MIT World Peace University

Python Programming
Second Year B. Tech, Semester 4

Basics of Python

Assignment No. 1

Prepared By

Krishnaraj Thadesar
Cyber Security and Forensics
Batch A1, PA 20

January 31, 2023

Contents

1 Aim

To learn the basics of the python programming language and understand fundamental syntax and semantics of Python Programming.

2 Objectives

1. 1. To learn the basics of the Python programming language.
2. 2. To learn the Variable declaration, User input and output of the Python programming language

3 Problem Statement

Learn basic statements of Python.

4 Theory

4.1 Introduction to Python

Python is a general-purpose programming language that is becoming increasingly popular in scientific computing. It is a high-level language, meaning that it is easy to read and write, and it is interpreted, meaning that it does not need to be compiled before it is run. Python is also an object-oriented language, meaning that it allows the user to define new types of objects and to create instances of those objects.

Python is a dynamically typed language, meaning that the type of a variable is determined at run time, rather than at compile time. Python is a multi-paradigm language, meaning that it supports multiple programming styles, including object-oriented, imperative, and functional programming. Python is a multi-platform language, meaning that it can be run on a variety of operating systems, including Windows, Mac OS X, and Linux.

4.2 Assigning Values to Variables

[2]:
```python
counter = 100 # An integer assignment
miles = 1000.0 # A floating point
name = "John" # A string
print(counter)
print(miles)
print(name)
```

```
100
1000.0
John
```

Here, 100, 1000.0 and "John" are the values assigned to counter, miles, and name variables, respectively.

4.3 Multiple Assignment

Python allows you to assign a single value to several variables simultaneously. For example -

[4]:
```
a = b = c = 1
d, e, f = 1, 2, "john"
print(a, b, c, d, e, f)
```

```
1 1 1 1 2 john
```

Here, two integer objects with values 1 and 2 are assigned to variables a and b respectively, and one string object with the value "john" is assigned to the variable c.

4.4 Standard Data Types

Python has five standard data types - * Numbers * String * List * Tuple * Dictionary

4.5 Python Numbers

Number data types store numeric values. Number objects are created when you assign a value to them. For example -

[6]:
```
var1 = 1
var2 = 10
```

Python supports four different numerical types -

- int (signed integers)
- long (long integers, they can also be represented in octal and hexadecimal)
- float (floating point real values)
- complex (complex numbers)

Here are some examples to show these types -

4.6 Python Strings

Strings in Python are identified as a contiguous set of characters represented in the quotation marks. Python allows for either pairs of single or double quotes. Subsets of strings can be taken using the slice operator ([ ] and [:]) with indexes starting at 0 in the beginning of the string and working their way from -1 at the end.

The plus (+) sign is the string concatenation operator and the asterisk (*) is the repetition operator. For example -

4.7 Python Lists

Lists are the most versatile of Python's compound data types. A list contains items separated by commas and enclosed within square brackets ([]). To some extent, lists are similar to arrays in C. One difference between them is that all the items belonging to a list can be of different data type.

The values stored in a list can be accessed using the slice operator ([ ] and [:]) with indexes starting at 0 in the beginning of the list and working their way to end -1. The plus (+) sign is the list concatenation operator, and the asterisk (*) is the repetition operator. For example -

4.8 Python Tuples

A tuple is another sequence data type that is similar to the list. A tuple consists of a number of values separated by commas. Unlike lists, however, tuples are enclosed within parentheses.

The main differences between lists and tuples are: Lists are enclosed in brackets ([ ]) and their elements and size can be changed, while tuples are enclosed in parentheses (( )) and cannot be updated. Tuples can be thought of as read-only lists. For example -

4.9 Python Dictionary

Python's dictionaries are kind of hash table type. They work like associative arrays or associative memories, depending on your background. Dictionaries consist of key-value pairs. A dictionary key can be almost any Python type, but are usually numbers or strings. Values, on the other hand, can be any arbitrary Python object.

Dictionaries are enclosed by curly braces ({ }) and values can be assigned and accessed using square braces ([]). For example -

5 User Input

While programming, we might want to take the input from the user. Taking input is a way of interact with users, or get data to provide some result. In Python two built-in methods are provided to read the data from the keyboard, the input() and raw_input() function.

input() - Python provides a simple framework for getting user input in the form of the *input()* function. The *input()* function reads a line from the console, converts it into a string, and returns it. When the *input()* function is called, the program flow halts until the user enters some input. The user then adds some information and presses the Enter key. The input function returns to the program with the entered string.

