

* Python Pandas:- a Library used for working with data sets

→ It has Functions for analyzing, cleaning, exploring and manipulating data

→ Pandas allows us to analyze big data and make conclusions based on statistical theories

→ Pandas can clean messy data sets, and make them readable and relevant

* Getting started:-

→ `import pandas as pd`

* Loading data

`df = pd.read_csv('csv File Location')` ex.
('data/survey_result_public.csv')

`df` → `data frame` يعرض بعض محتويات ال

`df.shape` → (89184, 84)

`df.info()` → `data types` يعرف عدد الصفوف والأعمدة ويعرض كل ال

`pd.set_option('display.max_rows', 85)` → ثم `df`

→ سيعرض جميع ال rows جنب الملباة

و بالمثل مع ال columns ← `'display.max_columns'`

`df.head()` → الملباة أول rows فقط أو عدد معين إذا كتب بين ()

`df.tail()` → الملباة آخر rows فقط أو عدد معين

* Pandas Data Frame & Series

```
people = {  
    "First": ["Essam", "Janet", "Will"],  
    "Last": ["Shenhab", "Doe", "Smith"],  
    "age": [20, 21, 22]  
}
```

people['age'] → [20, 21, 22]

* Creating a DataFrame:-

```
import pandas as pd
```

```
df = pd.DataFrame(people) * Load data into a DataFrame object.
```

	First	Last	age
0	Essam	Shenhab	20
1	Janet	Doe	21
2	Will	Smith	22

some output

```
df['age'] → 0 20  
           1 21  
           2 22  
df.age
```

← يفضل استخدام الـ [] لأن df.last يمكن تعارض مع اسم method

* Loc: Accessing a row by Label Location

`df.loc[[0,1], 'Last']` → 0 Shenhab
1 Doe

* When used in a survey

`df['...'].value_counts()` ^{ex.} → Yes 7128
No 1886

`df.loc[0]` → prints out one person's entire survey

`df.loc[0, 'Hobbyist']` → 'Yes' → Prints out a response for a specific Row

* Slicing (Not-inclusive)

`df.loc[0:2, 'Hobbyist': 'Employment']`

	Hobbyist	Age	Employment
→ 0	Yes	18-24 years old	Not Employed
1	No	25-34 " " "	Work From home
2	Yes	45-54 " "	Employed full-time

* Loc: Accessing a row by Label Location

dF.Loc[[0,1], 'Last'] → 0 Shenhab
 1 Doe

* When used in a survey

df['...'].value_counts() $\xrightarrow{\text{ex}}$ Yes 7128
No 1896

`df.loc[0]` → prints out one person's entire survey
`df.loc[0, 'Hobbyist']` → 'Yes' → Prints out a response for a specific Row

~~SL~~ Slicing (Not-inclusive)

dF.Loc [0:2, 'Hobbyist': 'Employment']

	Hobbyist	Age	Employment
→ 0	Yes	18-24 years old	Not Employed
1	No	25-34 " "	Work From home
2	Yes	45-54 " "	Employed full-time

* Indexes:- الرقم المكتوب في ال column الى على الشمال

		first	Last	Age
df →	0	Essam	shenhob	20
	1	Jannet	Doe	21
	2	Will	smith	22

set something as the index For the data Frame

df.set_index('Last') →	Last	First	Age
	Shenhob	Essam	20
	Doe	Jannet	21
	Smith	Will	22

← ال index بتاع df مش اتغير ولو ما بناله لسه بشكله الطبيعي

df.set_index('Last', inplace = True) لكن هيتغير اذا كتبنا

df →

Last

Shenhob Essam 20

Doe Jannet 21

Smith Will 22

df.index → Index(['Shenhob', 'Doe', 'smith'], dtype = 'object', name = 'Last')

df.loc['Shenhob'] →

First	Essam
Last	shenhob
Age	20

df.loc['Shenhob', 'First'] → 'Essam'

df.loc[0] → Error

df.iloc[0] → First Essam
Last Shenhob
Age 20

* Re-setting the index

df.reset_index(inplace=True)

df →

	Last	First	Age
0	Shenhob	Essam	20
1	Doe	Jannet	21
2	Smith	Will	22

← لتغيير index عند قراءة ملف csv.

