

Sveučilište u Splitu
Prirodoslovno-matematički fakultet

Programiranje 1 2013./2014. Vježba 2.



Programiranje 1 2013./2014.

Vježba 2

Sadržaj

- Print – naredba za ispis
- Varijable i pridruživanje vrijednosti
- Input – naredba za unos
- Elementi dijagrama toka



Print – naredba za ispis podataka

`print(<izraz>)`

Ključna (rezervirana) riječ

Niz znakova (string), Broj,
Aritmetički izraz

➤ Primjeri:

String: sve unutar navodnika će se točno tako i ispisati

```
print("Pozdrav")
```

Broj (jedan ili više brojeva odvojenih zarezom):

```
print(123) ili print(123,78,-25.56)
```

Aritmetički izraz: rezultat izraza će se ispisati

```
print(5*2-3)
```

Kombinacija gore navedenog:

```
print("a+b=",5+6, 11)
```

Ili:

```
print('a+b={}'.format(5+4))
```



Print – naredba za ispis podataka

- Posebni znakovi:
 - \n – prijelaz u novi redak (pritisnuti alt gr i Q za obrnutu kosu crtu)
 - \t – tabulator
 - \\ – ispis kose crte
 - \' – ispis jednostrukog navodnika
 - \” – ispis dvostrukog navodnika
- Ukoliko se ne naglasi **separator**, print naredba svaki član ispisuje s razmakom. Ukoliko članove želimo odvojiti drugačije, pišemo:

```
print("a+b=", 5+6, sep=', ')
```
- **Komentari** u Python programu počinju za znakom „#”:
#ova linija je komentar programera



Print – naredba za ispis podataka

➤ Primjer:

```
>>> print("Pjesma \"Voćka poslije kiše\" je lijepa pjesma.")
```

Pjesma "Voćka poslije kiše" je lijepa pjesma.

➤ Pogledajmo kako će biti ispisan sljedeći znakovni nizovi:

```
>>> 'prvi redak\ndrugi redak\ntreći redak'
```

```
'prvi redak\ndrugi redak\ntreći redak'
```

```
>>> print('prvi redak\ndrugi redak\ntreći redak')
```

prvi red

drugi red

treći red



Print – naredba za ispis podataka

➤ Uporaba tabulatora ilustrirana je sljedećim primjerom:

```
>>> 'jedan\tdva\ttri\nčetiri\tpet\tšest'
```

```
'jedan\tdva\ttri\nčetiri\tpet\tšest'
```

```
>>> print('jedan\tdva\ttri\nčetiri\tpet\tšest')
```

```
jedan dva tri
```

```
četiri pet šest
```



Print – naredba za ispis podataka

➤ Znakovni nizovi ograđeni s tri uzastopna navodna znaka:

U tom se slučaju tekst piše tako na način kako ga želimo ispisati (osim što treba voditi računa da će prvi redak biti pomaknut za tri mjesta ako se tekst počinje pisati iza prompta. Pogledajmo kako to izgleda:

```
>>> '''prvi redak  
drugi redak  
treći redak'''  
'prvi redak\ndrugi redak\ntreći redak'
```

```
>>> print("""prvi redak  
drugi redak  
treći redak""")  
prvi redak  
drugi redak  
treći redak
```



Print – naredba za ispis podataka

- Primjer: prva kitica spominjane pjesme Dobriše Cesarića ispisane na takav način:

```
>>> print("""
-----
VOĆKA POSLIJE KIŠE
Gle malu voćku poslije kiše:
Puna je kapi pa se njiše.
I bliješti suncem obasjana,
Čudesna raskoš njenih grana.
----->>>)
```

```
-----
VOĆKA POSLIJE KIŠE
Gle malu voćku poslije kiše:
Puna je kapi pa se njiše.
I bliješti suncem obasjana,
Čudesna raskoš njenih grana.
-----
```



Print – naredba za ispis podataka

- Obavljanje ovog primjera zahtijevalo je dosta tipkanja.
- Nezgodno je što će sve to što smo napisali netragom nestati kada ugasimo računalo ili samo zatvorimo sučelje.
- Bilo bi svakako korisno kada bismo sve što napišemo mogli pohraniti u datoteku/varijablu na disku/memoriji i kada nam zatreba ponovno dohvatiti posredstvom Python sučelja.



Variable

- Svi računalni programi na neki način obrađuju podatke.
- Podaci se sakupljaju, pohranjuju, obrađuju i mijenjaju tijekom izvođenja programa.
- Pojedinačni podatak nazivat ćemo **vrijednošću** (eng. value).
- Svi podaci koje program koristi moraju biti pohranjeni u memoriji.
- Problem pamćenja vrijednosti riješen je uvođenjem varijabli.
- **Variable** su imena koja će biti pridružena pojedinim vrijednostima.

