



Pandas For Beginners With Anas Arshad



ANAS ARSHAD

Why we use pandas?



anasarshad.py

Why we use Pandas?

Pandas is used in Python for data manipulation and analysis. It provides data structures for efficiently storing large datasets and tools for working with them. Some of the key features of pandas include:

- Data manipulation: subsetting, merging, and transforming data
- Data analysis: aggregating, summarizing, and calculating statistical values
- Data visualization: generating charts and graphs to visualize data
- Handling missing data: providing methods to handle missing or incomplete data

Overall, Pandas helps in cleaning, transforming, and analyzing large datasets in an efficient and streamlined way.

ANAS ARSHAD



How to create series?



anasarshad.py

Series

In pandas, a Series is created to represent a one-dimensional labeled array of data. The data in a Series can be of any type including integer, float, string, or a combination of types. Creating a Series in pandas allows for efficient data storage, manipulation and analysis, making it an important part of the data analysis and data manipulation process.

Code

```
import pandas as pd
db = ['C++', 'JAVA', 'RUST']
df = pd.Series(db)
print(df, type(df))
```

Output

```
0      C++
1     JAVA
2     RUST
dtype: object <class 'pandas.core.series.Series'>
```

ANAS ARSHAD



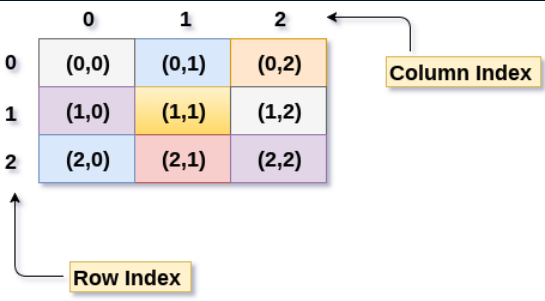
ANAS ARSHAD

What is DataFrame?

anasarshad.py

DataFrame

Pandas dataframes are created to efficiently handle and manipulate data in a tabular format. They allow us to organize and analyze large datasets with labeled rows and columns, making it easy to access and manipulate data based on specific features. The ability to perform operations such as filtering, aggregation, and transformation of data, as well as the integration with other libraries for data visualization, makes Pandas a valuable tool for data analysis and manipulation tasks.



The diagram illustrates a DataFrame as a 3x3 grid of cells. Each cell contains a coordinate pair (row index, column index). The columns are labeled 0, 1, and 2 at the top. The rows are labeled 0, 1, and 2 on the left. A yellow box labeled 'Column Index' with an arrow points to the column headers. A yellow box labeled 'Row Index' with an arrow points to the row headers.

	0	1	2
0	(0,0)	(0,1)	(0,2)
1	(1,0)	(1,1)	(1,2)
2	(2,0)	(2,1)	(2,2)

ANAS ARSHAD



How to create DataFrame?

anasarshad.py

Code

```
import pandas as pd
db = {
    'Language': ['C++', 'JAVA', 'RUST'],
    'OS': ['linux', 'Windows', 'MAC']
}
df = pd.DataFrame(db)
print(df)
```

Output

	Language	OS
0	C++	linux
1	JAVA	Windows
2	RUST	MAC

COLUMNS

ROWS

INDEX

ANAS ARSHAD



ANAS ARSHAD

What is head and tail in pandas?



anasarshad.py

Head and Tail

"head" and "tail" are methods in the Pandas library that allow you to retrieve the first (head) or last (tail) n rows of a dataframe. By default, "head" returns the first 5 rows and "tail" returns the last 5 rows of the dataframe, but you can specify the number of rows to retrieve by passing an argument to the method, e.g. `df.head(10)` returns the first 10 rows of the dataframe `df`. These methods are useful for quickly previewing the data in a dataframe, especially for large datasets.

`df.head()` First Five Rows

Unnamed: 0	Date	AveragePrice	Total Volume	4046
0	2015-12-27	1.33	64236.62	1036.74
1	2015-12-20	1.35	54876.98	674.28
2	2015-12-13	0.93	118220.22	794.70
3	2015-12-06	1.08	78992.15	1132.00
4	2015-11-29	1.28	51039.60	941.48

`df.tail()` Last Five Rows

Unnamed: 0	Date	AveragePrice	Total Volume	4046
18244	2018-02-04	1.63	17074.83	2046.96
18245	2018-01-28	1.71	13888.04	1191.70
18246	2018-01-21	1.87	13766.76	1191.92
18247	2018-01-14	1.93	16205.22	1527.63
18248	2018-01-07	1.62	17489.58	2894.77

ANAS ARSHAD



ANAS ARSHAD

What shape function do?



anasarshad.py

.shape

The shape property in Pandas returns the dimensions of a dataframe. It returns a tuple of the form (rows, columns), representing the number of rows and columns in the dataframe. The shape property is useful for quickly determining the size of a dataframe, especially when working with large datasets, and can be used to check the dimensions of the data before performing operations on it

