Submitted by:. **Saad Abdullah**

Roll No:. **SP20M2BB043**

Class **BSCS 7th Eve A**

(Assignment)

Introduction to Python:

What is Python?

Python is a popular programming language. It was created by Guido van Rossum, and released in 1991.

It is used for:

- web development (server-side),
- software development,
- mathematics,
- system scripting.

What can Python do?

Python can be used on a server to create web applications.

Python can be used alongside software to create workflows.

Python can connect to database systems. It can also read and modify files.

Python can be used to handle big data and perform complex mathematics.

Python can be used for rapid prototyping, or for production-ready software development.

Why Python?

Python works on different platforms (Windows, Mac, Linux, Raspberry Pi, etc).

Python has a simple syntax similar to the English language.

Python has syntax that allows developers to write programs with fewer lines than some other programming languages.

Python runs on an interpreter system, meaning that code can be executed as soon as it is written. This means that prototyping can be very quick.

Python can be treated in a procedural way, an object-oriented way or a functional way

Python Syntax compared to other programming languages

- Python was designed for readability, and has some similarities to the English language with influence from mathematics.
- Python uses new lines to complete a command, as opposed to other programming languages which often use semicolons or parentheses.
- Python relies on indentation, using whitespace, to define scope; such as the scope of loops, functions and classes. Other programming languages often use curly-brackets for this purpose.

Example:

```
marks1=int(input("Enter your marks :"))
marks2=int(input("Enter your marks :"))
marks3=int(input("Enter your marks :"))
marks4=int(input("Enter your marks :"))
if(marks1 <33 or marks2<33 or marks3<33):
    print("you have less than 33 marks")
elif (((marks1+marks2+marks3)/3)<40):
    print("Your average is less than 40")
else:
    print("Pass")
```

Modules:

In Python, Modules are simply files with the ". py" extension containing Python code that can be imported inside another Python Program. In simple terms, we can consider a module to be the same as a code library or a file that contains a set of functions that you want to include in your application. To create a module, we have to save the code that we wish in a file with the file extension ".py". Then, the name of the Python file becomes the name of the module.

For Example,

In this program, a function is created with the name "welcome" and save this file with the name mymodule.py i.e. name of the file, and with the extension ".py".

We saved the following code in a file named mymodule.py

def. welcome(name):

print("Hello, " + name +"

IF-Else Statement:

In computer programming, we use the `if` statement to run a block code only when a certain condition is met.

For example, assigning grades **(A, B, C)** based on marks obtained by a student.

1. if the percentage is above **90**, assign grade **A**
2. if the percentage is above **75**, assign grade **B**
3. if the percentage is above **65**, assign grade **C**

In Python, there are three forms of the `if...else` statement.

1. `if` statement
2. `if...else` statement
3. `if...elif...else` statement

Example:

```
def hello():
    marks=int (input("Enter your marks :"))
    if(marks>=80):
        print("Grade is A")
    elif(marks>=70):
        print("Grade is B")
    elif(marks>=50):
        print("Grade is C")
hello()
```

Getting Values from User:

```
var1=int (input("Enter the value :"))
var2=int (input("Enter the value :"))
var3=int (input("Enter the value :"))
var4=int (input("Enter the value :"))
print(max(var1,var2,var3,var4))
```

Comments

In computer programming, comments are hints that we use to make our code more understandable.

Comments are completely ignored by the interpreter. They are meant for fellow programmers. For example,

```
# declare and initialize two variables

num1 = 6

num2 = 9

# print the output

print('This is output')
```

Here, we have used the following comments,

declare and initialize two variables

print the output

Types of Comments in Python

In Python, there are two types of comments:

1. **single-line comment**
2. **multi-line comment**

Single-line Comment in Python

A single-line comment starts and ends in the same line. We use the # symbol to write a single-line comment.

For example,

```
# create a variable

name = 'Eric Cartman'

# print the value

print(name)
```

Run Code

Output

Here, we have created two single-line comments:

create a variable

print the value

Multi-line Comment in Python

Python doesn't offer a separate way to write multiline comments. However, there are other ways to get around this issue.

We can use # at the beginning of each line of comment on multiple lines. For example,

This is a long comment

and it extends

to multiple lines

Here, each line is treated as a single comment, and all of them are ignored.

