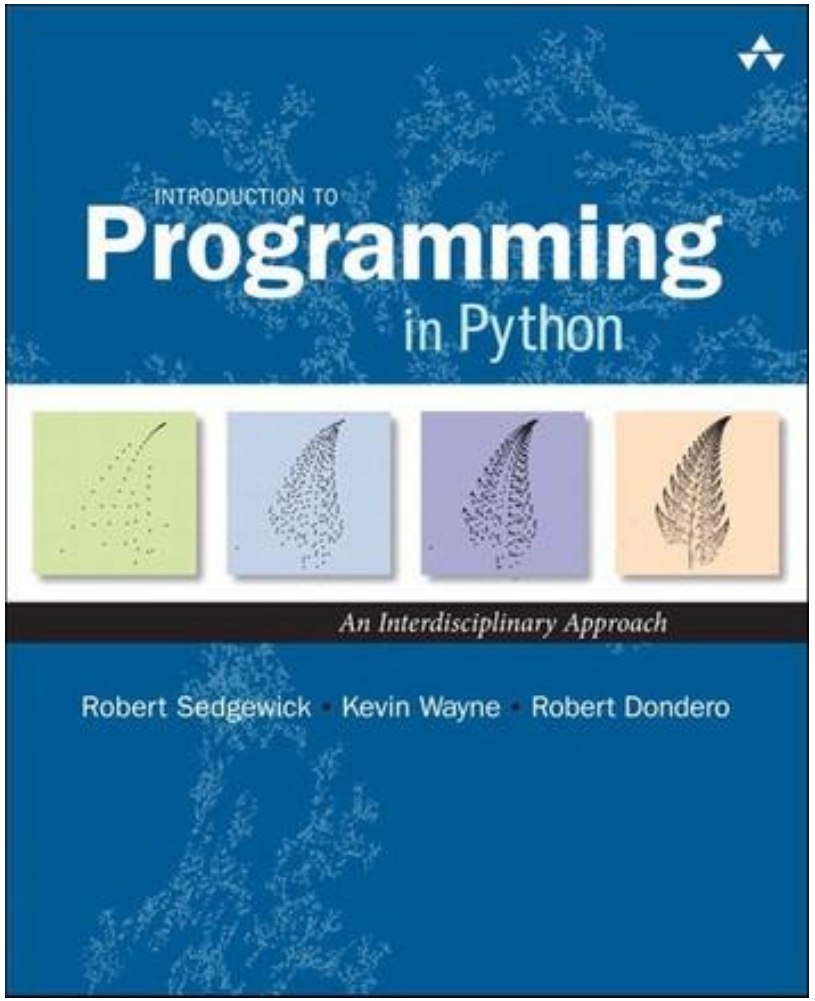
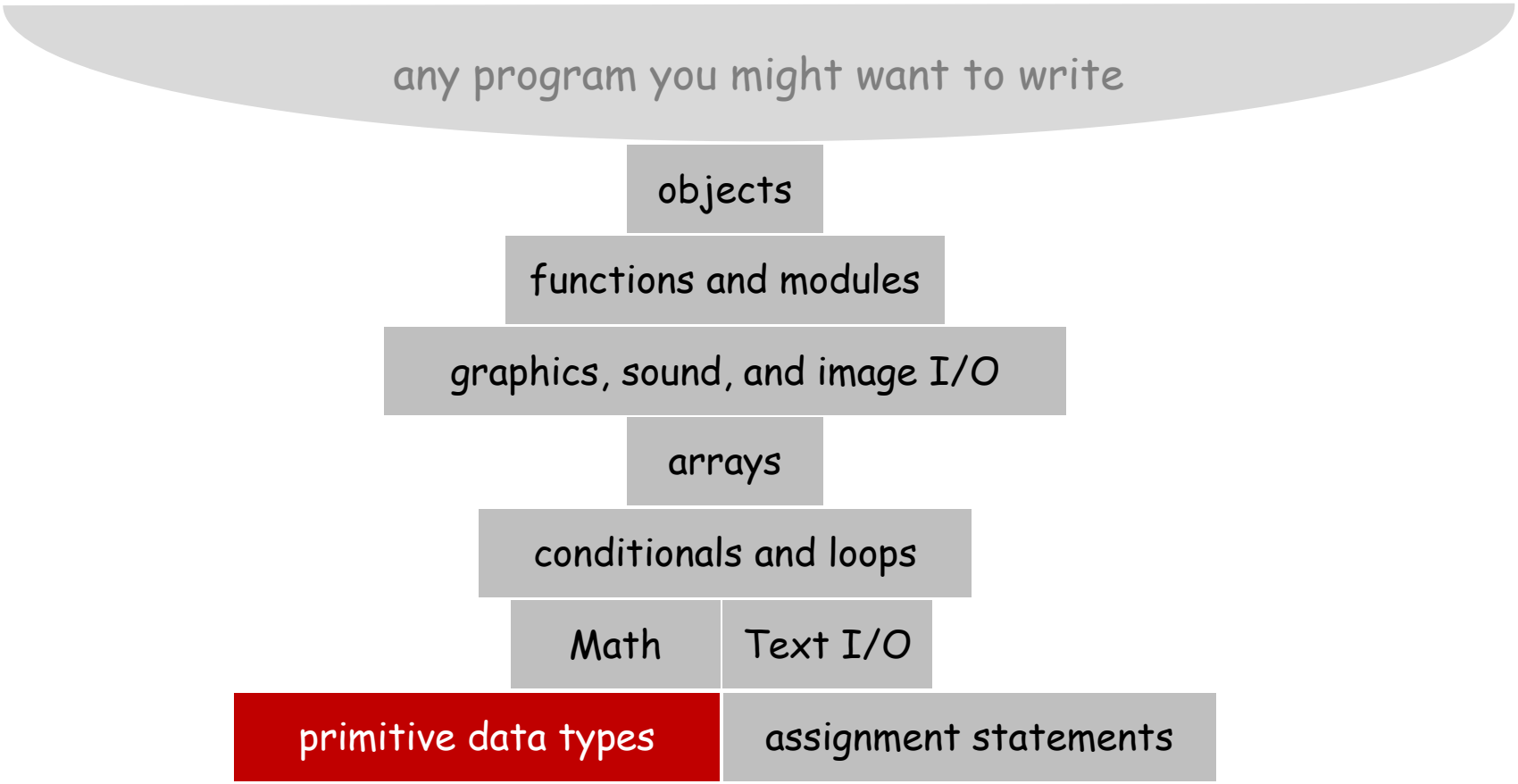
انواع داده‌ای پیش‌ساخته

