Python

File handling

File I/O

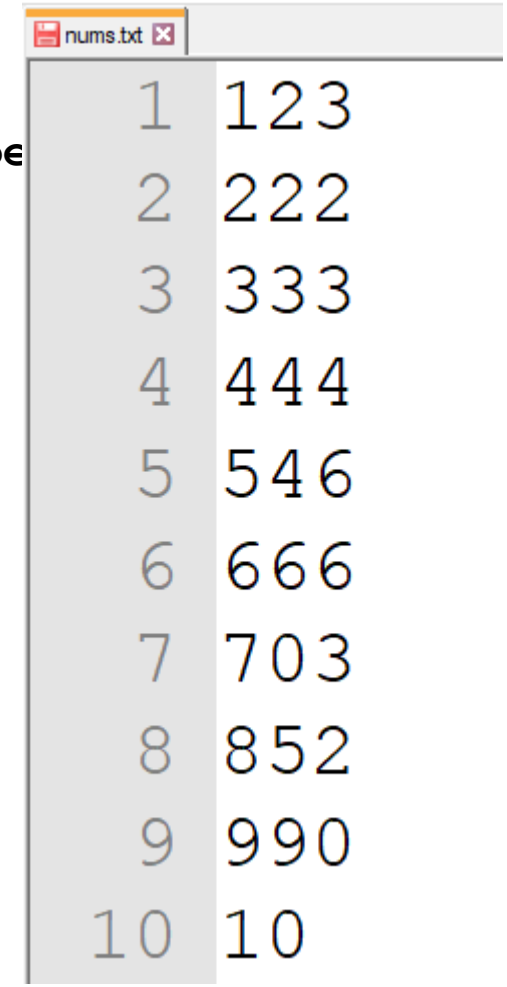
- Same **read** and **write** stream functions/operations of **sys** module are used to manipulate data in text files. For file output, even print can be used.
- Read and write functions can only process **strings**
- For files we have to
  - Open a file, before reading/writing data
  - Instructions for read or write data, as required
  - After reading/writing the desired data, close the file
- A variable (file variable) should be associated and used for above mentioned of processing files

File writing example

```
from random import randint

def main():
    numfile = "values.txt"
    ofile = open(numfile, "w")  # should be in try

    j = 0
    while j < 10:
        ofile.write(str(j) + ', ' +
                    str(randint(10, 10000)) + "\n")
        j = j + 1

    ofile.close()

    return

main()
```

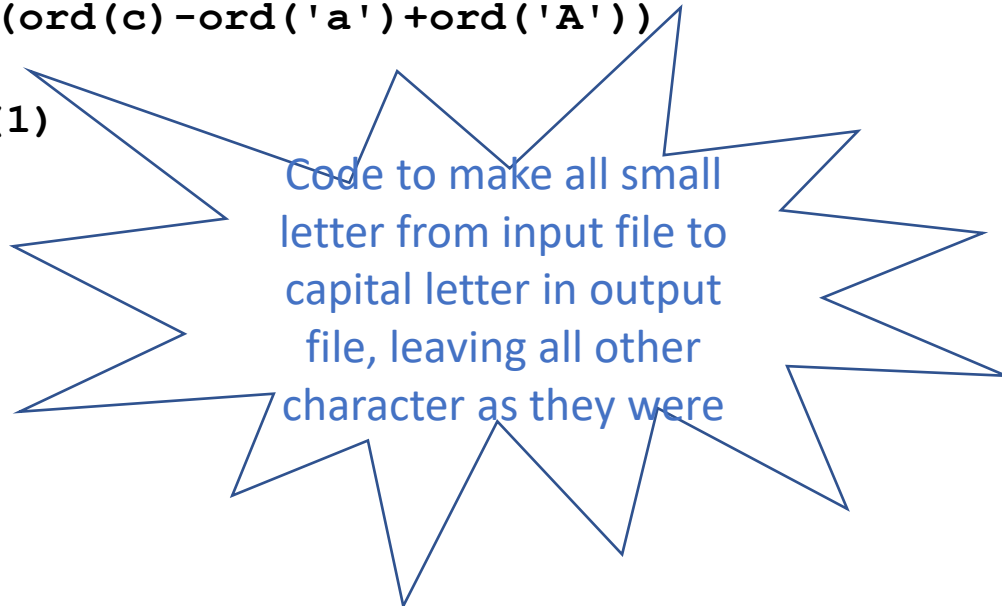
File reading example

```
def main():
        numfile = "nums.txt"
        ifile = open(numfile)   # should be

        a = [-1]*10

        print("Start")
        for j in range(10):
                a[j] = ifile.readline()
                print(a[j])   # comment it
        print("End")

        ifile.close()
        # print(a)   # uncomment it

        return 0

main()
```