Another way of doing this is to use triple quotes, either ''' or """.

These triple quotes are generally used for multi-line strings. But if we do not assign it to any variable or function, we can use it as a comment.

The interpreter ignores the string that is not assigned to any variable or function.

Let's see an example,

''' This is also a

perfect example of

multi-line comments '''

Here, the multiline string isn't assigned to any variable, so it is ignored by the interpreter. Even though it is not technically a multiline comment, it can be used as one.

What is PIP?

PIP is a package manager for Python packages, or modules if you like.

What is a Package?

A package contains all the files you need for a module.

Modules are Python code libraries you can include in your project.

Python Variable

Variables are the most important aspect of any programming language including python. In simple words, variables are names given to memory locations. The values stored in these memory locations can be altered throughout the programming life cycle. Variables are stored in the main memory and are generally required to carry forward various processes and transactions. Since we are talking about memory allocated to variables, it is quite obvious that once the variables are defined they will get associated with a physical address in the main memory. If we start accessing these values by physical addresses, the code will get complicated and that is why the addresses are associated with variable names.

Python Variable Types

- Numbers.
- String.
- List.
- Tuple.

Strings

Strings are Arrays

Like many other popular programming languages, strings in Python are arrays of bytes representing unicode characters. However, Python does not have a character data type, a single character is simply a string with a length of 1. Square brackets can be used to access elements of the string.For example, "hello" is a string containing a sequence of characters 'h' , 'e' , 'l' , 'l' , and 'o'

List

Lists are used to store multiple items in a single variable.

A list is an ordered data structure with elements separated by a comma and enclosed within square brackets. For example, list1 and list2 shown below contains a single type of data. Here, list1 has integers while list2 has strings. In Python, list and tuple are a class of data structures that can store one or more objects or values.

Lists are one of 4 built-in data types in Python used to store collections of data, the other 3 are Tuple, Set, and Dictionary, all with different qualities and usage.

Lists are created using square brackets:

Example

Create a List:

thislist = ["apple", "banana", "cherry"]

print(thislist)

```
numbers = [6, 9, 3, 1]
numbers.sort()
print(numbers)
```

Tuple

Tuples are used to store multiple items in a single variable. Tuple is one of 4 built-in data types in Python used to store collections of data, the other 3 are List, Set, and Dictionary, all with different qualities and usage. A tuple is a collection which is ordered and unchangeable.

A tuple in Python is similar to a list. The difference between the two is that we cannot change the elements of a tuple once it is assigned whereas we can change the elements of a list.

Creating a Tuple

A tuple is created by placing all the items (elements) inside parentheses (), separated by commas. The parentheses are optional, however, it is a good practice to use them.

A tuple can have any number of items and they may be of different types (integer, float, list, string, etc.).

Example

Different types of tuples

Empty tuple

my_tuple = ()

```
print(my_tuple)

# Tuple having integers

my_tuple = (1, 2, 3)

print(my_tuple)

# tuple with mixed datatypes

my_tuple = (1, "Hello", 3.4)

print(my_tuple)

# nested tuple

my_tuple = ("mouse", [8, 4, 6], (1, 2, 3))

print(my_tuple)
```

Dictionary

Dictionaries are used to store data values in key:value pairs.

A dictionary is a collection which is ordered*, changeable and do not allow duplicates.

Dictionary Items

Dictionary items are ordered, changeable, and does not allow duplicates.

Dictionary items are presented in key:value pairs, and can be referred to by using the key name.

Ordered or Unordered?

Dictionaries are changeable, meaning that we can change, add or remove items after the dictionary has been created.

Changeable

Dictionaries are changeable, meaning that we can change, add or remove items after the dictionary has been created.

Example

Create and print a dictionary:

```
thisdict = {
  "brand": "Ford",
  "model": "Mustang",
  "year": 1964
}
print(thisdict)
```

Create a dictionary in Python

Here's how we can create a dictionary in Python.

```
capital_city = {"Nepal": "Kathmandu", "Italy": "Rome", "England": "London"}
print(capital_city)
```

Run Code

Output

```
{'Nepal': 'Kathmandu', 'Italy': 'Rome', 'England': 'London'}
```