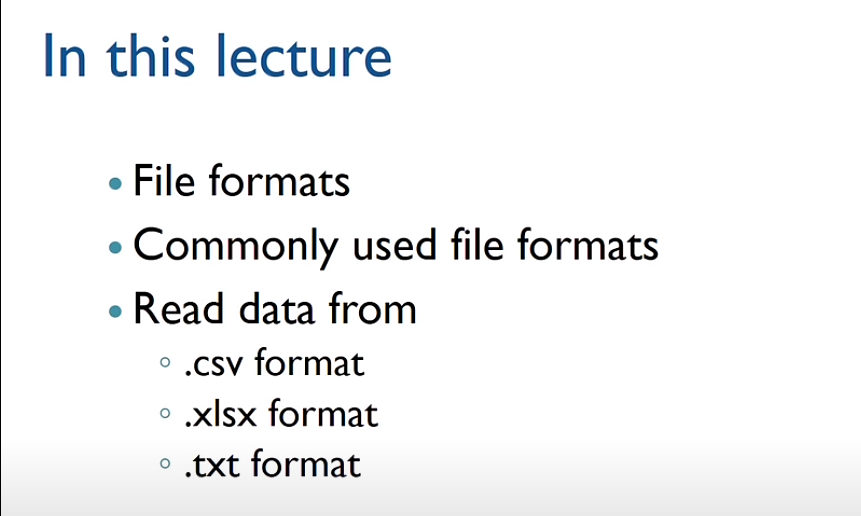
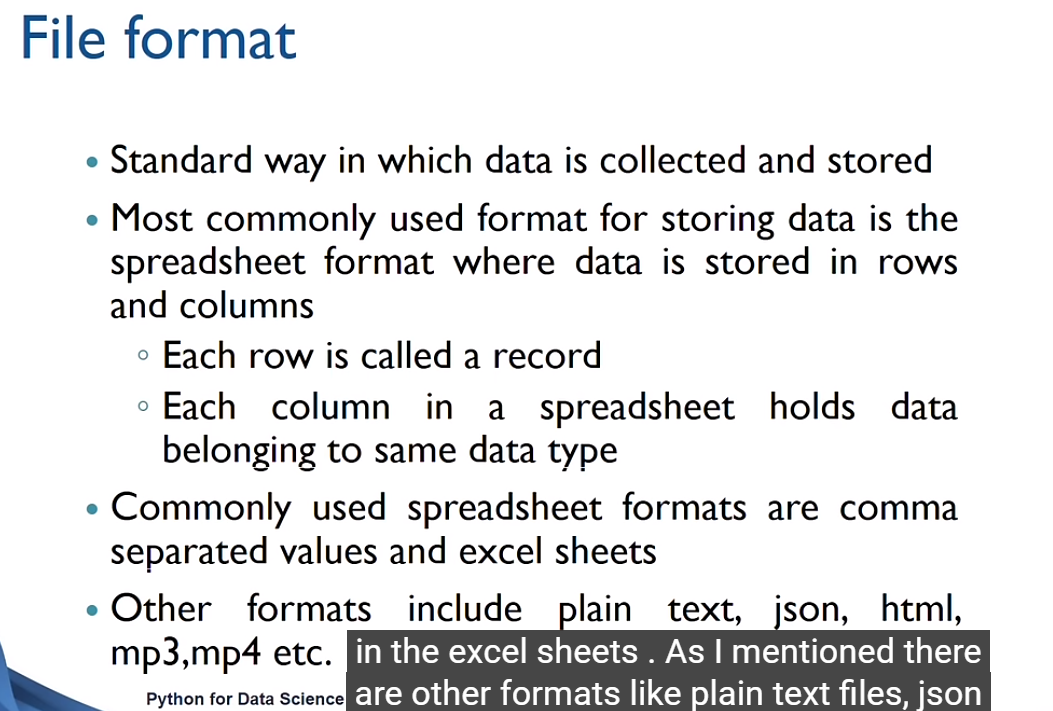
# Reading Data

