Binary Files

- **Read**:

Use the 'b' attribute when opening the file

```
img_file = open('img','rb')
img_data = img_file.read()
```

- **Write**:

Use the 'b' attribute when opening the file

```
img_file = open('img','wb')
img_file.write(img_data)
```

There may be times when textual data is not adequate for your needs, either because of limitations in functionality, or for performance reasons. Therefore, it may be useful to understand how to read and write binary data using Python.

Opening

In reading normal text files, you have been opening your files as follows:

```
file = open('myfile','r')
```

The second argument that is passed into the open function is the 'mode', which include 'r' for read and 'w' for write. By default, the file is assumed to be textual. However, if you want to open the file for binary reading or writing, you need to place a 'b' after the 'r' or 'w', which will tell Python that you are reading or writing a binary file.

To open a binary file for reading, the command would be:

```
file = open('bin_file','rb')
```

And to open a binary file for writing, the command would be:

```
file = open('bin_out_file','wb')
```

Reading and Writing

You will use the same generic `read()` and `write()` functions for writing binary data. Python takes care of the details for you.

```
bin_data = file.read()
file.write(bin_data)
```

Consider an example where, for every device, you must store the SSH key or some other token for accessing devices programmatically. After reading in the general information about each device, in textual format, the binary data must be read for each device, and attached to the device object. The following code will read the appropriate binary file and attach the binary data to the device:

```
        # Open SSH key file for this device
        key_file_path = "sshkeys/"+device_info['ip']+"/"+device_info['key']
        key_file = open(key_file_path,'rb')
        key_data = key_file.read()    # read ssh key data
        device.set_sshkey(key_data)   # store ssh key data in device object
```

The first two lines of code create the path to the key using the device IP address, and then open the file, using the mode `rb` for read-binary.

The third line reads the binary data into the variable key_data, and the fourth line references the device object, calling a method 'set_sshkey', passing the key_data, so that the SSH key data can be attached to the device object, for later access purposes.

Closing

It is best practices to close any files that you are finished working on. This ensures that any resources that are taken up by the open file are freed up. The command to do this is the `close()`.

```
key_file = open("info.txt", "r")
print key_file.read()
key_file.close()
```