Number Guessing Game Documentation

# Overview

This Python program is a simple number guessing game where the computer selects a random number between 1 and 200, and the player has to guess the number within a limited number of tries. The game offers feedback on whether the guess is too high or too low, and keeps track of the number of guesses made. This documentation explains the core functionalities of the program and its flow.

# Code Explanation

## 1. Importing Libraries

The program begins by importing the necessary libraries. It imports the `random` library for generating random numbers and the `time` library for adding delays during the game to enhance user experience.

## 2. Generating Random Numbers

The function `generate\_random\_number()` is responsible for generating a random integer between 1 and 200 using the `random.randint()` function. This number is then used as the target number that the player needs to guess.

## 3. Introduction Function

The function `intro()` introduces the player to the game by asking for their name and explaining the game rules. It informs the player that the program is thinking of a random number between 1 and 200, and prompts them to start guessing.

## 4. Guessing Logic

The function `pick(number)` handles the core guessing logic of the game. It allows the player up to 6 guesses to figure out the correct number. For each guess, the program gives feedback on whether the guess was too high, too low, or correct. If the player guesses correctly within the allowed number of guesses, the game congratulates them and ends. Otherwise, it informs them of the correct number if they fail to guess within the given attempts.

## 5. Error Handling

The program uses error handling with `try-except` blocks to ensure that the player's input is a valid number. If the player enters a non-integer value, the program catches the `ValueError` and informs the player that their input is invalid, without crashing.

## 6. Option to Play Again

At the end of each game, the player is asked if they would like to play again. The program checks if the player's input is 'yes' or 'y', and if so, the game restarts with a new randomly generated number. If the player chooses not to continue, the program exits the loop and ends the game.

## 7. Main Function

The `main()` function controls the flow of the game. It repeatedly calls the `intro()` and `pick()` functions, and asks if the player wants to play again. The `main()` function ensures that the game continues in a loop until the player decides to quit.

# Conclusion

This number guessing game is a simple yet engaging Python project that demonstrates basic concepts like loops, conditional statements, error handling, and random number generation. The user-friendly interaction and feedback make it a fun experience for the player.