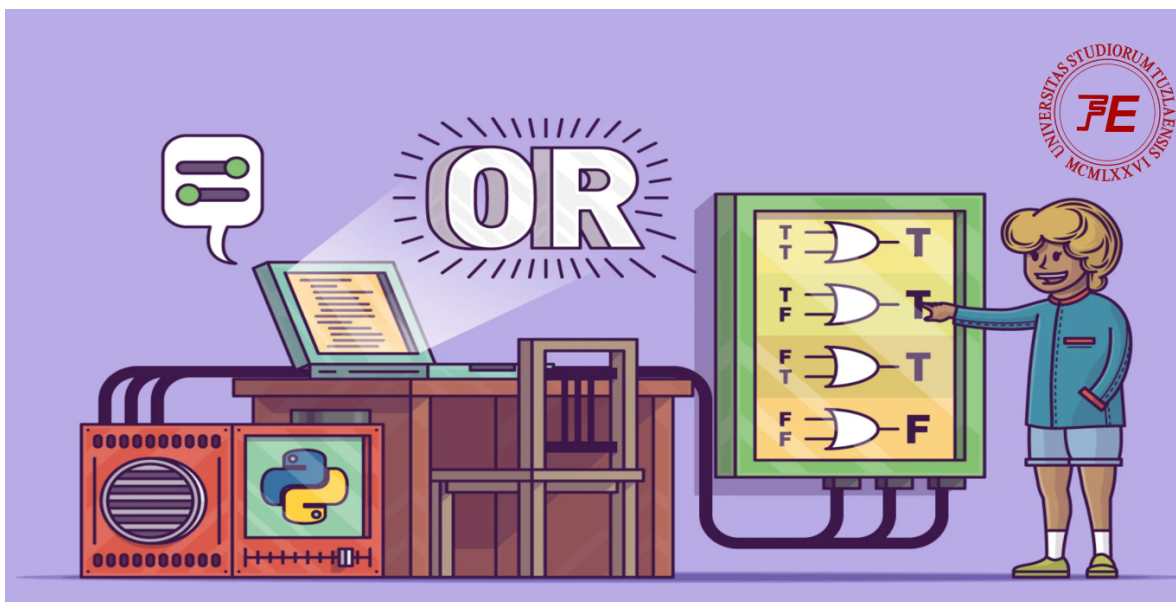


UNIVERZITET U TUZLI
FAKULTET ELEKTROTEHNIKE

Python tutoriali

Signali i sistemi

Emina Bajrektarević, Amara Sinanović



Tuzla, Septembar 2020. godine

Python Tutorijal 2

- Komentari
- Varijable
- Aritmetičke operacije
- Operatori poređenja
- Logičke operacije
- Stringovi

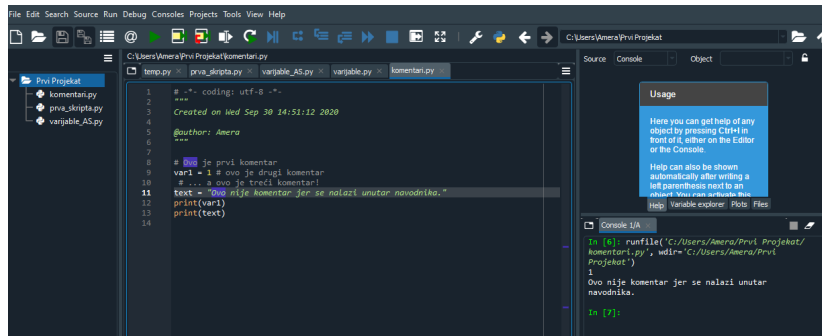
Komentari

Kao i u svakom programskom jeziku tako i u Pythonu postoje komentari koji programerima olakšavaju pisanje programskog koda. Korisno je opisati što se kojim od dijelova programskoga koda želi postići. Takvi opisi nazivaju se komentari. U Python-u se komentari mogu označiti na dva načina i to:

