# File Input and Output

Session 05

# Session Overview

In this session, you will be able to:

- Explain how to access, read, and write files in Python
- Describe pickle and the process to store the objects of Python in files

# File Related Operations

In Python, files are also considered as a type of object.

File objects serve as the main interface of the Python code to external files on the system.

They can be used to read and write text memos, Excel documents, audio clips, saved e-mail messages, and so on that are stored on the computer.

# Accessing a File (1-9)

Create a text output file by providing name and 'w' mode to write data:

**Code Snippet**:

```
>>> f = open('data.txt', 'w') #
create a new file to write
>>> f.write('Hello\n’) # Write
strings
>>> f.write('world\n')
>>> f.close() # Close the file and
flush output
```

# Accessing a File (2-9)

Know the current directory where the files are stored:

**Code Snippet A :**

```
>>> import os
>>> print (os.getcwd())
```

# Accessing a File (3-9)

**Code Snippet B**:

```
>>> import os
>>> os.chdir
('/path_of_the_directory_of_your_Choice')
>>> f = open('data.txt') # The default
processing mode is 'r'
>>> text = f.read() # This reads the
complete file in to a string
>>> text
>>> print(text) # print interprets control
characters
```

# Accessing a File (4-9)

File Modes:

| Mode | Description |
|---|---|
| 'r' | Opens a file for reading (default). |
| 'w' | Opens a file for writing. |
| 'x' | Opens a file for exclusive creation. |

# Accessing a File (5-9)

| Mode | Description |
|---|---|
| 'a' | Opens the file for appending content at the end of it and without truncating existing content. |
| 't' | Opens a file in text mode (default). |
| 'b' | Opens a file in binary mode. |
| '+' | Opens a file for updating (reading and writing r+). |

# Accessing a File (6-9)

Open and write data to a file:

**Code Snippet A**:

```
>>> file = open('data_new.txt', 'r')
```

**Code Snippet B**:

```
>>> f = open('data.txt','w')
>>> f.write('Hello\n')
>>> f.write('world\n')
>>> f.close()
```

# Accessing a File (7-9)

Data Read and Write Operations:

**Code Snippet A:**

```
>>> f=open('data.txt')
>>> text = f.read()
>>> print (text)
```

**Code Snippet B:**

```
>>>f2 = open('data.txt', 'w')
>>> f2.write('How are you ')
>>> f2.close()
```

**Code Snippet C:**

```
>>> f2= open('data.txt')
>>> text = f2.read()
>>> print (text)
```

# Accessing a File (8-9)

**Code Snippet A :**

```
>>> f= open('data.txt','w')
>>> f.write('Hello\n')
>>> f.write('world\n')
>>> f.close()
```

**Code Snippet B:**

```
# Open a file and check its entries
>>> f= open('data.txt')
>>> text= f.read()
>>> print(text)
```

**Code Snippet C:**

```
# The file contains Hello
# world
# Open the file in append
mode and add more strings to
it and close it
>>> f2=open('data.txt','a')
>>> f2.write('how are you\n')
>>> f2.close()
```

**Code Snippet D:**

```
#opening the file and printing the
content
>>> f2 =open('data.txt')
>>> text=f2.read()
>>> print (text)
```

# Accessing a File (9-9)

Opening file in text and binary mode:

**Code Snippet A**:

```
f = open ('data.txt','rb')
>>> text = f.read()
>>> print (text)
```

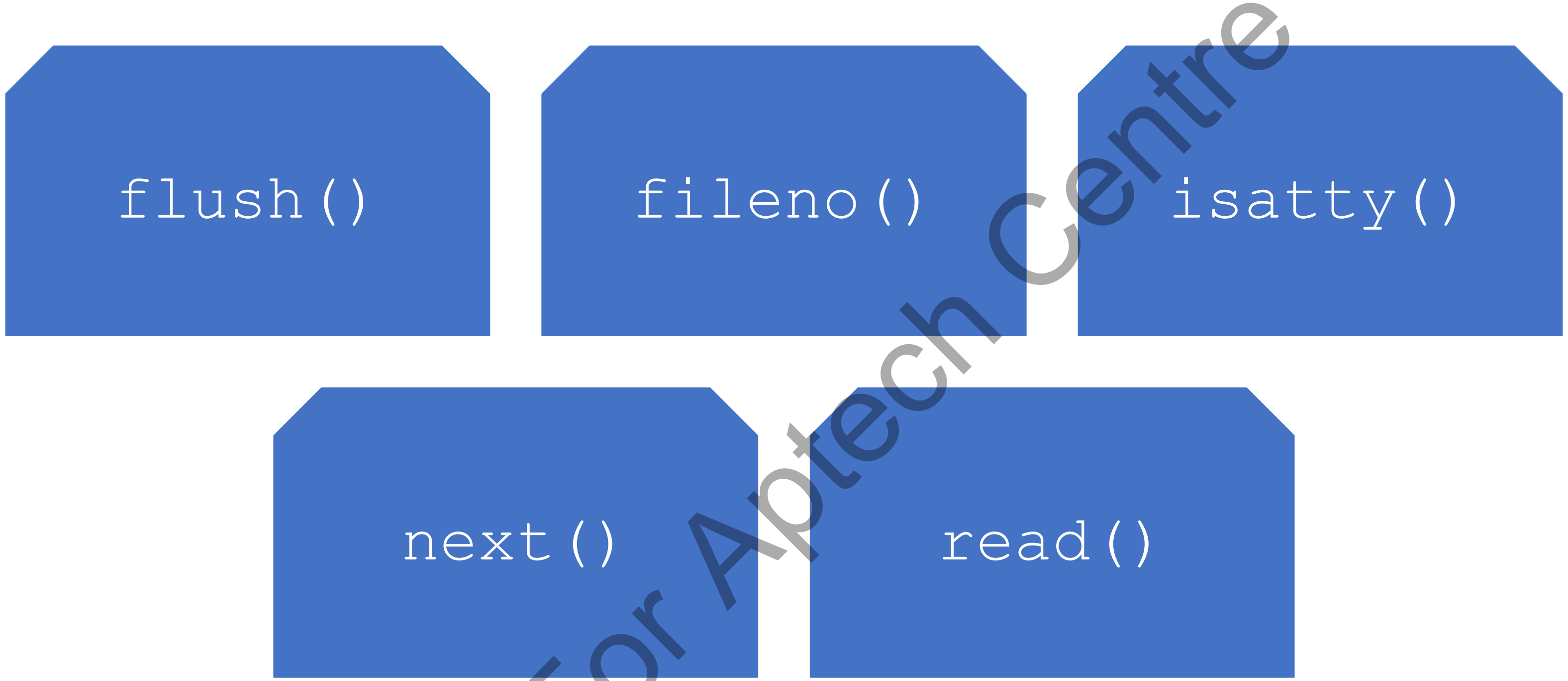
# Closing a File

Closing a file helps to release all the resources that were associated with the file and this is achieved using the `close()` method.

Though Python contains a garbage collector that cleans up unreferenced objects, one must not depend on it to close the file.

# Other File Methods (1-2)

Methods applied to files:



`flush()`

`fileno()`

`isatty()`

`next()`

`read()`

# Other File Methods (2-2)

`readline(size)`

`seek()`

`tell()`

`truncate(size)`

`write(str)`

`writelines(sequence)`

# Saving Objects in File

Ways to put different objects into a file:

- **Numbers:** `>>> X, Y, Z = 43, 44, 45`
- **Strings:** `>>> strng = 'Spam' # Must be strings to store in file`
- **Dictionaries:** `>>> dctnry = {'a': 1, 'b': 2}`
- **Lists:** `>>> lst = [1, 2, 3]`

# Storing Native Python Objects: pickle (1-3)

- Using the `eval()` function to convert strings to objects is a powerful tool.
- If there is a need to store native Python objects, but the source of the data in the file cannot be trusted, then pickle module is ideal.
- The pickle module is a cutting-edge tool that helps users to save most of the Python objects in a file directly.
- It does not require any to-string or from-string conversions.

# Storing Native Python Objects: pickle (2-3)

Pickle uses the following two main methods:

- Dump, which dumps an object to a file object.
- Load, which loads an object from a file object.

# Storing Native Python Objects: pickle (3-3)

Instances when pickling is used:

- Saving the state data of a program to a disk to make sure that it can continue from where it left off when restarted (persistence)
- Transmitting Python data through a TCP connection in a multi-core or distributed system (marshalling)
- Saving Python objects in a database
- Converting a random Python object to a string to make it usable as a dictionary key (for example, for caching and memoization)

# Storing Python Objects in JSON Format

- Though pickle module translates random Python objects to a format developed particularly for Python.
- On the other hand, JavaScript Object Notation (JSON) is the latest and evolving data interchange format.
- It is a programming-language-neutral and is supported by a range of systems.
- Though JSON does not support a wide range of Python object types, its advantage comes through its portability feature.

# Summary (1-4)

- File objects serve as the main interface of the Python code to external files on the system.
- To create a file object, use the built-in open function by providing an external filename and a processing mode in the form of strings, which is optional.
- There are different modes other than read and write in which a file can be opened.

# Summary (2-4)

- When an attempt is made to open a file, which does not exist in the directory in the read mode, `‘File Not Found’` error is displayed.
- Closing a file helps to release all the resources that were associated with the file and this is achieved using the `close()` method.
- The different methods that can be applied to files are `flush()`, `fileno()`, `isatty()`, `next()`, `read()`, `readline(size)`, `seek()`, `tell()`, `truncate(size)`, `write(str)` and `writelines(sequence)`.

# Summary (3-4)

- Python allows users to save a program’s object in the file for future use. One way to do this is by conversion. It is important to convert objects to strings with the help of conversion tools to save into file.
- If there is a need to store native Python objects, but the source of the data in the file cannot be trusted, then the standard library of Python called pickle module is ideal.

# Summary (4-4)

- JavaScript Object Notation (JSON) is the latest and evolving data interchange format. It is programming-language-neutral and is supported by a range of systems.