

COMMON PYTHON PANDAS INTERVIEW QUESTIONS



Habib Shaikh
AI Expert



Q1.

How can you exclude values from one Pandas Series (A) that do not appear in another Series (B)?

Python

```
import pandas as pd

# Sample Series
A = pd.Series([1, 2, 3, 4, 5])
B = pd.Series([3, 4, 5, 6, 7])

# Filter elements in A that are in B
filtered_A = A[A.isin(B)]
print(filtered_A)
```

Output:

```
go
2 3
3 4
4 5
dtype: int64
```





Q2.

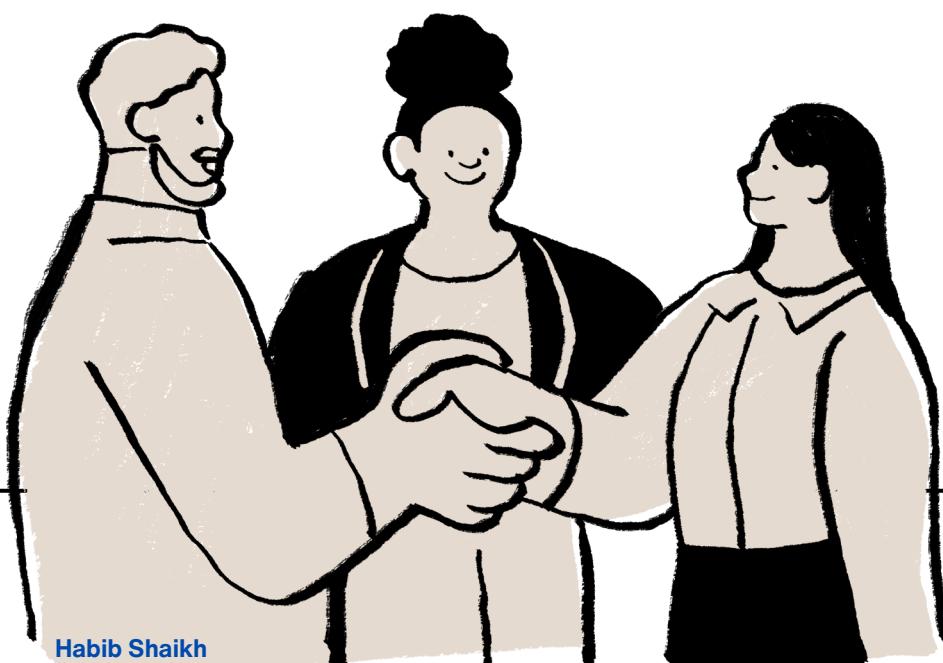
How to find elements unique to two Series (AAA and BBB)?

To identify elements unique to each Series (i.e., those that don't appear in both), you can use the `.symmetric_difference()` method.

Python

```
import pandas as pd
import numpy as np
AAA = pd.Series([1, 2, 3, 4, 5])
BBB = pd.Series([4, 5, 6, 7, 8])

# Find unique elements
unique_elements =
pd.Series(list(set(AAA).symmetric_difference(set(BBB))))
print(unique_elements)
```



Q3.

What is a Pandas DataFrame and how is it created?

A Pandas DataFrame is a 2D data structure for storing data in rows and columns.

You can create a DataFrame using dictionaries, lists, or arrays.

Python

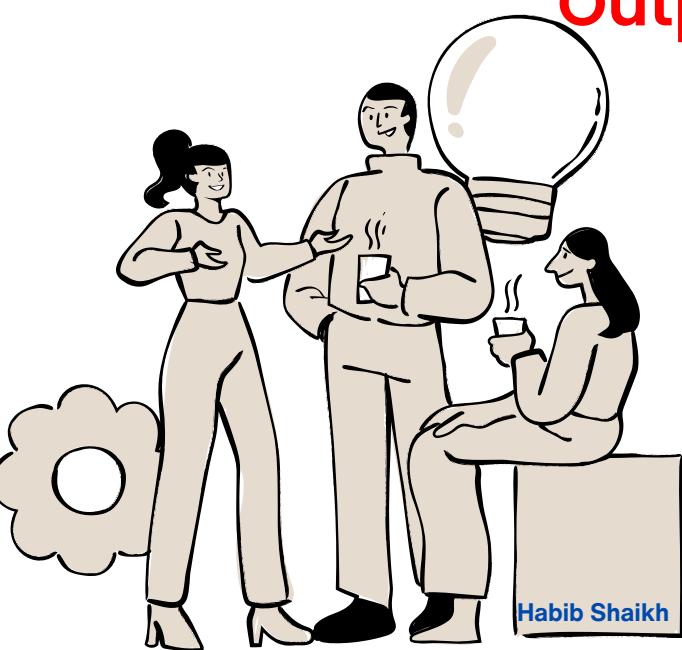
```
import pandas as pd

data = {'Name': ['Alice', 'Bob'], 'Age': [25, 30]}
df = pd.DataFrame(data)
print(df)
```

Output :

go

	Name	Age
0	Alice	25
1	Bob	30





Q4.

