Basic Data Types and Sequence Operations in Python

YE Yuxiao

Tsinghua University

yeyuxiao@outlook.com

2017/09/29

Overview

1. Python - What and Why
   - What is Python
   - Why should we learn Python
2. Basic Data Types in Python
   - Numbers
   - Strings
   - Lists
   - Tuples
   - Dictionaries
3. Sequence Operations
   - Indexing
   - Slicing

What is Python

- A high level programming language
- Similar to natural languages
- Easy to learn

Why should we learn Python

- Fun. Seriously.
- Power and control!
- It can help you get a job.

-- Korin Richmond

Basic Data Types in Python

- Numbers
- Strings
- Lists
- Tuples
- Dictionaries

Numbers

- Integers (int): a very self-explanatory name
  - e.g. 0, 1, -2896
- Floating Point (float): numbers with decimal points
  - e.g. 0.1, 1.0, -3.1415926
- Mathematical operations
  - try `print(6+4)`

Strings

- Extremely superficial (only appearance matters)
  - e.g. "print", "100", "6+4"
  - try `print("6+4")` and `"print"(6+4)`
- Anything inside a pair of single/double quotes will be treaded as a string.
  - '6+4'
  - "6+4"

Strings

- An unpaired quote always tries to pair with its nearest succeeding unpaired quote.
  - '"6+4", she said'
  - "'6+4', she said"
- The backslash "\": a quote right behind a "\" will lose its syntactic function; "\n" is a newline symbol.
  - "\"6+4\", she said"
  - `print('Hello \nworld')`

Lists

- A sequence (**ordered**) of items included in a pair of brackets: [ ].
  - [ ]
  - [1, 2, 3, 4]
  - [4, 3, 2, 1]
  - ['a', 'b', 'c']
- A left bracket always tries to pair with its furtherest succeeding right bracket.
  - [[1, 2], 3, "4"]

Tuples

- A sequence of items included in a pair of parentheses: ( ).
  - ((1, 2), [3, "4"])
- Tuples are very similar to lists, except they are **immutable**. Strings are also **immutable**.
  - `a = [1, 2, 3, 4]`
  - `b = (1, 2, 3, 4)`
  - `c = "1, 2, 3, 4"`
  - `a.append(5)`
  - `print(a)`
  - see what happens when `b.append(5)` and `c.append('5')`

Dictionaries

- Items stored in lists and tuples are **ordered**. You can always find an item in a list/tuple with their **position**.
  - e.g. the first item in [1, 2, 3] is 1; the first item in [3, 2, 1] is 3
- In dictionaries, items are stored (not ordered) with a name, in the form $\{\text{key}_1 : \text{value}_1, \text{key}_2 : \text{value}_2, ...\}$.

Dictionaries

- Keys can be any immutable objects (e.g., numbers, strings, tuples, but not lists or dictionaries). Values can be any objects.
  - { }
  - {'a': 1, (1, 2): '2', {1: 'a'}: [1, 2]}
- Get/change a value by querying the dictionary with its key.
  - `d = {'a': 1, (1, 2): '2', {1: 'a'}: [1, 2]}`
  - `print(d[(1, 2)])`
  - `d[(1, 2)] = 3`
  - `print(d)`

Sequence Operations

Strings, lists, and tuples are all sequence objects (sets of ordered items) in Python.

- Indexing
- Slicing

Indexing

- Each Item is indexed by their position (represented as integers).
- **ATTENTION**: in Python we always count from 0! So the first item in a sequence has an index of 0.
  - `l = ["a", "b", "c", "d"]`
  - `s = "Hello world"`
  - `print(l[0])`
  - `print(s[-1])`
  - how many characters in s? what is s[5]?

Slicing

- A slice is a sub-sequence of a sequence object.
- Syntax for slicing: sequence[start:end:step]. The start index is inclusive; the end index is exclusive.
  - `s = "Hello world"`
  - `print(s[1:6:2])`
  - `print(s[6:1:-1])`
- If the start/end index is not specified, its default value 0/-1 will be used. If the step index is not specified, its default value 1 will be used, and the colon before it can be omitted.
  - `s = "Hello world"`
  - `print(s[::2])`
  - `print(s[1:6])`
  - try `print(s[6:1])`

Practice

Try to solve these problems using Python. You are **soooo** allowed to search the internet. Getting help from online sources is a crucial skill for programmers.

1. Create a list containing three names of you friends, and assign it to a variable called MyFriend.
2. Make items in MyFriend ordered alphabetically.
3. Reverse MyFriend.

Q & A

If you have any questions, you can either post on Piazza, or:

- Check Python Documentation: https://docs.python.org/3.5/tutorial/index.html
- Search Stack Overflow: https://stackoverflow.com
- Ask Google/Baidu/Bing...
- ...