سید ناصر رضوی **www.snrazavi.ir**
۱۳۹۸

۱-۲ انواع داده‌ای پیش‌ساخته



اجزای برنامه‌نویسی



انواع داده‌ای پیش‌ساخته

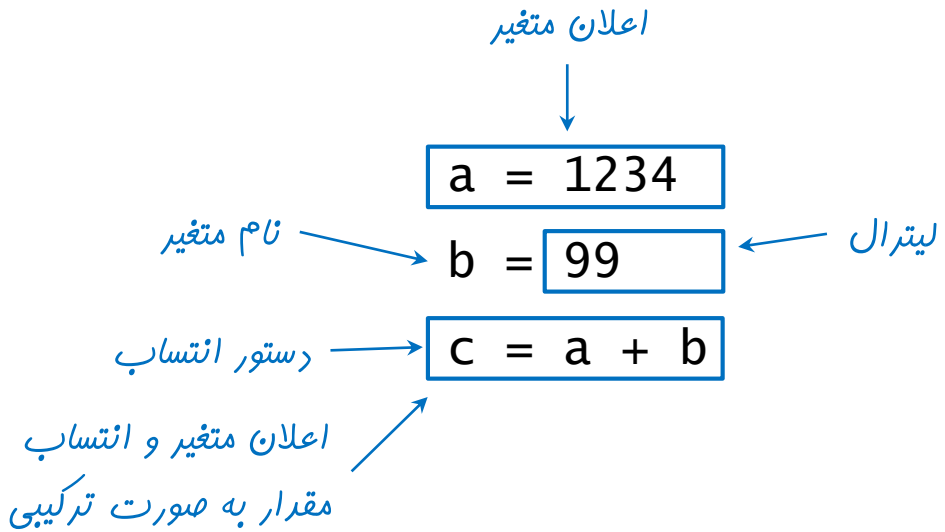
- نوع داده‌ای. یک مجموعه از مقادیر به همراه عملیات تعریف شده بر روی آن مقادیر.

