Lecture #12
CSE-2040 Programming IV

Transforming Education. Enriching Lives

Sin Thi Yar Myint

Professor

Faculty of Computer Science

MIIT

What will be cover today

- File

File

- A place on the disk with certain name to store related information (store in permanent memory known as hard disk or RAM). A file is a single entity pertains to a collection of related data structure.

OPERATIONS ON A FILE

- Open a file
- • Read or write
- • Close the file

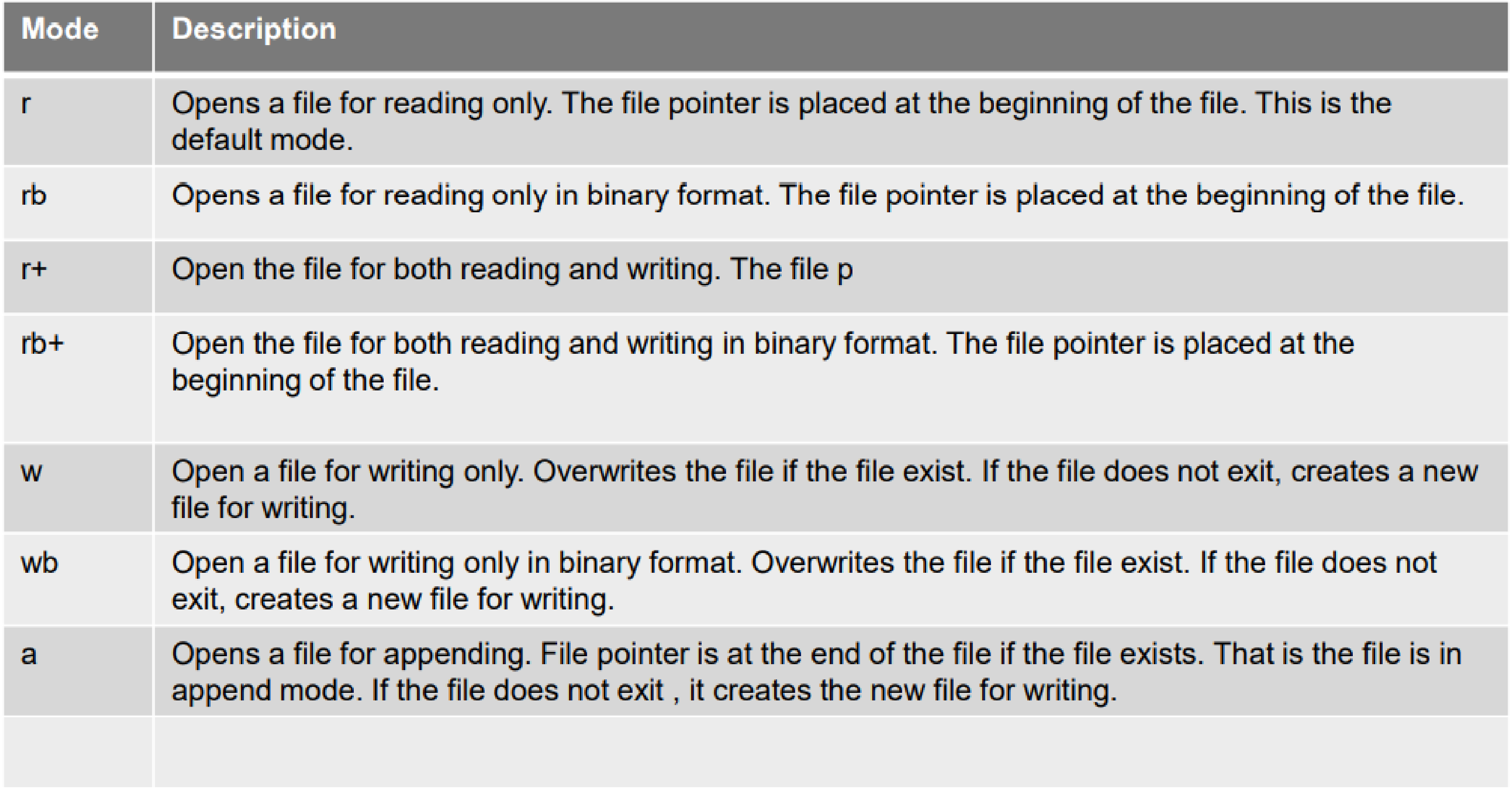
Opening a file

- The open() function
- contains three parameters, which are elaborated as under: File_name: Acess_mode: Buffering:
- fp=open(“data.txt”) fp=open(“C:/Pythonpgros/data.txt”)
- *Object open(fileName(M), accessMode(O));*

