An Introduction to Python programming

Chuck Li

STEM EduX

June 24, 2023

Introduction

- **Python is a simple, powerful and efficient interpreted language.**
- **Together with packages like NumPy, SciPy and Matplotlib, it provides a nice environment for scientific and business works.**
- **The language that we will use for all the homeworks and the projects.**

```
x = 34 - 23 # A comment.
y = "Hello" # Another one.
z = 3.45
if z == 3.45 or y == "Hello":
    x = x + 1
    y = y + " World" # String concat.

print (x)

print (y)
```

Basic Syntax

- Assignment uses = and comparison uses ==.
- For numbers + - * / % are as expected.
- Logical operators are words (and, or, not).
- Simple printing can be done with (print).
- Indentation matters to the meaning of the code.
- Block structure indicated by indentation.
- The first assignment to a variable creates it.
- Variable types don’t need to be declared. Python figures them out.

Whitespace is meaningful

- Use a newline to end a line of code.
- Two statements on the same line are separated with a semicolon ;
- A long line can continue on next with n
- Block structure is indicated by identation.
- The first line with less indentation is outside of a block.
- The first line with more identation starts a nested block.

Dynamic Typing

- Java: statically typed
- Variables are declared to refer to objects of a given type.
- Methods use type signatures to enforce contracts.
- Python: dynamic typed
- Variables come into existence when first assigned to.
- A variable can refer to an object of any type.
- All types are (almost) treated the same way.
- Type errors are only caught in runtime.

Objects and Types

- Every entity is an object.
- Strongly typed: Every object has an associated type, which it carries everywhere.
- Built-in object types:
  - Number 10
  - String ”hello”
  - List [1,’abc’,44]
  - Tuple (4,5)
  - Dictionary
  - Files
- Missing: Arrays

Sequence Types 1

- Tuple
- A simple immutable ordered sequence of items.
- Immutable: a tuple cannot be modified once created
- Items can be of mixed types including collection types
- Strings
- Immutable
- Conceptually very much like a tuple
- Lists
- Mutable ordered sequence of items of mixed types.

Sequence Types 2

- The sequence types share much of the same syntax and functionality.

```
# Tuples
tu = (23, 'abc', 4.56, (2,3), 'def')
# Lists
li = ["abc", 34, 4.34, 23]
# Strings
st = "Hello World"; st = 'Hello World'
# Accessing individual members of a sequence
# Starting with 0
tu[1] # 'abc'
# Negative lookup: count from right, starting with -1
tu[-3] # 4.56
```

Operations on Lists

```
li = [1, 11, 3, 4, 5]
li.append('a') # [1, 11, 3, 4, 5, 'a']
li.insert(2, 'i') # [1, 11, 'i', 3, 4, 5, 'a']
li = ['a', 'b', 'c', 'd']
li.index('b') # 1 - index of first occurence
li.count('b') # 2 - number of occurences
li.remove('b') # ['a', 'c', 'd'] - remove first occurence
li = [5, 2, 6, 8]
li.reverse() # reverse the list in place
li.sort() # sort the list in place
```

Dictionaries

- A mapping collection type

```
d = {'user':'bozo', 'pswd':1234};
d['user'] # returns 'bozo'
d['user'] = 'clown'
d['user'] # returns 'clown'
```

if, elif, else

```
if not done and (x > 1):
    doit()
elif done and (x <= 1):
    dothis()
else:
    dothat()
```

while, for

```
while True:
    line = input("Please enter a sentence: ")
    if len(line) == 0:
        break
    print(line)

for letter in 'Hello, world’:
    print (letter)

for i in range(2,10,2):
    print (i)
```

Defining functions

```
def get_final_answer(filename):
    """Documentation String"""
    line1
    line2
return total_counter

OutsideTheFunction()
```

- Function defining begins with def.
- Function name, its arguments and colon follow.
- No declaration of types of arguments or result.
- return indicates the value to be sent back to the caller.
- First line with less indentation is considered to be outside of the function definition.