type	*set of values*	*common operators*	*sample literals*
int	integers	+ - * // % **	99 12 2147483647
float	floating point numbers	+ - * / **	3.14 2.5 6.022e23
bool	true-false values	and or not	True False
str	sequence of characters	+	'AB' 'Hello' '2.5'

برنامه = داده + الگوریتم

تعاریف اولیه

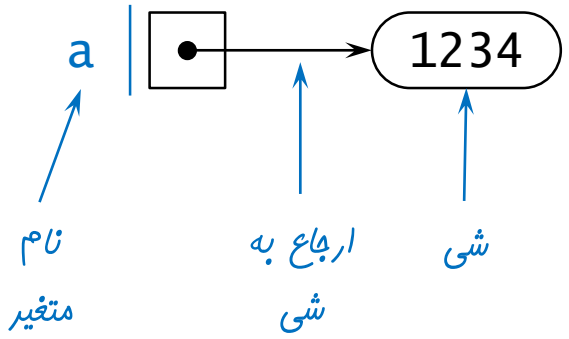
- اشیا. تمام مقادیر داده‌ای در یک برنامه پایتون به وسیله اشیا و روابط میان اشیا بازنمایی می‌شوند.
  - یک شی، یک بازنمایی درون-حافظه‌ای از یک مقدار از یک نوع داده‌ای خاص است.
    - مکان در حافظه
    - نوع داده‌ای: بیانگر رفتار شی (مقادیر و عملیات)
    - مقدار



تعاریف اولیه

- متغیر. نامی برای ارجاع به یک شی.
- دستور انتساب. مقید کردن متغیر سمت چپ به شی ایجاد شده در سمت راست عملگر انتساب.

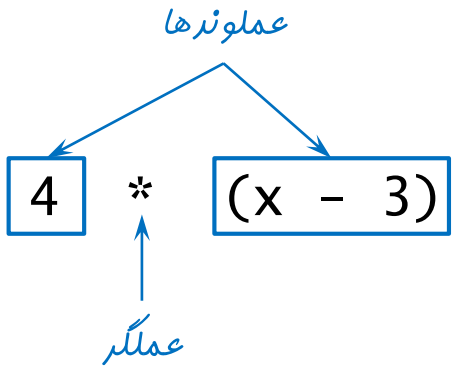
```
a = 1234
```



متغیر به یک شی ارجاع می‌کند

عبارت

- عبارت. ترکیبی از لیترال‌ها، متغیرها و عملگرها که پایتون آنها را برای تولید یک شی ارزیابی می‌کند.



ساختار یک عبارت

ردیابی (غیر رسمی)

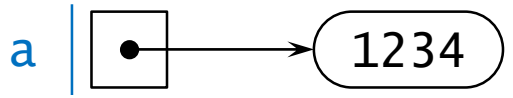
- ردیابی. یک جدول شامل مقادیر متغیرها پس از اجرای هر دستور.
  - به منظور دنبال کردن مقادیر نسبت داده شده به متغیرها در طول اجرای برنامه

	a	b	c
a = 1234	1234		
b = 99	1234	99	
c = a + b	1234	99	1333

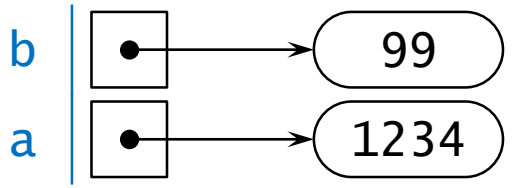
ردیابی (در سطح شی)

- ردیابی در سطح شی. دنبال کردن اشیا و ارجاع‌ها برای یک درک بهتر.