Cont’d

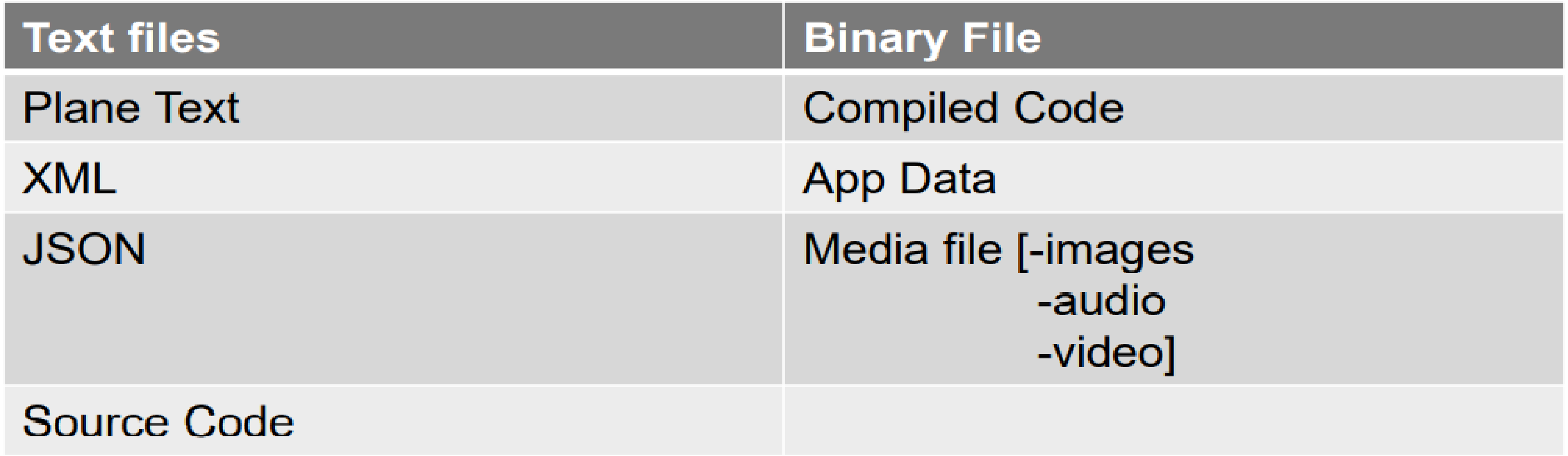
- (‘r’) – Read mode (default), open fails if the file does not exist.
- (‘w’) – Write mode, open a new file, if it does not exist. Else remove all contents of the existing file.
- (‘x’) – Exclusive creation, Fails if a file exists. Useful when always want to open a new file. So that the accidentally existing file is not overwritten or truncated.
- (‘a’) – Append mode, opens a new file, if it does not exist. Else added the new content to the end of the file without altering file contents.
- (‘t’) – Text mode (default)
- (‘b’) – Binary mode
- (‘+’) – Open a file for read and write mode.

File Mode



Mode	Description
r	Opens a file for reading only. The file pointer is placed at the beginning of the file. This is the default mode.
rb	Opens a file for reading only in binary format. The file pointer is placed at the beginning of the file.
r+	Open the file for both reading and writing. The file p
rb+	Open the file for both reading and writing in binary format. The file pointer is placed at the beginning of the file.
w	Open a file for writing only. Overwrites the file if the file exist. If the file does not exit, creates a new file for writing.
wb	Open a file for writing only in binary format. Overwrites the file if the file exist. If the file does not exit, creates a new file for writing.
a	Opens a file for appending. File pointer is at the end of the file if the file exists. That is the file is in append mode. If the file does not exit , it creates the new file for writing.

