Basic Programming in Python

Taehee Kim

May 6, 2022

University of Oldenburg

Basic programming in Python

Primitive Data Type

Units of information

- Smallest physical unit is 1 bit: 0 or 1
- Smallest logical unit is 1 byte: 8 bit
  –processor can usually only address entire bytes

Unicode

- current version 9.0 (as of June 2016) defines 128 172 of 1 114 112 possible characters
- 3 bytes are necessary for all possible characters
- UTF-8
  1 byte per character, covers most important Western characters

Primitive Data Type

	Type	byte	value range
	boolean	1	true or false
	char	1-4	All Unicode character
Integer	byte	1	$-2^7 ... 2^7 - 1$
Integer	short	2	$-2^{15} ... 2^{15} - 1$
Integer	int	4	$-2^{31} ... 2^{31} - 1$
Integer	long	8	$-2^{63} ... 2^{63} - 1$
Floating point numbers	float	4	$\pm 3.4 * 10^{38}$
Floating point numbers	double	8	$\pm 1.79 * 10^{308}$

Overflow: when we try to assign a value which beyond the range of the type.
e.g., assigning 130 to `byte`

Data Type in Python

Python

- Python automatically recognize data formats
- automatic conversion between types

Integer

- `int`
- Integers can be of any length, it is only limited by the memory available

Floating point number

- `1` = integer, `1.0` = floating point number
- A floating point number is accurate up to 15 decimal places. $\rightarrow$ use `decimal` module

String

- sequence of Unicode characters
- `' '` or `" "`
- use triple quotes for multi-line strings, `'''` or `"""`

Use `type()` to check data type

Identifier

- Identifier is the name given to entities like class, functions, variables etc.
- combination of..
  - a to z
  - A to Z
  - 0 to 9 (digits)
  - _ (underscore)
- An identifier cannot start with a digit.

Naming is important!

- `a = 25` vs. `age = 25`
- `myAge = 25`
- `my_age = 25`

Python Keywords

Python Keywords

- reserved words, i.e., we cannot use those as variable name, function name or any other identifier
- case sensitive

Table 1: Python keywords[1]

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	

[1] In case you are interested in the usage of them, see `https://www.programiz.com/python-programming/keyword-list`

Expression

- An expression is consisted of operands and operators

Expression

- An expression is consisted of operands and operators

Expression

- An expression is consisted of operands and operators

Operators

Table 2: Arithmetic operators

Operator	Meaning	Example
+	Add	x + y
–	Subtract	x – y
*	Multiply	x * y
/	Divide (always results into float)	x / y
%	Modulus - remainder of the division	x % y (remainder of x/y)
//	Floor division	x // y
**	Exponent	x**y (x to the power y)

Table 3: Comparison operators

Operator	Meaning	Example
>	True if left operand is greater than the right	x >y
<	True if left operand is less than the right	x <y
==	True if both operands are equal	x == y
!=	True if operands are not equal	x != y
>=	True if left operand is greater than or equal to the right	x >= y
<=	True if left operand is less than or equal to the right	x <= y

It returns True or False, i.e., boolean

Table 4: Logical operators

Operator	Meaning	Example
and, &	True if both the operands are true	x and y
or, |	True if either of the operands is true	x or y
not, !	True if operand is false	not x

Table 5: Assignment operators

Operator	Example	Equivalent to
=	x = 5	x = 5
+=	x += 5	x = x+5
−=	x −=5	x = x–5

Table 6: Identity operators

Operator	Example	Example
is	True if the operands are identical	x is True
is not	True if the operands are not identical	x is not True

Table 7: Membership operators

Operator	Example	Example
in	True if value/variable is found in the sequence	5 in x
not in	True if value/variable is not found in the sequence	5 not in x

Mutable vs. Immutable

- Mutable: you can change value
- Immutable: you cannot change value
- Immutable: int, float, str, tuple
- Mutable: list, dict, set

Mutable vs. Immutable

Immutable case example

Mutable vs. Immutable

Immutable case example

Mutable vs. Immutable

Immutable case example

Mutable vs. Immutable

Mutable case example

Mutable vs. Immutable

Mutable case example

Mutable vs. Immutable

Mutable case example

Data Structure

Data structure

- Assign a group of data to a variable
- several data structure is prepared
- Python: list, set, dictionary, tuple

Data Structure

Data Structure

tuple

- immutable
- memory efficient
- you cannot change value by mistake

Data structure

Set

- duplicated item is not allowed
- the order is not preserved
- mutable

Data structure

Data structure