******

JAHANGIR HUSSEN

SONARGAON UNIVERSITY

|  |
| --- |
| ***Python PANDAS Livery***  ***Here have some basic building function*** |

|  |  |
| --- | --- |
| # ১. pd.Series  # কাজ: একটি পান্ডাস সিরিজ তৈরি করে।  import pandas as pd  data = [1, 2, 3, 4]  series = pd.Series(data)  print(series) | [1, 2, 3, 4] |
| # ২. pd.DataFrame  # কাজ: একটি পান্ডাস ডেটা ফ্রেম তৈরি করে।  import pandas as pd  data = {'Name': ['Alice', 'Bob'], 'Age': [25, 30]}  df = pd.DataFrame(data)  print(df) | Name  Age  0 Alice   25  1   Bob   30 |
| # ৩. pd.read\_csv()  # কাজ: CSV ফাইল থেকে ডেটা পড়ে আনে।  import pandas as pd  csv\_data = pd.read\_csv('file.csv')  print(csv\_data) | (CSV ফাইল থেকে ডেটা প্রদর্শন করবে) |
| # ৪. pd.read\_excel()  # কাজ: Excel ফাইল থেকে ডেটা পড়ে আনে।  import pandas as pd  excel\_data = pd.read\_excel('file.xlsx')  print(excel\_data) | (Excel ফাইল থেকে ডেটা প্রদর্শন করবে) |
| # ৫. pd.read\_json()  # কাজ: JSON ফরম্যাটে ডেটা পড়ে আনে।  import pandas as pd  json\_data = pd.read\_json('file.json')  print(json\_data) | (JSON ফরম্যাটে ডেটা প্রদর্শন করবে) |
| # ৬. pd.read\_sql()  # কাজ: SQL ডেটাবেস থেকে ডেটা পড়ে আনে।  import pandas as pd  from sqlalchemy import create\_engine  engine = create\_engine('sqlite:///:memory:')  sql\_query = "SELECT \* FROM table\_name"  sql\_data = pd.read\_sql(sql\_query, engine)  print(sql\_data) | (SQL ডেটাবেস থেকে ডেটা প্রদর্শন করবে) |
| # ৭. pd.concat()  # কাজ: ডেটা ফ্রেম সংযোজন করে।  import pandas as pd  df1 = pd.DataFrame({'A': [1, 2], 'B': [3, 4]})  df2 = pd.DataFrame({'A': [5, 6], 'B': [7, 8]})  concatenated\_df = pd.concat([df1, df2])  print(concatenated\_df) | A  B  0  1  3  1  2  4  0  5  7  1  6  8 |
| # ৮. pd.merge()  # কাজ: ডেটা ফ্রেম মার্জ করে।  import pandas as pd  df1 = pd.DataFrame({'key': ['A', 'B'], 'value': [1, 2]})  df2 = pd.DataFrame({'key': ['A', 'B'], 'value': [3, 4]})  merged\_df = pd.merge(df1, df2, on='key')  print(merged\_df) | key  value\_x  value\_y  0   A        1        3  1   B        2        4 |
| # ৯. pd.groupby()  # কাজ: ডেটা ফ্রেম গ্রুপ করে কিছু কাজ করে।  import pandas as pd  df = pd.DataFrame({'key': ['A', 'B', 'A', 'B'], 'value': [1, 2, 3, 4]})  grouped\_df = df.groupby('key').sum()  print(grouped\_df) | value  key  A        4  B        6 |
| # ১০. pd.pivot\_table()  # কাজ: ডেটা ফ্রেম থেকে পিভট টেবিল তৈরি করে।  import pandas as pd  df = pd.DataFrame({'A': ['foo', 'foo', 'foo', 'bar', 'bar'], 'B': [1, 2, 3, 4, 5]})  pivot\_table\_df = pd.pivot\_table(df, values='B', index='A', aggfunc='sum')  print(pivot\_table\_df) | B  A  bar     9  foo     6 |
| # ১১. pd.isnull()  # কাজ: মিসিং ভ্যালু চেক করে এবং তাদের জন্য বুলিয়ান সিরিজ ফাইরে।  import pandas as pd  df = pd.DataFrame({'A': [1, 2, None, 4]})  is\_null\_series = pd.isnull(df['A'])  print(is\_null\_series) | 0    False  1    False  2     True  3    False  Name: A, dtype: bool |
| # ১২. pd.notnull()  # কাজ: মিসিং ভ্যালু চেক করে এবং তাদের জন্য বুলিয়ান সিরিজ ফাইরে (pd.isnull() এর উল্টা)।  import pandas as pd  df = pd.DataFrame({'A': [1, 2, None, 4]})  not\_null\_series = pd.notnull(df['A'])  print(not\_null\_series) | 0     True  1     True  2    False  3     True  Name: A, dtype: bool |
| # ১৩. pd.dropna()  # কাজ: মিসিং ভ্যালুগুলি সরাতে বা ডেটা ফ্রেম থেকে মিসিং ভ্যালু থাকলে সংগ্রহ করতে।  import pandas as pd  df = pd.DataFrame({'A': [1, 2, None, 4]})  cleaned\_df = df.dropna()  print(cleaned\_df) | A  0  1.0  1  2.0  3  4.0 |
| # ১৪. pd.fillna()  # কাজ: মিসিং ভ্যালুগুলির জন্য নির্দিষ্ট মান দিয়ে ভরা জাবে।  import pandas as pd  df = pd.DataFrame({'A': [1, 2, None, 4]})  filled\_df = df.fillna(0)  print(filled\_df) | A  0  1.0  1  2.0  2  0.0  3  4.0 |