Reading/writing in different files

```
def main():
        infile = input("Enter input file name: ")
        outfile = input("Enter output file name: ")

        ifile = open(infile)
        ofile = open(outfile, "w")
        c = ifile.read(1)
        while(c != ""):
                if c >= 'a' and c <= 'z':
                        c = chr(ord(c)-ord('a')+ord('A'))
                ofile.write(c)
                c = ifile.read(1)

        ifile.close()
        ofile.close()

        return

main()
```

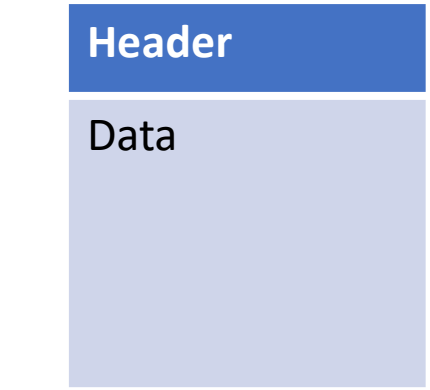


What is a file?

- A file is a sequence of bytes stored on storage media (disk, cd, flash drive, etc.). Every file in one folder has a unique name.
- Files are two types:
  - **Text files:** Contains formatted data, which is human readable, consists of alphabets, digits, punctuations, special characters, spaces, and new lines characters. Using any of commonly available text editors, one can view/edit data stored in text files.
  - **Binary files:** Contains application/software specific data. Only the specific software can use/display/modify data in binary files. E.g. Pictures stored on computer requires image viewing software to view them and image editing software is required to edit them. When these files are open in text editors, funny character are displayed and in general, humans are unable to read/understand them.

File structure/format

- File are used to store data, and reading/writing files generally requires no user interaction during the execution of the program.
- Data in the files generally has some structure/format. Which may be described in file itself in header area of file or in some other file or implicitly known to the programmer.
- Reading and writing data in files should strictly follow the structure/format of the file, otherwise later on file reading can results in errors.
- Some of the file related processing may be done without knowing the structure, e.g. counting lines in the file, counting characters in the file and capitalize small alphabets, etc.



BMP File Format
The 54 byte BMP header

offset	size	description
0	2	signature, must be 4D42 hex
2	4	size of BMP file in bytes (unreliable)
6	2	reserved, must be zero
8	2	reserved, must be zero
10	4	offset to start of image data in bytes
14	4	size of BITMAPINFOHEADER structure, must be 40
18	4	image width in pixels
22	4	image height in pixels
26	2	number of planes in the image, must be 1
28	2	number of bits per pixel (1, 4, 8, or 24)
30	4	compression type (0=none, 1=RLE-8, 2=RLE-4)
34	4	size of image data in bytes (including padding)
38	4	horizontal resolution in pixels per meter (unreliable)
42	4	vertical resolution in pixels per meter (unreliable)
46	4	number of colors in image, or zero
50	4	number of important colors, or zero

Well-known file formats

- There are variety of files in a computer system. Some of them, along with name of software that may manipulate them are mentioned below.
  - PDF: not possible to open in image viewer or text editor, only software recognize PDF file structure/format may view/edit it.
  - JPG: not possible to open in text editors, only software recognize JPG file structure/format may view/edit it. There are dozens of such software.
  - DOC or DOCX: generally Microsoft word is used to edit/view/print it.
  - Many examples, XLSX, PPTX, MP3, MP4, etc.
  - EXE: these files are created by compilers and operating system loads them to memory and executes them.
- Software capable recognizing structure of a file type can manipulate it.

Same file, different views

A file (without any change in stored bytes) may be understandable by different software and they treated data differently. E.g., A PPM (Portable Pixel Map) have following views using a Notepad++ and IrfanView software.

```
P2
# Shows the word "FEEP" (example from Netpbm man page on PGM)
24 7
15
0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
0  3  3  3  3  0  0  7  7  7  7  0  0 11 11 11 11  0  0 15 15 15 15  0
0  3  0  0  0  0  0  7  0  0  0  0  0 11  0  0  0  0  0 15  0  0 15  0
0  3  3  3  0  0  0  7  7  7  0  0  0 11 11 11  0  0  0 15 15 15 15  0
0  3  0  0  0  0  0  7  0  0  0  0  0 11  0  0  0  0  0 15  0  0  0  0
0  3  0  0  0  0  0  7  7  7  7  0  0 11 11 11 11  0  0 15  0  0  0  0
0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
```

Example (magnified)

```
P3
3 2
255
# The part above is the header
# "P3" means this is a RGB color image in ASCII
# "3 2" is the width and height of the image in pixels
# "255" is the maximum value for each color
# The part below is image data: RGB triplets
255   0   0     0 255   0     0   0 255
255 255   0   255 255 255     0   0   0
```

Image
(magnified)

https://en.wikipedia.org/wiki/Netpbm_format

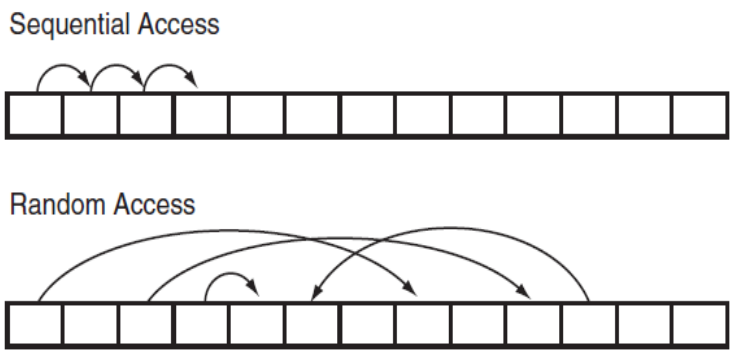
More file handling functions

- open, close, write, read and readline are already discussed and demonstrated
- readlines, writelines, truncate, tell, seek, flush, readable, writable, seekable
- File Attributes:
  - name, mode, encoding, closed
- File path:
  - open("C:/Documents/data.txt")
- OS Module:
  - open, close, remove, rename, getcwd, chdir, …

File handling modes and Random Access

Mode	Description
r	Opens a file for reading. (default)
w	Opens a file for writing. Creates a new file if it does not exist or truncates the file if it exists.
x	Opens a file for exclusive creation. If the file already exists, the operation fails.
a	Opens a file for appending at the end of the file without truncating it. Creates a new file if it does not exist.
t	Opens in text mode. (default)
b	Opens in binary mode.
+	Opens a file for updating (reading and writing)

Modes

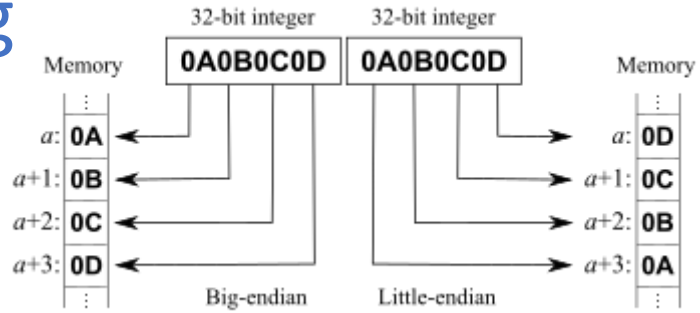


Sequential/Random Access

tell and seek functions and r+ mode

Binary File Handling

- str to byte, and vice versa
  - encode, decode
- int to bytes, , and vice versa
  - var.to_bytes(N, byteorder='big', signed=True) # N being number of bytes
  - type.from_bytes(fin.read(N), byteorder='little' , signed=False) # N being number of bytes
- struct Packing and Unpacking
- bytes and bytearray
- System Byte Order
  - sys.byteorder



https://www.devdungeon.com/content/working-binary-data-python
https://docs.python.org/3/library/struct.html