

23 Feb Ass

March 29, 2023

```
[ ]: Q1. Create a Pandas Series that contains the following data: 4, 8, 15, 16, 23, and 42. Then, print the series.
```

```
[ ]: ANS -
```

```
[26]: import pandas as pd
import numpy as np
```

```
[49]: Series = [[4,8,15,16,23,42]]
```

```
[51]: print(Series)
```

```
[[4, 8, 15, 16, 23, 42]]
```

```
[ ]:
```

```
[ ]: Q2. Create a variable of list type containing 10 elements in it, and apply pandas.Series function on the variable print it.
```

```
[ ]: ANS -
```

```
[2]: import pandas as pd
```

```
[3]: df = pd.Series(['Datascience' , 'Webapp' , 'Pwskills' , 'Programming' , 'Coding' , 'Skills' , 'Hardwork' , 'Shadab' , 'Khan' , 'Knowledge'])
```

```
[4]: print (df)
```

```
0    Datascience
1      Webapp
2    Pwskills
3  Programming
4      Coding
5      Skills
6    Hardwork
7      Shadab
8      Khan
```

9 Knowledge
dtype: object

[]:

[]:

[]: Q3. Create a Pandas DataFrame that contains the following data:

[]: ANS -

[44]: `import pandas as pd`

[45]: `data = [["Alice",25,"female"] , ["Bob",30,"male"] , ["clarie",27,"female"]]`

[46]: `df = pd.DataFrame(data, columns = ['Name' , 'Age' , 'Gender'])`

[47]: `print(df)`

	Name	Age	Gender
0	Alice	25	female
1	Bob	30	male
2	clarie	27	female

[]:

[]:

[]: Q4. What is 'DataFrame' in pandas and how is it different from pandas.series?
→ Explain with an example.

[]: ANS -

[]: A Pandas DataFrame is a two-dimensional data structure that can be thought of
→ as a spreadsheet. It can also be thought of as a collection of two or more series with common indices.

[]: A Pandas Series is a one-dimensional data structured that comprises of
→ key-value pair, where keys/labels are the values stored on that index. It is similar to a python
→ dictionary, except it provides more freedom to manipulate and edit the data.

[]:

[]:

```
[ ]:
```

```
[ ]:
```

```
[ ]:
```

```
[ ]: Q5. What are some common functions you can use to manipulate data in a Pandas DataFrame? Can you give an example of when you might use one of these functions?
```

```
[ ]: ANS -
```

```
[ ]: Apply Function in Pandas it is one of the commonly used pandas functions for manipulating a pandas dataframe and creating new variables. Pandas Apply function returns some value after passing each row/column of a data frame with some function. The function can be both default or user-defined.
```

```
[1]: import pandas as pd
```

```
[3]: df = pd.read_csv("https://raw.githubusercontent.com/datasciencedojo/datasets/master/titanic.csv")
```

```
[4]: df.head()
```

```
[4]:
```

	PassengerId	Survived	Pclass	\
0	1	0	3	
1	2	1	1	
2	3	1	3	
3	4	1	1	
4	5	0	3	

	Name	Sex	Age	SibSp	\
0	Braund, Mr. Owen Harris	male	22.0	1	
1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	
2	Heikkinen, Miss. Laina	female	26.0	0	
3	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	
4	Allen, Mr. William Henry	male	35.0	0	

	Parch	Ticket	Fare	Cabin	Embarked
0	0	A/5 21171	7.2500	NaN	S
1	0	PC 17599	71.2833	C85	C
2	0	STON/O2. 3101282	7.9250	NaN	S
3	0	113803	53.1000	C123	S
4	0	373450	8.0500	NaN	S

```
[5]: df.tail()
```

```
[5]: PassengerId  Survived  Pclass                                Name \
886          887         0         2                        Montvila, Rev. Juozas
887          888         1         1                  Graham, Miss. Margaret Edith
888          889         0         3  Johnston, Miss. Catherine Helen "Carrie"
889          890         1         1                  Behr, Mr. Karl Howell
890          891         0         3                  Dooley, Mr. Patrick
```

```
      Sex  Age  SibSp  Parch    Ticket   Fare Cabin Embarked
886  male  27.0     0     0    211536  13.00   NaN        S
887  female 19.0     0     0    112053  30.00   B42        S
888  female  NaN     1     2  W./C. 6607  23.45   NaN        S
889  male   26.0     0     0    111369  30.00  C148        C
890  male   32.0     0     0    370376   7.75   NaN        Q
```

```
[8]: df['shadab'] = 0
```

```
[9]: df.head()
```

```
[9]: PassengerId  Survived  Pclass  \
0              1         0         3
1              2         1         1
2              3         1         3
3              4         1         1
4              5         0         3
```

```
      Name      Sex  Age  SibSp  \
0      Braund, Mr. Owen Harris    male  22.0     1
1  Cumings, Mrs. John Bradley (Florence Briggs Th...  female  38.0     1
2      Heikkinen, Miss. Laina    female  26.0     0
3  Futrelle, Mrs. Jacques Heath (Lily May Peel)    female  35.0     1
4      Allen, Mr. William Henry    male  35.0     0
```

```
      Parch    Ticket   Fare Cabin Embarked  new_col  shadab
0         0  A/5 21171   7.2500   NaN        S         0         0
1         0    PC 17599  71.2833   C85        C         0         0
2         0  STON/O2. 3101282   7.9250   NaN        S         0         0
3         0    113803  53.1000  C123        S         0         0
4         0    373450   8.0500   NaN        S         0         0
```

```
[11]: df['shadab'] = df['PassengerId'] + df['Survived']
```

```
[12]: df.head()
```

```
[12]: PassengerId  Survived  Pclass  \
0              1         0         3
1              2         1         1
2              3         1         3
```

