

Promenljive, Stringovi, Liste



Zašto Python?

...zato što je:

- besplatan
- lagan za učenje
- moćan
- dobro dokumentovan

- Python se koristi u mnogim sferama primene računara: od zamene za kalkulator pa do kompleksnih korisničkih programa, stvaranja web prezentacija, upravljanje robotima.

- Python je softver otvorenog koda, postoji za više platformi - Windows, Unix, Linux, Mac OS itd

Radno okruženje Python-a

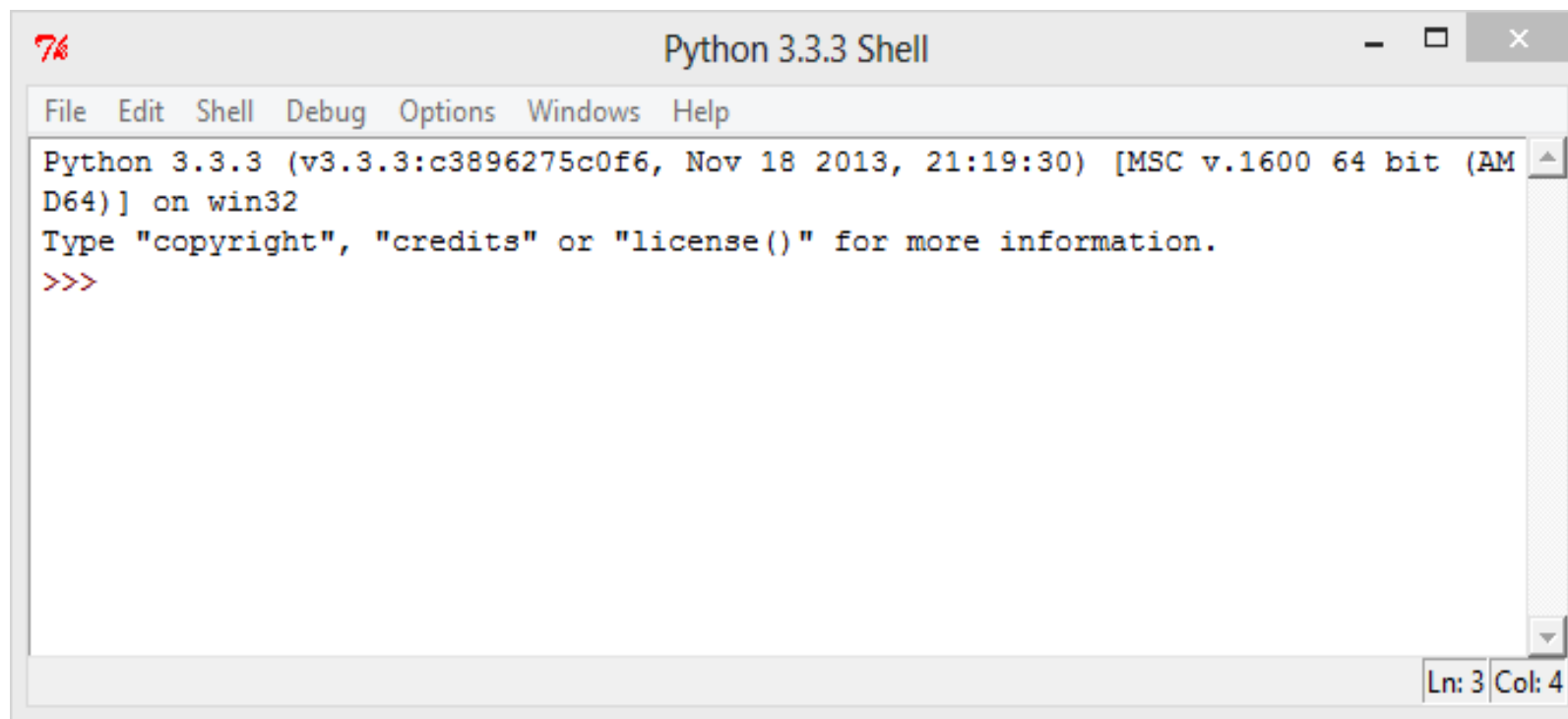
Python možete pokrenuti kliknuvši na "Start"->"Programs"->"Python"->"IDLE (Python GUI)".

>>> (tri strelice) nazivaju se *prompt* i označavaju da je Python spreman za unos naredbe.

Python je interpreterski jezik, što znači da nije moguće da napravite izvršne (.EXE) fajlove vašeg programa, već se program pokreće pomoću interpretera.

Za kreiranje datoteka sa .exe extenzijom može se koristiti py2exe program.

