficientom in prostim členom.
8. Sestavite funkcijo `resi(a,b)`, ki reši linearno enačbo: `a*x + b = 0`.
9. Sestavite funkcijo `ploscina(a,b,c,d,e,f)`, ki izračuna ploščino trikotnika z oglišči v točkah `A(a,b)`, `B(c,d)` in `C(e,f)`.
10. Sestavite funkcijo, ki ugotovi, ali so točke A, B in C iz prejšnjega primera kolinearne.
11. Sestavite funkcijo, ki ugotovi, ali si točke A, B in C sledijo v nasprotni smeri urinega kazalca.

5.11. Rešitve.

1)

```
def desetice(n):
    """ desetice(n) vrne desetice celega stevila"""
    return (n//10%10)
```