Introduction to Python

Jae Yun JUN KIM

February 21, 2018

Please follow the examples given to the below to learn Python.

1. Hello world

```
print("Hello world")

name = "Maria"
print(name)
```

2. Seven arithmetic operators

```
print("5 + 2 =", 5+2)
print("5 - 2 =", 5-2)
print("5 * 2 =", 5*2)
print("5 / 2 =", 5/2)
print("5 % 2 =", 5%2)
print("5 ** 2 =", 5**2)
print("5 // 2 =", 5//2)
```

3. Priorities in the application of arithmetic operators

```
print("1 + 2 - 3 * 2 = ", 1 + 2 - 3 * 2)
print("(1 + 2 - 3) * 2 = ", (1 + 2 - 3) * 2)
```

4. Strings I

```
quote = "\"Always remember you are unique\""

multi_line_quote =’’’ just
like everyone else’’’

# to joint two strings
new_string = quote + multi_line_quote
print(new_string)

print("%s %s %s" % (’I like the quote’, quote, multi_line_quote))
print(’\n’ * 5)

print("I don’t like ", end="")
print("newlines")
```

5. Lists (Mutable!)

```
grocery_list = [’Juice’, ’Tomatoes’, ’Potatoes’,
                ’Bananas’]

print(’First Item’, grocery_list[0])

grocery_list[0] = "Green Juice"
print(’First Item’, grocery_list[0])

print(grocery_list[1:3]) # remember that the index 3 is not included

other_events = [’Wash car’, ’Pick up kids’,
                ’Cash check’]

to_do_list = [other_events, grocery_list] # lists inside another list
print(to_do_list)

print((to_do_list[1][1]))

grocery_list.append(’Onions’)
print(to_do_list)

grocery_list.insert(1, "Pickle")
print(to_do_list)

grocery_list.remove("Pickle")
print(to_do_list)

grocery_list.sort()
print(to_do_list)

grocery_list.reverse()
print(to_do_list)

del grocery_list[4]
print(to_do_list)

to_do_list2 = other_events + grocery_list
print(to_do_list2)

print(len(to_do_list2))
print(max(to_do_list2))
print(min(to_do_list2))
```

6. Tuples (Immutable!)

```
pi_tuple = (3,1,4,1,5,9)
print(pi_tuple)

new_list = list(pi_tuple)
print(new_list)

new_tuple = tuple(new_list)
print(new_tuple)

print(len(new_tuple))
print(max(new_tuple))
print(min(new_tuple))
```

7. Dictionaries (Mutable!)

```
super_villains = {’Fiddler’ : ’Isaac Bowin’,
                  ’Captain Cold’ : ’Leonard Snart’,
                  ’Weather Wizard’ : ’Mark Mardon’,
                  ’Mirror Master’ : ’Sam Scudder’,
                  ’Pied Piper’ : ’Thomas Peterson’}

print(super_villains[’Captain Cold’])

del super_villains[’Fiddler’]

super_villains[’Pied Piper’] = ’Hartley Rathaway’

print(len(super_villains))

print(super_villains.get("Pied Piper"))

print(super_villains.keys())

print(super_villains.values())
```

8. Conditionals

```
age = 21

if age > 16 :
    print(’you are old enough to drive’)
```

```
else :
    print(’you are not old enough to drive’)

if age >= 21 :
    print(’you are old enough to drive a tractor trailer’)
elif age > 16 :
    print(’you are old enough to drive a car’)
else :
    print(’you are not old enough to drive’)

# combine the conditionals with logical operators: and, or, not

if ((age >= 1) and (age <= 18)) :
    print("You get a birthday")
elif ((age > 21) and (age <= 65)) :
    print("You get a birthday")
elif not(age==30) :
    print("You don’t get a birthday")
else :
    print("You get a birthday party yeah!")
```