Radno okruženje Python-a

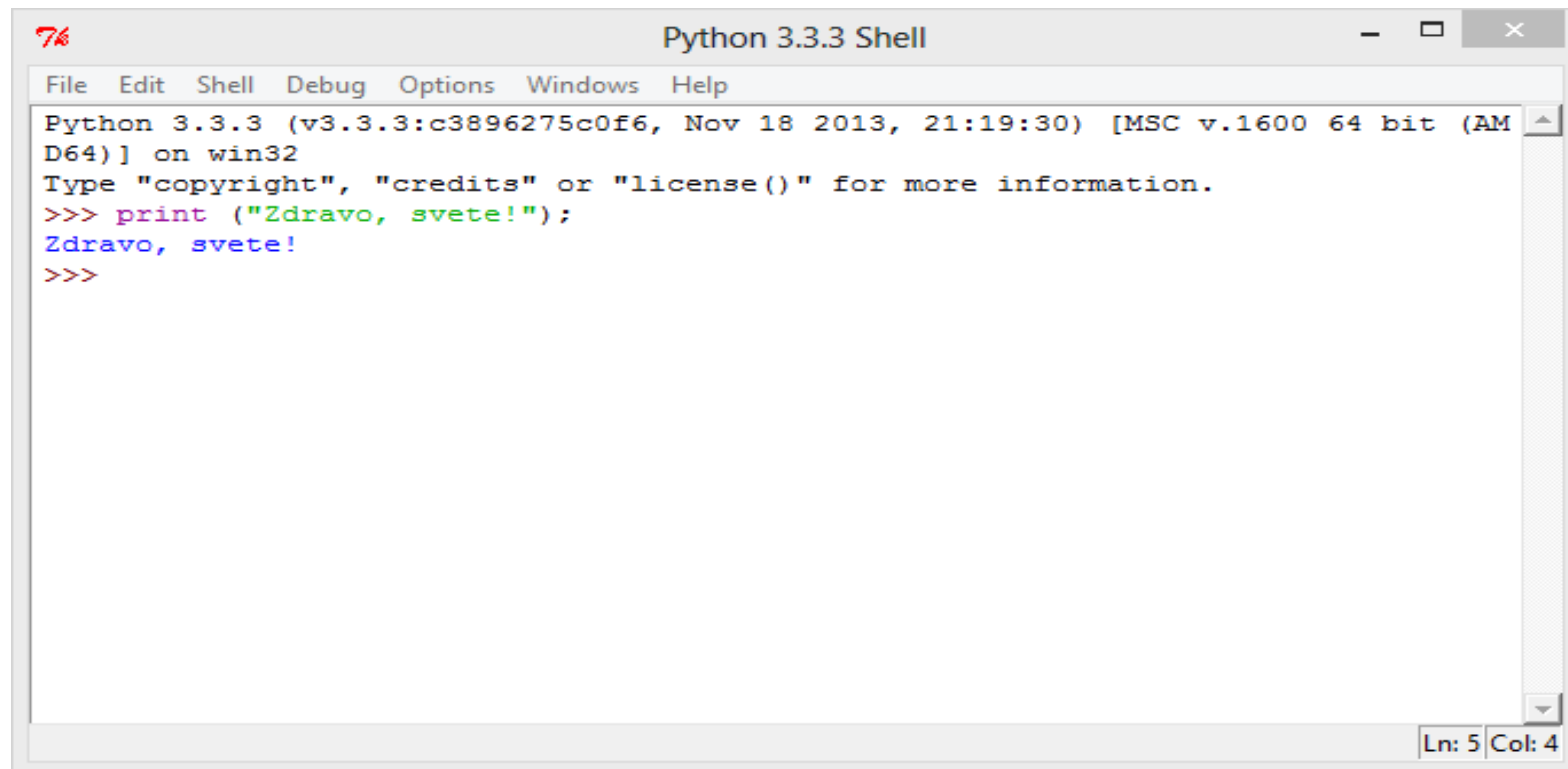
A screenshot of a Windows application window titled "Python 3.3.3 Shell". The window has a standard Windows interface with a title bar, a menu bar (File, Edit, Shell, Debug, Options, Windows, Help), and a scrollable text area. The text area contains the following text: "Python 3.3.3 (v3.3.3:c3896275c0f6, Nov 18 2013, 21:19:30) [MSC v.1600 64 bit (AMD64)] on win32", "Type \"copyright\", \"credits\" or \"license()\" for more information.", and a red prompt ">>>". The status bar at the bottom right shows "Ln: 3 Col: 4".

```
Python 3.3.3 Shell
File Edit Shell Debug Options Windows Help
Python 3.3.3 (v3.3.3:c3896275c0f6, Nov 18 2013, 21:19:30) [MSC v.1600 64 bit (AMD64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
```

Ln: 3 Col: 4

Prvi program - “Zdravo svete”

Postoji tradicija da se pri pisanju uputstva uvek prvo navede program koji ispisuje "Zdravo, svete!" na ekran. U Python-u , ovo je zaista lako :

A screenshot of a Python 3.3.3 Shell window. The window has a title bar with a red icon, the text "Python 3.3.3 Shell", and standard window controls. Below the title bar is a menu bar with "File", "Edit", "Shell", "Debug", "Options", "Windows", and "Help". The main text area contains the following text:

```
Python 3.3.3 (v3.3.3:c3896275c0f6, Nov 18 2013, 21:19:30) [MSC v.1600 64 bit (AMD64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> print ("Zdravo, svete!");
Zdravo, svete!
>>>
```

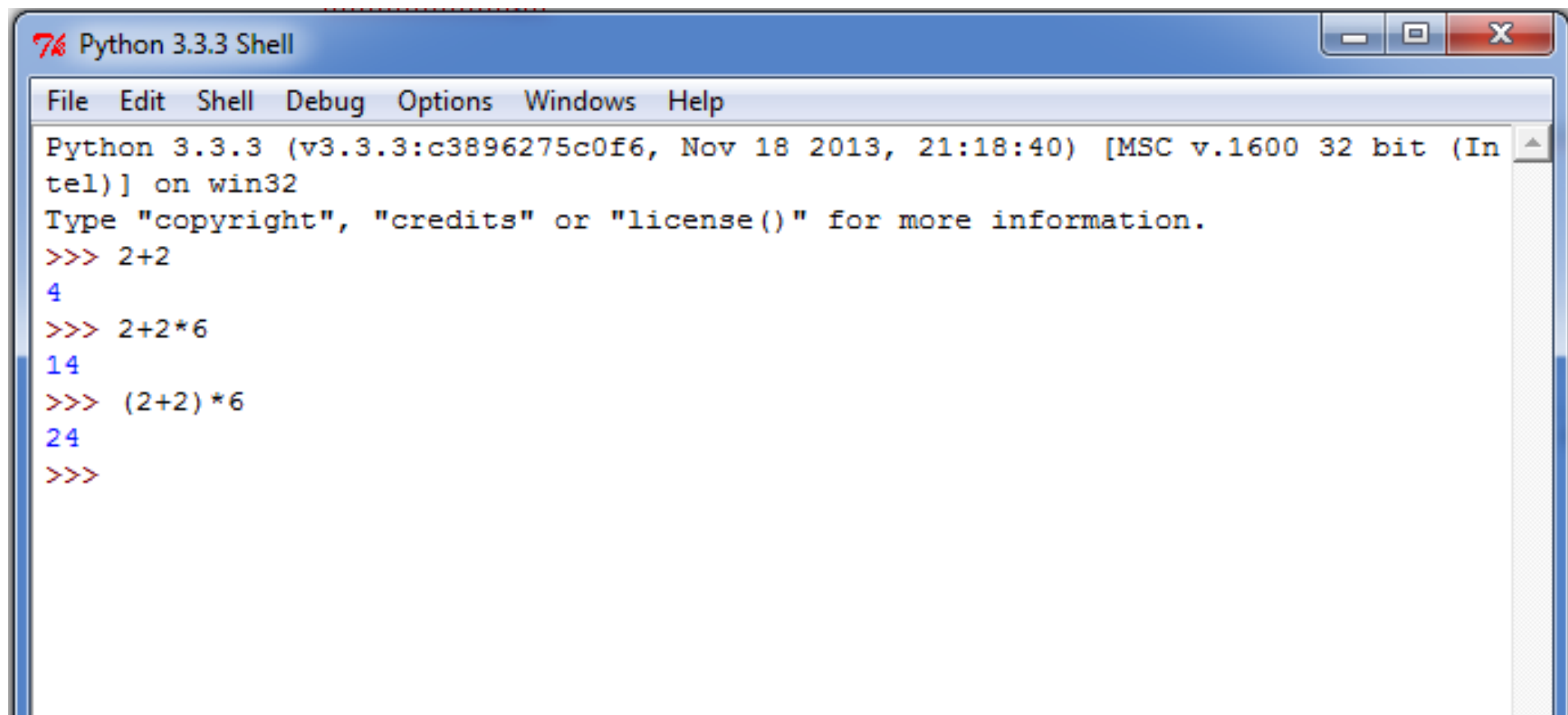
The text is color-coded: "print" is green, the string "Zdravo, svete!" is red, and the prompt ">>>" is blue. A vertical scrollbar is on the right side of the text area. At the bottom right of the window, a status bar shows "Ln: 5 Col: 4".