3	4	1	1
4	5	0	3

	Name	Sex	Age	SibSp	\
0	Braund, Mr. Owen Harris	male	22.0	1	
1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	
2	Heikkinen, Miss. Laina	female	26.0	0	
3	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	
4	Allen, Mr. William Henry	male	35.0	0	

	Parch	Ticket	Fare	Cabin	Embarked	new_col	shadab
0	0	A/5 21171	7.2500	NaN	S	0	1
1	0	PC 17599	71.2833	C85	C	0	3
2	0	STON/O2. 3101282	7.9250	NaN	S	0	4
3	0	113803	53.1000	C123	S	0	5
4	0	373450	8.0500	NaN	S	0	5

```
[13]: df[df["Age"] > 18]
```

```
[13]:
```

	PassengerId	Survived	Pclass	\
0	1	0	3	
1	2	1	1	
2	3	1	3	
3	4	1	1	
4	5	0	3	
..	
885	886	0	3	
886	887	0	2	
887	888	1	1	
889	890	1	1	
890	891	0	3	

	Name	Sex	Age	SibSp	\
0	Braund, Mr. Owen Harris	male	22.0	1	
1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	
2	Heikkinen, Miss. Laina	female	26.0	0	
3	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	
4	Allen, Mr. William Henry	male	35.0	0	
..	
885	Rice, Mrs. William (Margaret Norton)	female	39.0	0	
886	Montvila, Rev. Juozas	male	27.0	0	
887	Graham, Miss. Margaret Edith	female	19.0	0	
889	Behr, Mr. Karl Howell	male	26.0	0	
890	Dooley, Mr. Patrick	male	32.0	0	

	Parch	Ticket	Fare	Cabin	Embarked	new_col	shadab
0	0	A/5 21171	7.2500	NaN	S	0	1

1	0	PC 17599	71.2833	C85	C	0	3
2	0	STON/02. 3101282	7.9250	NaN	S	0	4
3	0	113803	53.1000	C123	S	0	5
4	0	373450	8.0500	NaN	S	0	5
..
885	5	382652	29.1250	NaN	Q	0	886
886	0	211536	13.0000	NaN	S	0	887
887	0	112053	30.0000	B42	S	0	889
889	0	111369	30.0000	C148	C	0	891
890	0	370376	7.7500	NaN	Q	0	891

[575 rows x 14 columns]

```
[14]: len(df) - len(df[df["Age"] > 18])
```

```
[14]: 316
```

```
[15]: df[df['Fare'] < 32.204208]
```

```
[15]:
```

	PassengerId	Survived	Pclass	Name \
0	1	0	3	Braund, Mr. Owen Harris
2	3	1	3	Heikkinen, Miss. Laina
4	5	0	3	Allen, Mr. William Henry
5	6	0	3	Moran, Mr. James
7	8	0	3	Palsson, Master. Gosta Leonard
..
886	887	0	2	Montvila, Rev. Juozas
887	888	1	1	Graham, Miss. Margaret Edith
888	889	0	3	Johnston, Miss. Catherine Helen "Carrie"
889	890	1	1	Behr, Mr. Karl Howell
890	891	0	3	Dooley, Mr. Patrick

