Python import csv to list

I have a CSV file with about 2000 records.

Each record has a string, and a category to it.

```
This is the first line, Line1
This is the second line, Line2
This is the third line, Line3
```

I need to read this file into a list that looks like this;

```
List = [('This is the first line', 'Line1'),
        ('This is the second line', 'Line2'),
        ('This is the third line', 'Line3')]
```

How can import this this `csv` to the list I need using Python?

python csv

edited Jun 3 '15 at 14:30 — starkeen — 23.3k 13 38 55

asked Jul 9 '14 at 19:48 — MorganTN — 411 1 4 8

1 Then use `csv` module: docs.python.org/2/library/csv.html – furas Jul 9 '14 at 19:53

2 If there is an answer that suits your question, please accept it. – Maciej Gol Mar 24 '15 at 21:37

Possible duplicate of How do I read and write CSV files with Python? – Martin Thoma Jan 11 at 7:47

7 Answers

Use the `csv` module (Python 2.x):

```
import csv
with open('file.csv', 'rb') as f:
    reader = csv.reader(f)
    your_list = list(reader)

print your_list
# [['This is the first line', 'Line1'],
#  ['This is the second line', 'Line2'],
#  ['This is the third line', 'Line3']]
```

If you need tuples:

```
import csv
with open('test.csv', 'rb') as f:
    reader = csv.reader(f)
    your_list = map(tuple, reader)

print your_list
# [('This is the first line', ' Line1'),
#  ('This is the second line', ' Line2'),
#  ('This is the third line', ' Line3')]
```

Python 3.x version (by @seokhoonlee below)

```
import csv

with open('file.csv', 'r') as f:
  reader = csv.reader(f)
  your_list = list(reader)

print(your_list)
# [['This is the first line', 'Line1'],
#  ['This is the second line', 'Line2'],
#  ['This is the third line', 'Line3']]
```

edited Feb 1 at 14:15

answered Jul 9 '14 at 19:55

Maciej Gol
8,628 2 17 35

3 Why do you use 'rb' instead of 'r'? – Drunken Master May 21 '15 at 14:28

3 @DrunkenMaster, `b` causes the file to be opened in binary mode as opposed to text mode. On some systems text mode means that `\n` will be convertes to platform-specific new line when reading or writing. See docs. – Maciej Gol May 24 '15 at 8:12

2 This does not work in Python 3.x : "csv.Error: iterator should return strings, not bytes (did you open the file in text mode?)" See below for the answer that works in Python 3.x – Gilbert May 30 '16 at 18:12

1 @Gilbert, thanks, updated the answer. – Maciej Gol May 31 '16 at 18:58

to save a few seconds of time debugging, you should probably add a note for the first solution, like "Python 2.x version" – paradite Jan 30 at 9:03

Update for **Python3**:

```
import csv

with open('file.csv', 'r') as f:
  reader = csv.reader(f)
  your_list = list(reader)

print(your_list)
# [['This is the first line', 'Line1'],
#  ['This is the second line', 'Line2'],
#  ['This is the third line', 'Line3']]
```

edited Apr 26 '16 at 20:11

Ryan Romanchuk
5,129 5 25 31

answered Feb 11 '16 at 13:43

seokhoonlee
288 4 14

Pandas is pretty good at dealing with data. Here is one example how to use it:

```
import pandas as pd

# Read the CSV into a pandas data frame (df)
#   With a df you can do many things
#   most important: visualize data with Seaborn
df = pd.read_csv('filename.csv', delimiter=',')

# Or export it in many ways, e.g. a list of tuples
tuples = [tuple(x) for x in df.values]

# or export it as a list of dicts
dicts = df.to_dict().values()
```

One big advantage is that pandas deals automatically with header rows.

If you haven't heard of Seaborn, I recommend having a look at it.

answered Nov 24 '16 at 13:52

Martin Thoma
20k 25 163 304

If you are sure there are no commas in your input, other than to separate the category, you can read the file line by line and split on `,` , then push the result to `List`

That said, it looks like you are looking at a CSV file, so you might consider using the modules for it

edited May 23 at 12:02
Community ♦
1 1

answered Jul 9 '14 at 19:53
Miquel
10.2k 6 34 75

```
result = []
for line in text.splitlines():
    result.append(tuple(line.split(",")))
```

answered Jul 9 '14 at 19:54
Acid_Snake
21 2

Can you please add a bit of explanation to this post? Code only is (sometimes) good, but code and explanation is (most times) better – Barranka Jul 9 '14 at 20:29

3 I know Barranka's comment is over a year old, but for anyone who stumbles upon this and can't figure it out: *for line in text.splitlines():* puts each individual line in temp variable "line". *line.split(",")* creates a list of strings that are split on the comma. *tuple(~)* puts that list in a tuple and *append(~)* adds it to the result. After the loop, *result* is a list of tuples, with each tuple a line, and each tuple element an element in the csv file. – Louis Oct 18 '15 at 10:05

A simple loop would suffice:

```
lines = []
with open('test.txt', 'r') as f:
    for line in f.readlines():
        l,name = line.strip().split(',')
        lines.append((l,name))

print lines
```

answered Jul 9 '14 at 19:54
Hunter McMillen
30k 11 73 133

What if some of the entries have commas in them? – Tony Ennis Feb 16 '16 at 17:59

@TonyEnnis Then you would need to use a more advanced processing loop. The answer by Maciej above shows how to use the csv parser that comes with Python to perform this operation. This parser most likely has all of the logic you need. – Hunter McMillen Feb 16 '16 at 18:21

Extending your requirements a bit, and assuming, you do not care about order of lines and want to get them grouped under categories, following solution can work for you:

```
>>> fname = "lines.txt"
>>> from collections import defaultdict
>>> dct = defaultdict(list)
>>> with open(fname) as f:
...     for line in f:
...         text, cat = line.rstrip("\n").split(",", 1)
...         dct[cat].append(text)
...
>>> dct
defaultdict(<type 'list'>, {' CatA': ['This is the first line', 'This is the another
line'], ' CatC': ['This is the third line'], ' CatB': ['This is the second line', 'This is
the last line']})
```

This way you get all relevant lines available in dictionary under key being the category.

answered Jul 9 '14 at 20:08
Jan Vlcinsky
22.1k 6 54 67