Osnovne matematičke operacije

Simbol	Operacija
+	Sabiranje
-	Oduzimanje
*	Množenje
/	Deljenje

Kao i u matematici, u programiranju je potrebno voditi računa o prednostima operacija. Operacije množenja i deljenja imaju veći prioritet izvršavanja od sabiranja i oduzimanja. Prilikom pisanja programa najčešće koristimo male zagrade kako bi smo lakše sagledali prioritete.

Osnovne matematičke operacije



The image shows a screenshot of a Python 3.3.3 Shell window. The window has a blue title bar with the text "Python 3.3.3 Shell" and standard window controls (minimize, maximize, close). Below the title bar is a menu bar with the following items: File, Edit, Shell, Debug, Options, Windows, and Help. The main text area displays the following content:

```
Python 3.3.3 (v3.3.3:c3896275c0f6, Nov 18 2013, 21:18:40) [MSC v.1600 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> 2+2
4
>>> 2+2*6
14
>>> (2+2)*6
24
>>>
```

The output shows the results of three arithmetic operations: $2+2=4$, $2+2*6=14$, and $(2+2)*6=24$. The prompt `>>>` is shown at the end of the last line.

Promenljive

Promenljive su imena koja dajemo delovima memorije računara. Imena se koriste da bi čuvala vrednosti.

Promenljive mogu biti brojevi, stringovi ili logičke vrednosti. Stringovi se sastoje od slova i simbola. Logičke promenljive mogu imati jednu od dve vrednosti, ili tačno ili netačno.

Uloga promenljivih je da u svakom trenutku možemo koristiti podatke koje smo direktno upisali u memoriju (konstante) ili podatke koje smo dobili prilikom izvršavanja programa.

Promenljive pozivamo tako što upisemo naziv promenljive u prompt i pritisnemo taster ENTER.

Promenljive

Za kreiranje promenjive koristimo operaciju jednakosti (=).

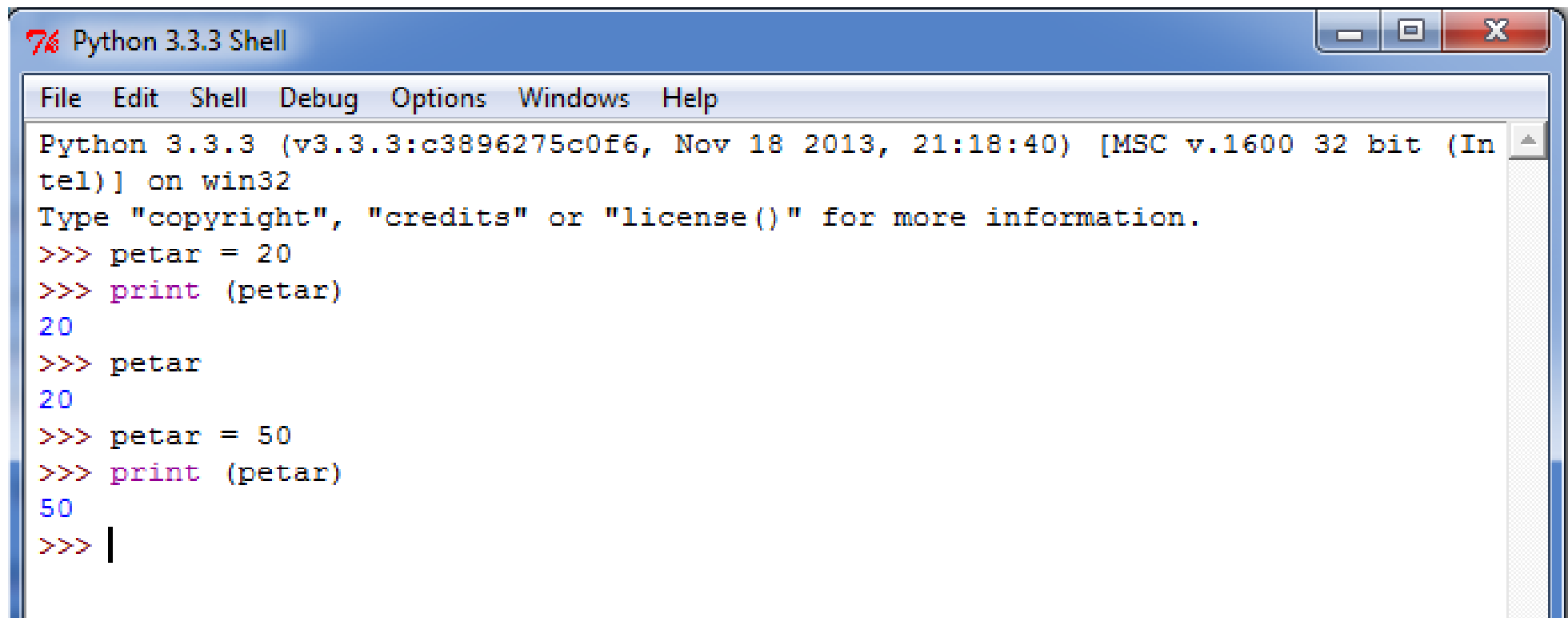
Napravimo promenjivu petar i dodelimo vrednost 20.

```
>>> petar = 20
```

Ova operacija naziva se operacijom dodele dodele. U slučaju petar = 20 memorijskoj lokaciji (koju smo nazvali petar) dodeljujemo vrednost 20.

Na ovaj način povezali smo termin “petar” sa brojevnom vrednošću 20, koju možemo da interpretiramo kao broj bodova koje je ostvario na takmičenju. Na sledećem primeru vidite na koji način dodeljujemo vrednosti promenljivoj kao i način na koji ispisujemo vrednost promenljive.