	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked \
0	male	22.0	1	0	A/5 21171	7.2500	NaN	S
2	female	26.0	0	0	STON/02. 3101282	7.9250	NaN	S
4	male	35.0	0	0	373450	8.0500	NaN	S
5	male	NaN	0	0	330877	8.4583	NaN	Q
7	male	2.0	3	1	349909	21.0750	NaN	S
..
886	male	27.0	0	0	211536	13.0000	NaN	S
887	female	19.0	0	0	112053	30.0000	B42	S
888	female	NaN	1	2	W./C. 6607	23.4500	NaN	S
889	male	26.0	0	0	111369	30.0000	C148	C
890	male	32.0	0	0	370376	7.7500	NaN	Q

	new_col	shadab
0	0	1

```

2         0         4
4         0         5
5         0         6
7         0         8
..      ...      ...
886        0      887
887        0      889
888        0      889
889        0      891
890        0      891

```

[680 rows x 14 columns]

```
[16]: df[df["Fare"] == 0]['Name']
```

```

[16]: 179          Leonard, Mr. Lionel
      263          Harrison, Mr. William
      271      Tornquist, Mr. William Henry
      277      Parkes, Mr. Francis "Frank"
      302      Johnson, Mr. William Cahoon Jr
      413      Cunningham, Mr. Alfred Fleming
      466          Campbell, Mr. William
      481      Frost, Mr. Anthony Wood "Archie"
      597          Johnson, Mr. Alfred
      633      Parr, Mr. William Henry Marsh
      674      Watson, Mr. Ennis Hastings
      732          Knight, Mr. Robert J
      806      Andrews, Mr. Thomas Jr
      815          Fry, Mr. Richard
      822      Reuchlin, Jonkheer. John George
      Name: Name, dtype: object

```

```
[17]: df[df['Sex'] == 'male']
```

```

[17]:   PassengerId  Survived  Pclass   Name  Sex \
0         1         0         3  Braund, Mr. Owen Harris  male
4         5         0         3  Allen, Mr. William Henry  male
5         6         0         3    Moran, Mr. James  male
6         7         0         1  McCarthy, Mr. Timothy J  male
7         8         0         3  Palsson, Master. Gosta Leonard  male
..      ...      ...      ...      ...  ...
883      884         0         2  Banfield, Mr. Frederick James  male
884      885         0         3    Sutehall, Mr. Henry Jr  male
886      887         0         2    Montvila, Rev. Juozas  male
889      890         1         1    Behr, Mr. Karl Howell  male
890      891         0         3    Dooley, Mr. Patrick  male

```

	Age	SibSp	Parch		Ticket	Fare	Cabin	Embarked	new_col	\
0	22.0	1	0		A/5 21171	7.2500	NaN	S	0	
4	35.0	0	0		373450	8.0500	NaN	S	0	
5	NaN	0	0		330877	8.4583	NaN	Q	0	
6	54.0	0	0		17463	51.8625	E46	S	0	
7	2.0	3	1		349909	21.0750	NaN	S	0	
..		
883	28.0	0	0	C.A./SOTON	34068	10.5000	NaN	S	0	
884	25.0	0	0	SOTON/OQ	392076	7.0500	NaN	S	0	
886	27.0	0	0		211536	13.0000	NaN	S	0	
889	26.0	0	0		111369	30.0000	C148	C	0	
890	32.0	0	0		370376	7.7500	NaN	Q	0	

	shadab
0	1
4	5
5	6
6	7
7	8
..	...
883	884
884	885
886	887
889	891
890	891

[577 rows x 14 columns]

[]:

[]:

[]:

[]:

[]: Q6. Which of the following is mutable in nature Series, DataFrame, Panel?

[]: ANS -

[]: All Pandas data structure are value mutable.(can be changed) and except Series, DataFrame, Panel all are size mutable.
 Series all are size mutable.series is size immutable.
 Dataframe is widely used and one of the most important data structures. Panel is used much less.

[]: Series is a one-dimensional array like structure with homogeneous data.


```
[ ]: DataFrame is a two-dimensional array with heterogeneous data.
```

```
[ ]: Panel is a three-dimensional data structure with heterogeneous data. it is hard
↳to represent the panel
in graphical representation. But a panel can be illustrated as a container of
↳DataFrame.
```

```
[ ]:
```

```
[ ]:
```

```
[ ]: Q7. Create a DataFrame using multiple Series. Explain with an example.
```

```
[ ]: ANS -
```

```
[34]: import pandas as pd
```

```
[35]: courses = pd.Series(["data_science" , "web_dev" , "programming"])
fees = pd.Series([2345 , 34323 , 767656])
discount = pd.Series([1000 , 1500 , 2000])
```

```
[36]: df = pd.concat([courses , fees] , axis=1)
```

```
[37]: df = pd.concat([courses , fees , discount] , axis=1)
```

```
[38]: print(df)
```

	0	1	2
0	data_science	2345	1000
1	web_dev	34323	1500
2	programming	767656	2000

```
[ ]:
```

```
[ ]:
```