Introduction to Files

In [3]:

```
fo = open("MyText.txt", "w")  # Opens file in r mode

print(fo, type(fo))

print ("Name of the file: ", fo.name) # Returns name of the file.
print ("Closed or not : ", fo.closed) # Returns true if file is closed, false otherwise.
print ("Opening mode : ", fo.mode) # Returns access mode with which file was opened.

fo.close()
```

```
<_io.TextIOWrapper name='MyText.txt' mode='w' encoding='cp1252'> <class '_i
o.TextIOWrapper'>
Name of the file:  MyText.txt
Closed or not :  False
Opening mode :  w
```

Opening modes

read mode ('r'), write mode ('w'), append mode ('a'),

Writing to a file

In [4]:

```
fo = open(r"C:\Users\Sia\Desktop\MyText.txt", "w")

fo.write("Hello Python")

fo.close()
```

Example

In [2]:

```
fo = open("IceCreamInventory.txt","w")

fo.write("Brand \t Flavor \t Price")

for i in range(3):
    s = "\n{} \t {} \t {}".format(input("Brand : "),input("Flavor : "),input("Price : "))
    fo.write(s)

fo.close()


# values = ["Brand : {} \t Flavor : {} \t Price : {}".format(input("Brand : "),input("Flavo
# fo.write("\n".join(values))

fo.close()
```

```
Brand : amul
Flavor : mango
Price : 50
Brand : naturals
Flavor : coconut
Price : 80
Brand : br
Flavor : choco
Price : 60
```

Read from a File

In [17]:

```
fo = open("IceCreamInventory.txt", "r")  # Opens file in r mode

# var = fo.read(6)

# print("Current cursor Location ", fo.tell())

# fo.seek(20)

# print("Current cursor Location ", fo.tell())

# var  = fo.readline()

# var  = fo.readlines()

# print(var)

fo.close()  # close the file object
```

```
['mango \tcandy \t50 \t\n', 'Choco Chips \tcombopacks \t90 \t\n', 'Buterscot
ch \tcups \t50 \t\n', 'American Nuts \tcombopacks \t120 \t\n', 'Almond Choco
bar \tcandy \t70 \t\n', 'Black current \tcups \t60 \t\n', 'Choco Brownie \tc
ombopacks \t130 \t\n', 'Strawberry \tcups \t40 \t\n', 'Chicoo \tcups \t60 \t
\n', 'malai kulfi \tcandy \t60 \t\n', 'Mango kulfi \tcandy \t80 \t\n', 'Stra
wberry \tcandy \t90 \t\n', 'Vanilla \tcups \t30 \t\n', 'Pista \tcombopacks
\t40 \t\n']
```

In [20]:

```
with open("IceCreamInventory.txt", "r") as f:
    data = f.readlines()

# print(data)

for line in data:
    print(line, end="")
```

```
mango   candy   50
Choco Chips     combopacks      90
Buterscotch     cups    50
American Nuts   combopacks      120
Almond Chocobar         candy   70
Black current   cups    60
Choco Brownie   combopacks      130
Strawberry      cups    40
Chicoo  cups    60
malai kulfi     candy   60
Mango kulfi     candy   80
Strawberry      candy   90
Vanilla         cups    30
Pista   combopacks      40
```

Ex1. Copy Contents from one file to another

In [ ]:

```
fr = open("ReadFile", "r")

var = fr.read()

fw = open("WriteFile","w")

fw.write(var)

fr.close()
fw.close()
```

