**PYTHON PROGRAMMING**

**Why Learn Python? Simple language, best packages, IPython notebooks, industry relevant.**

|  |  |  |  |
| --- | --- | --- | --- |
| **PYTHON DATA TYPES AND MOSTLY USED FUNCTIONS** | | | |
| **S/N** | **PYTHON DATA TYPES & SYNTAX -**  print(type(var\_name)) | **MEANING OR USES** | **FUNCTIONS** |
|  | **Integers**  **Syntax: num = int(input(“a”))** | Positive and negative whole numbers used for mathematical computations | Int(num) |
|  | **Floating point**  **Syntax: num=float(input(“a”))** | Decimal numbers used for mathematical computations | float(num) |
|  | **String**  **Syntax: str = ’statement’** | Sequence of Unicode characters i.e. a large collection of character unlike the ASCII code. It consist of letters, numbers and special characters. | str.upper(), str.lower(), str.split()  Indexing string:  print(str[index\_value])  Slicing string:  print(str[index\_value:]) |
|  | **List**  **Syntax: list = [a,b,c]**  where a,b,c can be integer, floating point numbers, strings etc. | This is an ordered sequence of items, one of the most used data type python due to its flexibility, it is mutable (can be changed) and all its items not necessarily be the same type | list.reverse(), sorted(list),  len(list), list.append(), list.insert(x,y)#add element y in location x where x is an index, list.remove(), list.extend(), list.pop() |
|  | **Tuple**  **Syntax: tuple =(a,b,c)** | This is an ordered sequence of items but is immutable (cannot be modified or changed) |  |
|  | **Set**  **Syntax: set = {a,b,c}**  where a,b,c can be integer, floating point numbers, strings, and list | Set is an unordered collection of unique items  Set is used to perform mathematical set operations like Union, Symmetric difference. Set is mutable, unordered and cannot be indexed | set.add(), set.update(), set.discard(), set.remove(), set.pop(),set.clear(), set1.issubset(set2)  Union of a set:  print(set1|set2) or  print(set1.union(set2))  Intersection of a set:  print(set1&set2)  print(set1.intersection(set2))  Set difference:  print(set1 – set2)  print(set1.difference(set2))  Symmetric difference:  print(set1^set2)  print(set1.symmetric\_difference(set2)) |
|  | Dictionary  **Syntax:**  **dict= {key1:value1,key2:value2}** | This is an unordered collection of key-value pairs. | print(dict[‘key’]) |
|  | Booleans |  |  |

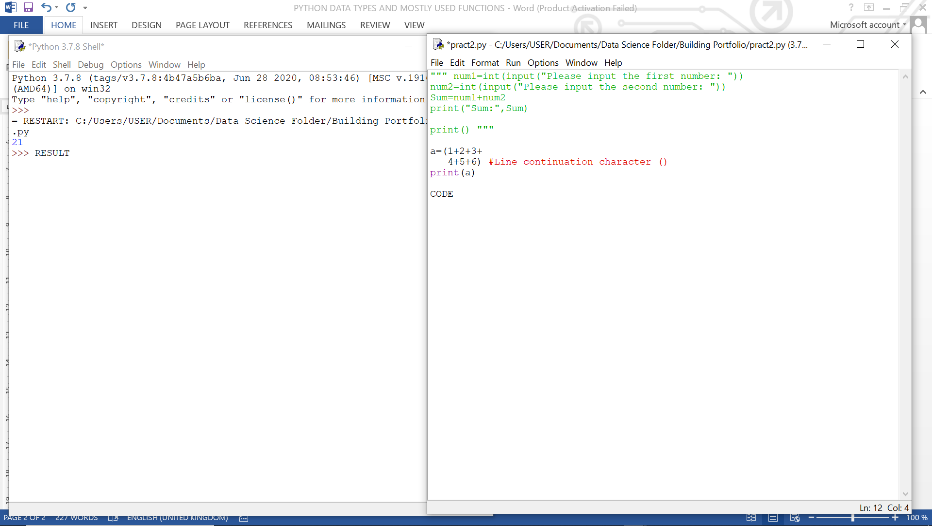
**Important Python Libraries:** pandas, matplotlib, nltk, numpy, scipy, scikit-learn, seaborn, tensorflow etc.

