File Handling

Mohammad Tasin (Tasin Coder)

Agenda

- Introduction to File Handling
- Types of files
- File opening modes
- Write, append, modify, read, search
- Removing and renaming or delete file data
- Various attributes office
- Pickle

Introduction to File Handling

- File handling is a process of reading and writing data to a file.
- In programming, this process is used to store and retrieve data in a computer.
- It is a fundamental concept in any programming language, including Python and C++.
- In such cases you need to store your data in secondary storage.

Types of files

- There are two types of File :-
  1. Text File
  2. Binary File

- Text File :- 👇
  - It stores data as a sequence of characters.
  - Data is stored in the form of ASCII characters.
  - Text files are organized around lines, each of which ends with a new line character ('\n')
  - Even the •py file (source file) is a text file

Binary File

- **Binary File :-** 👇
  - **Data is stored as a sequence of bytes.**
  - **It is normally used to store non - textual information.**

File opening modes

1. Step : open a file → f = open(FileName, OpeningMode)
2. Step : Read / Write
3. Step : close a file → f.close()

- r → FileNotFoundError when file not available. Read only.
- w → Create new empty file if not exists. Erase old data, write only.
- a → Append data , create if not exists write only.
- r+ → FileNotFoundError when file not available. Read and write.
- w+ → Write and Read . Create file when not exist . Erase old data
- a+ → Append and Read . Create file ' when not exist

Write

```python
def Write_in_File(File_Name,Data): # Type 1
    with open(File_Name,'w') as file:
        file.write(Data)

def Write(File_Name,Data): # Type 2
    file = open(File_Name,'w')
    file.write(Data)
    file.close()
```

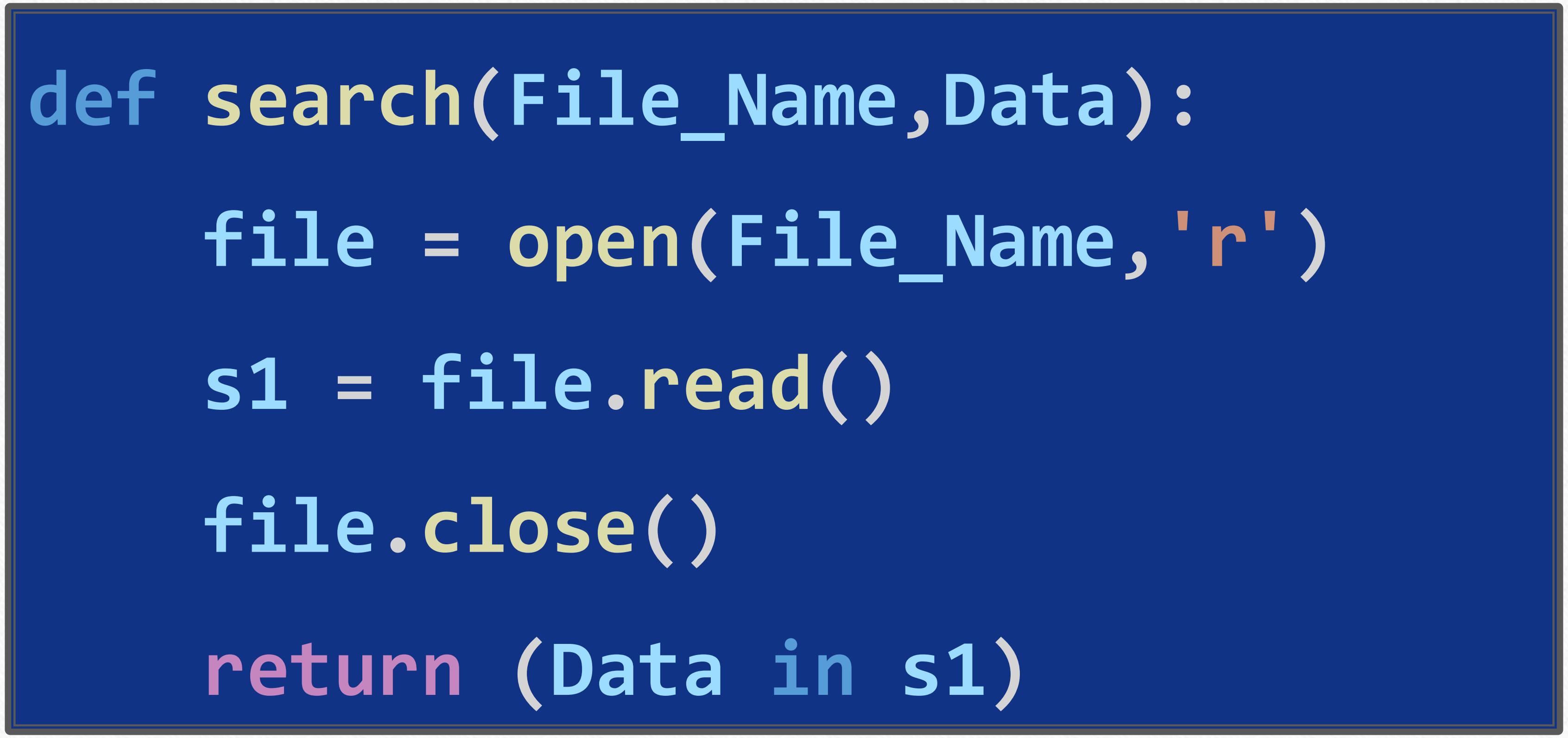
append

```
def Append(File_Name,Data):
    with open(File_Name,'a') as file:
        file.write(Data)
```

