

Print current working directory

```
os.getcwd()
```

Set new directory

```
os.chdir("/Users/ghosebishwajit/Desktop/")
```

Importing data from kaggle

Download kaggle.json and place it inside the user directory

```
mkdir ~/.kaggle  
cp kaggle.json ~/.kaggle/
```

Authenticating and connecting to kaggle

```
from kaggle.api.kaggle_api_extended import KaggleApi  
api = KaggleApi()  
api.authenticate()
```

Check for datasets

```
!kaggle datasets list
```

Check for datasets on a specific topic

```
!kaggle datasets list -s covid
```

The output

```
ref                                     title
                                     size  lastUpdated
downloadCount  voteCount  usabilityRating
-----
-----
-----
datasets/imdevskp/corona-virus-report  COVID-19
Dataset                                19MB  2020-08-07 03:47:47
197704          1611  1.0
datasets/sudalairajkumar/covid19-in-india  COVID-19 in
India                                758KB  2021-08-11 04:14:10
157038          1784  0.9117647
datasets/gpreda/covid-world-vaccination-progress  COVID-19 World
Vaccination Progress                2MB  2022-03-31 07:55:09
80573          2056  1.0
datasets/roche-data-science-coalition/uncover  UNCOVER COVID-
19 Challenge                          258MB  2022-05-02 16:35:45
21828          1283  0.875
```

datasets/tawsifurrahman/covid19-radiography-database				COVID-19
Radiography Database	778MB	2022-03-19	13:38:42	
46776	691	1.0		
datasets/andrewmvd/covid19-ct-scans				COVID-19 CT
scans	1GB	2020-04-23	12:29:33	
16211	495	1.0		
datasets/sudalairajkumar/covid19-in-usa				COVID-19 in
USA	10MB	2020-12-07	06:23:33	
22497	286	0.9705882		
datasets/sudalairajkumar/covid19-in-italy				COVID-19 in
Italy	441KB	2020-12-07	06:23:24	
15969	240	1.0		
datasets/hendratno/covid19-indonesia				COVID-19
Indonesia Dataset	902KB	2021-12-04	01:59:45	
16066	230	1.0		
datasets/iamhungundji/covid19-symptoms-checker				COVID-19
Symptoms Checker	912KB	2020-03-21	06:05:59	
14368	208	1.0		
datasets/georgesaavedra/covid19-dataset				COVID-19
dataset	9MB	2022-03-07	02:06:03	
6389	136	0.9705882		
datasets/gpreda/covid19-tweets				COVID19 Tweets
	28MB	2020-08-30	09:38:38	
15254	494	1.0		
datasets/allen-institute-for-ai/CORD-19-research-challenge				COVID-19 Open
Research Dataset Challenge (CORD-19)	15GB	2022-04-25	19:32:16	
144214	9894	0.88235295		
datasets/bachrr/covid-chest-xray				COVID-19 chest
xray	241MB	2020-05-15	00:30:50	
9757	219	0.9411765		
datasets/atilamadai/covid19				COVID-19
	297MB	2022-05-02	07:23:15	
2107	44	0.88235295		

datasets/mariaren/covid19-healthy-diet-dataset	COVID-19
Healthy Diet Dataset	82KB 2021-02-07 21:50:53
16230 376 1.0	
datasets/kaushiksuresh147/covidvaccine-tweets	Covid Vaccine
Tweets	62MB 2022-05-01 09:04:22
3133 72 1.0	
datasets/fireballbyedimyrnmom/us-counties-covid-19-dataset	US counties
COVID 19 dataset	27MB 2022-05-01 22:22:33
18424 417 0.9411765	
datasets/tunguz/covid19-genomes	COVID-19
Genomes	405KB 2021-06-17 19:59:48
1124 78 1.0	
datasets/anandhuh/latest-covid19-india-statewise-data	Latest Covid-
19 India Statewise Data	1KB 2022-05-06 18:46:02
20655 700 1.0	

Import data from Stata format

```
pd.read_stata('/Users/ghosebishwajit/Desktop/df/world.dta')
```

Import data from Excel format

```
df = pd.read_excel('untitled.xls')
```

Print top 5 rows

```
In [59]: df.head()
```

```
Out[59]:
```

	make	price	mpg	rep78	headroom	trunk	weight	length	turn	displacement	gear_ratio	foreign
0	AMC Concord	4099	22	3.0	2.5	11	2930	186	40	121	3.58	Domestic
1	AMC Pacer	4749	17	3.0	3.0	11	3350	173	40	258	2.53	Domestic
2	AMC Spirit	3799	22	NaN	3.0	12	2640	168	35	121	3.08	Domestic
3	Buick Century	4816	20	3.0	4.5	16	3250	196	40	196	2.93	Domestic
4	Buick Electra	7827	15	4.0	4.0	20	4080	222	43	350	2.41	Domestic