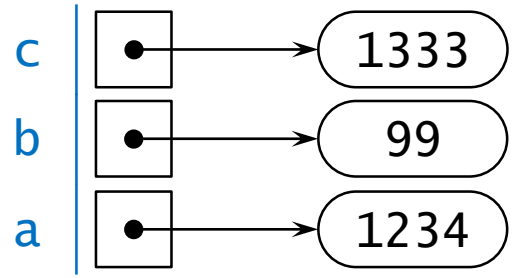
```
a = 1234
```




```
b = 99
```




```
c = a + b
```



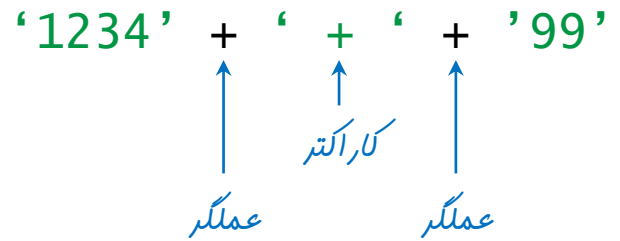
کار با داده‌های متنی

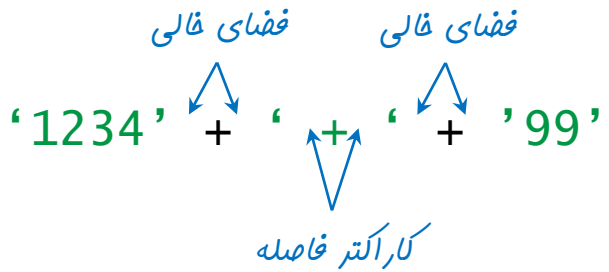
- نوع داده‌ای رشته. مفید برای عملیات ورودی و خروجی برنامه و پردازش متن.

values	sequence of characters
typical literals	‘Hello, world’
operation	concatenation
operator	+

expression	*value*
‘Hi, ‘ + ‘Bob’	‘Hi, Bob’
‘1’ + ‘ 2 ‘ + ‘1’	‘1 2 1’
‘1234’ + ‘ + ‘ + ’99’	‘1234 + 99’
‘1234’ + ’99’	‘123499’

هشدار. معنای کاراکترها به محل استفاده آنها بستگی دارد.





مثال: بخش‌بندی‌های یک خط کش


```
ruler1 = ‘1’
ruler2 = ruler1 + ‘ 2 ’ + ruler1
ruler3 = ruler2 + ‘ 3 ’ + ruler2
ruler4 = ruler3 + ‘ 4 ’ + ruler3

print(ruler1)
print(ruler2)
print(ruler3)
print(ruler4)
```

اتصال رشته‌ای

```
% python ruler.py
1
1 2 1
1 2 1 3 1 2 1
1 2 1 3 1 2 1 4 1 2 1 3 1 2 1
```

```
"1"
"1 2 1"
"1 2 1 3 1 2 1"
"1 2 1 3 1 2 1 4 1 2 1 3 1 2 1"
```

تبدیل نوع

- تبدیل اعداد به رشته برای عملیات خروجی.

```
print(str(a) + ' + ' + str(b) + ' = ' + str(a + b))
```

```
1234 + 99 = 1333
```

- تبدیل رشته به اعداد برای عملیات ورودی.

```
int('1234')
```

```
float('3.14')
```