5.1 Examples

[10]:
```
a = input("Enter name: ")
print("You Entered Name:", a)
```

```
You Entered Name: Krishnaraj
```

[12]:
```
num = input("Enter a number: ")
print("You Entered Number:",num)
print(type(num))
```

```
You Entered Number: 123
<class 'str'>
```

In the above example2, we have used the input() function to take input from the user and stored the user input in the num variable.It is important to note that the entered value 10 is a string, not a number. So, type(num) returns <class 'str'>.

5.1.1 Accepting Multiple Values from User

We can also ask for multiple values directly on a single line with only one call to the input() function. For example, let's get some information about a student from the user and store it in different variables. split()function helps in getting multiple inputs from users. It breaks the given input by the specified separator. If a separator is not provided then any white space is a separator. Generally, users use a split() method to split a Python string but one can use it in taking multiple inputs.

[14]:
```python
name, age, score = input("Enter student's name, age and score separated by space:
 ↪").split()
print("Student Name:", name)
print("Student Age:", age)
print("Student Score:", score)
```

```
Student Name: Krish
Student Age: 20
Student Score: 8
```

6 Platform

Operating System: Arch Linux x86-64
IDEs or Text Editors Used: Visual Studio Code with Jupyter
Interpreter: python 3.10.8

7 Libraries Used with pip

No additional Libraries are used with pip. The only libraries used are the default libraries that come with python.

8 Input

1. Numbers for Addition, and some Strings

9 Output

1. Basic Print Statements
2. Sum of 2 Numbers

10 Code

11 Learning Basics of Python - Assignment 1

Assignment 1

Krishnaraj Thadesar

Batch A1, PA20 CSE CSF

Semester 3 Second Year

[1]:
```
print("hello world")
```

```
hello world
```

11.0.1 Declaring Variables

[2]:
```
a = 1
print(a)
a = 1.5
print(a)
b = "hello"
print(b)
```

```
1
1.5
hello
```

11.0.2 To find the type of the Variable

[3]:
```
print(type(a))
print(type(b))
```

```
<class 'float'>
<class 'str'>
```

11.0.3 Taking Inputs

[5]:
```
a = int(input("enter a number"))
b = int(input("enter another number"))
print(a, b)
print("The sum of the two number is: ", a + b)
```

```
1 2
The sum of the two number is:  3
```

[6]:
```
a = float(input("enter a number"))
b = float(input("enter another number"))
print(a, b)
print("The sum of the two number is: ", a + b)
```

```
1.45 35.3
The sum of the two number is:  36.75
```

11.0.4 Multiple Inputs

[7]:
```
x, y, z = input("Enter 3 numbers").split()
print(x, y, z)
```

```
1 2 3
```

[8]:
```
# Trying to convert the input to int using list comprehension
a = "a, b, c, e, d, r"
some_list = [ord(i) + 3 % 26 for i in a.split(', ')]
print(some_list)
```

```
[100, 101, 102, 104, 103, 117]
```

[9]:
```
# Trying Piece of code to learn about list and the None type in python.
drink = "Available"
food = None
def menu(x):
if x == drink:
        print(drink)
else:
        print(food)

menu(drink)
menu(food)
```

```
Available
None
```

12 Conclusion

Understood the basic commands, statements and syntax of python programming language

13 FAQ

1. **What are the features of python language?**

   The Features of Python are:

   - Easy to Learn and Use
   - Free and Open Source
   - Portable
   - Extensible
   - Interpreted
   - Object-Oriented
   - High-Level Language
   - Automatic Memory Management
   - Large Standard Library

2. **Explain print () statement of python language.**

   The print() function is used to output the specified message to the screen. The message can be a string, or any other object, the object will be converted into a string before written to the screen.

   ```
   print("Hello World") # OUTPUT : Hello World

   ```

   1 shows the print statement in python programming language.

3. **How to write single line and multi-line comments in python programming language**

   Single line comments are written using the hash symbol at the start of the line.

   ```
   # This is a single line comment

   ```

   Multi-line comments are written using triple quotes (""") at the start and end of the comment.

   ```
   """
   This is a multi-line comment
   This is a multi-line comment
   This is a multi-line comment
   """

   ```