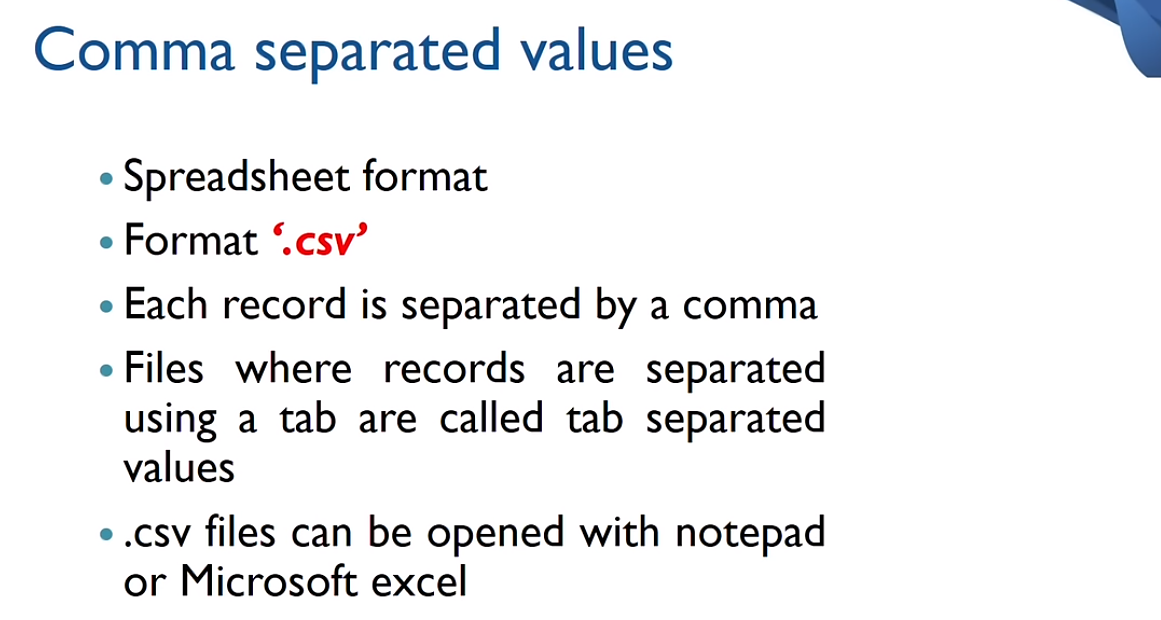
## Agenda



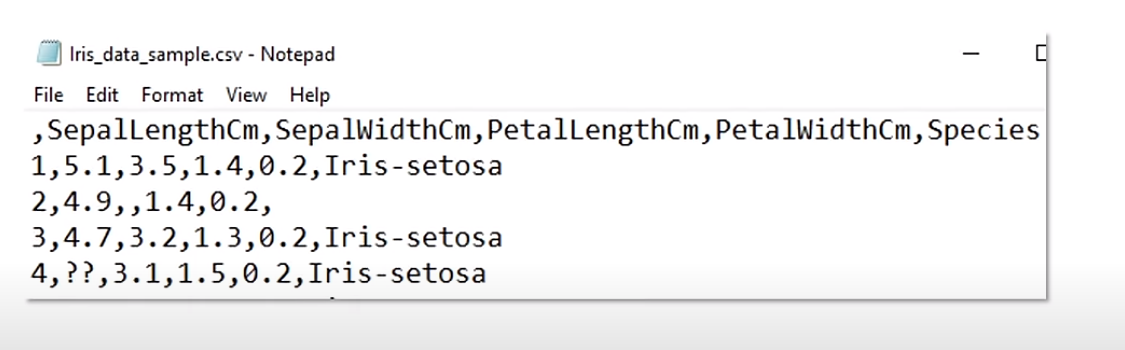
## File Format



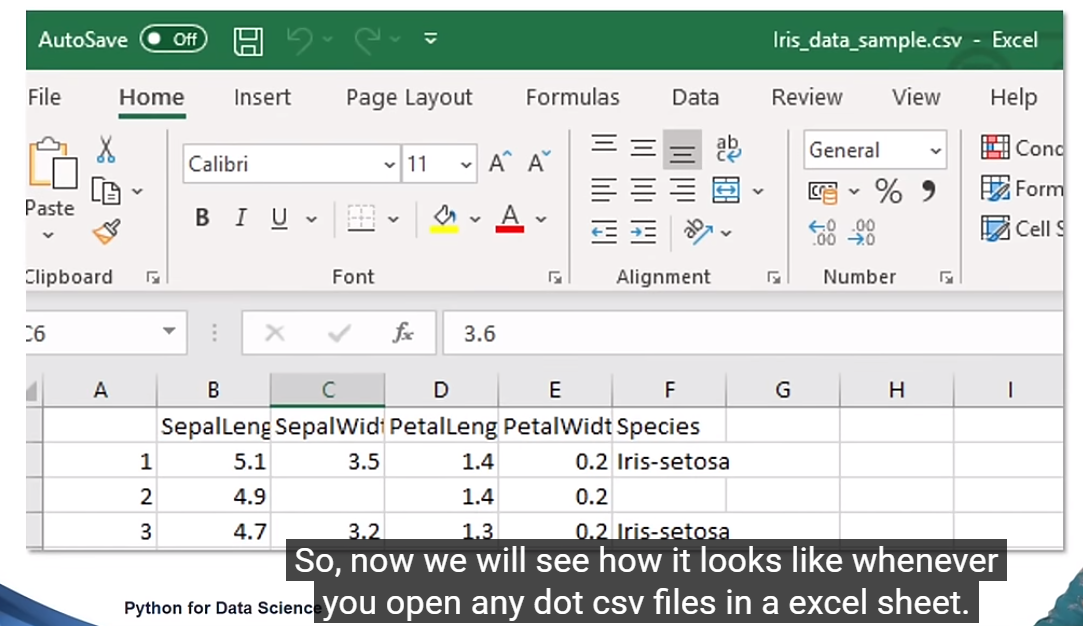
### Comma separated values (csv) format



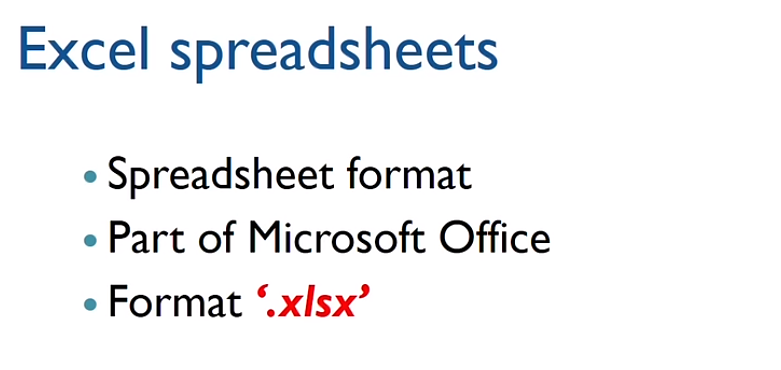