| # ১৫. pd.unique()  # কাজ: একটি সিরিজ বা লিস্টের ইউনিক মানগুলি প্রদর্শন করে।  import pandas as pd  df = pd.DataFrame({'A': [1, 2, 2, 3, 3, 4]})  unique\_values = pd.unique(df['A'])  print(unique\_values) | [1 2 3 4] |
| # ১৬. pd.value\_counts()  # কাজ: একটি সিরিজ বা লিস্টের মানগুলির সংখ্যা প্রদর্শন করে।  import pandas as pd  df = pd.DataFrame({'A': [1, 2, 2, 3, 3, 4]})  value\_counts = pd.value\_counts(df['A'])  print(value\_counts) | 2    2  3    2  1    1  4    1  Name: A, dtype: int64 |
| # ১৭. pd.duplicated()  # কাজ: একটি সিরিজের প্রতি মানের জন্য দ্বিতীয়বার প্রদর্শন করে বুলিয়ান সিরিজ।  import pandas as pd  df = pd.DataFrame({'A': [1, 2, 2, 3, 3, 4]})  duplicated\_series = pd.duplicated(df['A'])  print(duplicated\_series) | 0    False  1    False  2     True  3     True  4     True  5    False  Name: A, dtype: bool |
| # ১৮. pd.drop\_duplicates()  # কাজ: একটি সিরিজের মধ্যে দুটি বা ততোধিক একই মান থাকলে একটি মান বা দুটি মান বা আরো থাকলে শুধুমাত্র একটি বা তাদের একটি মান দিবে।  import pandas as pd  df = pd.DataFrame({'A': [1, 2, 2, 3, 3, 4]})  deduplicated\_df = df.drop\_duplicates()  print(deduplicated\_df) | A  0  1  1  2  3  3  5  4 |
| # ১৯. pd.to\_datetime()  # কাজ: ডেটা ফ্রেমের একটি কলামকে ডেটাটাইম অবজেক্টে রূপান্তর করে।  import pandas as pd  df = pd.DataFrame({'date': ['2022-01-01', '2022-02-01', '2022-03-01']})  df['date'] = pd.to\_datetime(df['date'])  print(df) | date  0 2022-01-01  1 2022-02-01  2 2022-03-01 |
| # ২০. pd.to\_numeric()  # কাজ: সিরিজের মানগুলি সংখ্যায় রূপান্তর করে।  import pandas as pd  df = pd.DataFrame({'A': ['1', '2', '3']})  df['A'] = pd.to\_numeric(df['A'])  print(df) | A  0  1  1  2  2  3 |
|  |  |
| # ২১. pd.cut()  # কাজ: সিরিজের মানগুলি একটি স্থির স্থানাংক বা ইনটারভালে বিভাজন করে একটি ক্যাটেগরি সিরিজ তৈরি করে।  import pandas as pd  df = pd.DataFrame({'A': [10, 25, 35, 50]})  df['Category'] = pd.cut(df['A'], bins=[0, 20, 40, 60], labels=['Low', 'Mid', 'High'])  print(df) | A Category  0  10      Low  1  25      Mid  2  35      Mid  3  50     High |
| # ২২. pd.qcut()  # কাজ: সিরিজের মানগুলি একটি স্থির স্থানাংক বা ইনটারভালে বিভাজন করে প্রযুক্তি দ্বারা সমান সংখ্যার ইনটারভালে একটি ক্যাটেগরি সিরিজ তৈরি করে।  import pandas as pd  df = pd.DataFrame({'A': [10, 25, 35, 50]})  df['Category'] = pd.qcut(df['A'], q=3, labels=['Low', 'Mid', 'High'])  print(df) | A Category  0  10      Low  1  25      Mid  2  35      Mid  3  50     High |
| # ২৩. pd.get\_dummies()  # কাজ: একটি সিরিজের ক্যাটেগরিক্যাল ভ্যারিয়েবল বা ডাটা ফ্রেমের কলামগুলির জন্য দুটি অসংখ্যিত ভ্যারিয়েবল হিসেবে দৈত্য তৈরি করে।  import pandas as pd  df = pd.DataFrame({'A': ['red', 'blue', 'green']})  dummies\_df = pd.get\_dummies(df['A'])  print(dummies\_df) | blue  green  red  0     0      0    1  1     1      0    0  2     0      1    0 |
| # ২৪. pd.melt()  # কাজ: ডাটা ফ্রেম তৈরি করতে মেল্ট করে ডাটা ফ্রেমকে রুপান্তর করে।  import pandas as pd  df = pd.DataFrame({'A': [1, 2, 3], 'B': [4, 5, 6]})  melted\_df = pd.melt(df, id\_vars=['A'], value\_vars=['B'])  print(melted\_df) | A variable  value  0  1        B      4  1  2        B      5  2  3        B      6 |
| # ২৫. pd.crosstab()  # কাজ: ডাটা ফ্রেম তৈরি করতে মেল্ট করে ডাটা ফ্রেমকে রুপান্তর করে।  import pandas as pd  df = pd.DataFrame({'A': ['foo', 'foo', 'foo', 'bar', 'bar', 'bar'],                     'B': ['one', 'one', 'two', 'two', 'one', 'one']})  crosstab\_df = pd.crosstab(df['A'], df['B'])  print(crosstab\_df) | B    one  two  A  bar    2    1  foo    2    1 |
|  |  |
|  |  |
|  |  |