* dictionary is a key value combination
* .DataFrame('name') is a function which converts the data into tabular form
* .read\_csv('filename') is to read a csv file
* .index() is used to change the original index of the table with the one we input
* .head(x) is used to print the first x values
* .tail(x) is used to print the last x values
* access columns -> table\_name["column\_name"], table\_name["column1\_name", "column2\_name", etc]
  + For multiple columns u can also define a list and call it by the list name
* convert the Series data to DataFrame -> table\_name[["column\_name"]]
  + basically use double square brackets for converting to table format
* View all the column names -> table\_name.columns
* Convert the column names to a list -> list\_name = list(table\_name.columns.values)
* Size of the table -> table\_name.shape
* [:3] -> prints the values from 0 to 3
* Row access -> .loc
  + .loc takes the name
  + .iloc takes the index
* Row and column loc -> can be done by loc and iloc where [rows, columns]
* For all values we use table\_name[start\_index : end\_index]
* Data specific -> table\_name[table\_name[‘column name’] == key]
* Np.logical\_and is used as AND operator [condition, condition]
* Add a new column -> table\_name[ column\_name] = [values]
* Sorting values 🡪 df.sort\_value(column\_name, ascending = True/False) based on ascending or descending order
* Multi level sort is also possible
* Df.sort\_values([col1, col2, etc], ascending = [False, True])
* .drop is used to delete a column
* .to\_csv is used to create a backup on a CSV file