Comma separated values in a notepad. It is hard to read which value belongs to which column(variable).

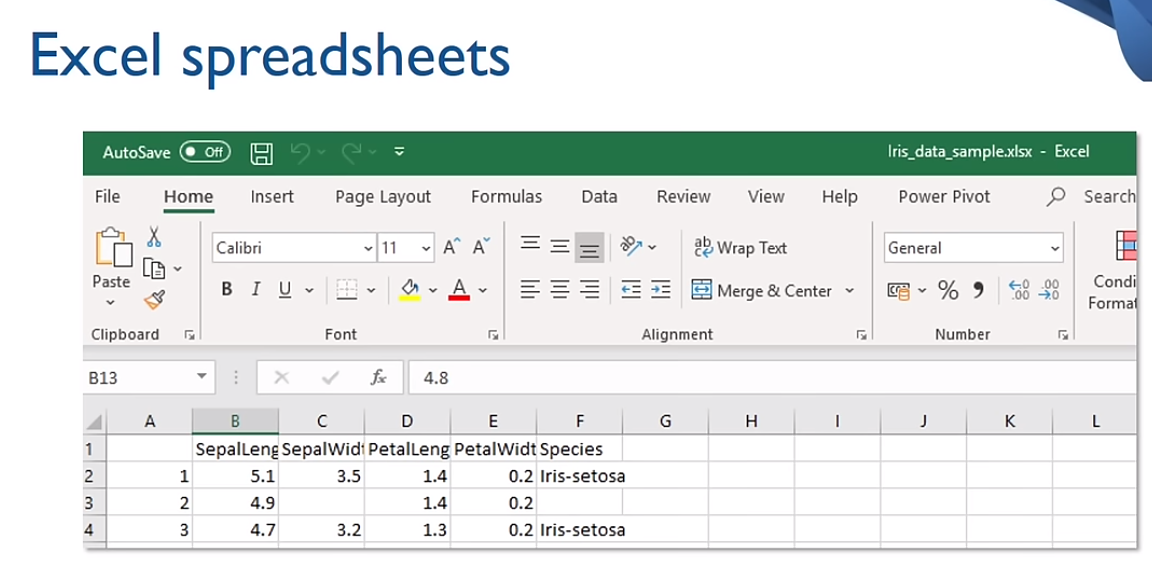


Comma separated values in an excel sheet. It is easy to read.

