# Import the randrange function from the random library

from random import randrange

def get\_user\_weapon():

    """

    Displays the weapon choices to the user, gets their input, validates it,

    and returns the valid choice.

    No parameters.

    Returns the user's integer choice (1, 2, or 3).

    """

    while True: # Loop for input validation

        print("Choose your weapon:")

        print("1. Rock")

        print("2. Paper")

        print("3. Scissors")

        # Using a try-except block is an advanced topic, but a simple check

        # for digits is within the scope of what you've learned.

        choice\_str = input("Enter your choice (1, 2, or 3): ")

        if choice\_str.isdigit():

            choice\_int = int(choice\_str)

            if 1 <= choice\_int <= 3:

                return choice\_int

            else:

                print("Invalid choice. Please enter a number between 1 and 3.\n")

        else:

            print("Invalid input. Please enter a number.\n")

def get\_opponent\_weapon():

    """

    Generates a random weapon choice for the opponent.

    No parameters.

    Returns a random integer between 1 and 3.

    """

    # randrange(1, 4) generates a random number from 1 up to (but not including) 4.

    # This results in a choice of 1, 2, or 3.

    return randrange(1, 4)

def determine\_winner(user\_weapon, opponent\_weapon):

    """

    Compares the user's and opponent's weapons to determine the winner.

    Accepts user\_weapon (int) and opponent\_weapon (int) as parameters.

    Prints the outcome of the game.

    """

    # Helper values to make the logic clearer

    ROCK = 1

    PAPER = 2

    SCISSORS = 3

    # Display choices to the user

    # This requires converting the numbers back to strings for a user-friendly message

    weapons = {ROCK: "Rock", PAPER: "Paper", SCISSORS: "Scissors"}

    print(f"You chose: {weapons[user\_weapon]}")

    print(f"Opponent chose: {weapons[opponent\_weapon]}")

    # Determine the winner

    if user\_weapon == opponent\_weapon:

        print("The game is a tie!\n")

    # User win conditions

    elif (user\_weapon == ROCK and opponent\_weapon == SCISSORS) or \

         (user\_weapon == SCISSORS and opponent\_weapon == PAPER) or \

         (user\_weapon == PAPER and opponent\_weapon == ROCK):

        print("You win!\n")

    # If it's not a tie and the user didn't win, the opponent must have won.

    else:

        print("You lose!\n")

def main():

    """

    The main function that controls the flow of the game.

    """

    # Initialize the choice variable to 'y' to start the first game

    play\_again = 'y'

    # Main game loop

    while play\_again.lower() == 'y':

        # Get weapons for the user and the opponent

        user\_weapon = get\_user\_weapon()

        opponent\_weapon = get\_opponent\_weapon()

        # Determine the winner of the round

        determine\_winner(user\_weapon, opponent\_weapon)

        # Ask the user if they want to play another round

        # Includes a validation loop

        while True:

            play\_again = input("Play again? (y/n): ")

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

                print() # Add a space for readability

                break

            else:

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

# This special if statement checks if the script is being run directly.

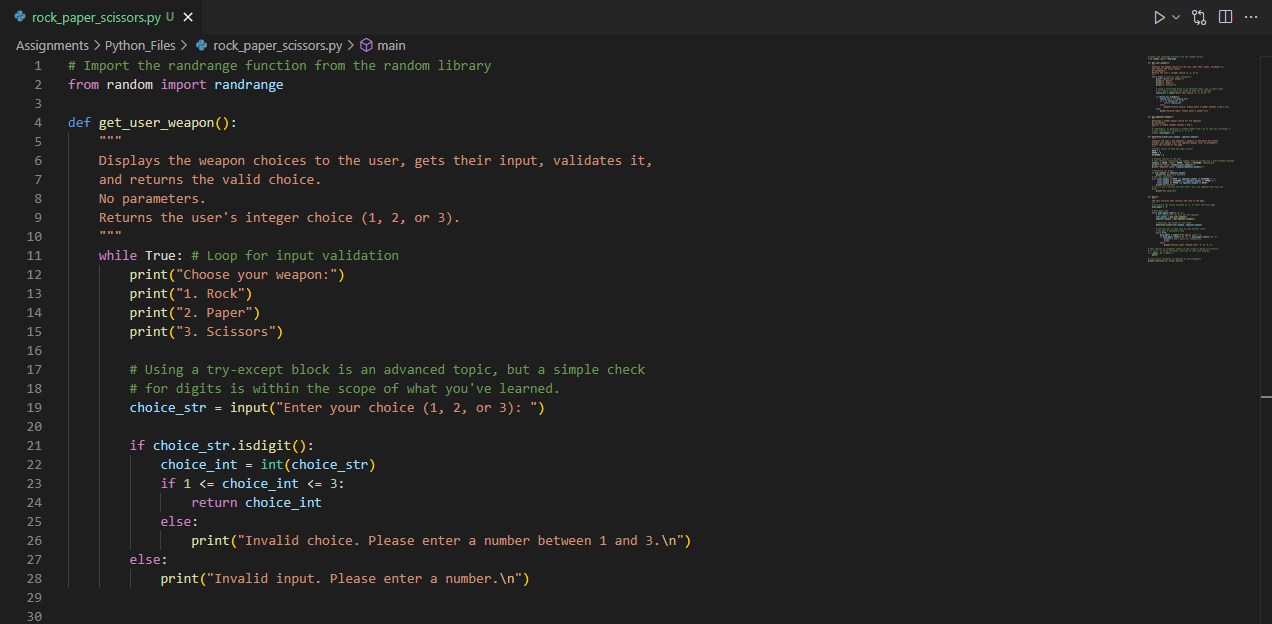
# If it is, it calls the main() function to start the program.

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

    main()

# Final print statement as required by the assignment

print("Completed by, Javier Silva")



A screenshot of a computer program

AI-generated content may be incorrect.

A screenshot of a computer program

AI-generated content may be incorrect.

A screenshot of a computer program

AI-generated content may be incorrect.

A computer screen with text

AI-generated content may be incorrect.