**Topics to cover**

a. Introduction to Python

b. Data types, expressions and statements

c. Control flow

d. Functions

e. Data types II

f. Files and exception,

g. Python modules and packages

**Introduction to Python**

Python is a general-purpose, high level programming language. It was designed by Guido van Rossum in 1991 and developed by Python Software Foundation. Python is a programming language that lets you work quickly and integrate systems more efficiently.

Features/Advantages of Python

1. Python is an object-oriented programming language: its structure supports concepts such as polymorphism and inheritance.
2. It is free (an open-source programming language): downloading and installation is free
3. Python has many built-in types and tools
4. Python has many library utilities and third-party library utilities such as NumPy, SciPy to make programming easier
5. Python is easy to learn and use
6. It is portable: it can run easily on any machine and operating system and is platform independent.
7. It is an interpreted language. (i.e., python is processed at runtime by the interpreter).

Data Types

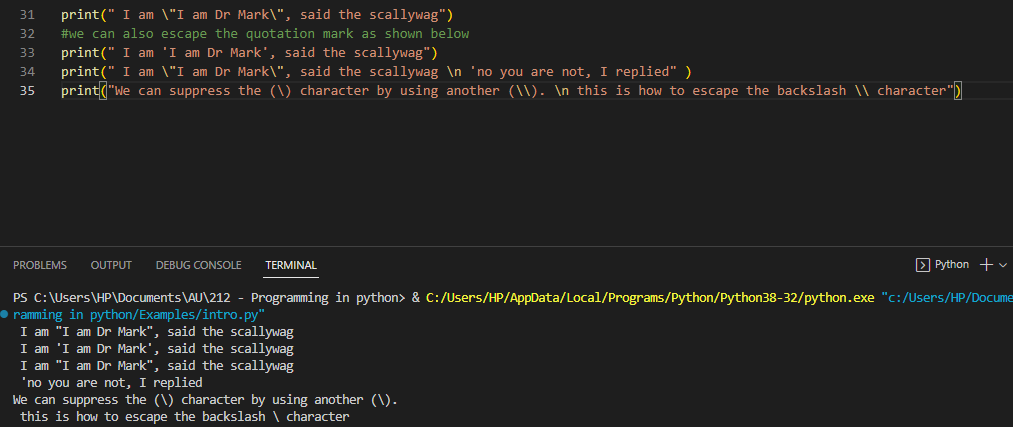
The following are data types that facilitate the storage and manipulation of data in the memory.

1. Int (integer, positive or negative whole number)
2. Float (a positive or negative number that contains one or more decimals)
3. Boolean (true or false)
4. String (a continuous set of characters bounded by single or double quotation marks)
5. List (a general-purpose collection of various data types)
6. Tuples

**Nota bien**

1. Suppressing Special Character:

Using a backslash (\) in front of the quote character in a string “escape” it and causes Python to suppress its usual special meaning. It is then interpreted simply as a literal single quote character



1. Comments:

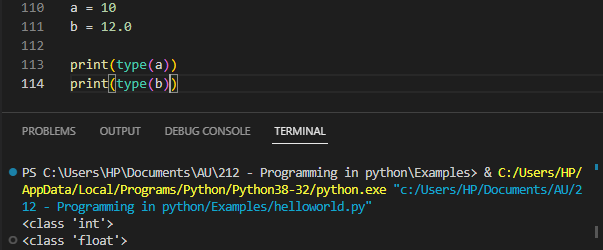
Single-line comments begins with a hash(#) symbol which indicates that the whole line should be considered as a comment

A Multi line comment is indicated by triple double quote(“ “ “) or single quote(‘ ‘ ‘)

**OPERATIONS INVOLVING DATA TYPES**

1. Numbers (Int & Floats)

The type() function gets what data type a variable is



**Arithmetic operators**

|  |  |
| --- | --- |
| **Operators** | **Description** |
| + | Add |
|  | Subtract |
| \* | Multiply |
| / | Divide |
| // | Floor |
| % | Modulus (remainder) |
| ++ | Exponent |
| < | Less than |
| > | Greater than |
| & | And |
| \ | Or |
| <= | Less than or equal to |
| >= | Greater than or equal to |
| == | Check equality |
| != | Not equal |

**Operator Order of Precedence in Python**

When we perform math in Python, equations are read in order of precedence. Operators such as \*, /, //, and % have a higher precedence and get evaluated before others such as + and – operators, which have lower precedence.

Parentheses () overrides the precedence of arithmetic operators

**Example**

print(7 + 3 \* 2)

>> 13

print((7+3)\*2)

>> 20

**Conversion between integer and float**

We can convert the instance of a variable to int or float using the int() or float() functions.

For instance

a = 10

b = 15.5

print(int(b))

>> 15

print(float(a))

>> 10.0

**Common inbuilt math functions and number methods**

1. round() - rounds numbers to some number of decimal places
2. abs() - gets the absolute value of a number
3. pow() - raises a number to some power

print(round(4.7))

>> 8

print(round(5.3))

>> 5

print(round(3.14777777, 3))

>> 3.148

print(round(3.14777777, 2))

>> 3.15

1. Strings: a sequence of characters bounded by single or double quotes

**Creating strings**

# string using double quotes

string1 = "Adeleke University"

# string using single quotes

string1 = 'is the best'

# multiline string

string3 = “““

Programming generally is fun

Yet it seems complex

But with practice, everything becomes easy

”””

Accessing String Characters

1. Indexing – characters in a string can be accessed by calling the index value

string1 = “Adeleke”

print(string1[1])

>> d

Python also allows negative indexing. For instance,

Print(string1[-2]) #prints the 2nd element from the rear

>> k

1. Slicing – a range of characters can be accessed by using the slicing operator – colon “:”

string1 = “Adeleke”

print(string1[0:3])

>> Adel

String Operations

1. Comparison – two strings can be compared using the == operator to return true or false

string1 = “Adeleke”

string2 = “Sade”

string3 = “Sade”

print(string == string2)

>> False

print(string3 == string2)

>> True

1. Concatenation – two or more strings can be joined using the + operator

string1 = “My name is ”

string2 = “Sade Adu”

print(string1 + string2)

>> My name is Sade Adu

1. Length – we can get the length of a string using the len() method

string2 = “Sade Adu”

print(len(string2))

>> 8

NB: space is a character

1. Reversal – the sequence of characters in string can be reversed using [::-1]

string2 = “Sade Adu”

print(string2[::-1])

>> udA edaS

1. Membership test – we can test if a character exists in a string using the keyword “in” which returns true or false

string2 = “Sade Adu”

print(‘a’ in string2)

>> True

print(‘q’ in string2)

>> False

1. Iteration – a for loop can be used to iterate through a string.

string2 = “Sade Adu”

for letter in string1:

    print(letter)

>> S

a

d

e

a

d

u

1. Other common string methods: upper(), lower(), index(), rstrip(), find(), split(), partition(), replace(), startswith(), isnumerical() etc.  
   Have fun checking out how this methods work.
2. List
3. Tuples