دریافت ورودی از خط فرمان


```
import sys


print('Hi, ' + sys.argv[1] + '.')
print('How are you?' )
```

sys.argv[0] sys.argv[1]

```
% python userargument.py Bob
Hi, Bob.
How are you?
```

دریافت ورودی از خط فرمان و تبدیل نوع


```
import sys


a = int(sys.argv[1])
b = int(sys.argv[2])

print(str(a) + ' + ' + str(b) + ' = ' + str(a + b))
```

```
% python userargument2.py 1234 99
1234 + 99 = 1333
```

اعداد صحیح

- نوع داده‌ای عدد صحیح. مفید برای بیان الگوریتم‌ها.

values	integers						
typical literals			1234	99	-99	0	1000000
operations	sign	add	subtract	multiply	division	remainder	power
operators	+ -	+	-	*	//	%	**

اعداد صحیح

- نوع داده‌ای عدد صحیح. مفید برای بیان الگوریتم‌ها.

expression	*value*	*comment*
5 + 3	8	جمع
5 - 3	2	تفریق
5 * 3	15	ضرب
5 // 3	1	تقسیم (بدون قسمت اعشاری)
5 % 3	2	باقیمانده
5 ** 3	125	توان‌رسانی
1 // 0	-	خطای زمان اجرا

مثال: عملیات بر روی اعداد صحیح


```
import sys

a = int(sys.argv[1])
b = int(sys.argv[2])

total = a + b
diff  = a - b
prod  = a * b
quot  = a // b
rem   = a % b
exp   = a ** b

print(str(a) + ' +  ' + str(b) + ' = ' + str(total))
print(str(a) + ' -  ' + str(b) + ' = ' + str(diff))
print(str(a) + ' *  ' + str(b) + ' = ' + str(prod))
print(str(a) + ' // ' + str(b) + ' = ' + str(quot))
print(str(a) + ' %  ' + str(b) + ' = ' + str(rem))
print(str(a) + ' ** ' + str(b) + ' = ' + str(exp))
```

```
% python intops.py 1234 99
1234 +  99 = 1333
1234 -  99 = 1135
1234 *  99 = 122166
1234 // 99 = 12
1234 %  99 = 46
1234 ** 99 = یک عدد با ۳۰۷ رقم
```

1234 = 12 * 99 + 46

اعداد اعشاری

- نوع داده‌ای **`float`**. مفید در محاسبات علمی و کاربردهای تجاری.

values	real numbers				
typical literals	3.14159	6.022e23	-3.0	1.414213562373095	
operations	add	subtract	multiply	division	exponentiation
operators	+	-	*	/	**

مثال: عملیات بر روی اعداد اعشاری


```
import sys

a = float(sys.argv[1])
b = float(sys.argv[2])

total = a + b
diff  = a - b
prod  = a * b
quot  = a // b
exp   = a ** b

print(str(a) + ' +  ' + str(b) + ' = ' + str(total))
print(str(a) + ' -  ' + str(b) + ' = ' + str(diff))
print(str(a) + ' *  ' + str(b) + ' = ' + str(prod))
print(str(a) + ' // ' + str(b) + ' = ' + str(quot))
print(str(a) + ' ** ' + str(b) + ' = ' + str(exp))
```

```
% python floatops.py 123.456 78.9
123.456 +  78.9 = 202.356
123.456 -  78.9 = 44.556
123.456 *  78.9 = 9740.6784
123.456 // 78.9 = 1.5647148289
123.456 ** 78.9 = 1.0478827916 7e165
```

مثال: عملیات بر روی اعداد اعشاری (معادله درجه دوم)


```
import sys
import math

b = float(sys.argv[1])
c = float(sys.argv[2])

discriminant = b*b - 4.0*c
d = math.sqrt(discriminant)

print((-b + d) / 2.0)
print((-b - d) / 2.0)
```

```
% python quadratic.py -3.0 2.0
2.0
1.0
```

