In [ ]:
```
#################################################################
## Pandas Cheat Sheet compiled by Paschal Chukwuemeka Amah.##
#################################################################
```

In [ ]:
```
#Create an empty dataframe with three columns
df = pd.DataFrame(columns = ["x", "y", "z"])
```

In [ ]:
```
#Quick summary of a dataframe: number of columns and rows
df.shape
```

In [ ]:
```
#Import a comma seperated file
df = pd.read_csv(filename.csv)

#Write a csv file
pd.to_csv(filename.csv)
```

In [ ]:
```
#Import a space delimited file
df = pd.read_csv("filename.txt", delim_whitespace=True)
```

In [ ]:
```
#Import an excel file
xls = pd.ExcelFile("file.extension")

#Peep the excel sheets
xls.sheet_names

#Load a sheet
df = xls.parse("sheetName")
```

In [ ]:
```
#Write to excel
from pandas import ExcelWriter

#Create a write object #Sample is the name of file written
writer = ExcelWriter('sample.xlsx')

#Write to a sheet 'sheet1'
df.to_excel(writer,'Sheet1')
writer.save()
```

In [ ]:
```
#Import a json file -  especially one sourced from a social media like Twitte
df = pd.read_jso("filename.json", lines = True, encoding = "utf-8")
```

In [ ]:
```
######################
## Clean up moves ##
####################
```

In [ ]:
```
#Remove non-characters from column
df['colName']= df['colName'](lambda x: x.replace('[^a-zA-Z]', ''))
```

In [ ]:
```
#Case conversion

#Convert whole dataframe to TitleCase
df = df.apply(lambda x: x.astype(str).str.title())

#To lower case
df = df.apply(lambda x: x.astype(str).str.lower())

#To UPPERCASE
df = df.apply(lambda x: x.astype(str).str.upper())
```

In [ ]:
```
#Drop duplicates leaving first occurence
df.drop_duplicates(keep = 'first')
```

In [ ]:
```
#Replace parts of column content for all rows. Here, all items but the first
df.loc[:,"colName"].map(lambda x: x.replace(x[4:], ''))
```

In [ ]:
```
#Replace parts of column content for all rows. Here, first 9 items are replac
df.loc[:,"colName"].map(lambda x: x.replace(x[:9], ''))
```

In [ ]:
```
#Replace parts of a column content for all rows. Here, all items but the firs
df.loc[:,"colName"].map(lambda x: x.replace(x[4:], '')+"abc")
```

In [ ]:
```
#Remove whitespaces in a column for all rows
df.loc[:,"colName"].map(lambda x: "".join(x.split()))
```

In [ ]:
```
#Merge two columns as str into a new column
df["NewColName"] = df["colName_x"].map(str)+ df["colName_y"].map(str)
```

In [ ]:
```
#Pick only the first 11 columns for all rows
df.iloc[:,:11]
```

In [ ]:
```
#Make a column of numbers from 0 to a particular range after importing numpy
df["NewCol"] = np.arange(0,len(df))
```

In [ ]:
```
#Sort a dataframe according to a column in descending order and reset the ind
df.sort_values("colName", ascending = True).reset_index(drop=True)
```

In [ ]:
```
#Return parts of df that satisfy a condition in a column
df[df['colName'] criteria] # eg1 df[df['colName'] == 'lagos'] eg2 df[df['colN
```

In [ ]:
```
#Drop part of a dataframe tail - 8 here
df.drop(df.tail(8).index)
```

In [ ]:
```
#Drop part of a dataframe head - 8 here
df.drop(df.head(8).index)
```

In [ ]:
```
#For a column, replace x ie row with y for all rows, if str in row/line, else
df['colName'].map(lamba x: replace(x, y) if 'str' in x else x)
```

In [ ]:
```
#Group dataframe by a column and count
df.groupby("colName").count()

#Group dataframe by 2 columns and count
df.groupby(["colName1", "colName2"]).count()
```

In [ ]:
```
#Group dataframe by 2 columns, count and sort in descending order
df.groupby("colName").count().sort_values("Other", ascending = False)

#Group dataframe by 2 columns, count and sort in descending order
df.groupby(["colName1", "colName2"]).count().sort_values("Other", ascending =
```

In [ ]:
```
#Convert database column to list
list1 = df.colName.tolist()
```