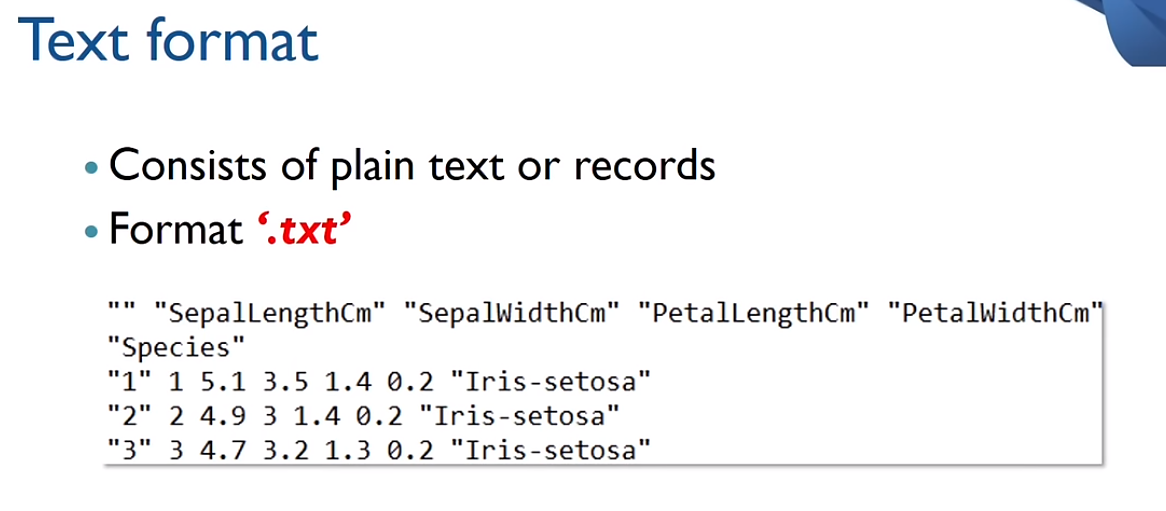


### Excel (xlsx) format

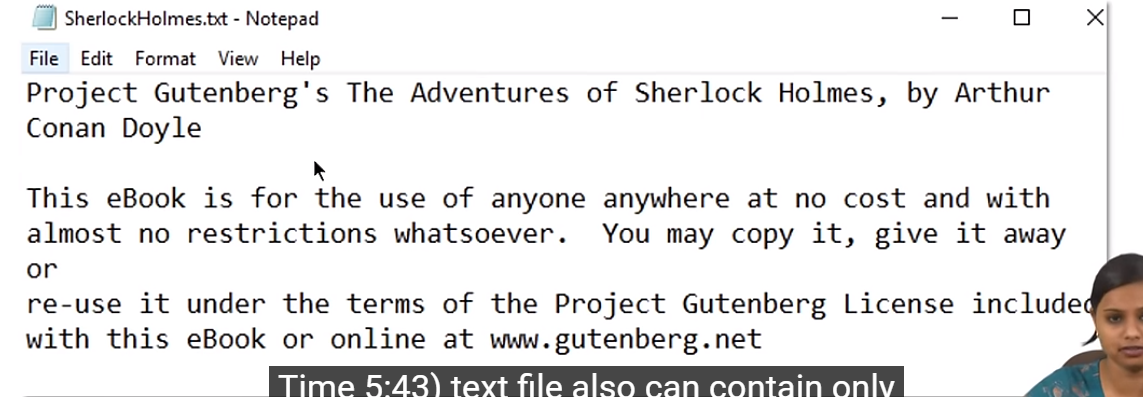




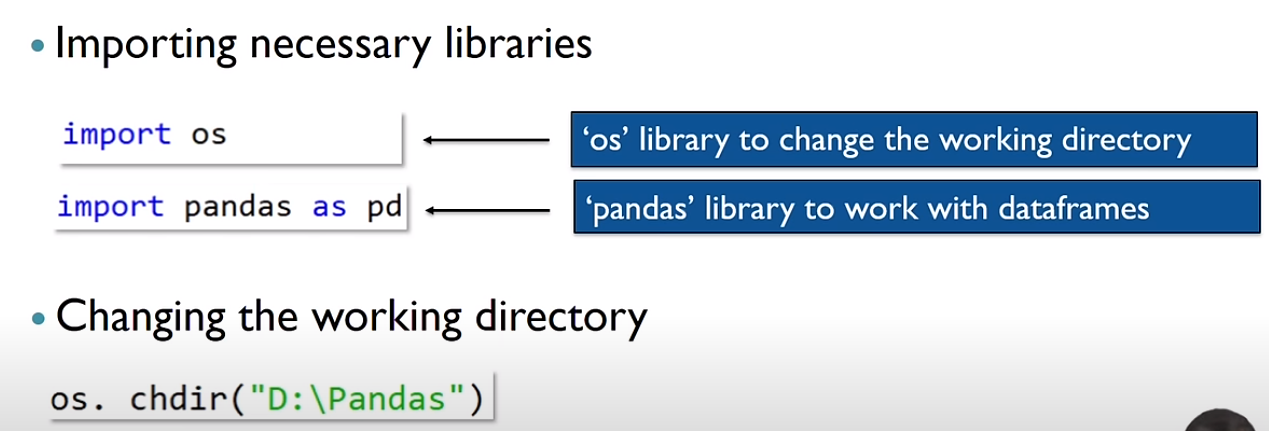
### Text (txt) format



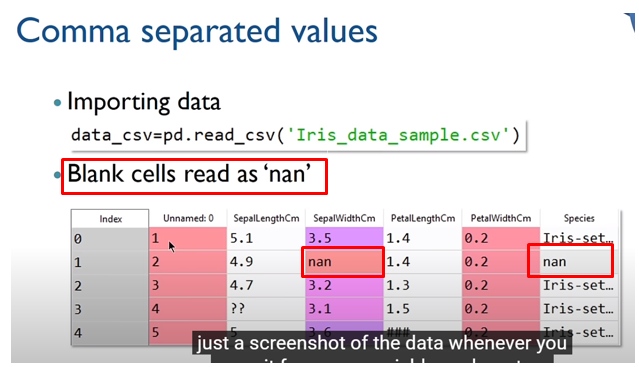
