

- = -> notebooks from this lecture: <https://github.com/krishnatray/RDP-Reading-Data-with-Python-and-Pandas>
- -> more advanced features of Pandas for exporting data
- -> we have imported csv data and from SQL databases
- -> **this is getting into more advanced use cases of importing data**
 - -> txt files
 - -> a csv file is a text file -> human readable text
 - -> CSV files are tabular
 - -> comma separated values
 - -> it's a text file which is in a tabular form
 - -> a CSV file is a tabulated text file
- -> **to import / read data from external sources without pandas**
 - -> just Python
 - -> for data scientists / analysts
 - -> file reading / writing in computers
 - -> operating systems process language
 - -> you can get a more advanced use case if you get a more
 - -> there are multiple concepts with this
 - -> you can get a more advanced use case if you know the exact details of the operating system
 - -> the open function
 - -> it creates a file pointer
 - -> grid line
 - -> read
 - -> its hard to parse the structure of the files
 - -> once you wanted to get more advanced usage, its going to get harder
 - -> so you use the pandas module
 - -> he opens a CSV file in an example
 - -> using the CSV module to parse the data
 - -> abstracting it away by the pandas module
- -> **question**

Given a file named certificates.csv with these contents:

```
Name$Certificates$Time (in months)
Tom$8$16
Kris$2$5
Ahmad$5$9
Beau$6$12
```

Fill in the blanks for the missing arguments below:

```
import csv
```

```
with open(__A__, 'r') as fp:
```

```
    reader = csv.reader(fp, delimiter=__B__)
```

```
    next(reader)
```

```
    for index, values in enumerate(reader):
```

```
        name, certs_num, months_num = values
```

```
        print(f"{name} earned {__C__} certificates in {months_num} months")
```

A: 'certificates.csv'

B: '-'

C: values

A: 'certificates.csv' <- This one

B: '\$'

C: certs_num

A: 'certificates'

B: '\$'

C: certs_num