Promenljive

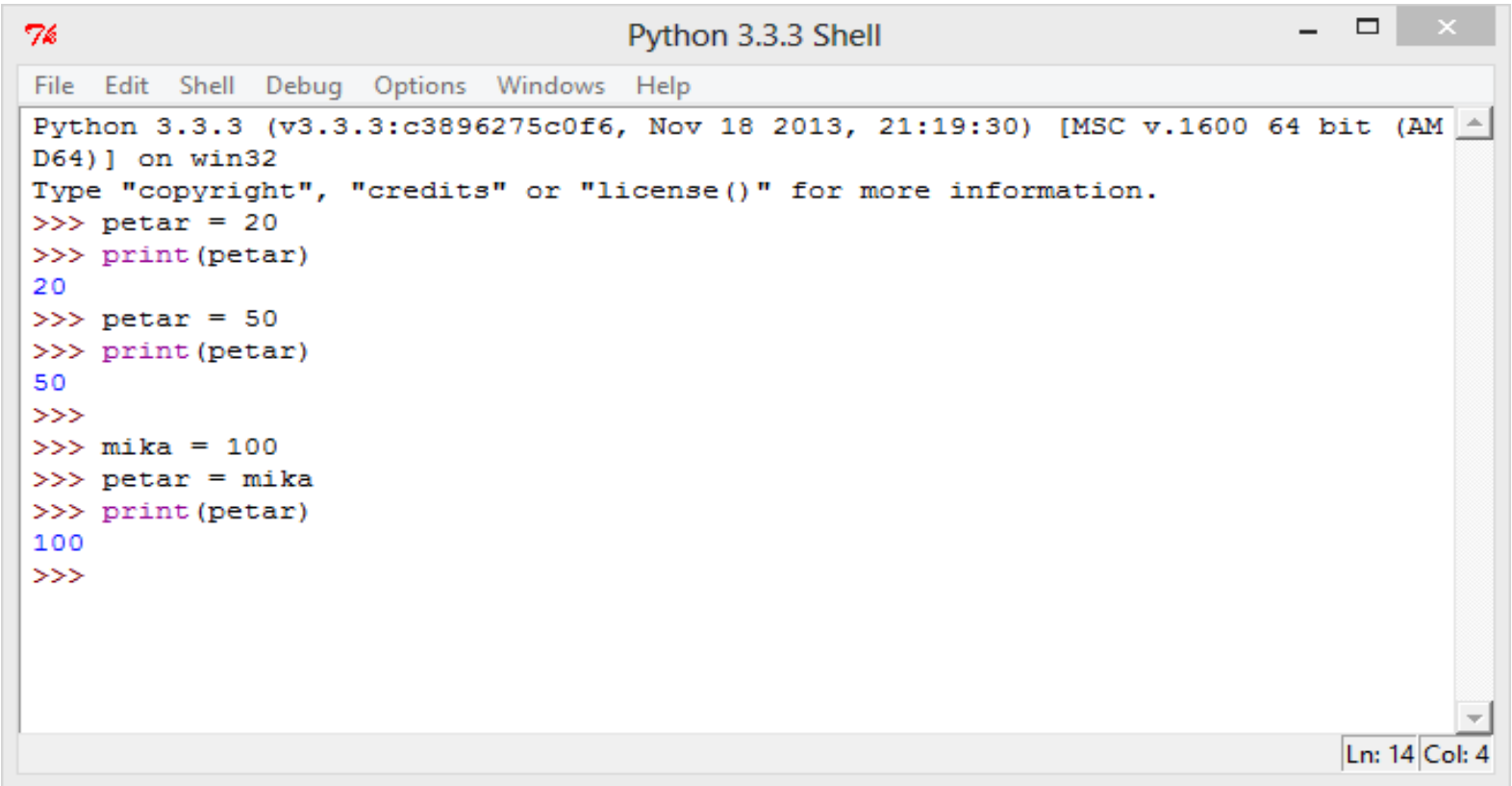


A screenshot of a Python 3.3.3 Shell window. The window has a blue title bar with the text "Python 3.3.3 Shell" and standard window controls (minimize, maximize, close). Below the title bar is a menu bar with options: File, Edit, Shell, Debug, Options, Windows, and Help. The main text area shows the following text:

```
Python 3.3.3 (v3.3.3:c3896275c0f6, Nov 18 2013, 21:18:40) [MSC v.1600 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> petar = 20
>>> print (petar)
20
>>> petar
20
>>> petar = 50
>>> print (petar)
50
>>> |
```

Promenljive i matematičke operacije

Definisanoj promenljivoj osim brojeva, stringova i logičkih izraza, možemo dodeliti i vrednosti drugih promenljivih.

A screenshot of a Python 3.3.3 Shell window. The window has a title bar with a red icon, the text "Python 3.3.3 Shell", and standard window controls. Below the title bar is a menu bar with "File", "Edit", "Shell", "Debug", "Options", "Windows", and "Help". The main text area contains the following code:

```
Python 3.3.3 (v3.3.3:c3896275c0f6, Nov 18 2013, 21:19:30) [MSC v.1600 64 bit (AMD64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> petar = 20
>>> print(petar)
20
>>> petar = 50
>>> print(petar)
50
>>>
>>> mika = 100
>>> petar = mika
>>> print(petar)
100
>>>
```

The status bar at the bottom right shows "Ln: 14 Col: 4".

Upotreba promenljivih

Uzećemo jednostavan primer. Svakog meseca dobijate 20 zlatnika. Tako da za godinu dana sakupite 240 zlatnika ($20 * 12$). Jednom godišnje pozajmite prijatelju 10 zlatnika i na kraju godine vama ostaje 230 zlatnika.

Evo kako bi to zapisali u python-u:

```
>>> 20 * 12
```

```
240
```

```
>>> 240 - 10
```

```
230
```