Code

```
import pandas as pd
db = {
    'Language': ['C++', 'JAVA', 'RUST'],
    'OS': ['linux', 'Windows', 'MAC']
}
df = pd.DataFrame(db)
df.shape
```

Output

(3, 2)

→ ROWS

→ COLUMNS

ANAS ARSHAD



ANAS ARSHAD

What Describe function do?



anasarshad.py

.describe()

In Pandas, the `.describe` function is used to generate descriptive statistics of a `DataFrame` or a `Series`. It returns a summary of the central tendency, dispersion, and shape of the distribution of a set of numerical data, excluding NaN values by default. The descriptive statistics include:

- `count`: the number of non-null observations
- `mean`: the mean of the non-null observations
- `std`: the standard deviation of the non-null observations
- `min`: the minimum of the non-null observations
- `25%`: the first quartile (25th percentile) of the non-null observations
- `50%`: the second quartile (50th percentile) of the non-null observations
- `75%`: the third quartile (75th percentile) of the non-null observations
- `max`: the maximum of the non-null observations

The syntax for using the `.describe` function is: `dataframe.describe()`. It can also be used on a specific column of a `DataFrame` by using the `dataframe['column_label'].describe()` syntax.

ANAS ARSHAD



ANAS ARSHAD

How to use describe function?

anasarshad.py

Code

```
import pandas as pd
db = {
    'Numbers': [1,2,3,4,5,6,7,8,9]
}
df = pd.DataFrame(db)
df.describe()
```

Output

	Numbers
count	9.000000
mean	5.000000
std	2.738613
min	1.000000
25%	3.000000
50%	5.000000
75%	7.000000
max	9.000000

ANAS ARSHAD



ANAS ARSHAD

Why do we transpose our data?



anasarshad.py

.T

In pandas, data can be transposed to change the orientation of the rows and columns. There are several reasons why one might transpose data in pandas, including:

1. Data visualization: Transposing the data can make it easier to visualize the data in a graph or chart, especially if the original arrangement of the data is not optimal for the desired visualization.
2. Matrix operations: If you want to perform matrix operations on the data, transposing the data can align the orientation of the data with the desired orientation for the operation.
3. Data reshaping: Transposing can be used as a step in reshaping data from a wide format to a long format, or vice versa, depending on the needs of a particular analysis.
4. Changing the orientation of the data for easier manipulation: In some cases, transposing the data can change the orientation of the data in a way that makes it easier to manipulate and analyze the data.

In summary, transposing in pandas is a useful tool for changing the orientation of data to better suit the needs of a particular data analysis or visualization task.

ANAS ARSHAD



ANAS ARSHAD

How to use Transpose function?



anasarshad.py

Code

```
import pandas as pd
db = {
    'Numbers':[1,2,3,4,5,6,7,8,9],
    'Fruit':['Apple', 'Peach', 'Orange', 'Mango']
}
df = pd.DataFrame(db)
df
df.T
```

Output

	Numbers	Fruit
0	1	Apple
1	2	Peach
2	3	Orange
3	4	Mango

without Transpose with Transpose

	0	1	2	3
Numbers	1	2	3	4
Fruit	Apple	Peach	Orange	Mango

ANAS ARSHAD



ANAS ARSHAD

How to select single column?



anasarshad.py

df['A'] OR df.A

Where df is your dataframe and column_name is the name of the column you want to select. For example, if you have a dataframe named df with a column named 'column_A', you can select this column by using:

Code

```
import pandas as pd
db = {
    'Numbers':[1,2,3,4,5,6,7,8,9],
    'Fruit':['Apple','Peach','Orange','Mango']
}
df = pd.DataFrame(db)
df['Fruit']
df.Numbers
```

Output

0	Apple
1	Peach
2	Orange
3	Mango

Name: Fruit, dtype: object

0	1
1	2
2	3
3	4

Name: Numbers, dtype: int64

df['A']

df.A

ANAS ARSHAD



ANAS ARSHAD

How to select multiple columnn?



anasarshad.py

```
df[['A','B']]
```

In Pandas, to select multiple columns, you can pass a list of column names to the indexing operator [].

Code

```
import pandas as pd
db = {
    'Numbers':[1,2,3,4,5,6,7,8,9],
    'Fruit':['Apple','Peach','Orange','Mango']
}
df = pd.DataFrame(db)
df[['Numbers','Fruit']]
```

Output

	Fruit	Numbers
0	Apple	1
1	Peach	2
2	Orange	3
3	Mango	4

ANAS ARSHAD



ANAS ARSHAD



Thank You

This is only for Beginners.

If you have any more
questions, feel free to ask!



<https://www.linkedin.com/in/anas-arshad-125059197/>

ANAS ARSHAD



ANAS ARSHAD