Print the bottom 5 rows

```
df.tail()
```

Printing rows of a specific range

```
In [70]: df[25:30]
```

```
Out[70]:
```

	make	price	mpg	rep78	headroom	trunk	weight	length	turn	displacement	gear_ratio	foreign
25	Linc. Continental	11497	12	3.0	3.5	22	4840	233	51	400	2.47	Domestic
26	Linc. Mark V	13594	12	3.0	2.5	18	4720	230	48	400	2.47	Domestic
27	Linc. Versailles	13466	14	3.0	3.5	15	3830	201	41	302	2.47	Domestic
28	Merc. Bobcat	3829	22	4.0	3.0	9	2580	169	39	140	2.73	Domestic
29	Merc. Cougar	5379	14	4.0	3.5	16	4060	221	48	302	2.75	Domestic

Print the colnames

```
In [60]: list(df)
```

```
Out[60]: ['make',  
          'price',  
          'mpg',  
          'rep78',  
          'headroom',  
          'trunk',  
          'weight',  
          'length',  
          'turn',  
          'displacement',  
          'gear_ratio',  
          'foreign']
```

Sorting column names alphabetically

```
df.reindex(sorted(df.columns), axis=1)
```

```
In [72]: df=df.reindex(sorted(df.columns), axis=1)
```

```
In [73]: df
```

```
Out[73]:
```

	displacement	foreign	gear_ratio	headroom	length	make	mpg	price	rep78	trunk	turn	weight
0	121	Domestic	3.58	2.5	186	AMC Concord	22	4099	3.0	11	40	2930
1	258	Domestic	2.53	3.0	173	AMC Pacer	17	4749	3.0	11	40	3350
2	121	Domestic	3.08	3.0	168	AMC Spirit	22	3799	NaN	12	35	2640
3	196	Domestic	2.93	4.5	196	Buick Century	20	4816	3.0	16	40	3250
4	350	Domestic	2.41	4.0	222	Buick Electra	15	7827	4.0	20	43	4080
...
69	97	Foreign	3.74	2.5	172	VW Dasher	23	7140	4.0	12	36	2160
70	90	Foreign	3.78	3.0	155	VW Diesel	41	5397	5.0	15	35	2040
71	89	Foreign	3.78	3.0	155	VW Rabbit	25	4697	4.0	15	35	1930
72	97	Foreign	3.78	2.0	156	VW Scirocco	25	6850	4.0	16	36	1990
73	163	Foreign	2.98	2.5	193	Volvo 260	17	11995	5.0	14	37	3170

74 rows x 12 columns

Print the dimension of the df

```
df.shape  
(74, 12) #74 rows and 12 columns
```

Create a sample df

```
import pandas as pd
```

Example 1

Input data

```
df = pd.DataFrame({'age': [31, 22, 34, 43, 12], 'city': ['vienne',  
'zurich', 'shenzhen', 'mumbai', 'katmandu']})
```

Print the df

```
In [4]: df
```

```
Out[4]:
```

	age	city
0	31	vienne
1	22	zurich
2	34	shenzhen
3	43	mumbai
4	12	katmandu

Check the column types

```
df.dtypes
```

```
age      int64  
city     object  
dtype: object
```

Example 2

Using tuples

```
countries = [('india', 'mumbai', '21 m'),  
             ('rusland', 'moskva', '11.92 m'),  
             ('china', 'peking', '21.5 m'),  
             ('canada', 'toronto', '3 m'),  
             ('usa', 'washington', '7.6 m'),  
             ('Danemark', 'copenhagen', '.6 m')  
            ]
```


Create the df

```
df = pd.DataFrame(countries, columns=['pays', 'ville', 'population'])
```

Print the top 5 rows

```
df.head()
```

```
In [17]: df.head()
```

```
Out[17]:
```

	pays	ville	population
0	india	mumbai	21 m
1	rusland	moskva	11.92 m
2	china	peking	21.5 m
3	canada	toronto	3 m
4	usa	washington	7.6 m

Print the column names

```
print(df.columns)
```

```
In [18]: print(df.columns)
```

```
Index(['pays', 'ville', 'population'], dtype='object')
```

Printing the column types as a dictionary

```
dict(df.dtypes)
```

```
In [16]: dict(df.dtypes)
```

```
Out[16]: {'pays': dtype('O'), 'ville': dtype('O'), 'population': dtype('O')}
```

Convert column names to uppercase

```
df.columns.str.upper()
```

```
In [20]: df.columns.str.upper()
```

```
Out[20]: Index(['PAYS', 'VILLE', 'POPULATION'], dtype='object')
```

Using the 'map' function

```
In [23]: df.columns=map(str.upper, df.columns)
```

```
In [24]: print(df)
```

	PAYS	VILLE	POPULATION
0	india	mumbai	21 m
1	rusland	moskva	11.92 m
2	china	peking	21.5 m
3	canada	toronto	3 m
4	usa	washington	7.6 m
5	Danemark	copenhagen	.6 m