Number Guessing Game: Code Explanation

Introduction

This document provides a detailed breakdown of a Python program that implements a Number Guessing Game.

The game randomly selects a number between 1 and 100, and the user must guess the correct number.

Hints are provided based on whether the guess is too high or too low. The player can continue playing or exit after winning.

Code Breakdown

Importing Required Module

```python
import random
```

- The `random` module is imported to generate a random number between 1 and 100.

Program Initialization

```python
input("Welcome to the Number Guessing Game. Press enter to begin.")
print()
print("-"*20)
print()
```

- The game starts with a welcome message and prompts the user to press Enter to begin.
- Visual separators (`"-"*20"`) are used to format the output cleanly.

Generating a Random Number

```python
number = random.randint(1,100)
guess = 0
sentry = "yes"
```

- `number`: Stores the randomly generated integer between 1 and 100.
- `guess`: Stores the user's inputted guess.
- `sentry`: A control variable that determines whether the game continues.

Game Loop

```python
while sentry.lower() == "yes":
```

- The `while` loop ensures that the game continues as long as the user inputs `"yes"`.

User Input and Condition Checks

```python
guess = int(input("What is your guess? "))
```

- The user is prompted to enter a guess.
- The input is converted to an integer to allow numerical comparison.

Conditional Statements

```python
if guess > number:
    print("Too High\n")
elif guess < number:
    print("Too Low\n")
elif guess == number:
    print("You Guessed It")
```

- If the guess is greater than the target number, the user is informed that the guess is "Too High."
- If the guess is smaller than the target number, the user is informed that the guess is "Too Low."
- If the guess is correct, the program congratulates the user.

Restarting or Ending the Game

```python
sentry = input("Do you want to play again? (Yes or No): ")
```

- After guessing correctly, the user is given the choice to play again.
- If the user enters `"yes"`, the game continues; otherwise, it terminates.

Handling Unexpected Errors

```python
else:
    input("Something went wrong. Please restart the program.")
```

- This line is redundant, as all possible inputs are covered by prior conditions.

Exiting the Program

```python
input("You have ended the program. Press enter to exit")
```

- Ensures the user has time to read the final message before the program closes.

Example Run

```
Welcome to the Number Guessing Game. Press enter to begin.

--------------------

What is your guess? 50
Too High

What is your guess? 25
Too Low

What is your guess? 37
Too Low

What is your guess? 42
You Guessed It
```

```
Do you want to play again? (Yes or No): No

You have ended the program. Press enter to exit.
```

Possible Improvements

1. **Error Handling:** Validate user input to ensure only integers are accepted.

2. **Replay Optimization:** Reset the random number when the user chooses to play again.

3. **Attempts Counter:** Track and display the number of attempts it took to guess the correct number.

4. **Difficulty Levels:** Allow users to choose a range (e.g., 1-50 for easy, 1-200 for hard).

This concludes the explanation of the Number Guessing Game program.