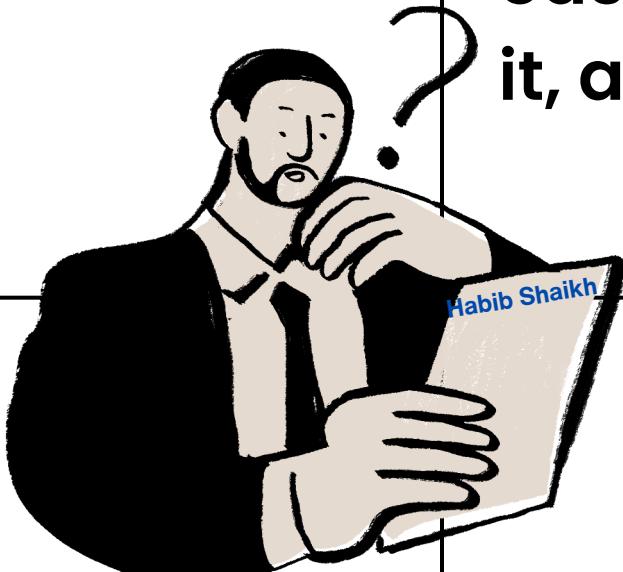
What is the role of Pandas in data analysis, and how is it used?

Pandas is a powerful Python library used for data manipulation and analysis. It provides data structures like **DataFrames** to handle and analyze structured data efficiently.

How is it used?

Pandas allows you to clean, filter, and transform data, making it essential for tasks like data analysis, exploration, and preprocessing.

For example, with a **DataFrame**, you can easily read data from CSVs, manipulate it, and perform calculations.



Q5.

Can you create a Pandas Series from a dictionary?

Yes, you can create a Series from a dictionary. The keys become the index, and the values are the data.

Python

```
import pandas as pd

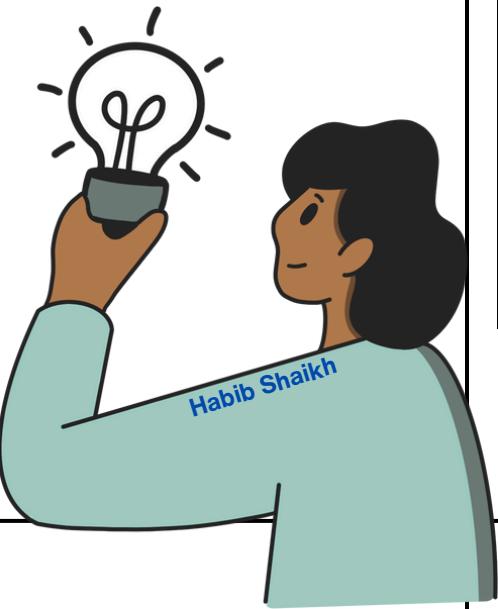
data = {'a': 10, 'b': 20, 'c': 30}
series = pd.Series(data)

print(series)
```

OUTPUT :

```
go

a 10
b 20
c 30
dtype: int64
```



Q6.

How to find items unique to either Series A or B?

Use `~isin()` to filter items not common in both Series.

Python

```
import pandas as pd
```

```
A = pd.Series([1, 2, 3, 4])  
B = pd.Series([3, 4, 5, 6])
```

```
# Items unique to A or B  
unique =  
A[~A.isin(B)].append(B[~B.isin(A)]).reset_index(drop=  
True)  
  
print(unique)
```



OUTPUT :

This returns elements that are not shared between A and B.

```
go  
0 1  
1 2  
2 5  
3 6  
dtype: int64
```



Q7.

How to delete rows or columns in a Pandas DataFrame?

Use `drop()` to remove rows or columns.