File Types



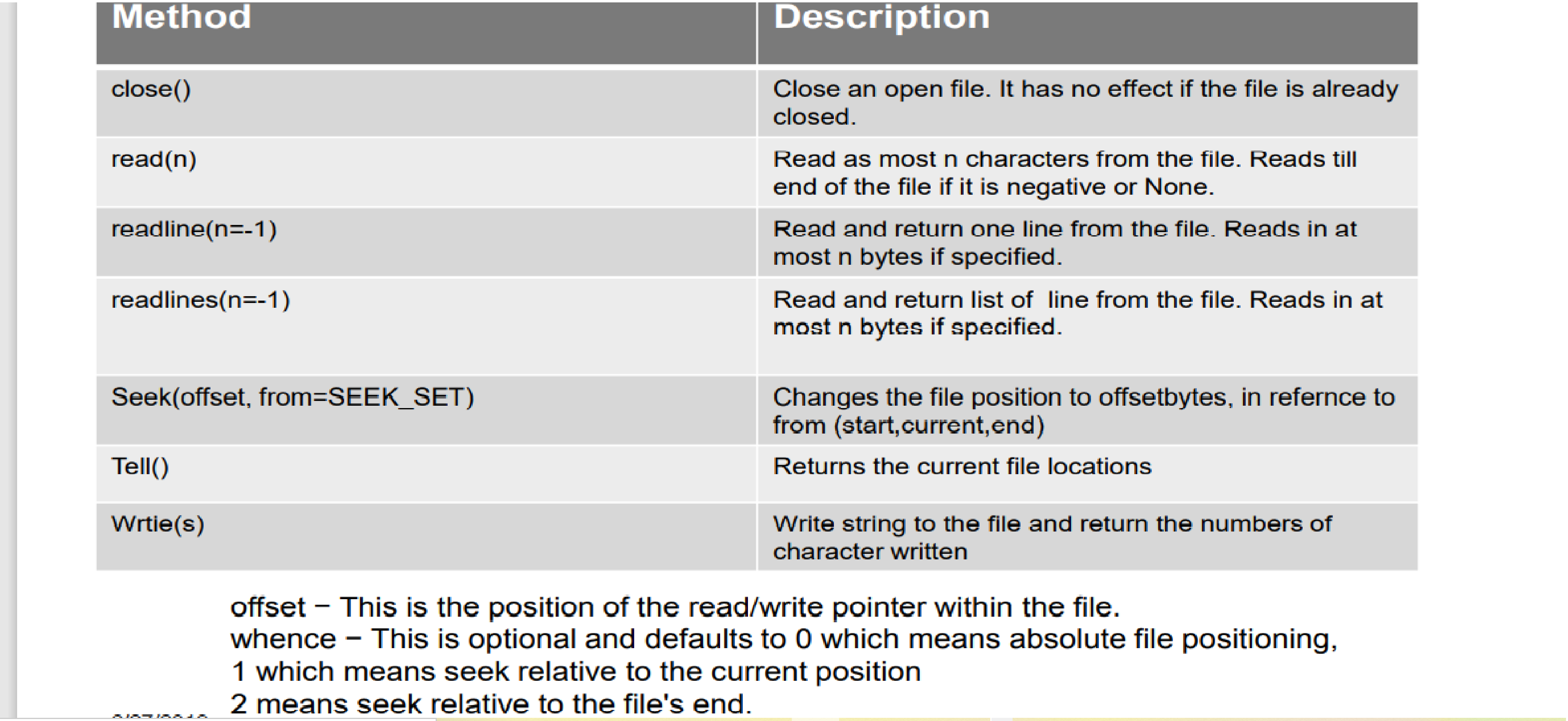
Text files	Binary File
Plane Text	Compiled Code
XML	App Data
JSON	Media file [-images -audio -video]
Source Code	

Encoding Scheme

- It is recommended to specify encoding type.
- File are store in bytes on the disk
- Need to encode them into str(txt) when we read into python.
- Default encoding scheme depend on platform.
  - Windows- ‘cp1252’ Linux-’utf-8’

