

CLASS -12



Computer Science

**DIRECTORATE OF EDUCATION
GOVT. OF NCT, DELHI**

2023-24



PUBLISHED BY: DELHI BUREAU OF TEXT BOOKS

**LIST OF GROUP LEADER AND SUBJECT – EXPERTS
FOR PREPARATION OF SUPPORT MATERIAL**

CLASS – XII (2023-24)

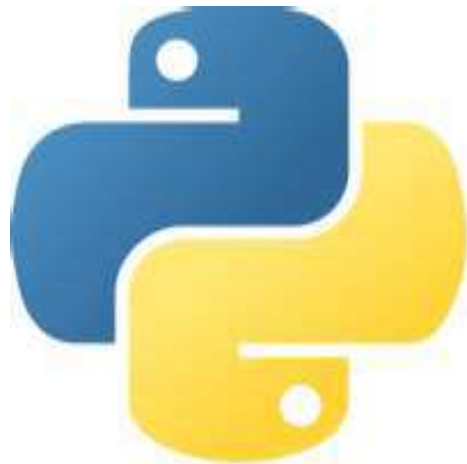
SUBJECT: COMPUTER SCIENCE

- | | |
|----------------------------|--|
| 1. MR. RAVI KANT | PRINCIPAL
GBSSS ,HASTSAL ,
DELHI(1618058) |
| 2. MS. PREETI DALAL | LECTURER COMPUTER SCIENCE
GOVT. CO-ED SSS,
SEC 2 DWARKA(1821029) |
| 3. MS. RUBY RANA | LECTURER COMPUTER SCIENCE,
SBV ASHOK NAGAR (1515002) |
| 4. MR. ANUJ KUMAR
DAGAR | LECTURER COMPUTER SCIENCE,
G CO.ED SV,JAFFARPUR KALAN
(1822055) |
| 5. MR. ROHIT AHLAWAT | LECTURER COMPUTER SCIENCE,
GBSSS ,HASTSAL (1618058) |
| 6. MR. SUNNY DAGAR | LECTURER COMPUTER SCIENCE
GBSSS No 3, NAJAFGARH
(1822057) |
| 7. MR. VINAY GAHLOT | LECTURER COMPUTER SCIENCE
GBSSS No 1, NAJAFGARH
(1822063) |

Table of Contents

S.No	Chapters	Page No.
1	Python revision tour	1-21
2	Functions	22-49
3	Exception Handing	50-61
4	File Handling	62-101
5	Data Structures	102-112
6	Computer Networks	113-156
7	Database Management system	157-224
8	CBSE Previous Year Question Paper 2023	225-239
9	CBSE Sample paper 2023-24	239-270
10	Practice set 1 (unsolved)	271-278
11	Practice set 2 (unsolved)	279-287

Python
Revision
Tour



CHAPTER- 1

PYTHON REVISION TOUR

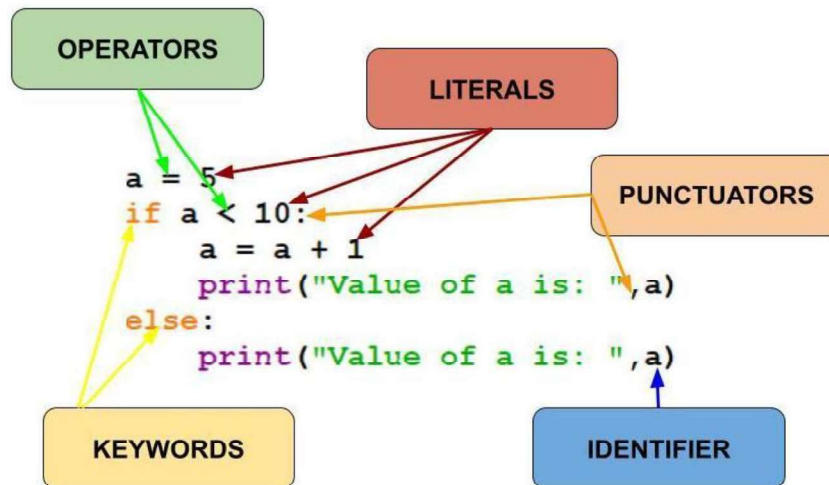
ABOUT PYTHON:

1. Python is a high-level programming language developed by Guido Van Rossum. The language was released in February 1991 and got its name from a BBC comedy series "Monty Python's Flying Circus".
2. It is an interpreted and platform independent language i.e. the same code can run on any operating system.
3. It can be used to follow both procedural and object-oriented approaches to programming.
4. It is free to use and based on two programming languages: ABC language and Modula-3.



