WhatsApp Message Formatter Using Python - Step-by-Step Code Explanation

This Python script simulates generating a WhatsApp message format using user input. The message includes sender name, receiver name, message content, emojis, and the current time.

from datetime import datetime

Importing the datetime module to capture and format the current time.

def generate\_whatsapp\_message(sender, receiver, message, emoji="", time=None):

Defining a function that takes sender, receiver, message, optional emoji, and time as parameters.

if not time:  
 time = datetime.now().strftime("%I:%M %p")

If time is not provided, it defaults to the current time formatted as 'Hour:Minute AM/PM'.

formatted = f"""  
📱 WhatsApp Message  
  
{receiver}, you have a new message from {sender}:  
  
💬 {message} {emoji}  
  
🕒 Sent at: {time}  
"""

Creating a formatted multi-line string that represents a WhatsApp-style message with all input values.

return formatted.strip()

Stripping any leading/trailing white space from the formatted message before returning.

# Example usage  
if \_\_name\_\_ == "\_\_main\_\_":

Checking if this script is being run as the main program.

print("📲 WhatsApp Message Generator")

Printing a heading for the WhatsApp Message Generator.

sender = input("Enter sender's name: ")

Prompting the user to enter the sender's name.

receiver = input("Enter receiver's name: ")

Prompting the user to enter the receiver's name.

message = input("Type the message: ")

Prompting the user to type the actual message.

emoji = input("Optional: Add emoji(s): ")

Prompting the user to optionally add an emoji.

result = generate\_whatsapp\_message(sender, receiver, message, emoji)

Calling the function with user input and storing the formatted result.

print("\nGenerated Message:\n")

Printing a label before displaying the generated message.

print(result)

Displaying the final generated message in the console.