9. Loops

```
#for loop
for x in range(0,10) : # x starts from 0 to 9 (not to 10!)
    print(x, ’ ’, end="")

print(’\n’)
grocery_list = [’Juice’, ’Tomatoes’, ’Potatoes’, ’Bananas’]

for y in grocery_list:
    print(y)

for x in [2,4,6,8,10]:
    print(x)

num_list = [[1,2,3],[10,20,30],[100,200,300]]

for x in range(0,3): # x ranges from 0 to 2 (not to 3!)
    for y in range(0,3): # y ranges from 0 to 2 (not to 3!)
        print(num_list[x][y])

# while loop
import random

random_num = random.randrange(0,100)

while(random_num != 15):
    print(random_num)
    random_num = random.randrange(0,100)
print(random_num)

i = 0

while(i <= 20):
    if(i%2 == 0):
        print(i)
    elif(i == 9):
        print(i)
        break
    else:
        i+= 1  #i=i+1
        continue

    i += 1
```

10. Functions

```
def addNumber(first, last):
    sum_num = first + last
    return sum_num

print(addNumber(1,4))

string = addNumber(1,5)
print(string)


print(’what is your name’)

import sys
name = sys.stdin.readline()

print(’hello’, name)
```

11. Strings II

```
# Strings

long_string = "I’ll catch you if you fall - The Floor"

# doesn’t print the letter corresponding to the 4th index
print(long_string[0:4])
print(long_string[-5:])

print(long_string[:-5])

print(long_string[:4], " be there")

print("%c is my %s letter and my number %d number is %.5f."
    % (’X’,’favorite’, 1, .14))

print(long_string.capitalize())

# returns the index of the first letter of the specified string
print(long_string.find("Floor"))

# returns true if all characters are alphabetic
print(long_string.isalpha())

# returns true if all characters are alphanumeric
# and there is at least one character.
print(long_string.isalnum())

print(len(long_string))

print(long_string.replace("Floor", "Ground"))

# strip out the initial white spaces prior to the actual string starts.
print(long_string.strip())

quote_list = long_string.split(" ")
print(quote_list)
```

12. File I/O

```
# use "ab+" to read and append file (it also creates and opens the file)
test_file = open("test.txt", "wb")

print(test_file.mode)

print(test_file.name)

test_file.write(bytes("Write me to the file\n", ’UTF-8’))

test_file.close()

test_file = open("test.txt", "r+") # for reading and writing
text_in_file = test_file.read()
test_file.close()
```

```
print(text_in_file)

import os
os.remove("test.txt")
```

13. Object Oriented Programming

```
# Objects

class Animal:
    __name = ""
    __height = 0
    __weight = 0
    __sound = 0

    # constructor
    def __init__(self, name, height, weight, sound):
        self.__name = name
        self.__height = height
        self.__weight = weight
        self.__sound = sound

    # encapsulation
    def setName(self,name):
        self.__name = name

    def getName(self):
        return self.__name

    def setHeight(self,height):
        self.__height = height

    def getHeight(self):
        return self.__height

    def setWeight(self,wheight):
        self.__weight = weight

    def getWeight(self):
        return self.__weight

    def setSound(self,sound):
        self.__sound = sound

    def getSound(self):
        return self.__sound

    # for polymorphism
    def getType(self):
        print("Animal")

    def toString(self):
        return "{} is {} cm tall and {} kilograms and say {}".format(
        self.__name, self.__height, self.__weight, self.__sound)


cat = Animal(’Whiskers’,33,10,’Meow’)

print(cat.toString())

# inheritance
class Dog(Animal):
    __owner = ""

    def __init__(self,name,height,weight,sound,owner):
        self.__owner = owner
        super(Dog, self).__init__(name,height,weight,sound)

    def setOwner(self, owner) :
        self.__owner = owner

    def getOwner(self):
        return self.__owner

    def getType(self):
        print("Dog")

    def toString(self):
        return "{} is {} cm tall and {} kilograms and say {} Hi {}".format(
self.getName(), self.getHeight(), self.getWeight(),
self.getSound(), self.__owner)

    def multipleSounds(self, how_many=None):
        if how_many is None:
            print(self.getSound())
        else:
            print((self.getSound() + ’ ’) * how_many)

spot = Dog("Spot", 53, 27, "Ruff", "Mike")
print(spot.toString())

# polymorphism
class AnimalTesting:
    def getType(self, animal):
        animal.getType()
```

```
test_animals = AnimalTesting()

test_animals.getType(cat)
test_animals.getType(spot)

spot.multipleSounds()

spot.multipleSounds(4)
```