df = pd.read_csv('data/survey_results_public.csv',
index_col='ResponseID')

df.head(3) →

	Country	Hobbyist
ResponseID		
1	Egypt	Yes
2	USA	No
3	UK	Yes

schema_df.sort_index() → Age
← لترتيب index أبجدياً
Age 1st code
Better Life
.....

* Filtering:-

$Filt = (df['Last'] \neq 'Doe') \rightarrow$

0	False
1	True
2	False

$df[Filt]$ \rightarrow

1	Jannet	Doe	21
2	John	Doe	24

 $df.loc[Filt]$ \rightarrow $df[Filt]$

$df.loc[Filt, 'Age'] \rightarrow$

1	21
2	24

* '&' \rightarrow and '|' \rightarrow or

$Filt = (df['Last'] == 'Doe') \& (df['First'] == 'John')$

$df.loc[Filt, 'Age'] \rightarrow$

3	24
---	----

$df.loc[\sim Filt, 'Age'] \rightarrow$

0	20
---	----

\neg يعكس الفلتر قيمه \rightarrow

1	21
2	22

* Examples :-

$\rightarrow countries = ['USA', 'India', 'UK']$

$Filt = df['Country'].isin(countries)$

$df.loc[Filt, 'Country'] \rightarrow$

2	USA
6	UK
12	India
27	India
107	USA

→ `Filt = df['Language Worked With'].str.contains('Python', na = False)`

`df.loc[Filt, 'Language Worked With']`

→ 1 HTML/CSS; Java; Python; C

5 C++; HTML/CSS; python

8 Bash/Shell/PowerShell; Python; Java

* Updating Rows and Columns :-

`df.columns → Index(['First', 'Last', 'Age'], dtype = 'object')`

لتعديل اسم ال columns ✖

`df.columns = ['First_name', 'Last_name', 'Age']`

`df → Index(['First_name', 'Last_name', 'Age'], dtype = 'object')`

لتعديل على ال columns باستخدام methods ✖

→ `df.columns = [x.method() for x in df.columns]`
↓ ex.
upper → `Index(['FIRST_NAME', 'L...'])`
Lower

→ `df.columns = df.columns.str.replace(' ', '-')`

لتعديل column ✖

→ `df.rename(columns = {'First_name': 'First', 'Last_name': 'Last'}, inplace=True)`