Nazivi varijabli

- Moraju započeti sa slovom ili znakom _
- Ostali znakovi mogu biti: slova, znamenke ili znak _
- Naziv varijable ne smije sadržavati razmake
- Python razlikuje mala i velika slova
- Naziv varijable ne može biti rezervirana riječ programskog jezika:

A ≠ a
Broj ≠ broj

and	del	for	is	raise
assert	elif	from	lambda	return
break	else	global	not	try
class	except	if	or	while
continue	exec	import	pass	yield
def	finally	in	print	



Pridruživanje vrijednosti

`<varijabla> = <izraz>`

Naziv varijable

Vrijednost ili izraz koji se prvo računa,
a tek zatim pridružuje varijabli

- Pridruživanje se obavlja tako da se najprije izračuna vrijednost izraza s **desne** strane znaka pridruživanja i nakon toga se ta vrijednost pridruži varijabli koja se nalazi s **lijeve** strane znaka pridruživanja.



Pridruživanje vrijednosti

- Nakon što naredbu utipkamo u interaktivnom sučelju i nakon toga pritisnemo tipku Enter, Python neće vratiti nikakvu vrijednost već će ispisati odzivni znak jer je trebao pridružiti vrijednost varijabli, a ne ispisati njenu vrijednost:

```
>>> x = 10
```

```
>>>
```

- Ako želimo ispisati vrijednost varijable x, tada treba utipkati x i Python će ju ispisati:

```
>>> x = 10
```

```
>>> x
```

```
10
```

```
>>>
```



Inicijalizacija varijabli

- Brojčane varijable se inicijaliziraju na vrijednost 0.
- Zašto?

- String varijable se inicijaliziraju na vrijednost "" (prazan string).
- Zašto?

Pridruživanje vrijednosti

- Primjer s preračunavanjem sekundi u minute, sate i dane možemo, napisati na sljedeći način:

```
>>> sekunde = 1000000
```

```
>>> ostatak_sekunde = sekunde % 60
```

```
>>> minute = sekunde // 60
```

```
>>> ostatak_minute = minute % 60
```

```
>>> sati = minute // 60
```

```
>>> ostatak_sati = sati % 24
```

```
>>> dani = sati // 24
```

```
>>> print(dani, ostatak_sati, ostatak_minute, ostatak_sekunde)
```

```
11 13 46 40
```



Pridruživanje vrijednosti

- Znak pridruživanja ne smije se poistovjetiti sa znakom jednakosti u matematici gdje on označava da je ono što stoji lijevo od znaka jednakosti jednako onome desno od znaka jednakosti.
- Prema tome, jednakost $x = x + 1$ u matematici nema nikakvog smisla dok je u Pythonu (a i ostalim programskim jezicima) potpuno smisljena konstrukcija. Tako se, primjerice, može pisati:

```
>>> x = 1000
```

```
>>> x = x + 1
```

```
>>> print(x)
```

```
1001
```



Pridruživanje vrijednosti

- Prema tome, preračunavanje sekundi u minute, sate i dane možemo obaviti i tako da ne uvodimo nove varijable za ostatke dijeljenja, već da pažljivim odabirom redoslijeda izračunavanja prethodnih sekundi, minuta i sati pridružimo nove vrijednosti:

```
>>> sekunde = 1000000
>>> minute = sekunde // 60
>>> sekunde = sekunde % 60
>>> sati = minute // 60
>>> minute = minute % 60
>>> dani = sati // 24
>>> sati = sati % 24
>>> print(dani, sati, minute, sekunde)
11 13 46 40
```



Input – naredba za unos podataka

- Uzima podatak od korisnika i sprema u varijablu
- U Pythonu se unos svodi na pridruživanje vrijednosti
- Izraz s desne strane je sada naredba input

`<varijabla> = input(<poruka>)`

String koji se korisniku ispiše na ekranu prilikom unosa

- Primjer:

```
x = input("Unesite string: ")
```

- Korisniku se ispiše poruka na ekran te se očekuje unos. Kraj unosa označen je unosom tipke Enter.

SVE ŠTO KORISNIK UNESE, PYTHON SMATRA STRINGOM (NIZOM ZNAKOVA).



Input – naredba za unos podataka

1) Unos jednog cijelog broja:

```
broj1 = int(input("Unesi cijeli broj:"))
```

Diagram showing the components of the code:
- **1.** points to the string `"Unesi cijeli broj:"` (prompt).
- **2.** points to the `int()` function (conversion).

2) Unos jednog realnog broja:

```
broj2 = float(input("Unesi realni broj:"))
```

Ako korisnik unese podatak koji nije odgovarajući, program se prekida.