- Linijski komentari

Linijski komentari počinju znakom "hash" - #. Nakon znaka za početak komentara, on se proteže do kraja linije.

```
1  # -*- coding: utf-8 -*-
2  """
3  Created on Wed Sep 30 14:51:12 2020
4
5  @author: Amera
6  """
7  # =====
8  # Ovo je blok komentar
9  # =====
10
11 # Ovo je prvi komentar
12 var1 = 1 # ovo je drugi komentar
13 # ... a ovo je tre i komentar!
14 text = "Ovo nije komentar jer se nalazi unutar navodnika."
15 print(var1)
16 print(text)
```



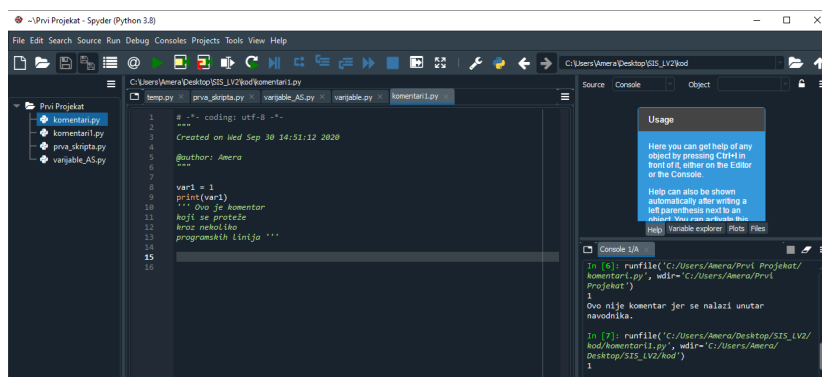
• Višelinijski komentari

Višelinijski komentari su komentari koji se protežu kroz nekoliko programskih linija a počinju i završavaju s ''' ili """ (tri jednostruka ili tri dvostruka navodnika). Bitno je naglasiti da znak za početak komentara koji se proteže kroz nekoliko programskih linija mora biti napisan na početku reda, tj. ispred znaka za početak komentara u nekoj liniji ne smije pisati nikakav programski kod.

```

1 # -*- coding: utf-8 -*-
2 """
3 Created on Wed Sep 30 14:51:12 2020
4
5 @author: Amera
6 """
7
8 var1 = 1
9 print(var1)
10 ''' Ovo je komentar
11 koji se proteže
12 kroz nekoliko
13 programskih linija '''

```



Varijable

Varijable su memorijske lokacije koje služe za pohranu vrijednosti, što znači da se prilikom stvaranja nove varijable u memoriji rezervišu određeni dio prostora za pohranu. Zavisno od tipa podatka, alokira se veći ili manji dio memorije i na taj način je moguće pohranjivati različite tipove podataka kao što su cijeli brojevi, racionalni brojevi, nizovi, stringovi, varijable i kompleksniji tipovi podataka s kojima Python može raditi. Važno je naglasiti da je Python objektno orijentisani programski jezik, odnosno tipovi podataka predstavljeni su kao objekti određenog tipa podatka.

Tipove varijabli u Pythonu nije potrebno eksplicitno pisati jer se deklaracija događa automatski pri-

likom upisivanja vrijednosti u varijablu. Znak jednakosti (=) koristi se za pridruživanje vrijednosti varijablama. S lijeve strane znaka jednakosti nalazi se varijabla dok s desne, njezina vrijednost.

Osnovi tipovi podataka u pythonu su:

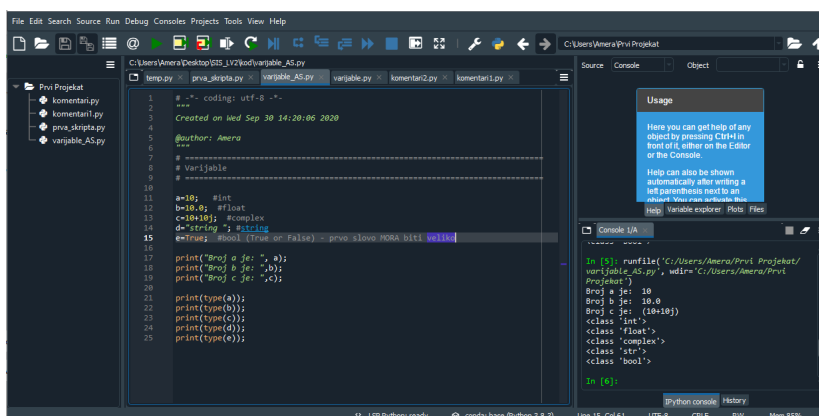
- Cijeli brojevi (int)
- Racionali brojevi (float)
- Kompleksni brojevi
- Logički podaci (bool)
- Nizovi karaktera (string)

Primjer kreiranja varijabli različitih tipova podataka prikazan je na slici ispod.

```

1  """
2  Created on Wed Sep 30 14:20:06 2020
3  @author: Amara
4  """
5  # =====
6  # Varijable
7  # =====
8
9  a=10;    #int
10 b=10.0;  #float
11 c=10+10j; #complex
12 d="string"; #string
13 e=True;  #bool (True or False) - prvo slovo MORA biti veliko
14
15 print("Broj a je: ", a);
16 print("Broj b je: ", b);
17 print("Broj c je: ", c);
18
19 print(type(a));
20 print(type(b));
21 print(type(c));
22 print(type(d));
23 print(type(e));

```



Prilikom kreiranja imena varijabli, potrebno je dobro razmisliti o najboljem imenu za neku varijablu, jer pametan odabir imena će mu omogućiti lakše pisanje programskoga koda, a posljedično i lakše snalaženje u programskom kodu. Naprimjer, kada je potrebno u varijablu spremiti trenutnu godinu, tada je tu varijablu pametnije nazvati trenutnaGodina, a ne trenutniMjesec.

Aritmetičke operacije

Python podržava nekoliko unaprijed defisanih aritmetičkih operatora za rad s brojevima koji će biti prikazani u tabeli ispod.

Operator	Namjena	Primjer upotrebe	Rezultat a=10, b=5
+	Sabiranje	x=a+b	x=15
-	Oduzimanje	x=a-b	x=5
*	Množenje	x=a*b	x=50
/	Dijeljenje	x=a/b	x=2
%	Ostatak cjelobrojnog dijeljenja	x=a%b	x=0
**	Potenciranje	x=a**b	x=100000
//	Cjelobrojno dijeljenje	x=a//b	x=2

Tablica 2.1: Aritmetički operatori

Također, uz gore navedene aritmetičke operatore u programskim jezicima, pa tako i u Pythonu, postoji skraćena verzija operatora pridruživanja. U nastavku je prikazana tabela tih operatora.

Operator	Primjer	Osnovni oblik	Rezultat a=10, b=5
+=	a+=b	a=a+b	a=15
-=	a-=b	a=a-b	a=5
=	a=b	a=a*b	a=50
/=	a/=b	a=a/b	a=2
%=	a%=b	a=a%b	a=0
=	a=b	a=a**b	a=100000
//=	a//=b	a=a//b	a=2

Tablica 2.2: Skraćena verzija operatora pridruživanja

Kao što se može vidjeti iz gornje tabele, skraćeni operatori omogućavaju kraći zapis jedne te iste stvari koja se u svim slučajevima može odraditi s osnovnim aritmetičkim operatorima.

```

1  #-*- coding: utf-8 -*-
2  # =====
3  # Aritmetički operatori
4  # =====
5  x = 15
6  y = 4
7
8  # Rezultat: x + y = 19
9  print('x + y =', x+y)
10
11 # Rezultat: x - y = 11
12 print('x - y =', x-y)
13
14 # Rezultat: x * y = 60
15 print('x * y =', x*y)
16
17 # Rezultat: x / y = 3.75
18 print('x / y =', x/y)
19
20 # Rezultat: x % y = 3
21 print('x % y =', x%y)
22
23 # Rezultat: x // y = 3

```

```

24 print('x // y =', x//y)
25
26 # Rezultat: x ** y = 50625
27 print('x ** y =', x**y)

```

Operatori poređenja

Operatori poređenja koriste se za poređenje vrijednosti s lijeve i s desne strane operatora. Operatori poređenja vraćaju vrijednosti 0 (netačno, engl. False) ili 1 (tačno, engl. True).