لتعديل بيانات column ✖

→ `df.loc[2] = {'Will', 'Ferrell', '45'}`

→ `df.loc[2, ['Last', 'Age']] = ['Ferrel', '45']`

→ `df.loc[2, 'Last'] = 'Ferrel'`

→ `df.at[,,,] = 'Ferrel'` ✖ When updating a single object

* Common Mistakes

`Filt = (df['Last'] == 'Smith')`

`df[Filt]['Last'] = 'Ferrel' → Error → Setting With Copy warning`

المح → `df.loc[Filt, 'Last'] = 'Ferrel'`

// تعديل البيانات داخل الـ columns

→ `df['Last'] = df['Last'].str.upper()`

`df['Last'] →`

0	SHE/HAB
1	DOE
2	FERREL

* Some Methods

→ `def update_email(email):`
`return email.upper()`

`df['email'].apply(update_email)`

→ `df['Last'] = df['email'].apply(lambda x: x.lower())`

→ `df.apply(len) →`

0	3
1	3
2	3

→ `df.applymap(len) →`

	first	Last	email
0	5	7	23
1	6	3	17
2	4	5	17

* `Age(int)` has no `len()`

→ `df['First'].map({'Essam': 'Yosef', 'Janet': 'John'})`

→ 0 Yosef

1 John

2 NaN → Not a number value

ولاحظ أنها عليها نستعمل 'replace' بدل map

→ `df['First'].replace({'Janet': 'John'})`

→ 0 Yosef

1 John

2 Will

ولكن df لم يتغير فيجب كتابته `df['first'] =` أو

* Add/Remove Rows and Columns From DataFrames:-

Combining Columns

`df['First'] + ' ' + df['Last']` →

0	Essam Shenhab
1	Jannet Doe
2	Will Smith

→ `df['Full-name'] = df['First'] + ' ' + df['Last']`

df →

	First	Last	Age	Full-name
0	Essam	Shenhab	20	Essam Shenhab
1	Jannet	Doe	21	Jannet Doe
2	Will	Smith	22	Will Smith

Removing Columns

→ `df.drop(columns=['First', 'Last'], inplace=True)`

df →

	Age	Full-name
0	20	Essam Shenhab
1	21	Jannet Doe
2	22	Will Smith

Splitting Strings

→ `df['full-name'].str.split(' ')`

→

0	[Essam, Shenhab]
1	[Jannet, Doe]
2	[Will, Smith]

* Expand

→ `df[['First', 'Last']] = df['Full-name'].str.split(' ', expand=True)`

df →	Age	Full-name	First	Last
0	20	Essam Shenhab	Essam	Shenhab
1	21	Jannet Doe	Jannet	Doe
2	22	Will Smith	Will	Smith

* Adding a Single Row

→ `df.append({'First': 'Tony'}, ignore_index=True)`

→	Age	Full-name	First	Last
0	20	Essam Shenhab	Essam	Shenhab
1	21	Jannet Doe	Jannet	Doe
2	22	Will Smith	Will	Smith
3	NaN	NaN	Tony	NaN

* Append Data Frames

```
people = {  
    'First': ['Tony', 'Steve'],  
    'Last': ['Stark', 'Rogers'],  
}
```

`df2 = Pd.DataFrame(people)`

→ `df=df.append(df2, ignore_index=True)` → يوجد ملق أفصل
 → `df=pd.concat([df,df2], ignore_index=True, sort=True)`
 ✖ Sort Columns ↗

df →

	age	First	Last
0	20	Essam	Shenhob
1	21	Jannet	Doe
2	22	Will	Smith
3	NaN	Tony	CLark
4	NaN	STeve	Rogers

✖ Remove Rows

→ `df.drop(index=3)` →

	age	First	Last
0	20	Essem	Shenhob
1	21	Jannet	Doe
2	22	Will	Smith
4	NaN	STeve	Rogers

→ `df.drop(index=df[df['Last']=='Doe'].index)`

→

	First	Last	age
0	Essam	Shenhob	20
2	Will	Smith	22
3	Tony	STark	NaN
4	STeve	Rogers	NaN

* Sorting Data

→ `df.sort_values(by='Last')` → Dataframe is sorted alphabetically depending on the (Last) and if it was numbers it would be sorted by the highest number.

→

	First	Last	age
1	Jannet	Doe	21
0	Essam	Shenhob	20
2	Will	Smith	22

* Sorting in descending order

→ `df.sort_values(by='Last', ascending=False)`

* If there duplicates →

→ `df.sort_values(by=['Last', 'First'], ascending=False)`

→ `df.sort_values(by=['Last', 'First'], ascending=[False, True])`

→

	First	Last	Age	Inplace = True
0	Corey	Barlog	45	
3	Adam	Doe	16	
1	Jane	Doe	20	
2	John	Doe	22	

→ `df.sort_index` → لترتيب لنفس الترتيب تاني

→ `df['last'].sort_values()` →

3	Doe
1	Doe
2	Doe
0	Barlog

* Sorting in a survey

→ `df['Salary USD'].nlargest(3)` →

Index	Salary USD
2607	20000
7516	20000
3219	20000

→ `df.nlargest(3, 'Salary USD')` → يعرض جميع البيانات الخاصة بالـ `Salary USD`

→ `df.nsmallest` بالـ `index`