**Reading data from csv file**

We can read data from csv file using reader() in csv module

csv.reader(csvfile, dialect='excel', delimeter= ?,\*\*fmtparams)

* Return a reader object which will iterate over lines in the given csvfile. csvfile can be any object which supports the iterator protocol and returns a string each time its \_\_next\_\_() method is called
* *If csvfile is a file object, it should be opened with newline=''*
* By default it takes comma as delimiter we can changed it by using delimiter argument
* *This method reads data from csv file and place each row data in a list*

import csv

with open('innovators.csv', 'r') as file:

    reader = csv.reader(file, delimiter = '\t')

    for row in reader:

        print(row)

**fmtparams** ---- these are optional formatting parameters( quoting, skipinitialspace etc)

**CSV files with Custom Delimiters**

We can use delimiter parameter to specify the delimiter, by default delimiter is comma.

**CSV files with initial spaces**

To remove these initial spaces from each column data then we need to pass an additional parameter called skipinitialspace

Let say we have below sample data, in which there is space for each col data (after each comma)

SN, Name, City

1, John, Washington

2, Eric, Los Angeles

3, Brad, Texas

import csv

with open('people.csv', 'r') as csvfile:

    reader = csv.reader(csvfile, skipinitialspace=True)

    for row in reader:

        print(row)

We can notice extra space is removed from each data

['SN', 'Name', 'City']

['1', 'John', 'Washington']

['2', 'Eric', 'Los Angeles']

['3', 'Brad', 'Texas']

**Note:**

*We can notice here that each row data is converted into a list*

**CSV files with quotes**

Some CSV files can have quotes around each or some of the entries, to remove use another optional parameter called **quoting=csv.QUOTE\_ALL**

Let say we have below sample data, and want output in proper csv format

"SN", "Name", "Quotes"

1, Buddha, "What we think we become"

2, Mark Twain, "Never regret anything that made you smile"

3, Oscar Wilde, "Be yourself everyone else is already taken"

import csv

with open('person1.csv', 'r') as file:

    reader = csv.reader(file, quoting=csv.QUOTE\_ALL, skipinitialspace=True)

    for row in reader:

        print(row)

Output will be below

['SN', 'Name', 'Quotes']

['1', 'Buddha', 'What we think we become']

['2', 'Mark Twain', 'Never regret anything that made you smile']

['3', 'Oscar Wilde', 'Be yourself everyone else is already taken']

csv.QUOTE\_ALL specifies the reader object that all the values in the CSV file are present inside quotation marks

There are 3 other predefined constants you can pass to the quoting parameter-

1. csv.QUOTE\_MINIMAL
2. csv.QUOTE\_NONNUMERIC
3. csv.QUOTE\_NONE

**Dialect in csv**

As we have seen above that there are some parameters(quote, delimiter, skipinitialspace) that we need to pass while reading csv file. This causes over work to specify for each read time. To overcome it we can create a dialect with all these parameter add use every time while reading

**Creating dialect**

We can create dialect using function csv.register\_dialect() at any time

csv.register\_dialect(name[, dialect[, \*\*fmtparams]])

name --- name of dialect

import csv

csv.register\_dialect('myDialect', delimiter='|',skipinitialspace=True,quoting=csv.QUOTE\_ALL)

above code creates a dialect by name 'myDialect' and this dialect will take care of delimiter , quoting , skipinitialspaec parameter

Example:

Let say we have below data and want to read it using csv module

"ID"| "Name"| "Email"

"A878"| "Alfonso K. Hamby"| "alfonsokhamby@rhyta.com"

"F854"| "Susanne Briard"| "susannebriard@armyspy.com"

"E833"| "Katja Mauer"| "kmauer@jadoop.com"

Solutions:

Step 1: ----- create dialect (mydialect)

Steps 2: ----- read data of csv file

import csv

#create dialect

csv.register\_dialect('myDialect',delimiter='|',skipinitialspace=True,quoting=csv.QUOTE\_ALL)

#read data usnig above dialect

with open('office.csv', 'r') as csvfile:

    reader = csv.reader(csvfile, dialect='myDialect')

    for row in reader:

        print(row)

Output will be below:-

['ID', 'Name', 'Email']

["A878", 'Alfonso K. Hamby', 'alfonsokhamby@rhyta.com']

["F854", 'Susanne Briard', 'susannebriard@armyspy.com']

