Faculty of
Information Technology
King Mongkut's Institute of Technology Ladkrabang

Python Crash Course

Dr. Sirasit Lochanachit

06036120 Data Structures and Algorithms Principles

Outline

- Python Overview
- Objects in Python
- Operators and Precedence

Python Overview

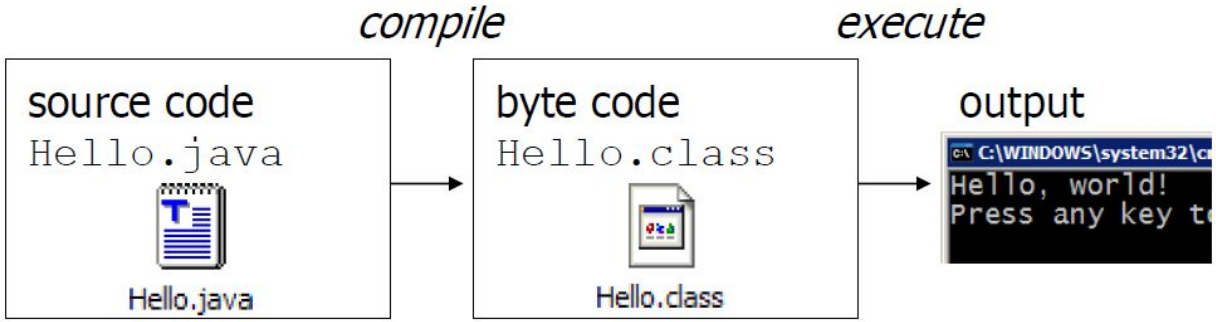
Python is formally an interpreted language.

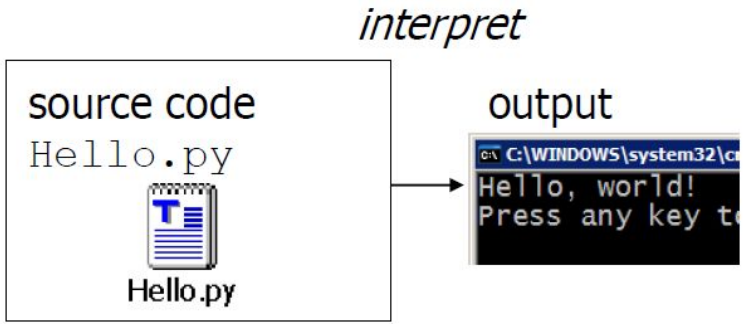
Commands are executed through a piece of software known as the Python interpreter.

- The interpreter receives a command -> evaluate -> reports the result

Programmer typically defines a series of commands in advance and saves those commands in a plain text file known as source code or a script.

For Python, source code is conventionally stored in a file named with the .py suffix (e.g., demo.py).


compile
execute
source code
Hello.java
Hello.java
byte code
Hello.class
Hello.class
output
C:\WINDOWS\system32\c
Hello, world!
Press any key t


interpret
source code
Hello.py
Hello.py
output
C:\WINDOWS\system32\c
Hello, world!
Press any key t

Objects in Python

Python is an object-oriented language and classes form the basis for all data types.

Python's built-in classes:

- **int** class for integers,
- **float** class for floating-point values,
- **str** class for character strings.

Assignment Statement

In Python, a variable can store any type of value/data without the need to declare a data type in advance.

- name = “Jane”
- age = 25

temperature = 35.6

(identifier/variable) (object)

Identifiers / Variables

Variables in Python are case-sensitive.

Variables can be composed of almost any combination of letters, numerals, and underscore characters.

- A variable cannot begin with a numeral

There are 33 specially reserved words that cannot be used as variables.