Python

```
import pandas as pd

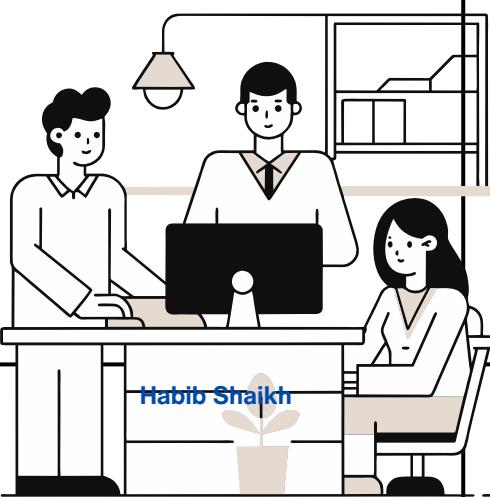
df = pd.DataFrame({'A': [1, 2, 3], 'B': [4, 5, 6]})

# Delete row with index 1
df = df.drop(index=1)

# Delete column 'B'
df = df.drop(columns='B')

print(df)
```

This removes specified rows or columns from the DataFrame.



Q8.

How to combine Pandas DataFrames?

Use `concat()` to stack DataFrames or `merge()` to join them on common columns.

Python

```
import pandas as pd
```

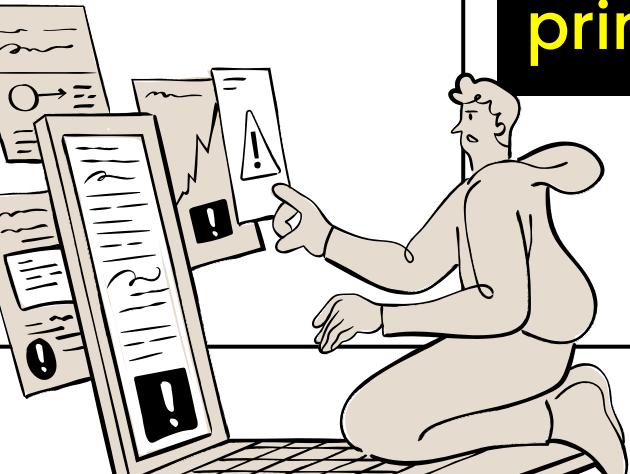
```
# Using concat
```

```
df1 = pd.DataFrame({'A': [1, 2]})  
df2 = pd.DataFrame({'A': [3, 4]})  
result = pd.concat([df1, df2])
```

```
# Using merge
```

```
df1 = pd.DataFrame({'ID': [1], 'Name': ['Alice']})  
df2 = pd.DataFrame({'ID': [1], 'Age': [25]})  
result = pd.merge(df1, df2, on='ID')
```

```
print(result)
```



Q9.

Can you create a Pandas Series from a dictionary?

Yes, you can create a Pandas Series directly from a dictionary. The dictionary keys become the index, and the values become the data.

Python

```
import pandas as pd
```

```
# Creating a Series from a dictionary
data = {'a': 10, 'b': 20, 'c': 30}
series = pd.Series(data)
print(series)
```

OUTPUT :

```
go
a  10
b  20
c  30
dtype: int64
```

This creates a Series with the dictionary keys as the index and values as the data.



Q10.

What is reindexing in Pandas?

Reindexing in Pandas is the process of changing the order or labels of a DataFrame or Series index. It can add, remove, or rearrange rows or columns.

Python

```
import pandas as pd
```

```
df = pd.DataFrame({'Name': ['Alice', 'Bob'], 'Age': [25, 30]},  
index=[0, 1])
```

```
# Reindexing to a new order  
df = df.reindex([1, 0, 2])  
print(df)
```

OUTPUT :

```
      go  
      Name  Age  
1  Bob  30.0  
0 Alice  25.0  
2  NaN  NaN
```



Q11.

How to add a new column to a Pandas DataFrame?

You can add a new column by assigning values to a new column name.

Python

```
import pandas as pd

df = pd.DataFrame({'Name': ['Alice', 'Bob'], 'Age': [25, 30]})

# Adding a new column
df['City'] = ['New York', 'Los Angeles']

print(df)
```

OUTPUT :

```
      Name  Age   City
0    Alice   25  NewYork
1     Bob   30  Los Angeles
```



Follow on Social Media



Habib Shaikh
AI Expert

Let's Get Connected for Our Latest News & Updates



https://medium.com/@aikadoctor_habibshaikh



<https://www.instagram.com/habib.shaikh2010/reels/>



<https://whatsapp.com/channel/0029Vb0PlJe3WHTWIRbE9x0J>



<https://www.youtube.com/@AiKaDoctor>