BASIC TERMS USED IN PYTHON:

1. **Token / Lexical Unit:** The smallest individual unit in a python program is known as Token or Lexical Unit. A token has a specific meaning for a python interpreter. Examples of tokens are: Keywords, Identifiers, Literals, Operators and Punctuators.



2. **Keywords:** Keywords are the reserved words and have special meaning for Python Interpreters. Each keyword can be used only for that purpose which it has been assigned. Examples: and, while, del, with, True, None, False, return, try etc.
3. **Identifiers:** These are the names given to variables, objects, classes or functions etc. there are some predefined rules for forming identifiers which should be followed else the program will raise Syntax Error.

9. **Dynamic Typing:** It means that it will be decided at the run time that which type of value the variable will store. It is also called implicit conversion. For example,

```
a = 'hello'    #a is of string type
a = 12        #a is of integer type
a = 12.56     # a is of float type
b = 10
c = a + b
a
12.56
b
10
c
22.560000000000002
```

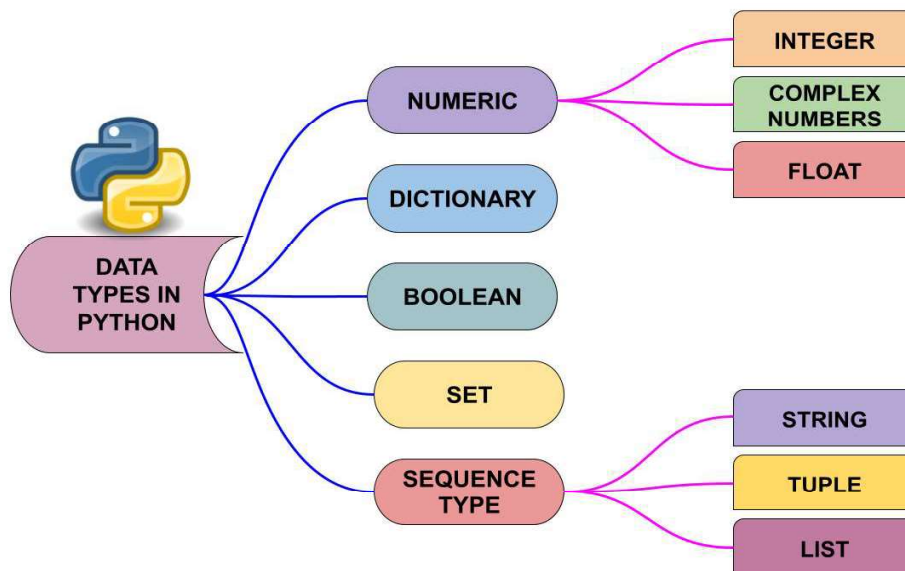
Here, we need not to specify the type of value a will store besides we can assign any type of value to a directly. Similarly, the data type of c will be decided at run time on the basis of value of a and b.

10. **Type Casting:** In Type casting, the data type conversion of a variable is done explicitly by using some built-in functions. Here, we can say that we force the variable by applying a built-in function to change the data type and it is not done at run time. Some common type casting functions are int(), float(), str(), list(), tuple(), dict() etc.

```
a = 12        #a is of integer type
b = float(a)  #changing a from integer to float
b            #now b will contain float value of a
12.0

t = (1,2,3,4)
s = list(t)
s
[1, 2, 3, 4]
```

DATA TYPES IN PYTHON:

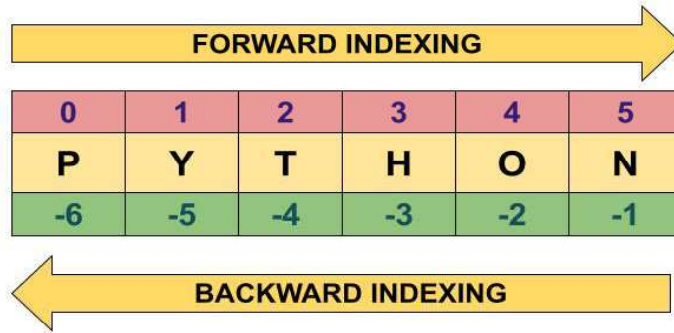


Above data types are classified in two basic categories: Mutable data types and Immutable data types. **Mutable data types** are those data types whose value can be changed without creating a new object. It means mutable data types hold a specific memory location and changes are made directly to that memory location. **Immutable data types** are those data types whose value cannot be changed after they are created. It means that if we make any change in the immutable data object then it will be assigned a new memory location.