read

```
def Read(File_Name):
    try:
        with open(File_Name,'r') as file:
            data = file.read()
            return data
    except FileNotFoundError:
        print("FileNotFoundError")
```

search



```python
def search(File_Name,Data):
    file = open(File_Name,'r')
    s1 = file.read()
    file.close()
    return (Data in s1)
```

modify

```python
def Modify(FileName, oldWord, newWord):
    s1 = ''
    try:
        with open(FileName,'r') as file :
            s1 = file.read()
            s1 = s1.replace(oldWord,newWord)
    except FileNotFoundError:
        print("FileNotFoundError")
    with open(FileName,'w') as file :
        file.write(s1)
```

delete file data

```python
def Delete(FileName, oldWord):
    newWord = ''
    s1 = ''
    try:
        with open(FileName,'r') as file :
            s1 = file.read()
            s1 = s1.replace(oldWord,newWord)
    except FileNotFoundError:
        print("FileNotFoundError")
    with open(FileName,'w') as file :
        file.write(s1)
```

Removing and Renaming a file

```
import os

os.rename("Handling.txt","Handl.txt")

os.remove("Handl.txt")
```

Various attributes of file object

Variables

- f.name
- f.mode
- f.closed

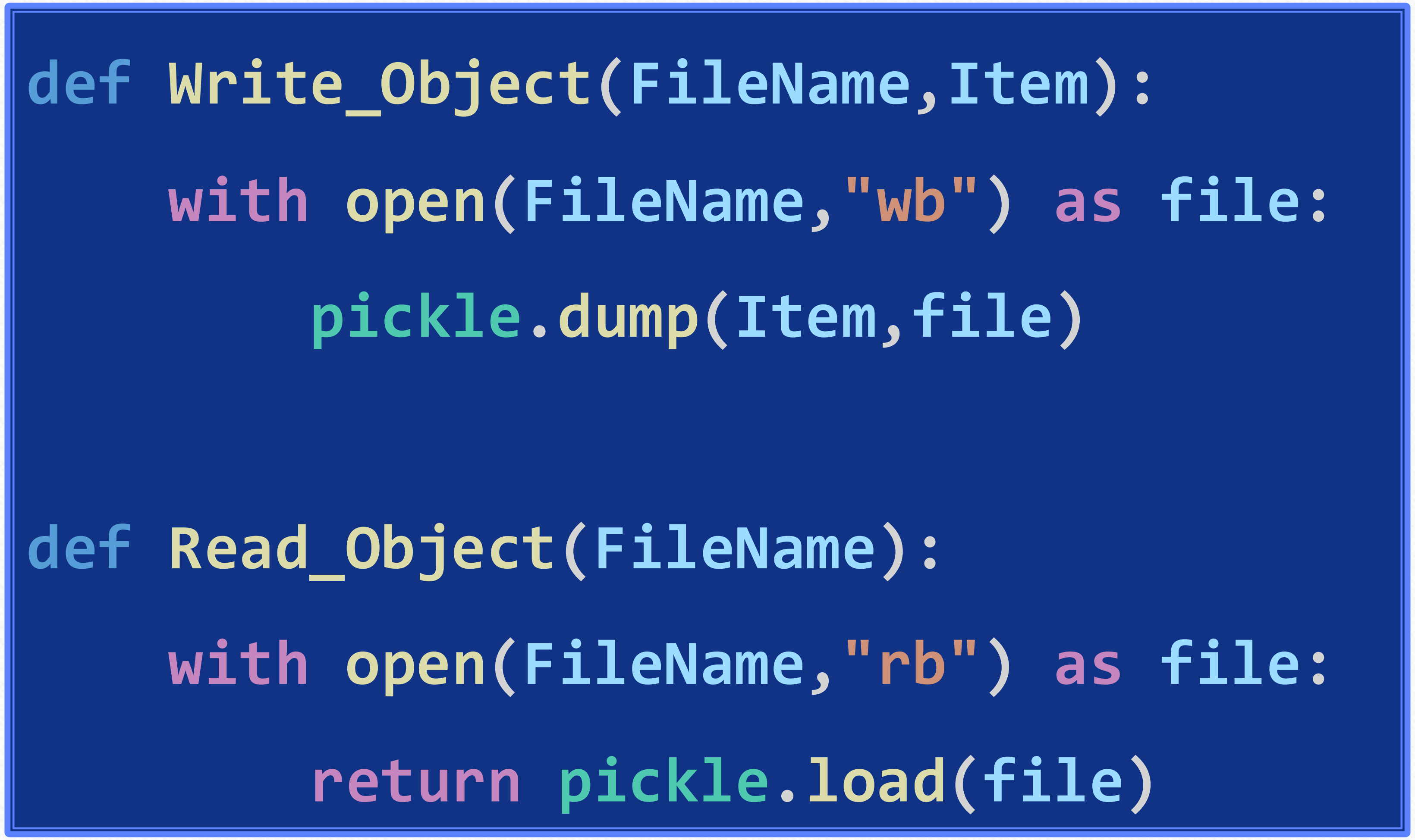
Functions

- f.write(text)
- f. writelines(list of lines)
- f. read()
- f. read (n)
- f.readline()
- f.readlines()

Pickle

- pickle is a python module
- pickling is a way to convert a Python object into a character stream.
- For pickling and unpicking, file opening mode should be binary that is rb, wb, ab, rb+, wb+, ab+

dump load methods



```
def Write_Object(FileName,Item):
    with open(FileName,"wb") as file:
        pickle.dump(Item,file)


def Read_Object(FileName):
    with open(FileName,"rb") as file:
        return pickle.load(file)
```