It can contain the data purely in the text format.



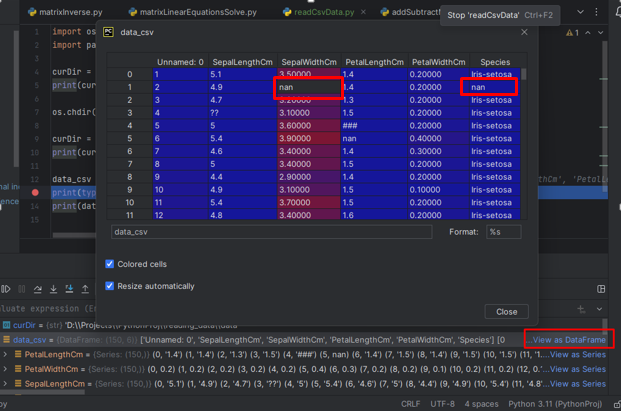
## Importing Data

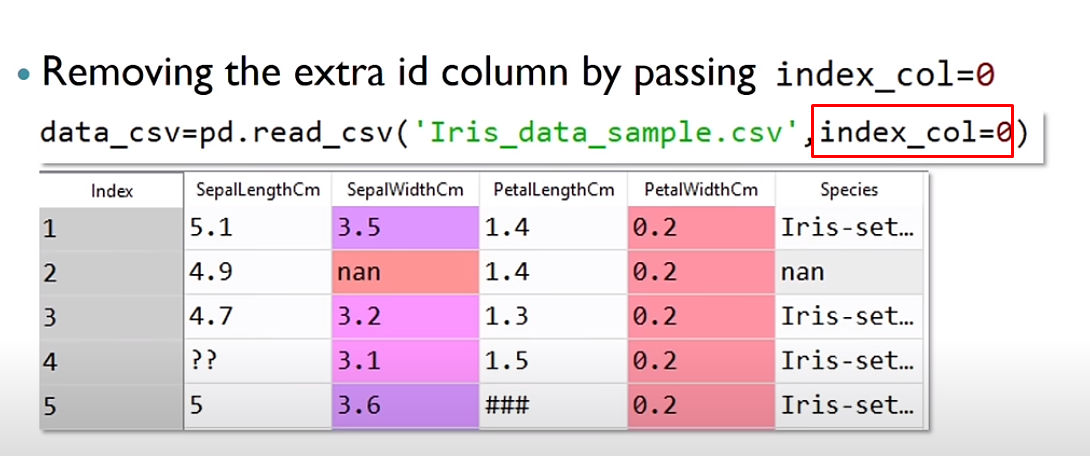


Use Panda’s dataframes to import the data from the file.