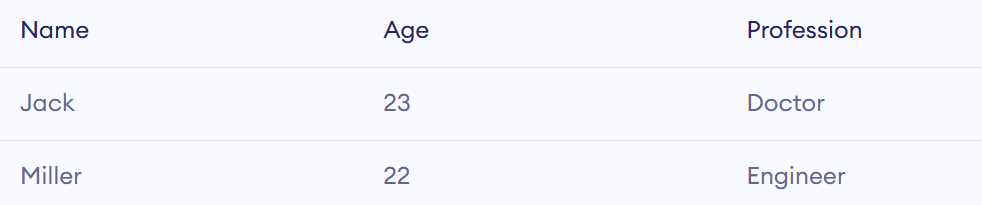
["E833", 'Katja Mauer', 'kmauer@jadoop.com']

**Read CSV files with csv.DictReader()**

The csv.DictReader class operates like a regular reader but maps the information read into a dictionary. The keys for the dictionary can be passed in with the fieldnames parameter or inferred from the first row of the CSV file

csv.DictReader(file, fieldnames=None, restkey=None, restval=None, dialect='excel', \*args, \*\*kwds)

Let say we have data in people.csv file in below format.



We can convert it to key value pair(key will be header and value will be value/data of column for each row)

with open('D:\\sample.csv', 'r') as file:

    csv\_file = csv.DictReader(file)

    for row in csv\_file:

        print(row)

Now we will have result in below format

{Name: 'Jack', Age: ' 23, Profession: ' Doctor’}

{Name: Miller, Age: ' 22, Profession: ' Engineer’}

**csv writer**

For writing data into csv file we use writer function

csv.writer(csvfile\_path, dialect=?, newline=””,\*\*fmtparams)

Return a writer object responsible for converting the user’s data into delimited strings on the given file-like object. csvfile can be any object with a write() method.

If csvfile is a file object, it should be opened with newline=''

Let say we want to write below data in to a csv file we can use it

import csv

with open('innovators.csv', 'w', newline='') as file:

    writer = csv.writer(file)

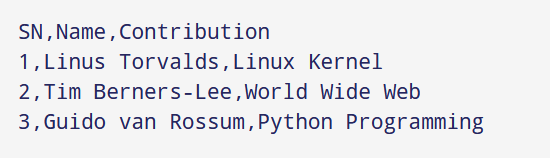
    writer.writerow(["SN", "Name", "Contribution"])

    writer.writerow([1, "Linus Torvalds", "Linux Kernel"])

    writer.writerow([2, "Tim Berners-Lee", "World Wide Web"])

    writer.writerow([3, "Guido van Rossum", "Python Programming"])

Output will be in below format



basically csv.writer() gives back a writer object which can be used to write data using writerow() or writerows()

**csvwriter.writerow(row)**

Write the row parameter to the writer’s file object, formatted according to the current dialect. Return the return value of the call to the write method of the underlying file object

*Actually each row data is given in a list object.*

**csvwriter.writerows(rows)**

Write all elements in rows (an iterable of row objects as described above) to the writer’s file object, formatted according to the current dialect

*It takes each row data in list of list form.*

import csv

row\_list = [["SN", "Name", "Contribution"],

             [1, "Linus Torvalds", "Linux Kernel"],

             [2, "Tim Berners-Lee", "World Wide Web"],

             [3, "Guido van Rossum", "Python Programming"]]

with open('protagonist.csv', 'w', newline='') as file:

    writer = csv.writer(file)

    writer.writerows(row\_list)

How to write data/csv data in a csv file.

Steps 1: ------ First Open a file in write/append mode.

Steps 2: ------ Create a csv writer object using writer().

Steps 3: ------- Write data into csv file using writerow() or writerows().

**Creating CSV Files with Custom Delimiters**

Same as read object, just specify the delimiter in writer()

**Creating CSV files with Quotes**

It’s same as reading just specify 'quote' attribute while creating csv writer object, that’s all

**Creating dialect for csv writer**

It’s same as how to do in reader()

**CSV files with csv.DictWriter()**

csv.DictWriter(f, fieldnames, restval='', extrasaction='raise', dialect='excel', \*args, \*\*kwds)

**file** - CSV file where we want to write to

**fieldnames** - a list object which should contain the column headers specifying the order in which data should be written in the CSV file

While using csv.DictWriter() while writing data by using writerow() or writerows() data must be in key-value pair.

Key will be header and value will be value for corresponding header as shows below

with open('players.csv', 'w', newline='') as file:

    fieldnames = ['player\_name', 'fide\_rating']

    writer = csv.DictWriter(file, fieldnames=fieldnames)

    writer.writeheader()

    writer.writerow({'player\_name': 'Magnus Carlsen', 'fide\_rating': 2870})

    writer.writerow({'player\_name': 'Fabiano Caruana', 'fide\_rating': 2822})

    writer.writerow({'player\_name': 'Ding Liren', 'fide\_rating': 2801})