**Number Guessing Game Documentation**

**Overview**

This Python script implements a simple number guessing game where the player tries to guess a randomly selected number between 1 and 100. The game provides feedback on whether the guess is too high or too low until the user guesses correctly.

**Functions**

1. **number\_guessing\_game()**:
   * **Purpose**: Runs the number guessing game.
   * **Parameters**: None.
   * **Returns**: None.
   * **Usage**: The function generates a random number between 1 and 100 and prompts the user to guess the number. The function provides feedback on whether the guess is too high or too low until the user guesses correctly.

**Detailed Explanation**

* **Welcome Message**: The game starts by welcoming the player and explaining the rules.

**Python Code :**

print("Welcome to the Number Guessing Game!")

print("I have selected a number between 1 and 100. Can you guess what it is?")

* **Generating a Random Number**: A random number between 1 and 100 is selected using random.randint().

**Python Code :**

number\_to\_guess = random.randint(1, 100)

* **Tracking Attempts**: A variable attempts is initialized to count the number of guesses the player makes.

**Python Code :**

attempts = 0

* **Game Loop**: The game runs in a loop, continuously prompting the player for their guess until they guess correctly.

**Python Code :**

while True:

* + **Input Handling**: The player is prompted to enter a guess, which is converted to an integer.

**Python Code :**

guess = int(input("Enter your guess: "))

attempts += 1

* + **Input Validation**: The game checks if the guess is within the valid range (1 to 100). If not, it prompts the player to enter a valid number.

**Python Code :**

if guess < 1 or guess > 100:

print("Please enter a number between 1 and 100.")

* + **Guess Feedback**: The game provides feedback on whether the guess is too low or too high. If the guess is correct, it congratulates the player and breaks out of the loop.

**Python Code :**

elif guess < number\_to\_guess:

print("Too low! Try again.")

elif guess > number\_to\_guess:

print("Too high! Try again.")

else:

print(f"Congratulations! You've guessed the correct number in {attempts} attempts.")

break

* + **Error Handling**: If the player enters an invalid input (e.g., a non-numeric value), the game prompts them to enter a valid number.

**Python Code :**

except ValueError:

print("Invalid input. Please enter a valid number.")

**Example Usage :**

To run the number guessing game, execute the script. The user will be prompted to guess the number until they find the correct answer.

**Python Code :**

if \_\_name\_\_ == "\_\_main\_\_":

number\_guessing\_game()

**Complete Python Code:**

import random

def number\_guessing\_game():

print("Welcome to the Number Guessing Game!")

print("I have selected a number between 1 and 100. Can you guess what it is?")

number\_to\_guess = random.randint(1, 100)

attempts = 0

while True:

try:

guess = int(input("Enter your guess: "))

attempts += 1

if guess < 1 or guess > 100:

print("Please enter a number between 1 and 100.")

elif guess < number\_to\_guess:

print("Too low! Try again.")

elif guess > number\_to\_guess:

print("Too high! Try again.")

else:

print(f"Congratulations! You've guessed the correct number in {attempts} attempts.")

break

except ValueError:

print("Invalid input. Please enter a valid number.")

if \_\_name\_\_ == "\_\_main\_\_":

number\_guessing\_game()

This documentation provides an overview and detailed description of the number guessing game's function, its purpose, parameters, and usage. It includes the complete code, making it easy to understand and execute.