In Pycharm, start in debug mode

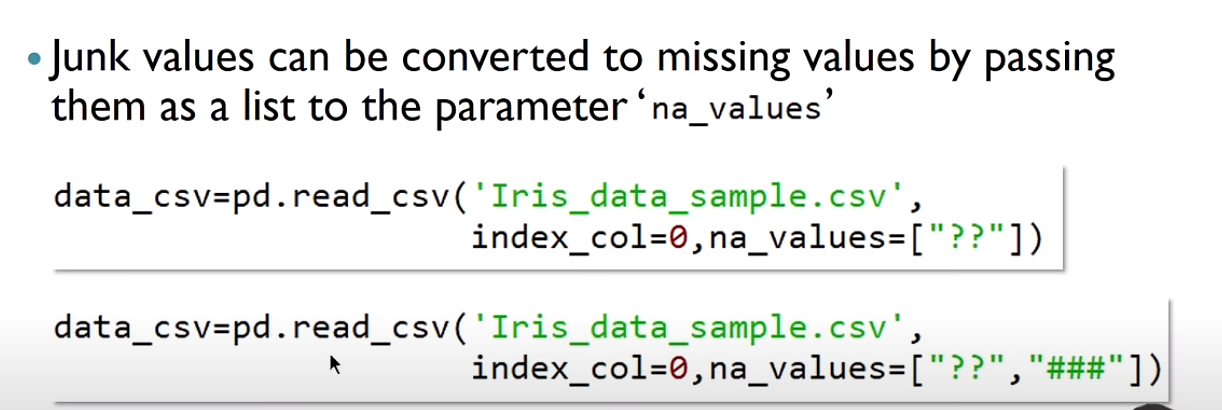


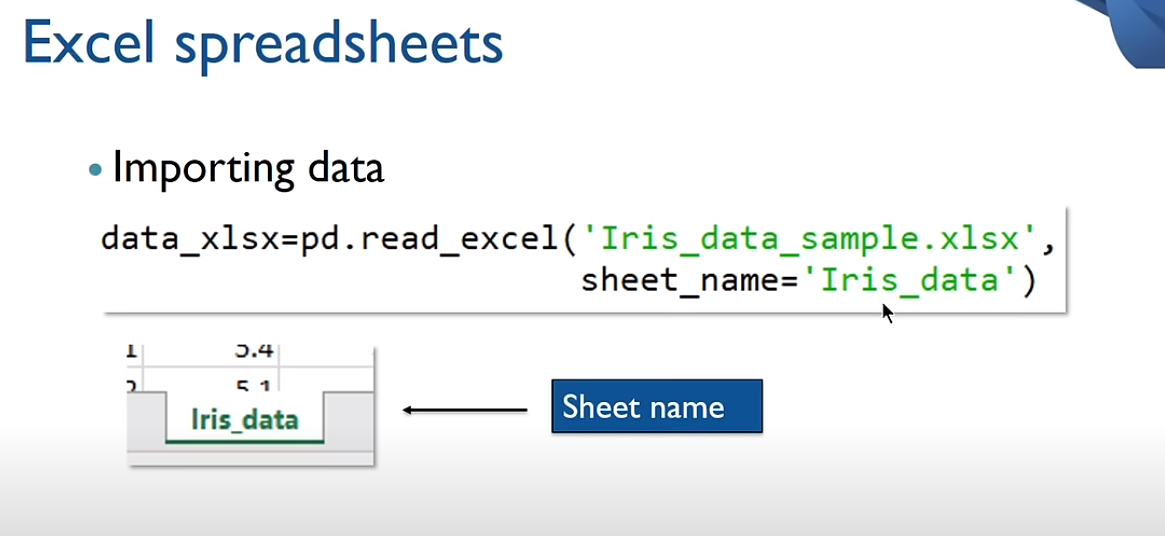


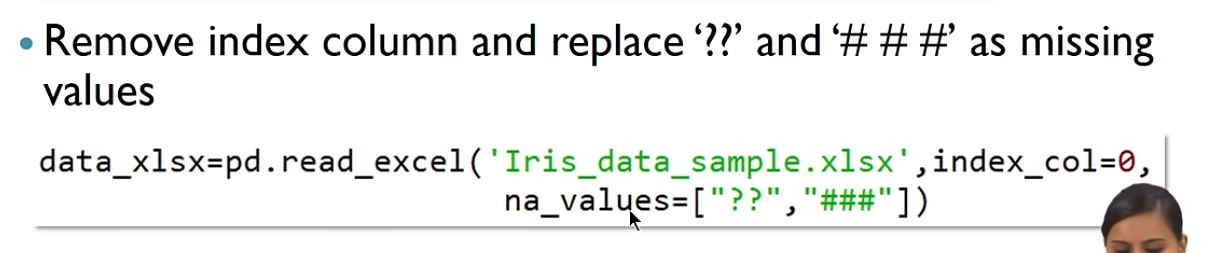
0th column of csv file will be considered as an index col, so Pandas won't add its own index column