Sada možemo brojeve prebaciti u promenjive:

```
>>> broj_zlatnika = 20
```

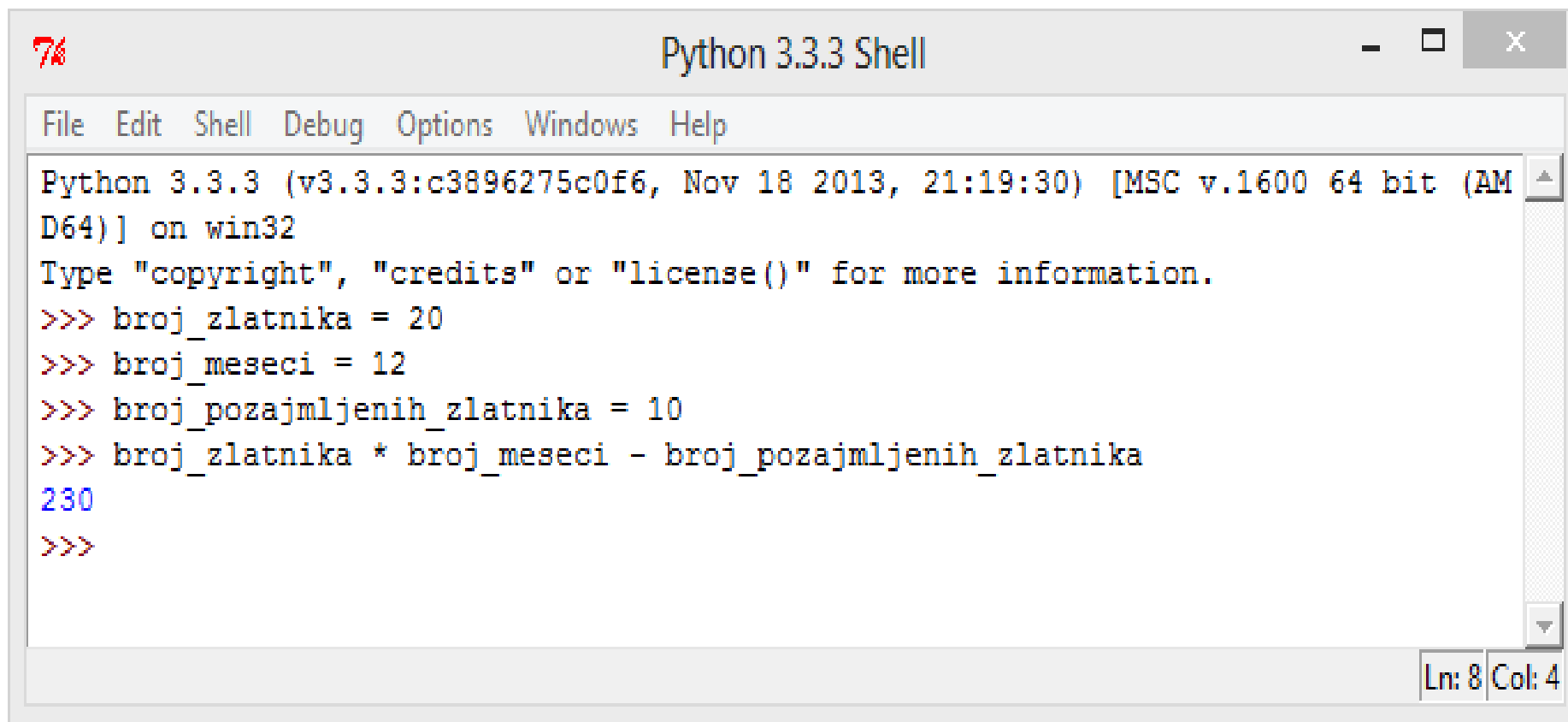
```
>>> broj_meseci = 12
```

```
>>> broj_pozajmljenih_zlatnika = 10
```

```
>>> broj_zlatnika * broj_meseci - broj_pozajmljenih_zlatnika
```

```
230
```

Upotreba promenljivih



```
Python 3.3.3 Shell
File Edit Shell Debug Options Windows Help
Python 3.3.3 (v3.3.3:c3896275c0f6, Nov 18 2013, 21:19:30) [MSC v.1600 64 bit (AMD64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> broj_zlatnika = 20
>>> broj_meseci = 12
>>> broj_pozajmljenih_zlatnika = 10
>>> broj_zlatnika * broj_meseci - broj_pozajmljenih_zlatnika
230
>>>
```

Ln: 8 Col: 4

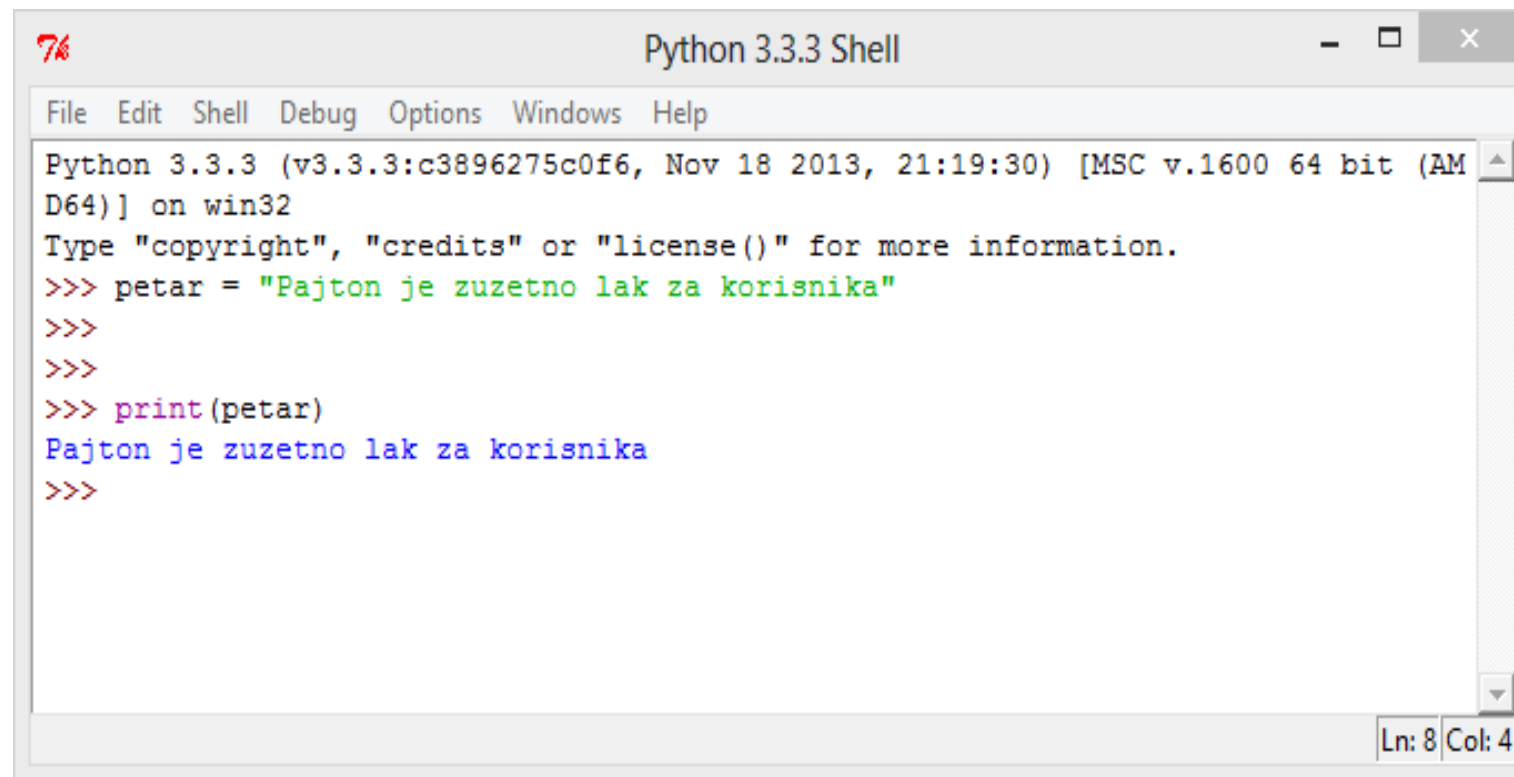
String (tekstualni podaci)

U programiranju tekst nazivamo stringom. String možete zamisliti kao kolekciju znakova. Sva slova, brojevi i simboli mogu biti string, u uvodnom primeru Zdravo svete prikazali smo jedan od načina za prikazivanje stringova.