Reserved Words

False	class	finally	is	return
None	continue	for	lambda	try
True	def	from	nonlocal	while
and	del	global	not	with
as	elif	if	or	yield
assert	else	import	pass	
break	except	in	raise	

Dynamically Typed

temperature = 35.6

(identifier/variable) (object)

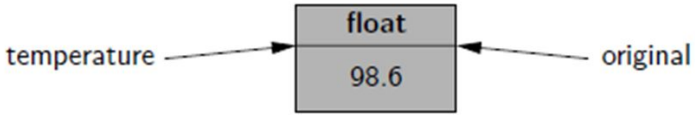
- Each identifier is implicitly associated with the memory address of the object to which it refers.
- No advance declaration associating a variable with a particular data type.
- A variable can be associated with any type of object, and it can later be reassigned to another object of the same (or different) type.

Dynamically Typed

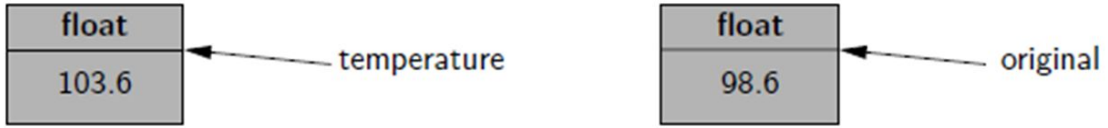
temperature = 35.6

(identifier/variable) (object)

original = temperature



temperature = temperature + 5



- A programmer can establish an alias by assigning a second identifier to an existing object.

Built-in Python Data Types

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

Ref: https://www.w3schools.com/python/python_datatypes.asp

Operators and Precedence

- Logical Operators

not	unary negation
and	conditional and
or	conditional or

- Equality Operators

is	same identity
is not	different identity
==	equivalent
!=	not equivalent

Operators and Precedence

- Comparison Operators

<	less than
<=	less than or equal to
>	greater than
>=	greater than or equal to

- Arithmetic Operators

+	addition
–	subtraction
*	multiplication
/	true division
//	integer division
%	the modulo operator

Operators and Precedence

- Bitwise Operators

~	bitwise complement (prefix unary operator)
&	bitwise and
|	bitwise or
^	bitwise exclusive-or
<<	shift bits left, filling in with zeros
>>	shift bits right, filling in with sign bit

- Sequence Operators

s[j]	element at index *j*
s[start:stop]	slice including indices [start,stop)
s[start:stop:step]	slice including indices start, start + step, start + 2*step, ..., up to but not equalling or stop
s + t	concatenation of sequences
k * s	shorthand for s + s + s + ... (k times)
val **in** s	containment check
val **not in** s	non-containment check

Operators and Precedence

- Dictionary Operators

d[key]	value associated with given key
d[key] = value	set (or reset) the value associated with given key
del d[key]	remove key and its associated value from dictionary
key **in** d	containment check
key **not in** d	non-containment check
d1 == d2	d1 is equivalent to d2
d1 != d2	d1 is not equivalent to d2

Operators and Precedence

- Operator Precedence

Ordered from highest to lowest

Operator Precedence		
	Type	**Symbols**
1	member access	expr.member
2	function/method calls container subscripts/slices	expr(...) expr[...]
3	exponentiation	**
4	unary operators	+expr, −expr, ~expr
5	multiplication, division	*, /, //, %
6	addition, subtraction	+, −
7	bitwise shifting	<<, >>
8	bitwise-and	&
9	bitwise-xor	^
10	bitwise-or	|
11	comparisons containment	is, is not, ==, !=, <, <=, >, >= in, not in
12	logical-not	not expr
13	logical-and	and
14	logical-or	or
15	conditional	val1 if cond else val2
16	assignments	=, +=, −=, *=, etc.

Exercise 1

Write a program to calculate an age based on a birth year (AD)

- Example input: 2001
- Expected output:
  - Your age is 22

Exercise 2

Write a program to calculate an area of a triangle by accepting a height and a width (base) of a triangle.

- Example input: triangle(10, 12)
- Expected output:
  - A triangular area is 60