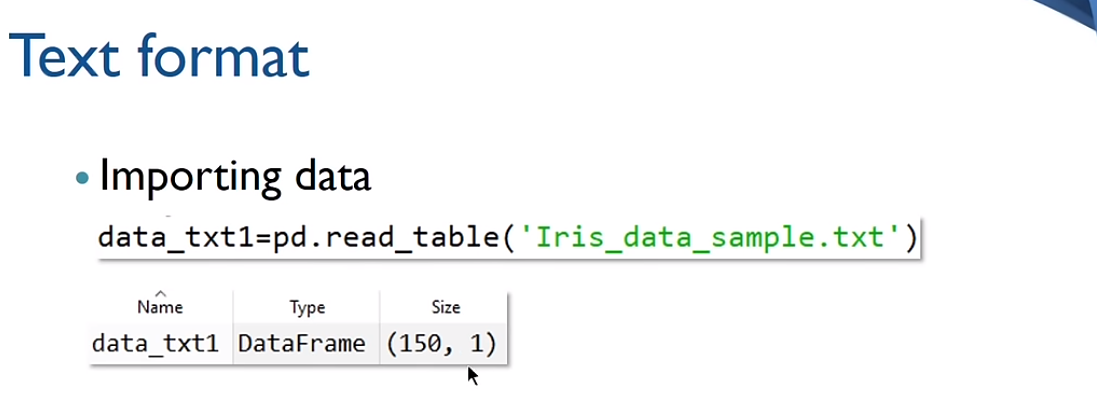


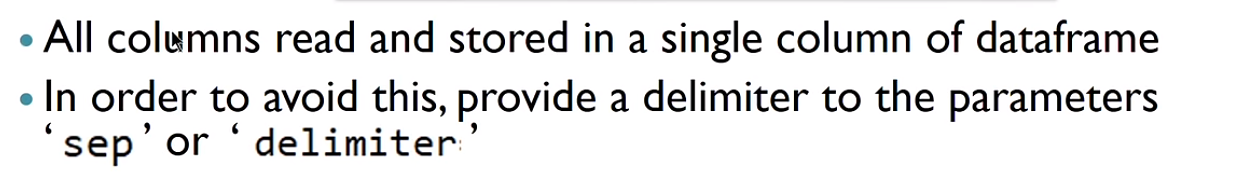
Let’s say, you got the data in such a way that ?? and ### are represented as missing values. Pandas will replace only blanks as NaN. You can replace all ?? and ### with NAN.