1. In **Numeric data types**, Integers allow storing whole numbers only which can be positive or negative. Floating point numbers are used for storing numbers having fractional parts like temperature, area etc. In Python, Floating point numbers represent double precision i.e. 15-digit precision. Complex numbers are stored in python in the form of $A + Bj$ where A is the real part and B is the imaginary part of complex numbers.
2. **Dictionary** is an unordered set of comma separated values where each value is a **key:value** pair. We represent the dictionary using curly brackets {}. Keys in the dictionary should be unique and they cannot be changed while values of the keys can be changed.
3. **Boolean** allows to store only two values True and False where True means 1 and False means 0 internally.
4. **Sequence data types** store a collection or set of values in an ordered manner. We can traverse the values/elements using indexing. The sequence data types are:

A. **STRING:**

- In Python, string is a sequence having Unicode characters. Any value written/assigned in “ ” or ‘ ’ is considered as a string in python.
- Each character is at a particular place called Index having starting value 0 if traversing forward. Indexing can also be done in backward direction starting from the last element/character of string where starting value will be -1 in backward direction.
- Strings are immutable.
- Joining operation in a string can be done between two strings only. We cannot join a number and a string using ‘+’.
- Concatenation operation in a string can be done between a string and a number only using ‘*’.
- Slicing is defined as extracting a part of string from the main string using unique index positions starting from 0. In slicing we can specify start, stop and step values to extract substring from the given string.



B. LIST:

- Lists can hold multiple elements of the same or different data types.
- Lists are mutable in nature which means the values can be updated without assigning a new memory location. It is denoted by square brackets [].
- 'for' loop can be used to traverse a list.
- We can add (join) a list with another list only and not with int, float or string type. Joining of 2 or more can be done using '+'.
 - We can concatenate a list with an integer only. Concatenation (Replication) can be done in a list using '*'.
 - Slicing means extracting a part of list. Slicing in a list is done in same way as in String.

C. TUPLE:

- Tuples can also hold multiple elements of the same or different data types like lists but tuples are immutable in nature. These are denoted by round brackets ().
- If multiple values are assigned to a single variable then by default the type of variable will be tuple.
- Tuples can be traversed in same way as Strings using 'for' loop.
- Slicing, Concatenation (Replication) and Joining operations can also be performed on Tuples in same way as of Strings.

WIDELY USED BUILT-IN FUNCTIONS IN PYTHON:

STRING FUNCTIONS:

len (string)	It returns the number of characters in any string including spaces.
capitalize()	It is used to convert the first letter of sentences in capital letter.
title()	It is used to convert first letter of every word in string in capital letters.
upper()	It is used to convert the entire string in capital case letters.
lower()	It is used to convert entire string in small case letters.
count(substring, [start], [end])	It is used to find the number of occurrences of substring in a string. We can also specify starting and ending index to specify a range for searching substring.
find(substring, [start],[end])	This function returns the starting index position of substring in the given string. Like count(), we can specify the range for searching using starting and ending index. It returns -1 if substring not found.
index(substring)	It returns the starting index position of substring. If substring not found then it will return an error " Substring not found ".
isalnum()	It is used to check if all the elements in the string are alphanumeric or not. It returns either True or False.
islower()	It returns True if all the elements in string are in lower case, otherwise returns False.
isupper()	It returns True if all the elements in string are in upper case, otherwise returns False.
isspace()	It returns True if all the elements in string are spaces, otherwise returns False.
isalpha()	It returns True if all the elements in string are alphabets, otherwise returns False.
isdigit()	It returns True if all the elements in string are digits, otherwise returns False.
split([sep])	This function is used to split the string based on delimiter/separator value which is space by default. It returns a list of n elements where the value of n is based on delimiter. The delimiter is not included in the output.
partition(sep)	It divides the string in three parts: head, separator and tail, based on the sep value which acts as a delimiter in this function. It will always return a tuple of 3 elements. The delimiter/separator will be included as 2 nd element of tuple in the output.
replace(old, new)	It is used to replace old substring inside the string with a new value.
strip([chars])	It returns a copy of string after removing leading and trailing white spaces by default. We can also provide chars value if we want to remove characters instead of spaces. If chars is given then all possible combination of given characters will be checked and removed.
lstrip([chars])	It returns a copy of string after removing leading white spaces. If chars value is given then characters will be removed.
rstrip([chars])	It returns a copy of string after removing trailing white spaces. If chars value is given then characters will be removed.