Customer and Transaction example

In [10]:

```
with open("customers", "r") as f:
    data = f.readlines()
d = {}

for line in data:
    l = line.split(",")
    d[int(l[0])] = l[1] + " " +l[2]

print(d)
```

```
{4000001: 'Kristina Chung', 4000002: 'Paige Chen', 4000003: 'Sherri Melto
n', 4000004: 'Gretchen Hill', 4000005: 'Karen Puckett', 4000006: 'Patrick
Song', 4000007: 'Elsie Hamilton', 4000008: 'Hazel Bender', 4000009: 'Malc
olm Wagner', 4000010: 'Dolores McLaughlin', 4000011: 'Francis McNamara',
4000012: 'Sandy Raynor', 4000013: 'Marion Moon', 4000014: 'Beth Woodard',
4000015: 'Julia Desai', 4000016: 'Jerome Wallace', 4000017: 'Neal Lawrenc
e', 4000018: 'Jean Griffin', 4000019: 'Kristine Dougherty', 4000020: 'Cry
stal Powers', 4000021: 'Alex May', 4000022: 'Eric Steele', 4000023: 'Wesl
ey Teague', 4000024: 'Franklin Vick', 4000025: 'Claire Gallagher', 400002
6: 'Marian Solomon', 4000027: 'Marcia Walsh', 4000028: 'Dwight Monroe', 4
000029: 'Wayne Connolly', 4000030: 'Stephanie Hawkins', 4000031: 'Neal Mi
ddleton', 4000032: 'Gretchen Goldstein', 4000033: 'Tim Watts', 4000034:
'Jerome Johnston', 4000035: 'Shelley Weeks', 4000036: 'Priscilla Wilkerso
n', 4000037: 'Elsie Barton', 4000038: 'Beth Walton', 4000039: 'Erica Hal
l', 4000040: 'Douglas Ross', 4000041: 'Donald Chung', 4000042: 'Katherine
Bender', 4000043: 'Paul Woods', 4000044: 'Patricia Mangum', 4000045: 'Loi
s Joseph', 4000046: 'Louis Rosenthal', 4000047: 'Christina Bowden', 40000
48: 'Darlene Barton', 4000049: 'Harvey Underwood', 4000050: 'William Jone
s', 4000051: 'Frederick Baker', 4000052: 'Shirley Merritt', 4000053: 'Jas
```

In [12]:

```
with open("customers", "r") as f :
    data = f.readlines()

customers = list(map(lambda x : (x.strip()).split(","), data))
# print(customers)

l2 = list(filter(lambda x : int(x[3]) > 25 and int(x[3])<60 ,customers))
print(l2)
```

```
[['4000001', 'Kristina', 'Chung', '55', 'Pilot'], ['4000003', 'Sherri',
'Melton', '34', 'Firefighter'], ['4000006', 'Patrick', 'Song', '42', 'Vet
erinarian'], ['4000007', 'Elsie', 'Hamilton', '43', 'Pilot'], ['4000009',
'Malcolm', 'Wagner', '39', 'Artist'], ['4000011', 'Francis', 'McNamara',
'47', 'Therapist'], ['4000012', 'Sandy', 'Raynor', '26', 'Writer'], ['400
0013', 'Marion', 'Moon', '41', 'Carpenter'], ['4000015', 'Julia', 'Desa
i', '49', 'Musician'], ['4000016', 'Jerome', 'Wallace', '52', 'Pharmacis
t'], ['4000018', 'Jean', 'Griffin', '45', 'Childcare worker'], ['400002
1', 'Alex', 'May', '39', 'Environmental scientist'], ['4000023', 'Wesle
y', 'Teague', '42', 'Carpenter'], ['4000024', 'Franklin', 'Vick', '28',
'Dancer'], ['4000025', 'Claire', 'Gallagher', '42', 'Musician'], ['400002
6', 'Marian', 'Solomon', '27', 'Lawyer'], ['4000028', 'Dwight', 'Monroe',
'45', 'Economist'], ['4000029', 'Wayne', 'Connolly', '40', 'Real estate a
gent'], ['4000030', 'Stephanie', 'Hawkins', '50', 'Human resources assist
ant'], ['4000031', 'Neal', 'Middleton', '59', 'Civil engineer'], ['400003
3', 'Tim', 'Watts', '58', 'Lawyer'], ['4000034', 'Jerome', 'Johnston', '3
8', 'Childcare worker'], ['4000036', 'Priscilla', 'Wilkerson', '35', 'Agr
icultural and food scientist'], ['4000037', 'Elsie', 'Barton', '27', 'Chi
ldcare worker'], ['4000039', 'Erica', 'Hall', '33', 'Police officer'],
```