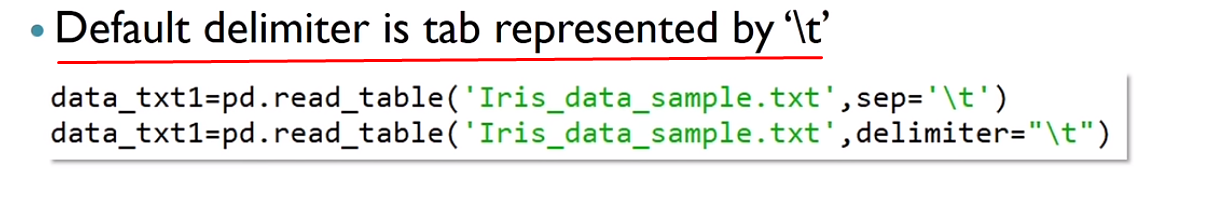


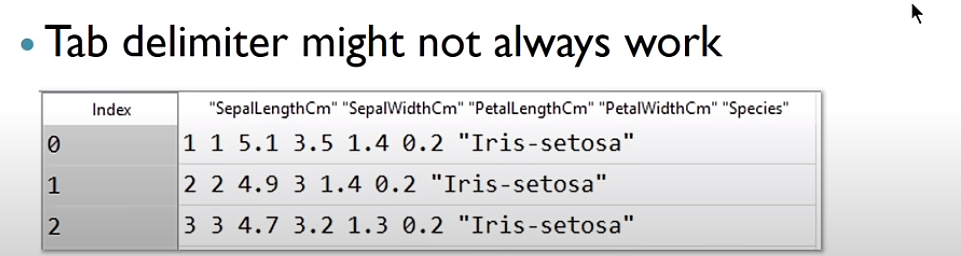


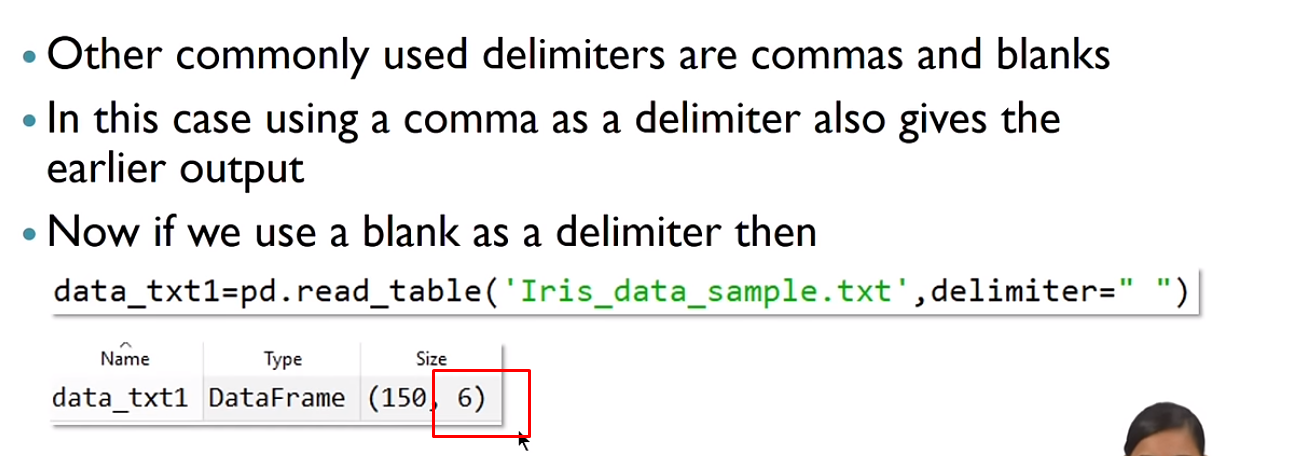


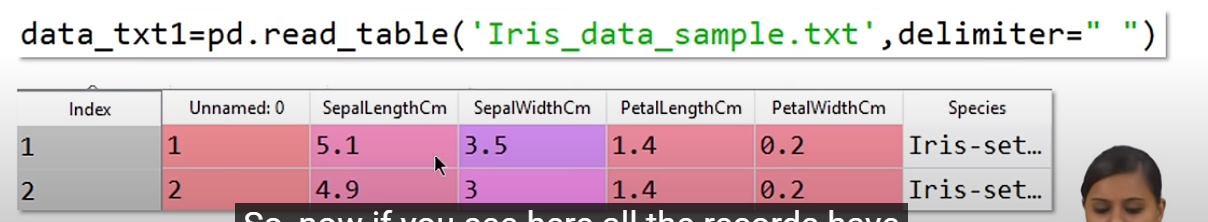
Sometimes, when you read data from text file, it reads all columns as one column. So, it creates only one column in dataframes.

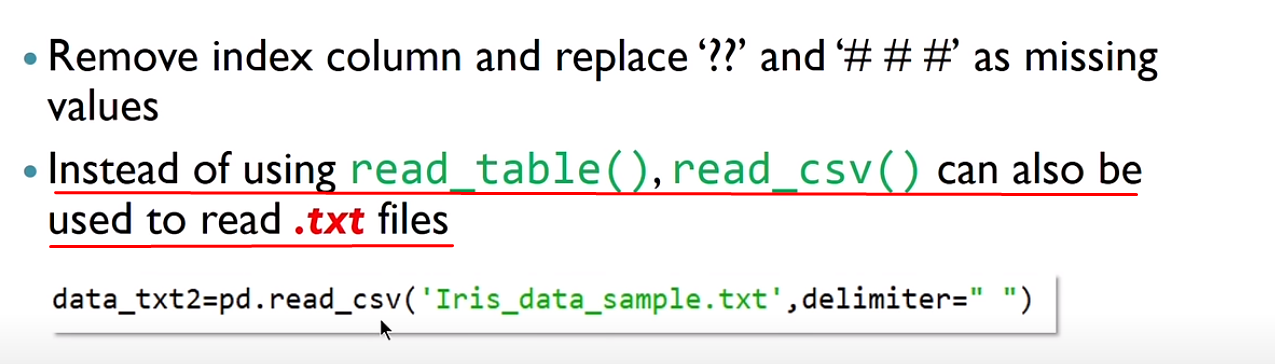
To overcome this problem, you can either use a 'sep' or 'delimiter' parameter. Normal delimiters are tab, blank, comma etc.











If you use, read\_csv() instead of read\_table() to read a text file, it is mandatory to give a delimiter.