$x^2 - 3x + 2 = 0$

```
% python quadratic.py -1.0 -1.0
1.6180339887498895
-0.6180339887498949
```

$x^2 - x - 1 = 0$

```
% python quadratic.py 1.0 1.0
ValueError: math domain error
```

$x^2 + x + 1 = 0$

کتابخانه math در پایتون


```
In [1]: import math
```

```
In [2]: dir(math)
```

یک دستور بسیار مفید به منظور کسب اطلاعات اولیه در مورد کتابخانه‌ها

```
['__doc__',
 '__loader__',
 '__name__',
 '__package__',
 '__spec__',
 'acos',
 'acosh',
 'asin',
 'asinh',
 'atan',
 'atan2',
 'atanh',
 'ceil',
 'copysign',
 'cos',
 'cosh',
```

- کتابخانه math
  - توابع متداول ریاضی
  - لگاریتم و توان‌رسانی
  - توابع مثلثاتی

مقادیر بولی

- نوع داده‌ای bool. مفید برای کنترل منطق و روند اجرای برنامه.

values	true or false
literals	`True` `False`
operations	`and` `or` `not`
operators	`and` `or` `not`

a	not a
False	True
True	False

a	b	a and b	a or b
False	False	False	False
False	True	False	True
True	False	False	True
True	True	True	True

عملگرهای مقایسه‌ای

- عملگرهای مقایسه‌ای. عملوندهایی از یک نوع را دریافت و یک عملوند از نوع بولی تولید می‌کنند.

op	*meaning*	`true`	`false`
==	*equal*	`2 == 2`	`2 == 3`
!=	*not equal*	`3 != 2`	`2 != 2`
<	*less than*	`2 < 13`	`2 < 2`
<=	*less than or equal*	`2 <= 2`	`3 <= 2`
>	*greater than*	`13 > 2`	`2 > 13`
>=	*greater than or equal*	`3 >= 2`	`2 >= 3`

مثال: سال کبیسه

- پرسش. آیا یک سال داده شده، یک سال کبیسه است؟
  - بله اگر (۱) بر ۴ بخش‌پذیر باشد و بر ۱۰۰ بخش‌پذیر نباشد یا (۲) بر ۴۰۰ بخش‌پذیر باشد.

```
import sys

year = int(sys.argv[1])

is_leap_year = (year % 4 == 0)
is_leap_year = is_leap_year and (year % 100 != 0)
is_leap_year = is_leap_year or  (year % 400 == 0)

print(is_leap_year)
```

```
% python leapyear.py 2016
True

% python leapyear.py 1900
False

% python leapyear.py 2000
True
```

توابع و APIها

- توابع پیش‌ساخته.
  - در هر برنامه‌ای به صورت مستقیم قابل استفاده هستند.

```
print()   str()   int()   float()   abs()   max()   min()
```

- توابع استاندارد.
  - توابع تعریف شده در ماجول‌های استاندارد پایتون
  - قابل استفاده در هر برنامه‌ای که آن ماجول را `import` می‌کند.

```
math.sqrt()   math.log()   math.exp()   math.sin()

random.random()   random.randrange(x, y)   random.randint()
```

توابع و APIها

- توابع تعریف شده به وسیله کاربر.
  - مانند توابع تعریف شده در کتابخانه‌های مربوط به کتاب

```
stdio.write()    stdio.writeln()

stddraw.point()    stddraw.line()
```

- تعریف توابع.
  - در فصل دوم یاد خواهید گرفت چگونه توابع مورد نیاز خود را تعریف و استفاده نمایید.

