Python From Scratch

Python Data Types & Python Numbers & Python Casting

Lesson 3 Content

Python Data Types

- Built-in Data Types
- Getting the Data Type
- Setting the Data Type
- Setting the Specific Data Type
- Test Yourself With Exercises

Python Numbers

- Numeric types in Python
  - int
  - float
  - Complex
- Type Conversion
- Random Number
- Test Yourself with Exercises

Python Casting

- Specify a Variable Type
- Constructor functions
  - int()
  - float()
  - str()

Python Data Types

Built-in Data Types

In programming, data type is an important concept.

Variables can store data of different types, and different types can do different things.

Python has the following data types built-in by default, in these categories:

Text Type:	str	Set Types:	set, frozenset
Numeric Types:	int, float, complex	Boolean Type:	bool
Sequence Types:	list, tuple, range	Binary Types:	bytes, bytearray, memoryview
Mapping Type:	dict	None Type:	NoneType

Getting the Data Type

You can get the data type of any object by using the type() function:

Example

Print the data type of the variable x:

```
x = 5
print(type(x))
```

Setting the Data Type

In Python, the data type is set when you assign a value to a variable:

Example	Data Type	Example	Data Type
x = "Hello World"	**str**	x = {"apple", "banana", "cherry"}	**set**
x = 20	**int**	x = frozenset({"apple", "banana", "cherry"})	**frozenset**
x = 20.5	**float**	x = True	**bool**
x = 1j	**complex**	x = b"Hello"	**bytes**
x = ["apple", "banana", "cherry"]	**list**	x = bytearray(5)	**bytearray**
x = ("apple", "banana", "cherry")	**tuple**	x = memoryview(bytes(5))	**memoryview**
x = range(6)	**range**	x = None	**NoneType**
x = {"name" : "John", "age" : 36}	**dict**		

Setting the Specific Data Type

If you want to specify the data type, you can use the following constructor functions:

Example	Data Type	Example	Data Type
x = str("Hello World")	**str**	x = dict(name="John", age=36)	**dict**
x = int(20)	**int**	x = set(("apple", "banana", "cherry"))	**set**
x = float(20.5)	**float**	x = frozenset(("apple", "banana", "cherry"))	**frozenset**
x = complex(1j)	**complex**	x = bool(5)	**bool**
x = list(("apple", "banana", "cherry"))	**list**	x = bytes(5)	**bytes**
x = tuple(("apple", "banana", "cherry"))	**tuple**	x = bytearray(5)	**bytearray**
x = range(6)	**range**	x = memoryview(bytes(5))	**memoryview**

Python Numbers

Python Numbers

There are three numeric types in Python: ➡

- `int`
- `float`
- `complex`

- Variables of numeric types are created when you assign a value to them:

Example

```
x = 1    # int
y = 2.8  # float
z = 1j   # complex
```

- To verify the type of any object in Python, use the type() function:

Example

```
print(type(x))
print(type(y))
print(type(z))
```

Int

Int, or integer, is a whole number, positive or negative, without decimals, of unlimited length.

Example

Integers:

```
x = 1
y
= 35656222554887711
z = -3255522

print(type(x))
print(type(y))
print(type(z))
```

Float

Float, or "floating point number" is a number, positive or negative, containing one or more decimals. can also be scientific numbers with an "e" to indicate the power of 10.

Example

Floats:

```
x = 1.10
y = 1.0
z = -35.59

print(type(x))
print(type(y))
print(type(z))
```

Complex

Complex numbers are written with a "j" as the imaginary part:

Example

Complex:

```
x = 3+5j
y = 5j
z = -5j

print(type(x))
print(type(y))
print(type(z))
```

Type Conversion

You can convert from one type to another with the int(), float(), and complex() methods:

Example

Convert from one type to another:

```
x = 1    # int
y = 2.8  # float
z = 1j   # complex

a = float(x)   #convert from int to float:
b = int(y)     #convert from float to int:
c = complex(x) #convert from int to complex:

print(a)
print(b)
print(c)

print(type(a))
print(type(b))
print(type(c))
```

Note: You cannot convert complex numbers into another number type.

Random Number

Python does not have a random() function to make a random number, but Python has a built-in module called random that can be used to make random numbers:

Example

Import the random module, and display a random number between 1 and 9:

```
import random

print(random.randrange(1, 10))
```

In our Random Module Reference you will learn more about the Random module.

Python Casting

Specify a Variable Type

There may be times when you want to specify a type on to a variable. This can be done with casting. Python is an object-orientated language, and as such it uses classes to define data types, including its primitive types.

Casting in python is therefore done using constructor functions:

- int() - constructs an integer number from an integer literal, a float literal (by removing all decimals), or a string literal (providing the string represents a whole number)
- float() - constructs a float number from an integer literal, a float literal or a string literal (providing the string represents a float or an integer)
- str() - constructs a string from a wide variety of data types, including strings, integer literals and float literals

Examples

Integers:

```
x = int(1)    # x will be 1
y = int(2.8)  # y will be 2
z = int("3")  # z will be 3
```

Floats:

```
x = float(1)     # x will be 1.0
y = float(2.8)   # y will be 2.8
z = float("3")   # z will be 3.0
w = float("4.2") # w will be 4.2
```

Strings:

```
x = str("s1") # x will be 's1'
y = str(2)    # y will be '2'
z = str(3.0)  # z will be '3.0'
```

Test Yourself With Exercises

Data Type Exercise:

The following code example would print the data type of x, what data type would that be?

```
x = 5
print(type(x))
```

Number Exercise:

Insert the correct syntax to convert x into a floating point number.

```
x = 5
x = ____(x)
```