Awesome! 🚀 Let's dive into **Day 3 of your JARVIS (Violet) project**.

✅ Day 3 Goal: Add Basic Wake Word Detection (Offline)

You've already done:

- ✅ Day 1: Text-to-speech (pyttsx3)
- ✅ Day 2: Speech-to-text (speech_recognition)

Now, we want Violet to **wake up when called** with a name like "Violet" instead of always listening.

🎯 What You'll Learn Today

- What is a **wake word**
- Simple offline wake word detection using Python
- Integrating it with your current speech recognition setup

🌟 Concept: What's a Wake Word?

A **wake word** is a trigger word (like "Alexa" or "Hey Siri") that activates the assistant. Until the wake word is heard, the assistant stays passive.

🧠 Strategy for Day 3

We'll build a loop where:

1. Violet **keeps listening for the wake word** ("violet").
2. If it hears the wake word → it runs your take_command() function to respond.

🛠️ Code Walkthrough

🔁 Step 1: Continuous Wake Word Listening

```
import speech_recognition as sr

import pyttsx3
```

```
engine = pyttsx3.init()
recognizer = sr.Recognizer()

def speak(text):
    engine.say(text)
    engine.runAndWait()

def take_command():
    with sr.Microphone() as source:
        print("Listening for command...")
        audio = recognizer.listen(source)

    try:
        command = recognizer.recognize_google(audio).lower()
        print(f"You said: {command}")
        return command
    except:
        return ""

def listen_for_wake_word(wake_word="violet"):
    with sr.Microphone() as source:
        print("Listening for wake word...")
        audio = recognizer.listen(source)

    try:
        text = recognizer.recognize_google(audio).lower()
        print(f"Heard: {text}")
        return wake_word in text
```

```
    except:
        return False

# Main loop
while True:
    if listen_for_wake_word():
        speak("Yes, I am here!")
        command = take_command()
        # Process the command here
        if "stop" in command:
            speak("Goodbye!")
            break
```

💉 How to Test It

1. Run the code.
2. Say “Violet”.
3. It should respond with “Yes, I am here!” and then wait for a real command.
4. If you say “stop”, it exits.

📝 Optional Challenge

- Change wake word to anything you want.
- Add a **response sound** or LED blink (for future robot version).

🎯 Day 3 Complete When:

- Wake word (“violet”) detection is working
- Command is processed only after saying the wake word
- You can exit using “stop”

Would you like help with improving accuracy or want to go one step further with offline wake word models (like using snowboy or Porcupine)?