Operator	Opis	Primjer	Rezultat 1 a=10, b=5	Rezultat 2 a=5, b=5
==	Jednako	a==b	False	True
!=	Različito	a!=b	True	False
>	Veće	a>b	True	False
<	Manje	a<b	False	False
>=	Veće ili jednako	a>=b	True	True
<=	Manje ili jednako	a<=b	False	True

Tablica 2.3: Operatori poređenja

Operatori poređenja se najčešće koriste za kontrolu toka programa.

```

1 # -*- coding: utf-8 -*-
2 # =====
3 # Operatori poredjenja
4 # =====
5 x = 10
6 y = 12
7
8 # Rezultat: x > y - False
9 print('x > y is', x>y)
10
11 # Rezultat: x < y - True
12 print('x < y is', x<y)
13
14 # Rezultat: x == y - False
15 print('x == y is', x==y)
16
17 # Rezultat: x != y - True
18 print('x != y is', x!=y)
19
20 # Rezultat: x >= y - False
21 print('x >= y is', x>=y)
22
23 # Rezultat: x <= y - True
24 print('x <= y is', x<=y)

```

Logički operatori

Logički operatori koriste se za kreiranje složenijih logičkih izraza. Za kreiranje složenijih izraza koriste se tri logička operatora, a to su:

- Operator and – operacija I (konjunkcija)
- Operator or – operacija ILI (disjunkcija)
- Operator not – operacija NE (negacija).

Kod složenijih logičkih izraza moguće je istovremeno kombinirati više vrsta operacija koje se izvršavaju sljedećim rasporedom:

1. Aritmetičke operacije
2. Operacije poredjenja
3. Logičke operacije

```

1  # -*- coding: utf-8 -*-
2  # =====
3  # Logicki operatori
4  # =====
5
6  x = True
7  y = False
8
9  print('x and y is',x and y)
10
11 print('x or y is',x or y)
12
13 print('not x is',not x)

```

Stringovi

Kako smo već ranije naglasili, string je jedan od osnovnih tipova podataka u Paythonu. String se kreira stavljanjem teksta unutra navodnika, a mogu se koristiti jednostruki, ' , ili dvostruki, " navodnici. Trostruki navodnici se koriste kada se string proteže na više od jedne linije. U primjeru možemo da vidimo neke od mogućih operacija nad stringovima.

```

1  # -*- coding: utf-8 -*-
2
3  a="prva";
4  b="druga";
5
6  c=a+b; #spajanje dva stringa
7  print(c)
8
9  d=input("Unesite rije : "); #unos sa tastature
10
11 e=len(c); # funkcija koja se koristi za odredjivanje duzine stringa
12
13 niz=' '*10; #string koji se sastoji od 10 *
14 niz2= '-'*20; #string koji se sastoji od 20 -
15
16 print(niz)
17 print(niz2)
18
19 #indeksiranje stringova po inje od 0
20 print(a[0]); #prvi karakter stringa a
21 print(a[20]); # error messageindeks van opsega
22
23 maksimalan=max(a); #karakter iz stringa koji ima najve u ASCII vrijednost
24 minimalan=min(a); #karakter iz stringa koji ima najmanju ASCII vrijednost
25
26
27 # =====
28 # Segmentacija stringova - Izdvajanje dijela stringa
29 # =====
30
31 w='abcdefghijk';
32 w1= w[1:3] # string koji se sastoji od prvog i drugog karaktera stringa w
33 w2= w[:5] # prvih 5 karaktera stringa w
34 w3 = w[5:] #elementi stiga od 5 karaktera do kraja
35 w4= w[-2:] #zadnja dva karaktera stringa
36 w5= w[:] #citav sting
37 w6 = w[1:7:2]; # Slova od indeksa 1 do 7, svako drugo
38
39