تبدیل نوع

- تبدیل نوع. تبدیل از یک نوع به نوع دیگر.
  - تبدیل نوع صریح: با استفاده تبدیل نوع یا توابع

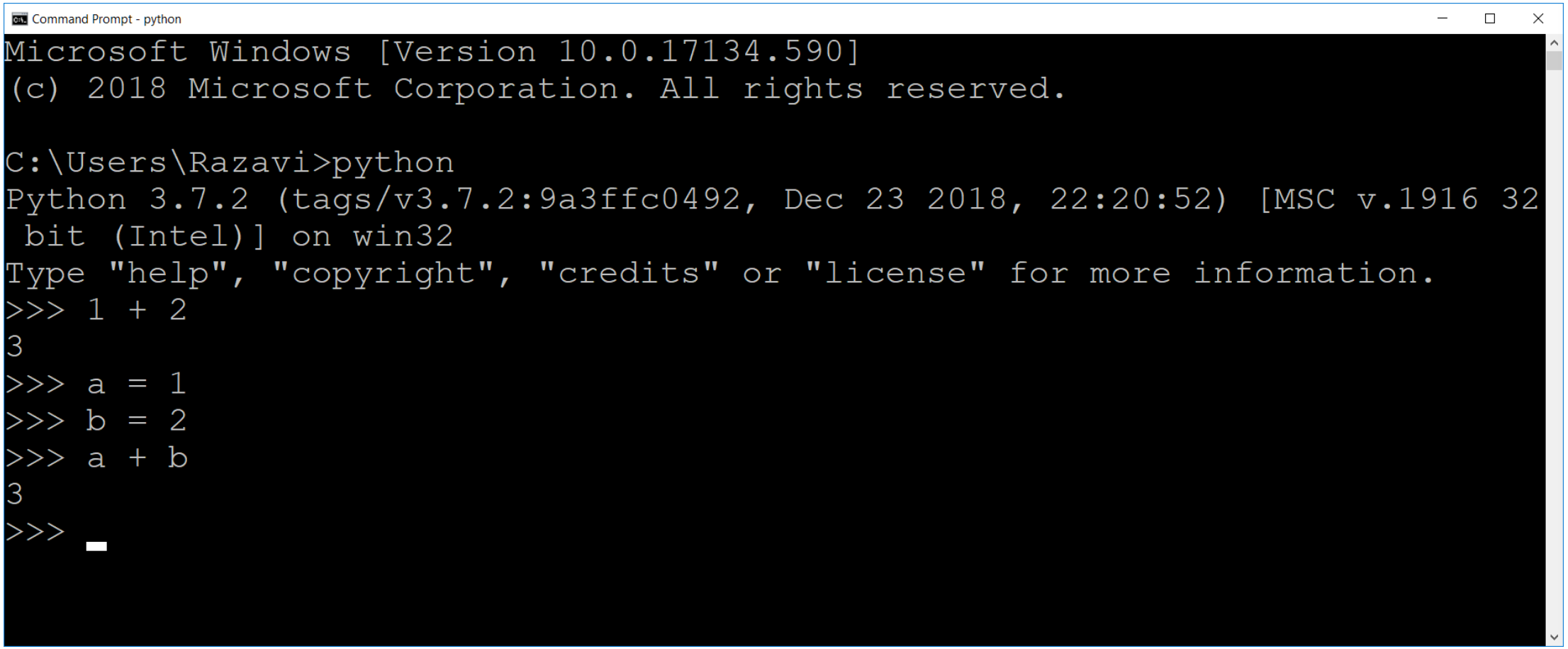
function call	*description*
`str(x)`	*تبدیل شی* **X** *به یک رشته*
`int(x)`	*تبدیل رشته یا عدد اعشاری* **X** *به یک عدد صحیح*
`float(x)`	*تبدیل رشته یا عدد صحیح* **X** *به یک عدد اعشاری*
`round(x)`	*نزدیک‌ترین عدد صحیح به عدد* **X**

  - تبدیل نوع ضمنی (خودکار).

```
x = 10 / 4.0
```

عدد صحیح ۱۰ به صورت خودکار به یک عدد اعشاری تبدیل می‌شود

پایتون به صورت تعاملی: آزمایش سریع دستورات و دستیابی به مستندات



پایتون به صورت تعاملی: آزمایش سریع دستورات و دستیابی به مستندات