Country Example

In [ ]:

```
import string
from random import choice
import csv

with open("countries data.csv",'r',newline="") as file:
    countries = list(csv.reader(file, delimiter=","))

alphabets = string.ascii_uppercase

f = lambda x : [country[1] for country in countries if country[1][0] == x]
dict_alphabets = {x : f(x)  for x in alphabets}

name  = (input("Enter your name : ")).upper()

for letter in name:
    print(letter," - ", choice(dict_alphabets[letter]))
```

Storing Data

Many of your programs will ask users to input certain kinds of information. You might allow users to store preferences in a game or provide data for a visualization. Whatever the focus of your program is, you'll store the information users provide in data structures such as lists and dictionaries. When users close a program, you'll almost always want to save the information they entered. A simple way to do this involves storing your data using the json module.

The json module allows you to dump simple Python data structures into a file and load the data from that file the next time the program runs. You can also use json to share data between different Python programs. Even better, the JSON data format is not specific to Python, so you can share data you store in the JSON format with people who work in many other programming languages. It's a useful and portable format, and it's easy to learn. Not e The JSON (JavaScript Object Notation) format was originally developed for JavaScript. However, it has since become a common format used by many languages, including Python.

In [ ]:

```
import json

numbers = [2, 3, 5, 7, 11, 13]
filename = 'numbers.json'
with open(filename, 'w') as f_obj:
    json.dump(numbers, f_obj)
```

In [ ]:

```
import json

filename = 'numbers.json'
with open(filename) as f_obj:
    numbers = json.load(f_obj)
print(numbers)
```

The pickle module

In order to put values into a file, you have to convert them to strings. You have already seen how to do that with str

f.write (str(12.3))
f.write (str([1,2,3]))

The problem is that when you read the value back, you get a string. The original type information has been lost. In fact, you can't even tell where one value ends and the next begins:
f.readline()
'12.3[1, 2, 3]'

The solution is pickling, so called because it \preserves" data structures. The pickle module contains the necessary commands. To use it, import pickle and then open the file in the usual way:

In [ ]:

```
import pickle

f = open("pfile.pck","wb")
pickle.dump(12.3,f)
pickle.dump([1,2,3], f)
f.close()
```

In [ ]:

```
f = open("pfile.pck","rb")
print(pickle.load(f))
print(pickle.load(f))
```

The os Module

List all files in a directory in Python

In [ ]:

```
from os import listdir
from os.path import isfile, join
files_list = [f for f in listdir(r'D:\Teaching material\Python\RGIT') if isfile(join(r'D:\T
print(files_list);
```

Sort files by date

In [ ]:

```
import glob
import os

files = glob.glob("*.txt")
files.sort(key=os.path.getmtime)
print("\n".join(files))
```

Get a directory listing sorted by creation date

In [ ]:

```
from stat import S_ISREG, ST_CTIME, ST_MODE
import os, sys, time

#Relative or absolute path to the directory
dir_path = sys.argv[1] if len(sys.argv) == 2 else r'.'

#all entries in the directory w/ stats
data = (os.path.join(dir_path, fn) for fn in os.listdir(dir_path))
data = ((os.stat(path), path) for path in data)

# regular files, insert creation date
data = ((stat[ST_CTIME], path)
           for stat, path in data if S_ISREG(stat[ST_MODE]))

for cdate, path in sorted(data):
    print(time.ctime(cdate), os.path.basename(path))
```

Example : Check if file exists or not

In [ ]:

```
import os.path
open('abc.txt', 'w')
print(os.path.isfile('abc.txt'))
```

Write a Python program to check if a file path is a file or a directory.

In [ ]:

```
import os
path="abc.txt"
if os.path.isdir(path):
    print("\nIt is a directory")
elif os.path.isfile(path):
    print("\nIt is a normal file")
else:
    print("It is a special file (socket, FIFO, device file)" )
print()
```

Write a Python program to get the size of a file.

In [ ]:

```
import os
file_size = os.path.getsize("abc.txt")
print("\nThe size of abc.txt is :",file_size,"Bytes")
```