```



```

40 # =====
41 # Metodi nad stringovima
42 # =====
43
44 f= 'ksdfnjsdf';
45 g= 'AAAAAAA';
46
47 f1=f.upper(); # konverzija svih malih slova u velika
48 g1=g.lower() #konverziha svih velikih slova u mala
49
50 print(f1, '+++', g1)
51
52 m='matematika';
53 l=m.replace('a','o') #zamijeni svaki karakter a sa karakterom o u stringu m
54 print(l)
55
56 slova=m.count('t') # broj karaktera t u stringu m
57
58 indeks= m.index('a'); #vraca indeks prvog pojavljivanja 'a' u stringu m

```

Zadatak 2.1

Napišite program koji će u varijable a i b spremiti dva dvocifrena broja. U varijablu a spremite posljednju cifru broja koji se nalazi u varijabli b, a u varijablu b spremite zadnju cifru broja koja se nalazi u varijabli a. Ispišite sadržaj varijabli a i b.

Rješenje

```

1 # -*- coding: utf-8 -*-
2
3 # =====
4 # Napišite program koji će u varijable a i b spremiti dva
5 #dvocifrena broja. U varijablu a spremite posljednju cifru broja koji se
6 #nalazi u varijabli b, a u varijablu b spremite zadnju cifru
7 # broja koja se nalazi u varijabli a. Ispišite sadržaj varijabli a i b.
8 # =====
9
10 #a=input("Unesite prvi cijeli dvocifreni broj:");
11 #b=input("Unesite drugi cijeli dvocifreni broj:");
12
13 a=input('Unesite prvi dvocifreni broj:') # a - string
14 b=input('Unesite drugi dvocifreni broj:') # b- string
15
16 a=eval(a); #konverzija stringa u numericku vrijednost
17 b=eval(b); #konverzija stringa u numericku vrijednost
18
19
20 a1= a//10; #desetice
21 a10= a%10; #jedinice
22
23 b1=b//10; #desetice
24 b10=b%10; #jedinice
25
26 a=b10;
27 b=a10;
28
29 print(a);
30 print(b);

```

Zadatak 2.2

U varijablu upišite neki proizvoljni niz znakova (string). Nad varijablom pozovite odgovarajuću funkciju koja će vratiti dužinu upisanoga niza znakova te rezultat spremite u varijablu. Na temelju dužine niza ispišite sve znakove do polovine niza. Na temelju dužine niza ispišite sve znakove od polovine niza. Kreirati novu varijablu koju će činiti elementi početnog stringa sa parnim indeksima. Kreirati novu varijablu koju će činiti elementi početnog stringa sa neparnim indeksima.

Rješenje

```

1  # -*- coding: utf-8 -*-
2  # =====
3  """
4  U varijablu upišite neki proizvoljni niz znakova (string). Nad varijablom
5  pozovite odgovaraju u funkciju koja će vratiti dužinu upisanoga niza
6  znakova te rezultat spremite u varijablu. Na temelju dužine niza
7  ispišite sve znakove do polovine niza. Na temelju dužine niza
8  ispišite sve znakove od polovine niza. Kreirati novu varijablu koju će
9  činiti elementi početnog stringa sa parnim indeksima. Kreirati novu varijablu koju će
10 činiti elementi početnog stringa sa neparnim indeksima.
11
12 """
13 # =====
14
15 a= 'abcdefghijk'
16 print(a)
17
18 l=len(a); #odredjivanje duzine stringa
19
20 mean= l//2; #indeks srednjeg elementa
21
22 a2=a[:mean]; # a2 se sastoji od prve polovine stringa a
23 print(a2)
24 a5=a[-mean:] # a5 se sastoji od druge polovine stringa a
25
26 print(a5)
27
28 a3=a[0:(l-1):2]; # neparni elementi stringa a
29
30 a4=a[1:(l-1):2]; # parni elementi stringa a
31
32 print(a3)
33 print(a4)

```