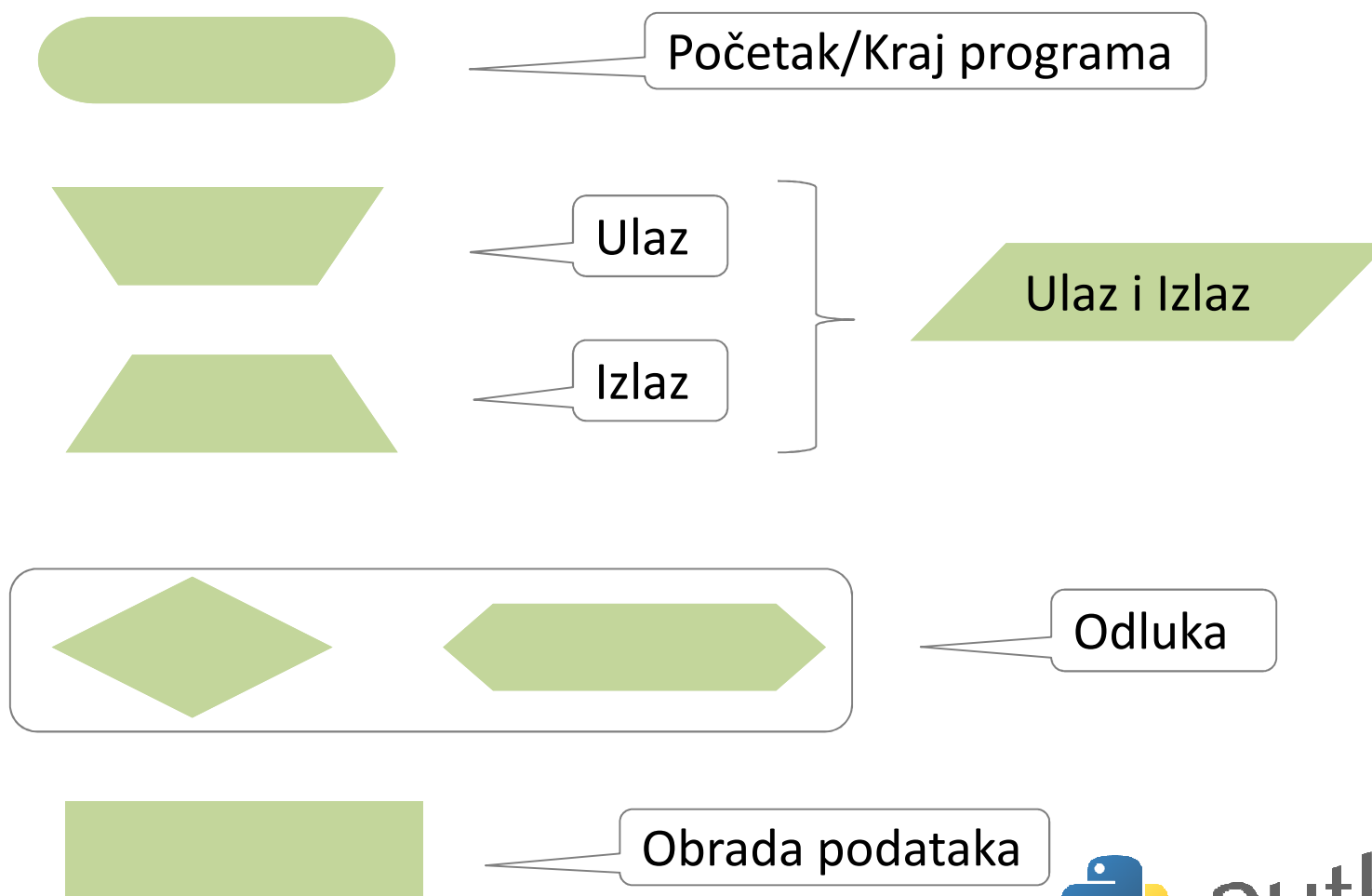
3) Ako želimo da Python raspozna kojeg je tipa uneseni broj:

```
a = eval(input("Unesi nesto: "))
```

- Dolazi od riječi “evaluate” (hrv. procijeni)
- Ne radi sa slovima (stringovima)
- Može prepoznati aritmetički izraz
- Ako unosimo nekoliko brojeva odjednom (odvojenih zarezom), koristimo eval



Elementi dijagrama toka



Programiranje 1 2013./2014.

Vježba 2

Zadatak 2.1.

Napiši program u kojem ćeš varijablama dodijeliti sljedeće vrijednosti i ispisati ih:

100, -1000, a, 3.123, 3.123456789, true, True, 0.000000000001,
9.99999, 1/3, Ivana, 2/5, false, False

Kojeg je tipa pojedina varijabla?



Zadatak 2.2.

Maja ima x kuna. Napiši program koji će izračunati koliko Maja može kupiti čokolade cijene y kuna i koliko će joj nakon kupnje ostati novca.



Programiranje 1 2013./2014.

Vježba 2

Zadatak 2.3.

Napiši program u kojem ćeš izračunati i ispisati površinu kruga radijusa 5cm (dijagram toka i kod).



Zadatak 2.4.

Napiši program koji će zatražiti unos stranice pravokutnika. Za unesenu stranicu izračunati i ispisati opseg i površinu pravokutnika (dijagram toka i kod).



Programiranje 1 2013./2014.

Vježba 2

Zadatak 2.5.

Napiši program koji će zatražiti unos imena i godine rođenja te ispisati:

“Pozdrav, Ana. Rođeni ste 1992. godine i sada imate 21 godinu.”

(dijagram toka i kod)



Programiranje 1 2013./2014.
Vježba 2

Zadaci za rješavanje

Budin, L., Brođanac, P., Markučić, Z., Perić, S. (2012)
Rješavanje problema programiranjem u Pythonu,
Element, Zagreb



4.4. Zadaci za ponavljanje i vježbu, strana 55-60

