Why files?

When you want to work with a file, the first thing to do is to open it. This is done by invoking the open() built-in function.

open() has a single return: the file object.

It’s important to remember that it’s your responsibility to close the file. In most cases, upon termination of an application or script, a file will be closed eventually. However, there is no guarantee when exactly that will happen.

open()

The key function for working with files in Python is the open() function.

The open() function takes two parameters; filename, and mode.

There are four different methods (modes) for opening a file:

"r" - Read - Default value. Opens a file for reading, error if the file does not exist

"a" - Append - Opens a file for appending, creates the file if it does not exist

"w" - Write - Opens a file for writing, creates the file if it does not exist

"x" - Create - Creates the specified file, returns an error if the file exists

"r+" - for both reading and writing

In addition you can specify if the file should be handled as binary or text mode

"t" - Text - Default value. Text mode

"b" - Binary - Binary mode (e.g. images)

https://docs.python.org/3/tutorial/inputoutput.html#reading-and-writing-files (https://docs.python.org/3/tutorial/inputoutput.html#reading-and-writing-files)

In [1]:

```
f = open("demo.txt")
# Operations
f.close()

# This is same because read (r) and text (t) is default mode.
# f = open("demofile.txt", "rt")
```

In [2]:

```
# It is a good practice to always close the file when you are done with it.

try:
    x = 0
    f= open("demo.txt")
    # Some operations
except:
    print('error')
else:
    print("File Closed")
    f.close()
```

```
File Closed
```

In [3]:

```python
# no need of finally block
with open("demo.txt") as f:
    for i in f:
        print(i)
    # data processing
    # data processing

# file is closed here
```

```
Lorem Ipsum is simply dummy

text of the printing and typesetting

industry. Lorem Ipsum has been the

industry's standard dummy text ever

since the 1500s, when an unknown

printer took a galley of type and

scrambled it to make a type specimen

book. It has survived not only five

centuries, but also the

leap into electronic typesetting,

remaining essentially unchanged.
```

In [4]:

```
# Printing each line in file:
with open("demo.txt") as my_file:
    for i in my_file:
        print(i)
```

```
Lorem Ipsum is simply dummy

text of the printing and typesetting

industry. Lorem Ipsum has been the

industry's standard dummy text ever

since the 1500s, when an unknown

printer took a galley of type and

scrambled it to make a type specimen

book. It has survived not only five

centuries, but also the

leap into electronic typesetting,

remaining essentially unchanged.
```

Reading contents from file:

- read() takes a number as a parameter and reads from the file based on that number of bytes. If no argument is passed, or None or -1 is passed, then the entire file is read.
- readline() takes a number as a parameter and reads at most that number of characters from the line. This continues to the end of the line and then wraps back around. If no argument is passed, or None or -1 is passed, then the entire line (or rest of the line) is read.
- readlines() reads the remaining lines from the file object and returns them as a list.

In [5]:

```
# read()

with open("demo.txt") as f:
    print(f.read(20))
```

```
Lorem Ipsum is simpl
```

In [6]:

```
# readline()
with open("demo.txt") as f:
    for i in f:
        data = f.readline(10)
        print(data)
```

```
text of th
industry.
industry's
since the
printer to
scrambled
book. It h
centuries,
leap into
remaining
```

In [7]:

```
# readline()
with open("demo.txt") as f:
    data = f.readline(10)
    print(data, type(data))
```

```
Lorem Ipsu <class 'str'>
```

In [8]:

```
# By calling readline() two times, you can read the two first lines

with open("demo.txt") as f:
    data = f.readline()
    print(data)
    print(f.readline())
```

```
Lorem Ipsum is simply dummy

text of the printing and typesetting
```

In [9]:

```
# By looping through the lines of the file, you can read the whole file, line by line.

with open("demo.txt") as f:
    for i in f:
        print(i, type(i))
```

```
Lorem Ipsum is simply dummy
 <class 'str'>
text of the printing and typesetting
 <class 'str'>
industry. Lorem Ipsum has been the
 <class 'str'>
industry's standard dummy text ever
 <class 'str'>
since the 1500s, when an unknown
 <class 'str'>
printer took a galley of type and
 <class 'str'>
scrambled it to make a type specimen
 <class 'str'>
book. It has survived not only five
 <class 'str'>
centuries, but also the
 <class 'str'>
leap into electronic typesetting,
 <class 'str'>
remaining essentially unchanged. <class 'str'>
```

