Base Completion:

# Import the randrange function to generate random numbers.

from random import randrange

def main():

    """

    Main function to encapsulate the base version of the number guessing game logic.

    """

    # --- Game Setup ---

    # Announce the purpose of the program.

    print("--- Guess the Number! ---")

    # Generate a random number from a range of 0 to 10.

    # randrange(0, 11) includes 0 and goes up to, but not including, 11.

    random\_number = randrange(0, 11)

    # Initialize a counter to track the number of guesses.

    guess\_count = 0

    # --- Main Game Loop ---

    # A 'while True' loop will repeat indefinitely until the user

    # guesses correctly and the 'break' statement is executed.

    while True:

        # Get user input and convert it to an integer.

        guess\_str = input("Guess the number between 0 and 10: ")

        guess = int(guess\_str)

        # Increment the guess counter for each attempt.

        guess\_count += 1

        # --- Guess Checking Logic ---

        # Check if the user's guess matches the random number.

        if guess == random\_number:

            # If the guess is correct, print a success message and exit the loop.

            print(f"Success! You guessed the number correctly.")

            print(f"It took you {guess\_count} trie(s).")

            break

        else:

            # If the guess is incorrect, provide feedback.

            print("Incorrect. Please try again.")

    # Final statement showing who completed the program.

    print("\nCompleted by, Javier Silva")

# This standard line runs the main function when the script is executed.

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

    main()

A screenshot of a computer program

AI-generated content may be incorrect.

A screenshot of a computer program

AI-generated content may be incorrect.

Advanced Completion:

# Import the randrange function to generate random numbers.

from random import randrange

def main():

    """

    Main function to encapsulate the advanced version of the number guessing game logic.

    """

    # Outer loop to control whether the user wants to play again.

    play\_again = 'y'

    while play\_again.lower() == 'y':

        # --- Game Setup ---

        # Announce the purpose of the program and the rules for the new game.

        print("--- New Game: Guess the Number! ---")

        print("I'm thinking of a number between 0 and 10.")

        print("You have 5 tries to guess it.")

        print("-----------------------------------")

        # Generate a random number from a range of 0 to 10.

        # randrange(0, 11) includes 0 and goes up to, but not including, 11.

        random\_number = randrange(0, 11)

        # --- Main Game Loop ---

        # A 'for' loop limits the user to a specific number of guesses (5).

        # The range starts at 1 and goes to 6 to give us guess numbers 1 through 5.

        for guess\_count in range(1, 6):

            # Get user input and tell them which guess they are on.

            guess\_str = input(f"Guess #{guess\_count}: Enter your guess: ")

            guess = int(guess\_str)

            # --- Guess Checking Logic ---

            # Check if the user's guess matches the random number.

            if guess == random\_number:

                print(f"Success! You guessed the number correctly.")

                print(f"It took you {guess\_count} trie(s).")

                break  # Exit the for loop immediately since the game is won.

            # If the user is out of guesses, inform them they lost.

            elif guess\_count == 5:

                print("\nYou've run out of tries. You lose.")

                print(f"The correct number was {random\_number}.")

            # If the guess is incorrect but there are tries left.

            else:

                print("Incorrect. Please try again.")

        print("-----------------------------------")

        # --- Play Again Logic ---

        # Ask the user if they want to play again and validate the input.

        while True:

            play\_again = input("Would you like to play again? (y/n): ")

            # Standardize input to lowercase to handle 'Y' or 'y'.

            if play\_again.lower() == 'y' or play\_again.lower() == 'n':

                break # Exit validation loop on valid input.

            else:

                print("Invalid choice. Please enter 'y' or 'n'.")

        print() # Add a space for readability before the next game.

    # Final statement showing who completed the program.

    print("Thank you for playing!")

    print("Completed by, Javier Silva")

# This standard line runs the main function when the script is executed.

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

    main()

A computer screen shot of a program code

AI-generated content may be incorrect.

A screen shot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

Advance Completion:

# Import the randrange function to generate random numbers.

from random import randrange

def main():

    """

    Main function to encapsulate the full version of the number guessing game logic.

    """

    # Outer loop to control whether the user wants to play again.

    play\_again = 'y'

    while play\_again.lower() == 'y':

        # --- Game Setup ---

        # Announce the purpose of the program and the rules for the new game.

        print("--- New Game: Guess the Number! ---")

        print("I'm thinking of a number between 0 and 100.")

        print("You have 10 tries to guess it.")

        print("-------------------------------------")

        # Generate a random number from a range of 0 to 100.

        # randrange(0, 101) includes 0 and goes up to, but not including, 101.

        random\_number = randrange(0, 101)

        # --- Main Game Loop ---

        # A 'for' loop limits the user to a specific number of guesses (10).

        # The range starts at 1 and goes to 11 to give us guess numbers 1 through 10.

        for guess\_count in range(1, 11):

            # This inner while loop validates that the user enters a valid number.

            while True:

                try:

                    # Get user input and tell them which guess they are on.

                    guess\_str = input(f"Guess #{guess\_count}: Enter your guess: ")

                    guess = int(guess\_str)

                    break # Exit the validation loop if the input is a valid integer.

                except ValueError:

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

            # --- Guess Checking Logic ---

            # Check if the user's guess matches the random number.

            if guess == random\_number:

                print(f"Success! You guessed the number {random\_number} correctly.")

                print(f"It took you {guess\_count} trie(s).")

                break  # Exit the for loop immediately since the game is won.

            # If the user is out of guesses, inform them they lost.

            elif guess\_count == 10:

                print("\nYou've run out of tries. You lose.")

                print(f"The correct number was {random\_number}.")

            # If the guess is incorrect, provide feedback on whether it was too high or too low.

            elif guess > random\_number:

                print("Incorrect. Your guess was too high. Try again.")

            else: # The only remaining possibility is the guess was too low.

                print("Incorrect. Your guess was too low. Try again.")

        print("-------------------------------------")

        # --- Play Again Logic ---

        # Ask the user if they want to play again and validate the input.

        while True:

            play\_again = input("Would you like to play again? (y/n): ")

            # Standardize input to lowercase to handle 'Y' or 'y'.

            if play\_again.lower() == 'y' or play\_again.lower() == 'n':

                break # Exit validation loop on valid input.

            else:

                print("Invalid choice. Please enter 'y' or 'n'.")

        print() # Add a space for readability before the next game.

    # Final statement showing who completed the program.

    print("Thank you for playing!")

    print("Completed by, Javier Silva")

# This standard line runs the main function when the script is executed.

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

    main()

A computer screen shot of a program code

AI-generated content may be incorrect.

A screen shot of a computer

AI-generated content may be incorrect.

A screen shot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.