Centre for Differently Abled Persons
Bharathidasan University

III BCA – V SEMESTER

PYTHON PROGRAMMING
(20UCA5CC6)

UNIT – IV

Prepared by
Dr. M. Prabavathy
Ms. M. Hemalatha

UNIT – IV

Set:

- Sets are used to store multiple items in a single variable.
- Set is one of 4 built-in data types in Python used to store collections of data.
- A set is a collection which is unordered, unchangeable*, and unindexed.
- Sets are written with curly brackets.
- Example:

```
thisset = {"apple", "banana", "cherry"}
print(thisset)
```

Set Items

- Set items are unordered, unchangeable, and do not allow duplicate values.
- Unordered
  - Unordered means that the items in a set do not have a defined order.
  - Set items can appear in a different order every time you use them, and cannot be referred to by index or key.
- Unchangeable

    - Set items are unchangeable, meaning that we cannot change the items after the set has been created.
    - Once a set is created, you cannot change its items, but you can remove items and add new items.
- Duplicates Not Allowed
    - Sets cannot have two items with the same value.
    - Example:

```
thisset = {"apple", "banana",
"cherry", "apple"} print(thisset)
```

Tuples:

- A tuple is a collection which is ordered and unchangeable.
- In Python tuples are writtenwith round brackets.
  - Supports all operations for sequences.
  - Immutable, but member objects may be mutable.
  - If the contents of a list shouldn’t change, use a tuple to prevent items from accidently being added, changed, or deleted.
  - Tuples are more efficient than list due to python’s implementation.
- We can construct tuple in many ways:
  - X=() #no item tupleX=(1,2,3)
  - X=tuple(list1)

X=1,2,3,4

- Example:

`>>> x=(1,2,3)`
`>>> print(x)`
	Output : (1, 2, 3)

`>>> x=[4,5,66,9]`
`>>> y=tuple(x)`
`>>> y`
	Output: (4, 5, 66, 9)

Some of the operations of tuple are:

- Access tuple items
- Change tuple items
- Loop through a tuple
- Count()
- Index()
- Length()

Dictionaries:

- A dictionary is a collection which is unordered, changeable and indexed.
- In Pythondictionaries are written with curly brackets, and they have keys and values.
  - ○ Key-value pairs ○ Unordered

- We can construct or create dictionary like:
    - X={1:’A’,2:’B’,3:’c’}
    - X=dict([(‘a’,3) (‘b’,4)]
    - X=dict(‘A’=1,’B’ =2)
- Example:
    - >>> dict1 = {"brand":"mrcet","model":"college","year":2004}
    - >>> dict1{'brand': 'mrcet', 'model': 'college', 'year': 2004}
- Operations and methods:
- Methods that are available with dictionary are tabulated below.

Method	**Description**
clear()	Remove all items form the dictionary. Return a
copy()	shallow copy of the dictionary Return a new
items()	view of the dictionary's items(key, value).
keys()	Return a new view of the dictionary's keys.

Functions

- A function is a block of code which only runs when it is called.
- You can pass data, known as parameters, into a function.
- A function can return data as a result.

Creating a Function:

- In Python a function is defined using the def keyword:
- Example:

```
defmy_function():
print("Hello from a function")
```

Calling a Function:

- To call a function, use the function name followed by parenthesis
- Example:

```
defmy_function():
print("Hello from a
function")
my_function()
```

")

Arguments

- Information can be passed into functions as arguments.
- Arguments are specified after the function name, inside the parentheses.
- You can add as many arguments as you want, just separate them with a comma.
- The following example has a function with one argument (fname).
- Example:
  - defmy_function(fname):
    print(fname + " Refsnes

```
my_function("Emil")
my_function("Tobias")
my_function("Linus")
```

Recursion

- Python also accepts function recursion, which means a defined function can call itself.
- Recursion is a common mathematical and programming concept.
- It means that a function calls itself.
- This has the benefit of meaning that you can loop through data to reach a result.

Files& Directories:

- A file is some information or data which stays in the computer storage devices.
- Python givesyou easy ways to manipulate these files.
- Generally files divide in two categories,text file and binary file.
- Text files are simple text whereas the binary files contain binarydata which is only readable by computer.

  - Text files:
    - In this type of file, Each line of text is terminated with a special charactercalled EOL (End of Line), which is the new line character (‘\n’) in python by default.

- Binary files:
    - In this type of file, there is no terminator for a line and the data is storedafter converting it into machine understandable binary language.

Creating Files

- To create a new file in Python, use the open() function.
- The open() function takes two parameters; filename, and mode.
- There are four different methods (modes) for opening a file:
    - "r" - Read - Default value. Opens a file for reading, error if the file does not exist
    - "a" - Append - Opens a file for appending, creates the file if it does not exist
    - "w" - Write - Opens a file for writing, creates the file if it does not exist
    - "x" - Create - Creates the specified file, returns an error if the file exists
- In addition you can specify if the file should be handled as binary or text mode
    - "t" - Text - Default value. Text mode
    - "b" - Binary - Binary mode (e.g. images)
- Syntax: f = open("demofile.txt", "rt")

Operations on files

- **Read Only Parts of the File**
    - By default the read() method returns the whole text:

```
f = open("demofile.txt", "r")
print(f.read(5))
```

- **Read Lines**
    - You can return one line by using the readline() method:

      ```
      f = open("demofile.txt", "r")
      print(f.readline())
      ```
- **Write to an Existing File**
    - To write to an existing file, you must add a parameter to the open() function:
        - "a" - Append - will append to the end of the file
        - "w" - Write - will overwrite any existing content
    - Example: Open the file "demofile2.txt" and append content to the file:
        - f = open("demofile2.txt", "a")
        - f.write("Now the file has more content!")
- **Close Files**
    - It is a good practice to always close the file when you are done with it.

      ```
      f = open("demofile.txt", "r")
      print(f.readline())
      f.close()
      ```

The file Object Attributes:

S. No.	Attribute	Description
1	file.closed	Returns true if file is closed, false otherwise.
2	file.mode	Returns access mode with which file was opened.
3	file.name	Returns name of the file.

Example

- fo = open("foo.txt", "wb")
- print "Name of the file: ", fo.name print "Closed or not : ", fo.closed print "Opening mode : ", fo.mode

Output:

- Name of the file: foo.txt
  Closed or not : False
- Opening mode :wb

File Positions

- There are two more methods of file objects used to determine or get files positions.

    - tell():
        - This method is used to tell us the current position within the file which means the next read or write operation will be performed that many bytes away from the start of the file.
        - **Syntax :**obj.tell()

Directory

- A Directory also sometimes known as a folder is a unit organizational structure in a computer’s file system for storing and locating files or more folders.
- Python now supports a number of APIs to list the directory contents.
- Python has several built-in modules and functions for handling files.
- These functions are spread out over several modules such as os, os.path, shutil, and pathlib, to name a few.

seek():

- This method is used to change the current position of file.This method has two main parameters offset and from.

Syntax :

- obj.seek(offset,from)
    - offset: number of bytes to be moved.
    - from: reference position from where the bytes needs to be moved.

Listing Files in a Directory

- os.listdir() returns a Python list containing the names of the files and subdirectories in the directory given by the path argument:
- Example:>>>os.listdir('my_directory/')

THANK YOU