In [10]:

```python
# If you want to read all the lines of a file in a list you can also use list(f) or f.readlines(

with open("demo.txt") as f:
    data= f.readlines()
    print(data, type(data))
```

```
['Lorem Ipsum is simply dummy \n', 'text of the printing and typesetting \n', 'industry. Lorem Ipsu
m has been the \n', "industry's standard dummy text ever \n", 'since the 1500s, when an unknown
\n', 'printer took a galley of type and \n', 'scrambled it to make a type specimen \n', 'book. It h
as survived not only five \n', 'centuries, but also the \n', 'leap into electronic typesetting,
\n', 'remaining essentially unchanged.'] <class 'list'>
```

Writing to an existing file

To write to an existing file, you must add a parameter to the open() function:

"a" - Append - will append to the end of the file

"w" - Write - will overwrite any existing content

"r+"

In [11]:

```python
with open("demo.txt","a") as f:
    f.write("Content appended")
#     print(f.readlines(), type(data))
```

In [12]:

```python
with open("demo.txt","a") as f:
    f.write("\nContent appended with new line character")
```

In [13]:

```python
with open("demo.txt","r") as f:
    print(f.read())
```

```
Lorem Ipsum is simply dummy
text of the printing and typesetting
industry. Lorem Ipsum has been the
industry's standard dummy text ever
since the 1500s, when an unknown
printer took a galley of type and
scrambled it to make a type specimen
book. It has survived not only five
centuries, but also the
leap into electronic typesetting,
remaining essentially unchanged.Content appended
Content appended with new line character
```

In [18]:

```python
with open("demo.txt","r+") as f:
    f.write("Added new line using r+ mode\n")
    print(f.read())
```

```
text of the printing and typesetting
industry. Lorem Ipsum has been the
industry's standard dummy text ever
since the 1500s, when an unknown
printer took a galley of type and
scrambled it to make a type specimen
book. It has survived not only five
centuries, but also the
leap into electronic typesetting,
remaining essentially unchanged.Content appended
Content appended with new line characterAdded new line using r+ mode
```

In [19]:

```python
with open("demo.txt","w") as f:
    f.write("Content overwritten\n")
```

In [20]:

```python
with open("demo.txt","r") as f:
    print(f.read())
```

```
Content overwritten
```

Delete file:

https://www.w3schools.com/python/python_file_remove.asp (https://www.w3schools.com/python/python_file_remove.asp)

Question 1

In [21]:

```
# Assume a file city.txt with details of 5 cities in the given format:
# cityname  population(in lakhs) area(in sq KM) ):
# Example:
# Dehradun 5.78 308.20
# Delhi 190 1484
# ..............
# Open file city.txt and read to:
# Display details of all cities  [3]
# Display city names with population more than 10Lakhs [4]
# Display sum of areas of all cities  [3]
```

In [22]:

```
with open('city.txt','r') as f:

    print('Displaying details of all cities'.center(70,'-'))
    for line in f:
        print(line, end='')

    f.seek(0)

    print('\n' + 'Displaying city names with population more than 10 Lakhs'.center(70,'-'))
    l = f.readlines()
    data = []
    for i in range(5):
        data.append(l[i].rstrip('\n').split(' '))

    for i in data:
        if(float(i[1])>10):
            print(i[0])

    print('\n' + 'Displaying sum of areas of all cities'.center(70,'-'))
    sum = 0.0
    for i in data:
        x = float(i[2])
        sum = sum + x
    print('Sum of areas of all cities = {} sq Km'.format(sum))
```

```
-------------------Displaying details of all cities-------------------
Dehradun 5.78 308.20
Rishikesh 190 1484
Chennai 240 2005
Lucknow 30.58 1576
Haridwar 8 1500

-------Displaying city names with population more than 10 Lakhs-------
Rishikesh
Chennai
Lucknow

----------------Displaying sum of areas of all cities-----------------
Sum of areas of all cities = 6873.2 sq Km
```

Question 2

Question 2

In [23]:

```
# Assuming suitable data create a file “temp.txt” which stores the sales of 10 products quarterl
# format where sales_first is no of sales in the first quarter:
# Productname sales_first sales_second sales_third sales_fourth

# e.g.  TV 45 78 89 90
#         mobile 123 678 781 772
# ........
# Write python script to:
# Display all product details
# Find the average sale of all products
# Find a product with maximum sales
```