  - Credit:https://docs.python.org/3/howto/unicode.html

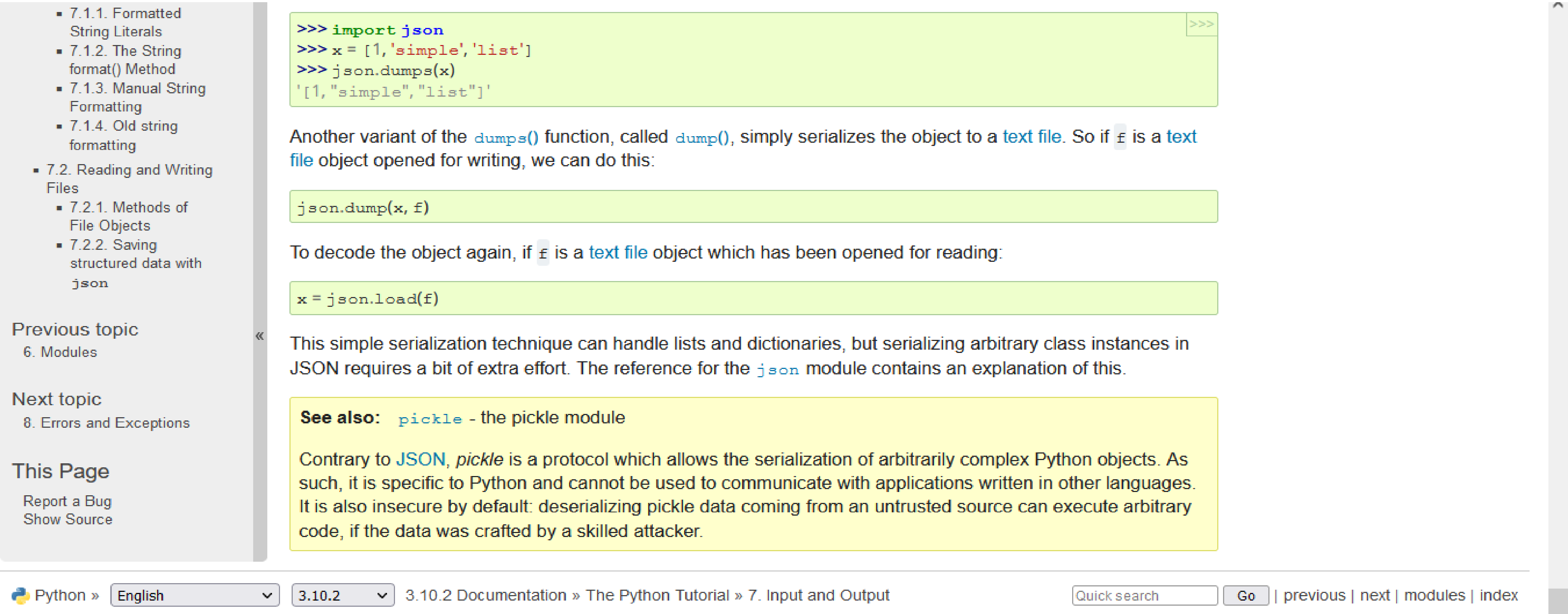
Method



Method	Description
close()	Close an open file. It has no effect if the file is already closed.
read(n)	Read as most n characters from the file. Reads till end of the file if it is negative or None.
readline(n=-1)	Read and return one line from the file. Reads in at most n bytes if specified.
readlines(n=-1)	Read and return list of line from the file. Reads in at most n bytes if specified.
Seek(offset, from=SEEK_SET)	Changes the file position to offsetbytes, in refernce to from (start,current,end)
Tell()	Returns the current file locations
Wrtie(s)	Write string to the file and return the numbers of character written

offset − This is the position of the read/write pointer within the file.
whence − This is optional and defaults to 0 which means absolute file positioning,
1 which means seek relative to the current position
2 means seek relative to the file's end.

JSON



Previous topic

6. Modules

Next topic

8. Errors and Exceptions

This Page

Report a Bug
Show Source

```
>>> import json
>>> x = [1, 'simple', 'list']
>>> json.dumps(x)
'[1, "simple", "list"]'
```

Another variant of the `dumps()` function, called `dump()`, simply serializes the object to a text file. So if `f` is a text file object opened for writing, we can do this:

```
json.dump(x, f)
```

To decode the object again, if `f` is a text file object which has been opened for reading:

```
x = json.load(f)
```

This simple serialization technique can handle lists and dictionaries, but serializing arbitrary class instances in JSON requires a bit of extra effort. The reference for the `json` module contains an explanation of this.

See also: `pickle` - the pickle module

Contrary to JSON, *pickle* is a protocol which allows the serialization of arbitrarily complex Python objects. As such, it is specific to Python and cannot be used to communicate with applications written in other languages. It is also insecure by default: deserializing pickle data coming from an untrusted source can execute arbitrary code, if the data was crafted by a skilled attacker.

Python »
English
3.10.2
3.10.2 Documentation » The Python Tutorial » 7. Input and Output
Quick search
Go
| previous | next | modules | index

JSON

- JSON is a text serialization format (it outputs unicode text, although most of the time it is then encoded to utf-8), while pickle is a binary serialization format;

- JSON is human-readable, while pickle is not;

- JSON is interoperable and widely used outside of the Python ecosystem, while pickle is Python-specific;

- JSON, by default, can only represent a subset of the Python built-in types, and no custom classes; pickle can represent an extremely large number of Python types (many of them automatically, by clever usage of Python’s introspection facilities; complex cases can be tackled by implementing specific object APIs);

- Unlike pickle, deserializing untrusted JSON does not in itself create an arbitrary code execution vulnerability.

Pickle

Pickle-

The process of converting the class object into a byte stream that can be stored into a file

Object serialization

pickle.dumb(object,file)

Unpickle

A byte stream is converted in class object.

Reading the class objects from the file.

Deserialization

Object=pickle.load(file)

REFERENCES

- https://docs.python.org/3/tutorial/inputoutput.html#reading-and-writing-files
- https://docs.python.org/3/library/pickle.html#comparison-with-json
- Core Python Programming- Dr.R.Nageswaro Rao, second edition

Thank you

Successful and unsuccessful people do not vary greatly in their abilities. They vary in their desires to reach their potential. – *John Maxwell*