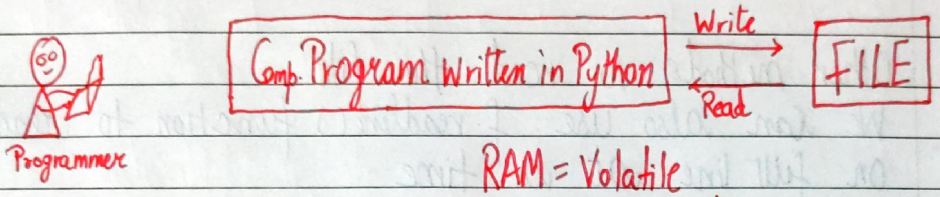
URBAN EDGE

Chapter 9 - File I/O

The random Access memory is volatile and all its contents are lost once a program terminates
In order to persist the data forever, we use files

A file is data stored in a storage device. A Python program can talk to the file by reading content from it and writing content to it



RAM = Volatile
HDD = Non Volatile

Types of Files

There are 2 types of files:

1. Text files (.txt, .c etc)
2. Binary files (.jpg, .dat, etc)

Python has a lot of functions for reading, updating and deleting files.

Opening a file

Python has an open() function for opening files. It takes 2 parameters: filename and mode.

```
open("this.txt", "r")
```

- open is a built-in function
- "this.txt" → Filename
- "r" → mode of opening (read mode)

Reading a file in python

```
f = open("this.txt", "r")  → open the file in r mode
text = f.read()            → Read its contents
print(text)                → Print its contents
f.close()                  → Close the file
```

We can also specify the number of characters in read() function : f.read(2)
↳ Reads first 2 characters

Other methods to read the file
We can also use f.readline() function to read on full line at a time

f.readline() → Reads one line from the file

Modes of opening a file

r → open for reading
w → open for writing
a → open for appending
+ → open for updating

'rb' will open for read in binary mode
'rt' will open for read in text mode

Writing files in Python

In order to write to a file, we first open it in write or append mode after which, we use the python's f.write() method to write to the file!

URBAN EDGE

```
f = open("this.txt", "w")
f.write("This is nice")
f.close()
```

→ Can be called multiple times

With statement

The best way to open and close the file automatically is the with statement

```
with open("this.txt") as f:
    f.read()
```

↳ Dont need to write f.close() as it is done automatically.