In [24]:

```python
from collections import defaultdict

with open('temp.txt','r') as f:

    print('Displaying all product details'.center(70,'-'))
    for line in f:
        print(line, end='',sep='\t')

    f.seek(0)
    l = f.readlines()
    data = []
    for i in range(10):
        data.append(l[i].rstrip('\n').split(' '))

    # Find the average sale of all products
    print('\n' + 'Displaying average sales of all products'.center(70,'-'))
    for i in data:
        print('Average sale of {} = {}'.format(i[0],((int(i[1]) + int(i[2]) + int(i[3]) + int(i[

    # Find a product with maximum sales
    d = defaultdict(int)
    for i in data:
        d[i[0]] = int(i[1]) + int(i[2]) + int(i[3]) + int(i[4])
    m = sorted(d.items(), key = lambda x: x[1])[-1]
    print('\nProduct having maximum sales is {}'.format(m[0]))
```

```
--------------------Displaying all product details--------------------
TV 10 20 30 15
Fridge 25 1 21 0
Microwave 1 2 3 4
AC 5 4 8 7
Cooler 2 5 12 6
Mobiles 12 32 21 14
Mixer 17 4 11 8
Grinder 13 6 5 11
Fan 15 1 6 14
Laptop 16 9 8 9

---------------Displaying average sales of all products---------------
Average sale of TV = 18.75
Average sale of Fridge = 11.75
Average sale of Microwave = 2.5
Average sale of AC = 6.0
Average sale of Cooler = 6.25
Average sale of Mobiles = 19.75
Average sale of Mixer = 10.0
Average sale of Grinder = 8.75
Average sale of Fan = 9.0
Average sale of Laptop = 10.5

Product having maximum sales is Mobiles
```

Question 3

In [ ]:

```
# Assume a file movie.txt with movie details, separated by spaces, in the given format:
# (movie_name director_first_name production_cost(in crores) Year_of_release ):

# Example:
# Lagaan Ashutosh 98 2001
# Dangal Nitesh 110 2016
# ..............
# Open file movie.txt and write a python script to:
# Count number of movies in the file.
# Add a new movie detail (War Amit 180 2019) at the end of the file.
# Display details of all movies where production cost is more than 80 Crores
# Display the first five movie details.
# Display director name who has worked in more than two movies.
```

In [39]:

```
from collections import Counter

with open('movie.txt','r+') as f:

    # Counting no of lines in file
    count = 0
    for line in f:
        count+=1
    print('No. of movies in the file is {}'.format(count))

    # Adding new movie details
#     f.write('\nWar Amit 180 2019')
    f.seek(0)

    count = 0
    for line in f:
        count+=1
    print('No. of movies in the file after write operation is {}'.format(count))
    f.seek(0)

    # All movies with production cost more than 80 crores
    print('\n'+'Displaying details of all movies where production cost is more than 80 Crores'.c

    l = f.readlines()
    data = []
    for i in range(count):
        data.append(l[i].rstrip('\n').split(' '))
    f.seek(0)

    for i in data:
        if(float(i[2])>80):
            print(*i)
    f.seek(0)

    # Displaying first 5 movie details
    print('\n' + 'Displaying details of first 5 movies'.center(90,'-'))
    for i in range(5):
        print(f.readline(),end='')

     # Displaying director who has worked in more than 2 films
    temp = []
    for i in data:
```

```python
        temp.append(i[1])
    c = Counter(temp)
    print('\n' + 'Displaying name of directors who have worked in more than 2 movies'.center(90,
    for x,y in c.items():
        if y>2:
            print(x)
```

```
No. of movies in the file is 11
No. of movies in the file after write operation is 11

------Displaying details of all movies where production cost is more than 80 Crores-------
Lagaan Ashutosh 98 2001
Dangal Nitesh 110 2016
Bahubali Rajmouli 400 2015
Bahubali2 Rajmouli 200 2017
Panipat Ashutosh 100 2019
Bhootnath Nitesh 150 2014
War Amit 180 2019

---------------------------Displaying details of first 5 movies---------------------------
Lagaan Ashutosh 98 2001
Dangal Nitesh 110 2016
K3G Karan 75 2000
Bahubali Rajmouli 400 2015
Drive Karan 50 2019

------------Displaying name of directors who have worked in more than 2 movies------------
Ashutosh
Karan
```

In [ ]: