**Package Installation**

* Built-in Libraries
  + available with python interpreter
  + without any supporting statements user can use the methods
  + Example
    - * input()
      * int()
      * float()
      * complex()
      * len()
      * max()
      * min()
      * sum()
* Internal libraries
  + available with python software i.e Secondary storage devices
  + need of supporting statements to use the functions
  + **import modulename**
  + then use the functions in the programme

Example

* + - * os
      * sys
      * arrays
      * random
* External Libraries
  + available in the repository
  + **download the module, install and then use import statement and the use functions**

**Example**

* + - * xlwrite
      * xlread
      * PyMYSQL
      * cx\_Oracle
      * socket
      * boto3
      * pands

**Tool for installing the packages**

pip → used to install the python external packages in linux and windows

easyxf→ used to install the python external packages in windows

**In Linux**

1. sudo apt-get install Python3-pip

2. pip install modulename

pip list → list all the installed external modules

pip show modulename→ description about the particulur module

pip uninstall modulename → uninstall the specified module

pip install modulename → install the specified module

**Arrays**

Array is used to store more than one value in a single name.

Array can store the same data type values in a single name.

In Python, arrays are implemented using the package **array.**

While declaring array programmer has to specify the type of values to be stored in an array

**import array**

**Syntax**

**import array**

**variable=array.array(“typecode”,[val1,val2,val3....])**

The following table shows the type code to be specified and its corresponding data type

| Type code | Type | Minimum size in bytes |
| --- | --- | --- |
| 'c' | character | 1 |
| 'b' | int | 1 |
| 'B' | int | 1 |
| 'u' | Unicode character | 2 |
| 'h' | int | 2 |
| 'H' | int | 2 |
| 'i' | int | 2 |
| 'I' | long | 2 |
| 'l' | int | 4 |
| 'L' | long | 4 |
| 'f' | float | 4 |
| 'd' | float | 8 |

**Attributes and Methods for arrays**

**1. typecode**

This attribute is used to show the data type of array in type code format

**Syntax**

arrayvariable.typecode

**2. itemsize**

This attribute is used to show the size occupied by data type of array

**Syntax**

arrayvariable.itemsize

**3. buffer\_info()**

This method is used to show the memory address and number of values in an array

**Syntax**

arrayvariable.buffer\_info()

**4. append()**

This method is used to add a new value in an array at the end. The value to be appended should be of same type of Array.

**Syntax**

arrayvariable.append(value/variable)

**5. count()**

This method returns the number of occurrence of the given value

**Syntax**

arrayvariable.count(value/variable)

**6. extend()**

This method is used to add more than one values at the end of the array. It extend the values from another array or any data structures like list, tuples, etc., but the data type should be same

**Syntax**

arrayvariable.extend(array/list/tuples/values)

**7. fromlist()**

This method append items from the list to an array.

This is equivalent to

for x in y:

a.append(x)

**Syntax**

arrayvariable.fromlist(listvariable/listvalues)

**8. index()**

This method returns the first occurrence position(index) of the specified value

**Syntax**

arrayvariable.index(variable/values)

**9. insert()**

This method is used to insert a new value in the array before the specified position .If the position is negative value then it treated as the end of the array.

**Syntax**

arrayvariable.insert(position,value/variable)

**10. pop()**

This method removes specified position item from the array and returns the value removed. If the position is not specified by defaults it takes -1, so last item is removed and returned.

**Syntax**

variable=arrayvariable.pop([position])

**11. remove()**

This method removes the first occurrence of specified value from the array.

**Syntax**

arrayvariable.remove(variable/value)

**12. reverse()**

This method reverses the order of the items in the array.

**Syntax**

arrayvariable.reverse()

**13. tolist()**

This method is used to convert the array to an ordinary list with the same items.

**Syntax**

variable=arrayvariable.tolist()

**Set**

A set is an **unordered collection** of items.

Every value in set is unique i.e., without duplicates.

Sets can be used to perform mathematical set operations like union, intersection, symmetric difference etc.

A set is created by placing all the items inside curly braces **{}**, separated by comma or by using the built-in function set().

**Syntax**

variable={val1,val2,val3.....}

variable=set(val1,val2,val3...)

Empty curly braces {} will make an empty dictionary in Python. To make a set without any elements the set() function is used without any argument.

variable=set()

There is no index attached to any value in a python set. So they do not support any indexing or slicing operation.

**1. add()**

This method is used to add the new element to the set

**Syntax**

setvariable.add(value)

**2. update()**

This method is used to add more than one values in the set but the elements should be unique. Programmer can add the values from a tuple, list , string and from other set

**Syntax**

setvariable.update(list/tuple/string/set)

**3. discard() or remove()**

These methods are used to remove the specified item from the set. The discard() method does not produce any error message when the specified value is not in the set. The remove() method raises the error message if the value is not in the set.

**Syntax**

setvariable.discard(value)

sevariable.remove(value)

**4. clear()**

This method removes all the values from set and make it as empty.

**Syntax**

setvariable.clear()

**Set Operations**

Sets can be used to carry out mathematical set operations like union, intersection, difference and symmetric difference. These operations can be done with either operators or methods.

**1. Union**

Union operation of the sets A and B is a set of all elements from both sets. It is performed using | operator or using the method union()

**Syntax**

setvariable1.union(setvariable2)

setvariable1|setvariable2

**2. Intersection**

Intersection operation of sets A and B is a set of elements that are common in both sets. It is performed using & operator or using the method intersection().

**Syntax**

setvariable1.intersection(setvariable2)

setvariable1 & setvariable2

**3. Difference**

Difference operation of sets A and B

(A - B) is a set of elements that are only in A but not in B.

(B - A) is a set of element in B but not in A.

Difference is performed using – operator or using the method difference().

**Syntax**

setvariable1.difference(setvariable2)

setvariable1 - setvariable2

**4. Symmetric Difference**

Symmetric Difference of set A and B is a set of elements in both A and B except those that are common in both i.e except the common elements it returns all the elements. Symmetric difference is performed using ^ operator or using the method symmetric\_difference().

**Syntax**

setvariable1.symmetric\_difference(setvariable2)

setvariable1 ^ setvariable2

**Functions/Methods**

* code re usability
* written once execute many times within the program or among programs
* function has 2 parts
  + **Definition part** → statements which does the task of a functions
  + **Calling part** → execute the functions
* classified into 2 types
  + **Built-in functions**
    - definition part is written by the developer
    - programmer just call the functions
  + **user-defined functions**
    - both definition and calling opart has to be written by the programmers as per their needs

**Syntax**

**def functionanme([parameter1,parameter2….]):**

**statements**

**……**

**………..**

**Calling part syntax**

**[variable=]functionname([arguments…..])**

* + - classified into 4 types
      * function without argument without return type
      * function with argument with return type
      * function with argument without return tyype
      * function without argument with return type

**Special Features of functions**

**1.function with default arguments**

**2. function with multiple return values**

**3. function with dynamic number of arguments**

**4. function with positional arguments**

**Modules**

* used to call the function in another program

Python to excel

Excel to python

Files

Database connection

Exception Handling

Packages

Pre-defined modules

os/sys

Math

Datetime

Data visualization

ail sending

Boto3 package

Example scripts