```
from PyDictionary import PyDictionary

dictionary = PyDictionary()

while True:
    word = input("Enter your word: ")
    if word == "":
        break

    print(dictionary.meaning(word))
```

This Python code uses the PyDictionary library to look up the meanings of words entered by the user. Here's a step-by-step explanation of what each part of the code does:

1. **`from PyDictionary import PyDictionary`: This line imports the `PyDictionary` class from the PyDictionary library, which is used for word-related tasks like word definitions, synonyms, and antonyms.**

2. **`dictionary = PyDictionary()`: This line creates an instance of the `PyDictionary` class and assigns it to the variable `dictionary`. This instance will be used to perform word-related operations.**

3. **The code enters a `while True:` loop, which means it will continue running until a break condition is met. In this case, it's an infinite loop.**

4. **`word = input("Enter your word: ")`: This line prompts the user to enter a word and stores the input in the `word` variable.**

5. **`if word == "":`: This line checks if the user entered an empty string (i.e., they pressed Enter without typing a word).**
    - **If the user entered an empty string:**
        - **`break`: The `break` statement is used to exit the loop, effectively ending the program.**
    - **If the user entered a word:**
        - **`print(dictionary.meaning(word))`: The program looks up the meaning of the entered word using the `meaning` method of the `dictionary` object and prints the result to the screen.**
6. **The loop continues, allowing the user to enter more words for lookup.**

In summary, this code creates a simple interactive program that repeatedly asks the user to enter a word. It uses the PyDictionary library to look up the meaning of the entered word and displays the meaning on the screen. The program continues running until the user enters an empty string and then exits.