U Python-u, string kreiramo tatko sto na početku i kraju teksta, koji želimo da bude tipa string, stavljamo dvostruke navodnike.

Za primer uzmimo promenjivu petar i njoj dodelimo sledeću rečenicu "Pajton je izuzetno lak za korisnika"

String - tekstualni podaci



The image shows a screenshot of a Python 3.3.3 Shell window. The window has a title bar that says "Python 3.3.3 Shell" and a menu bar with "File", "Edit", "Shell", "Debug", "Options", "Windows", and "Help". The main text area contains the following code:

```
Python 3.3.3 (v3.3.3:c3896275c0f6, Nov 18 2013, 21:19:30) [MSC v.1600 64 bit (AMD64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> petar = "Pajton je zuzetno lak za korisnika"
>>>
>>>
>>> print(petar)
Pajton je zuzetno lak za korisnika
>>>
```

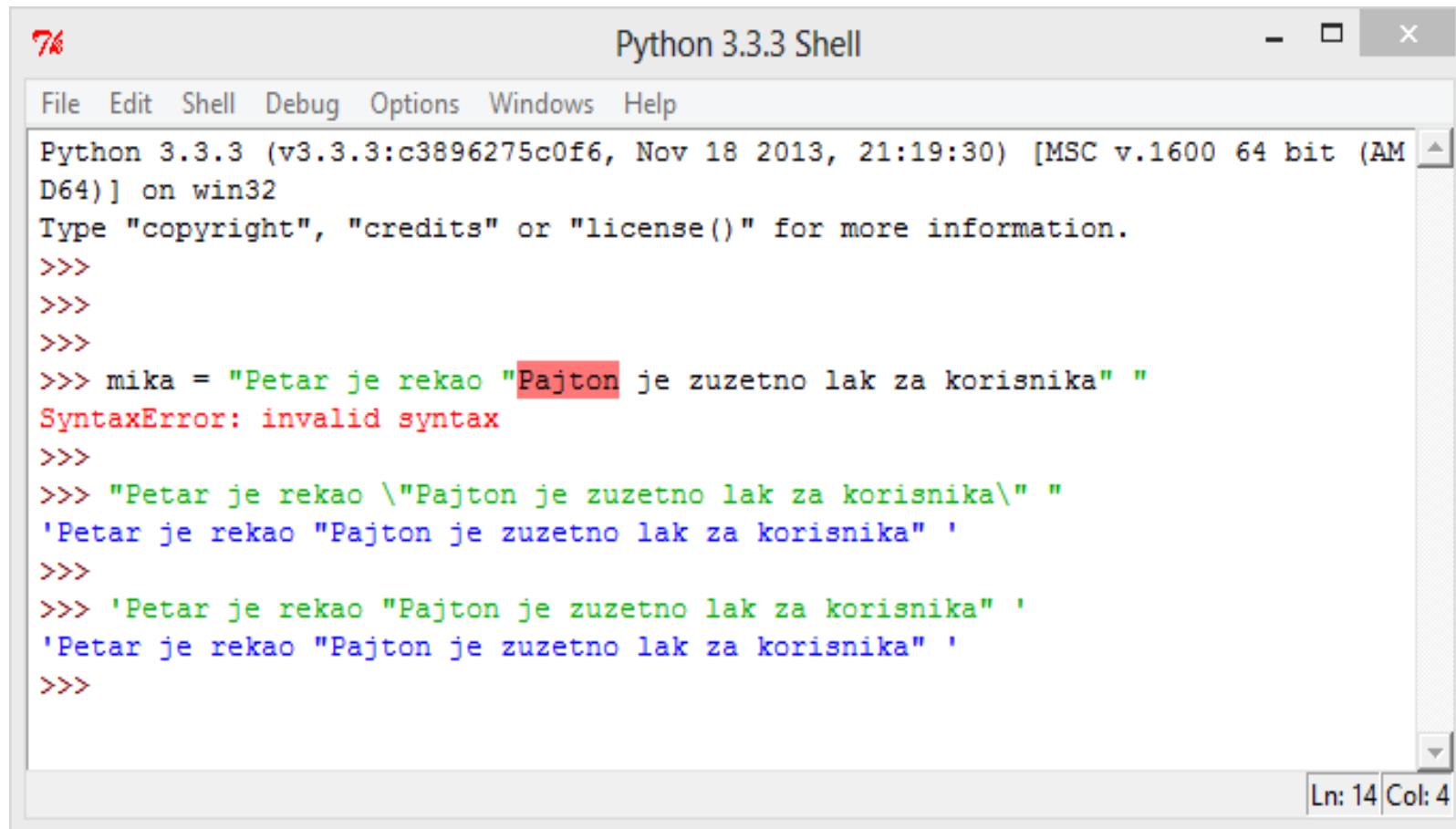
The output of the `print` statement is "Pajton je zuzetno lak za korisnika". The status bar at the bottom right shows "Ln: 8 Col: 4".

String (tekstualni podaci)

Na sledećem primeru možete videti da dodavanjem dvostrukih navodnika u okviru stringa dobijamo grešku. Ako želimo da dodamo dvostruke navodnike u okviru stringa, potrebno je da se ceo string nalazi pod jednostrukim navodnicima.

Postoji i drugo rešenje, a to je dodavanjem *backslash-a* znaka ispred dvostrukih navodnika.

String (tekstualni podaci)

A screenshot of a Python 3.3.3 Shell window. The window has a title bar with a red icon, the text "Python 3.3.3 Shell", and standard window controls. Below the title bar is a menu bar with "File", "Edit", "Shell", "Debug", "Options", "Windows", and "Help". The main text area shows the following text:

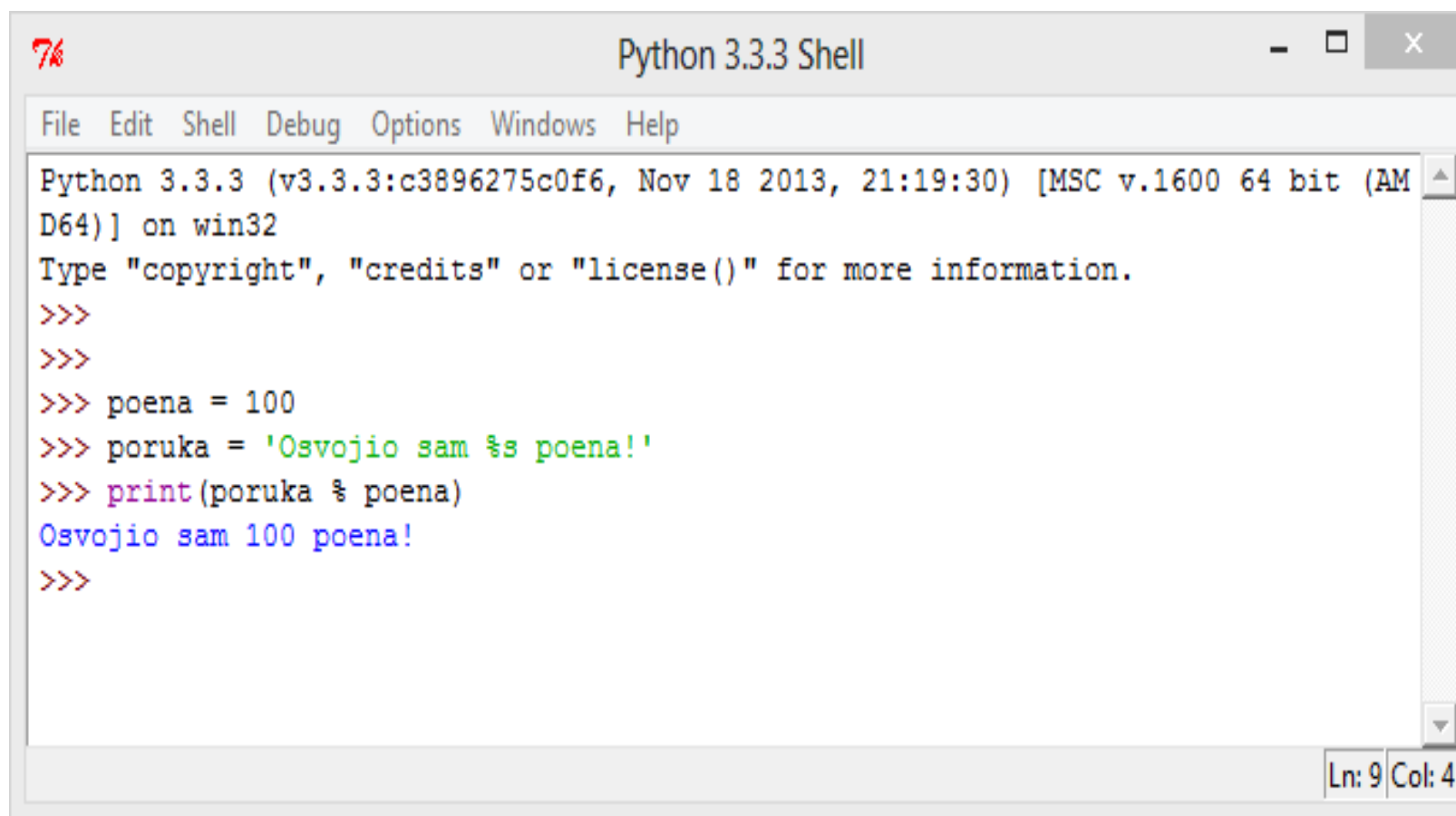
```
Python 3.3.3 (v3.3.3:c3896275c0f6, Nov 18 2013, 21:19:30) [MSC v.1600 64 bit (AMD64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
>>>
>>>
>>> mika = "Petar je rekao "Pajton je zuzetno lak za korisnika" "
SyntaxError: invalid syntax
>>>
>>> "Petar je rekao \"Pajton je zuzetno lak za korisnika\" "
'Petar je rekao "Pajton je zuzetno lak za korisnika" '
>>>
>>> 'Petar je rekao "Pajton je zuzetno lak za korisnika" '
'Petar je rekao "Pajton je zuzetno lak za korisnika" '
>>>
```

The word "Pajton" in the first string is highlighted in red. At the bottom right of the window, a status bar shows "Ln: 14 Col: 4".

```
76 Python 3.3.3 Shell
File Edit Shell Debug Options Windows Help
Python 3.3.3 (v3.3.3:c3896275c0f6, Nov 18 2013, 21:19:30) [MSC v.1600 64 bit (AMD64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
>>>
>>>
>>> mika = "Petar je rekao "Pajton je zuzetno lak za korisnika" "
SyntaxError: invalid syntax
>>>
>>> "Petar je rekao \"Pajton je zuzetno lak za korisnika\" "
'Petar je rekao "Pajton je zuzetno lak za korisnika" '
>>>
>>> 'Petar je rekao "Pajton je zuzetno lak za korisnika" '
'Petar je rekao "Pajton je zuzetno lak za korisnika" '
>>>
Ln: 14 Col: 4
```

String (tekstualni podaci)

Primer: ubacivanje vrednosti promenljivih u string



```
Python 3.3.3 Shell
File Edit Shell Debug Options Windows Help
Python 3.3.3 (v3.3.3:c3896275c0f6, Nov 18 2013, 21:19:30) [MSC v.1600 64 bit (AMD64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
>>>
>>> poena = 100
>>> poruka = 'Osvojio sam %s poena!'
>>> print(poruka % poena)
Osvojio sam 100 poena!
>>>
```

Ln: 9 Col: 4

String (tekstualni podaci)

Množenje stringova

Sta je rezultat množenja 10 i 5? Naravno, odgovor je 50. Ali ako broj 10 pomnožimo sa stringom a dobićemo sasvim drugačije rešenje:

```
>>> print(10 * 'a')  
aaaaaaaaaa
```

Kao što možete videti na ovom primeru, slovo a ispisano je 10 puta.