**Python Keywords (Reserved words, they are case sensitive):** print. If, class, input, try, except etc.

**Python Identifiers (names given to entities like class, functions, variables in Python, it helps to differentiate entities from one another, they could be called memory location)**: var1, sum, diff, etc.

**Multiline comment in Python: Use three quotation mark e.g.** “”” comment “””

**Indentation in Python (helps in code readability): four spaces can be used to indent in Python**

**Python statements (instruction a Python interpreter can execute)**

**Line continuation character ()**

**Multiple statement in a single line using delimiter (;):** a=10; b=20

**Assignment operator in Python (=):** Single or Multiple Assignment e.g. a = 10 #Single assignment; a, b= 5, 10 or a= b=”var” #Multiple assignment

**Storage location in Python** (print variable address): e.g. x=3; print (id(x))

**Data types fact in Python:** Every value in Python has a data type. Since everything is an object in Python programming. Data types are actually **classes** and variables are **instance (object)** of these classes.

We can use the **type () function** to determine *the class a variable or a value belongs to* and the **isinstance () function** to check if *an object belongs to a particular class*.

**Conversion between Data types:** Different data types can be converted to each other e.g. float (5), int (100.5), str (20) etc.

**Print formatting**: The output of data can be in the following forms: print (); print (“The value of a is :”, a); a=10;b=20 ; print(“Statement {} and statement {}”.format(a,b)) or print(“Statement {0} and statement {1}”.format(a,b)).

**Python Operators** (Special symbols in Python used in carrying out arithmetic or logical computation, **the value that the operator operates on is called the operand**) e.g. arithmetic (+,-,\*, \*\*, %,//), relational (comparison: <, >,!=), Boolean (logical: AND, OR, NOT), bitwise, assignment and special operators.

* The **floor division operator (//)** on a number returns the integer close to the number but less than the number e.g. 15/2 = 7.5; 15//2 = 7; -15//2 = -8
* **Bitwise operators** (acts on operands as if they were string of binary digits, it operates bit by bit) e.g. Bitwise AND(&),Bitwise OR(|),Bitwise NOT(~),Bitwise XOR(^),Bitwise Right shift(>>),Bitwise Left shift(<<)etc.
* **Assignment operators** (used in Python to assign values to variables e.g. =, +=,-=,\*=,/=,%=,//=,\*\*=,&=,|=,^=,>>=,<<= etc.)

**Special operators** (Identity and Membership operators)

* **Identity operators** are used to check if two values (or variables) are located on the same part of the memory. The identity operators in Python are **is and is not**. Identity operators are only true for single variables.
* **Membership operators** are used to test whether a value or variable is found in a sequence (string, tuple, list, set and dictionary). The membership operators in Python are only **in and not in**. NB: The membership operator can only check the truth value of the key in a **dictionary** and not the value.

**Python if...else statement** (used for decision making): These includes the if statement, if...else statement, if...elif...else statement.

* **Nested if statements** (When we have if...elif...else statement inside another if...elif...else statement, it is called nesting)

**Python while loop** (used to iterate over a block of code as long as the test expression/condition is true  
). Syntax: while test expression:

Body of while

NB: The while loop can be terminated with a break statement

**Python for loop** (used to iterate over a sequence: list, tuple, string or other iterable objects). In python iterating over a sequence is called **transversal**.

Syntax: for *element* in *sequence*:

*Body of for*

**Python range() function** (this generate a sequence of numbers). The start, stop and step-size can be defined while using the range () function. Syntax: range (start, stop, step-size).

**Python Built-in functions** (these functions are used to manipulate strings). A function is a block of reusable code that performs a certain task.

**Placeholders in Python**(%s for strings, %d for integers, and %f for floats, %4.2f means 4 total length and 2d.p)

**Type casting in Python** (This is simply the conversion from one data type to another, such as from an integer to a string).Data types that allow for Type casting in python are int (), float () and str ().

**Python Escape character** (Syntax: \Hello, this removes the first character and print the remaining character)

**Python Inline if statement** (this a simpler form of an if statement and is more convenient if performing a simple task)

Syntax: do Task A if condition is **true** else do Task B

***Continue…***

* Prepared by **Oluronbi D.E.**