

(13) ord()

Returns the ASCII code of a character.

(14) ord()

Returns the ASCII code of a character given by user.

(15) chr()

(16) chr()

Returns the character of any ASCII code given by user.

List → A mutable sequence of values is called a list. Its values are called List Item & are enclosed in []

*5

Chapter 2

Review of Python Basic

1) Difference b/w Iterative mode & Script mode of Python

Iterative Mode → In this mode commands are directly type in the (>>) Command prompt & output is displayed as soon as enter key is pressed. Commands do not saved in the memory.

Script Mode → In this mode commands are saved for future use. Once typed they cannot be erased from the same window & needs to be run for a better display by accepting input for the user.

Variable & Datatype

Variable → These are the storage containers for accepting & pressing data for a program. They are the named location from where values can be only changed or fetched.

ii) Identity → It refers to the address in the memory which once created cannot be changed.
Ex: $b = c, d$

→ To retrieve the address (identity) of a variable.

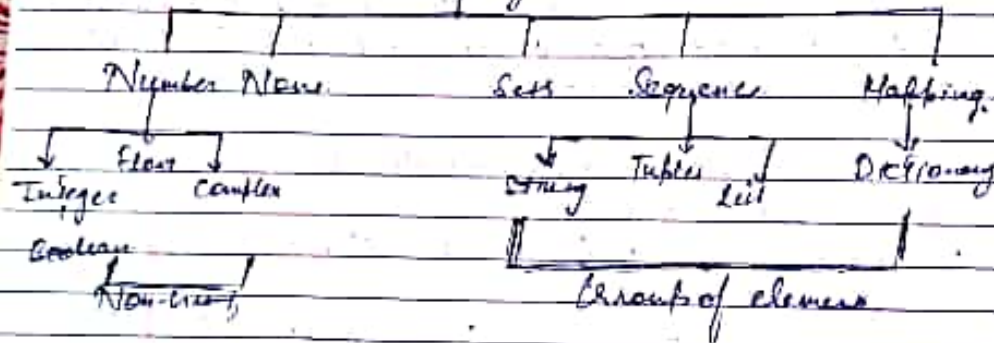
We can use the command

⇒ id (Variable name)

iii) Data type or Type :- In Python we don't have to declare any datatype explicitly as it automatically assigns a datatype depending on the value it is assigned but if we want to fix a particular type for a variable we can do so by using few key words like int, float, str, etc.

Classification of Datatype

Datatype



① Assignment operators $\rightarrow =$
 $\rightarrow ! =$

② Relational operators

③ Logical operators \rightarrow and
 \rightarrow B.A.
 \rightarrow NOT

④ Identity operators

⑤ Bitwise operators

⑥ Membership

I/O Functions

Input \rightarrow output \leftarrow ^{not (Notion)}
eval (string/Num)

Output \rightarrow print () - + ,

Comments \rightarrow The small messages in quotes which
is used to explain any code of part of program
but it is not compile.

Types \rightarrow

1) Single Line Comment \rightarrow using (#)

2) Multiline Comment \rightarrow using (/* */)

Flow of Execution

- 1) Sequence
- 2) Selection or Conditional \rightarrow if, if-else, if... else, else
- 3) Iteration or Looping \rightarrow for, while

Sequence :- The statements of a program are executed one after other.

Selection or Conditional statements :- The execution of any statement depends upon a situation which is either true or false.

It comprises of if, if... else, if... else if... else

Iteration or Looping constructs :- A execution of a statement or group of statements depends upon a situation which is either true or false.

It comprises of :-

Iteration or Looping construct :- A statement or group of statements are repeated a fixed no. of times depending upon a condition.

This construct is essentially implemented in programs related to finding or less or greater than, for & while are the two iterative forms.

\rightarrow Range () \rightarrow is a built in func of iteration construct & used to create a sequence of no. for iteration starting from a start value & ending to a no. smaller than the stop value.

Strings → It is a sequence of characters enclosed within the quotes. Python accepts two types of quotes: single quote & double quote. Strings are immutable in nature.

String Slicing → It is used to extract a subset of characters from a given string. The syntax is: string name [start:stop:step]

The start & end indices must be within the range of the length of the string. Slicing of a -ve will slice the string in rev. order.

String Operations

* Condition with loop in a list:

list = [i for i in range(0, 10) if i % 2 == 0]

[0, 2, 4, 6, 8]

Built-in Functions of Lists

Function

Description

1) Comp (list2, list1)

Compares the elements from two lists

2) Len (list)

Finds the length of the list or returns total no. of elements in a list

3) Max (list)

Returns the maximum element from a list

4) Min (list)

Returns the smallest element from a list

5) List (seq)

Converts the tuple in a list

6) Sum (list)

Adds up all the elements of list

Method

Description

1) Append (list, item)

Adds an item at the end of list

2) Index (list, item)

Returns the pos. of element in a list

3) Insert (list, item, pos)

Inserts a value at a given pos. & shifts the value of this pos towards right

| | | |
|-----|-------|---|
| [:] | Range | Extracts characters from a given string |
| [] | slice | Extracts the characters from the given indices. |

Built in String Method

| Method | Function |
|--------------|--|
| 1) isalpha() | Returns true if the string contains only letters. |
| 2) isdigit() | Returns true if only numbers. |
| 3) islower() | Check only lowercase letters. |
| 4) isupper() | Check only uppercase letters. |
| 5) lower() | Converts to lowercase. |
| 6) upper() | " to uppercase. |
| 7) lstrip() | Removes leading spaces. |
| 8) strip() | " trailing " |
| 9)rstrip() | " white |
| 10) title() | This function does not take any argument and returns true if the string is properly formatted i.e. every first letter after space is in capital. |

updating

Description:

1) `clear()`

Removes the elements from both sides

2) `items()`

Produces a printable string representation of a dictionary

3) `type()`

Returns the the type of value used.

Method

Description:

1) `clear()`

Removes all elements of dictionary

2) `dict.setdefault()`

Creates a copy of a elements of "

3) `dict.items()`

Returns the key value pairs as a "

4) `dict.setdefault(key, default)`

Attaches the key value pair of default value at the end of dict

5) `dict.values()`

Returns list of dictionary values

Tuples A set of immutable values is called tuples & values are enclosed in ()

All the operations in a tuple are same as that of a list.

list (-rs) Removes the immutability nature of a tuple & makes it mutable & list.

tuple (list) → Convert a list to tuple.

Dictionary A set of mutable values paired with immutable keys.

It is declared using {} & referred by values pairs are referred by :