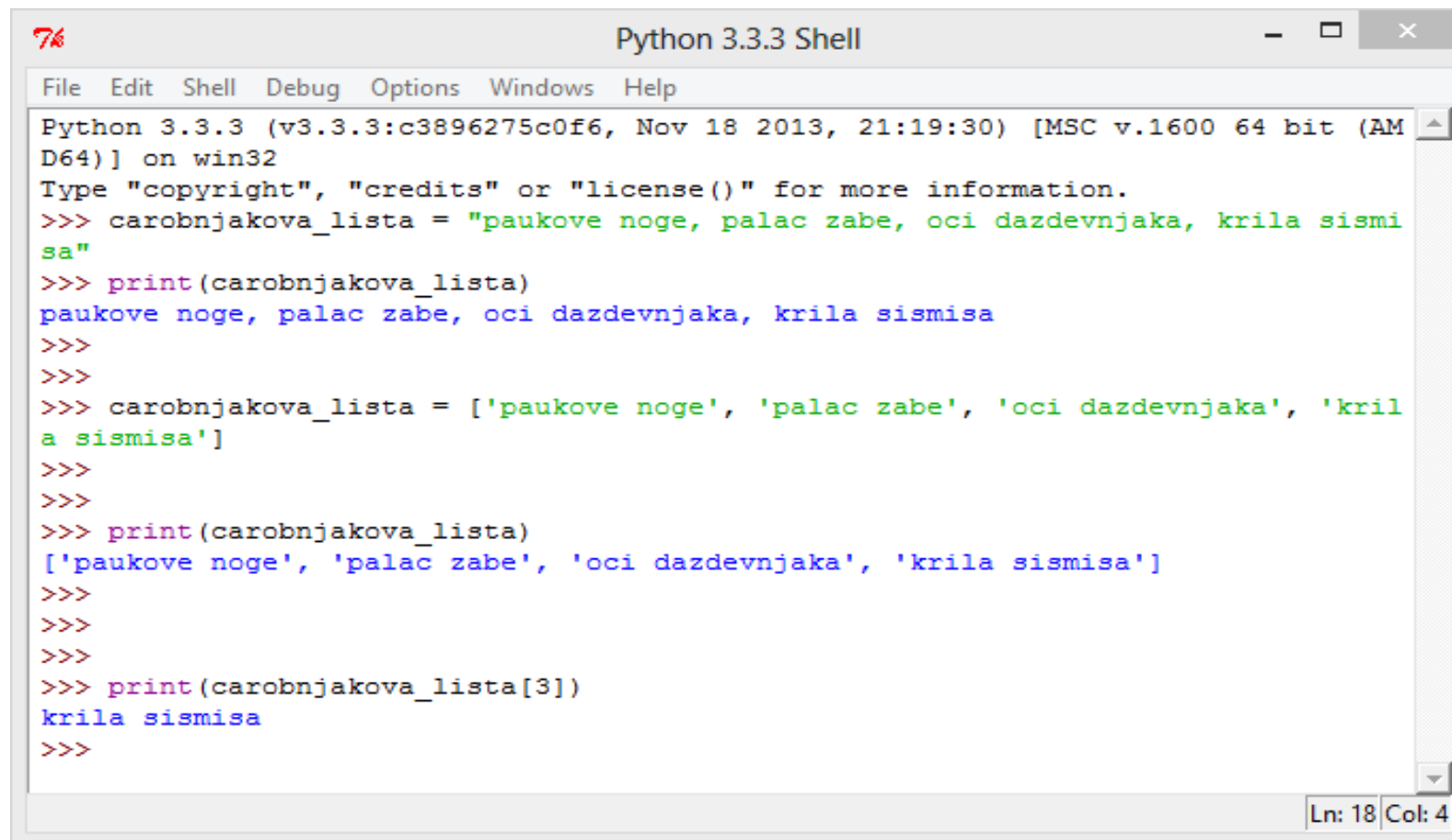
Liste

Lista je poseban objekat Python-a koja može sadržati veći broj tekstualnih podataka. Bitna karakteristika listi je mogućnost manipulacije sadržajem, odnosno prikazivanje samo određene grupe podataka.

Kreiranje liste zahteva malo više kucanja za razliku od string-a, ali lista je korsnija od stringa zbog toga što sa njom možemo upravljati na različite načine (dodavati nove elemente, brisati članove liste..).

Na sledećem primeru uočimo razliku između stringa i liste.

Liste

A screenshot of a Python 3.3.3 Shell window. The window has a title bar with a red icon, the text "Python 3.3.3 Shell", and standard window controls. Below the title bar is a menu bar with "File", "Edit", "Shell", "Debug", "Options", "Windows", and "Help". The main area contains a Python prompt and several lines of code. The code defines a string, prints it, defines a list, prints the list, and prints an element from the list. The status bar at the bottom right shows "Ln: 18 Col: 4".

```
Python 3.3.3 (v3.3.3:c3896275c0f6, Nov 18 2013, 21:19:30) [MSC v.1600 64 bit (AMD64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> carobnjakova_lista = "paukove noge, palac zabe, oci dazdevnjaka, krila sismisa"
>>> print(carobnjakova_lista)
paukove noge, palac zabe, oci dazdevnjaka, krila sismisa
>>>
>>>
>>> carobnjakova_lista = ['paukove noge', 'palac zabe', 'oci dazdevnjaka', 'krila sismisa']
>>>
>>>
>>> print(carobnjakova_lista)
['paukove noge', 'palac zabe', 'oci dazdevnjaka', 'krila sismisa']
>>>
>>>
>>> print(carobnjakova_lista[3])
krila sismisa
>>>
```

Ln: 18 Col: 4

Liste

Korišćenje listi

```
>>> carobnjakova_lista[3] = 'jezik puža'  
['Paukove noge ', 'palac žabe ', 'oči daždevnjaka ', 'jezik puža']
```

Liste se mogu koristiti i za čuvanje brojeva:

```
>>> brojevi = [1, 2, 5, 10, 20]
```

Kao i za kombinaciju brojeva i stringova:

```
>>> brojevi_i_stringovi = ['plava', 'bela', 6, 'crvena', 'siva', 7, 8, 9]
```

Liste mogu da čuvaju i druge liste:

```
>>> brojevi = [1, 2, 5, 10, 20]  
>>> brojevi_i_stringovi = ['plava', 'bela', 6, 'crvena', 'siva', 7, 8, 9]  
>>> nova_lista = [brojevi, brojevi_i_stringovi]  
>>> print(nova_lista)  
[[1, 2, 5, 10, 20], ['plava', 'bela', 6, 'crvena', 'siva', 7, 8, 9]]
```

Liste - operacija dodavanja

Za dodavanje nove stavke u listu koristimo `append` funkciju. Funkcija je komad koda koji govori pajtonu da uradi nešto.

Na primer, ako želimo da dodamo jezik puža u našu listu, potrebno je da uradimo sledeće:

```
>>> carobnjakova_lista.append('jezik puža')
>>> print(carobnjakova_lista)
['Paukove noge ', 'palac žabe ', 'oči daždevnjaka ', 'krila sismisa', 'jezik puža']
```

Liste - operacija brisanja elementa

Za brisanje stavke iz liste koristimo del komandu. Del je skraćénica od delete.

```
>>> del carobnjakova_lista[2]  
>>> print(carobnjakova_lista)  
['Paukove noge ', 'palac žabe ', 'krila sismisa', 'jezik puža']
```

Liste - spajanje dve liste

```
>>> brojevi = [1, 2, 5, 10, 20]
>>> brojevi_i_stringovi = ['plava', 'bela', 6, 'crvena', 'siva', 7, 8, 9]
>>> nova_lista = brojevi + brojevi_i_stringovi
>>> print(nova_lista)
[1, 2, 5, 10, 20, 'plava', 'bela', 6, 'crvena', 'siva', 7, 8, 9]
```
