**22.String manipulation functions in python**

# Sample string

text = "  Hello, World! Python is amazing.  "

# Uppercase

print("Upper:", text.upper())

# Lowercase

print("Lower:", text.lower())

# Title case (capitalizes each word)

print("Title:", text.title())

# Strip (removes leading and trailing whitespaces)

print("Strip:", text.strip())

# Replace (replace a substring)

print("Replace:", text.replace("Python", "AI"))

# Split (splits the string into a list)

print("Split:", text.split())

# Join (joins a list into a string)

words = ["Coding", "is", "fun"]

print("Join:", " ".join(words))

# Find (finds the position of a substring)

print("Find position of 'Python':", text.find("Python"))

# Count occurrences of a substring

print("Count occurrences of 'i':", text.count("i"))

# Startswith (checks if the string starts with a specific substring)

print("Starts with 'Hello':", text.strip().startswith("Hello"))

# Endswith (checks if the string ends with a specific substring)

print("Ends with 'amazing.':", text.strip().endswith("amazing."))

# Zfill (pads the string with zeroes)

num = "42"

print("Zero-filled:", num.zfill(5))

# Isalnum (checks if the string contains only alphanumeric characters)

print("Is 'Python123' alphanumeric?", "Python123".isalnum())

# Isspace (checks if the string contains only whitespace)

print("Is '  ' only spaces?", "  ".isspace())

# Format method

age = 25

print("Using format method:", "I am {} years old.".format(age))

# f-string method

name = "Alice"

print(f"Using f-string: My name is {name} and I am {age} years old.")

# Casefold (converts to lowercase, similar to lower() but more aggressive)

print("Casefold:", text.casefold())

**Output:**

Upper:   HELLO, WORLD! PYTHON IS AMAZING.

Lower:   hello, world! python is amazing.

Title:   Hello, World! Python Is Amazing.

Strip: Hello, World! Python is amazing.

Replace:   Hello, World! AI is amazing.

Split: ['Hello,', 'World!', 'Python', 'is', 'amazing.']

Join: Coding is fun

Find position of 'Python': 16

Count occurrences of 'i': 2

Starts with 'Hello': True

Ends with 'amazing.': True

Zero-filled: 00042

Is 'Python123' alphanumeric? True

Is '  ' only spaces? True

Using format method: I am 25 years old.

Using f-string: My name is Alice and I am 25 years old.

Casefold:   hello, world! python is amazin

**23.Data wrangling in python:join,combine,and reshape**

import pandas as pd

import numpy as np

# Sample datasets

df1 = pd.DataFrame({'ID': [1, 2, 3], 'Name': ['Alice', 'Bob', 'Charlie'], 'Age': [25, 30, 35]})

df2 = pd.DataFrame({'ID': [3, 4, 5], 'Name': ['Charlie', 'David', 'Eve'], 'Age': [35, 40, 45]})

df3 = pd.DataFrame({'ID': [1, 2, 3], 'Salary': [50000, 60000, 70000]})

print("DataFrames before joining:\n", df1, "\n", df2, "\n", df3)

# 1. Joining (Merging data on a common column)

merged\_df = pd.merge(df1, df3, on="ID", how="left")

print("\nData after merging (Left Join):\n", merged\_df)

# 2. Combining (Concatenating data vertically)

combined\_df = pd.concat([df1, df2], ignore\_index=True)

print("\nData after concatenation:\n", combined\_df)

# 3. Reshaping (Pivot & Melt)

reshaped\_df = df1.pivot(index="ID", columns="Name", values="Age")

print("\nPivoted Data:\n", reshaped\_df)

melted\_df = df1.melt(id\_vars=["ID"], var\_name="Attribute", value\_name="Value")

print("\nMelted Data:\n", melted\_df)

# 4. Reshaping using Stack & Unstack

stacked\_df = df1.set\_index("ID").stack()

print("\nStacked Data:\n", stacked\_df)

unstacked\_df = stacked\_df.unstack()

print("\nUnstacked Data:\n", unstacked\_df)

**Output:** DataFrames before joining:

    ID     Name  Age

0   1    Alice   25

1   2      Bob   30

2   3  Charlie   35

    ID     Name  Age

0   3  Charlie   35

1   4    David   40

2   5      Eve   45

    ID  Salary

0   1   50000

1   2   60000

2   3   70000

Data after merging (Left Join):

    ID     Name  Age  Salary

0   1    Alice   25   50000

1   2      Bob   30   60000

2   3  Charlie   35   70000

Data after concatenation:

    ID     Name  Age

0   1    Alice   25

1   2      Bob   30

2   3  Charlie   35

3   3  Charlie   35

4   4    David   40

5   5      Eve   45

Pivoted Data:

 Name  Alice   Bob  Charlie

ID

1      25.0   NaN      NaN

2       NaN  30.0      NaN

3       NaN   NaN     35.0

Melted Data:

    ID Attribute    Value

0   1      Name    Alice

1   2      Name      Bob

2   3      Name  Charlie

3   1       Age       25

4   2       Age       30

5   3       Age       35

Stacked Data:

 ID

1   Name      Alice

    Age          25

2   Name        Bob

    Age          30

3   Name    Charlie

    Age          35

dtype: object

Unstacked Data:

        Name Age

ID

1     Alice  25

2       Bob  30

3   Charlie  35