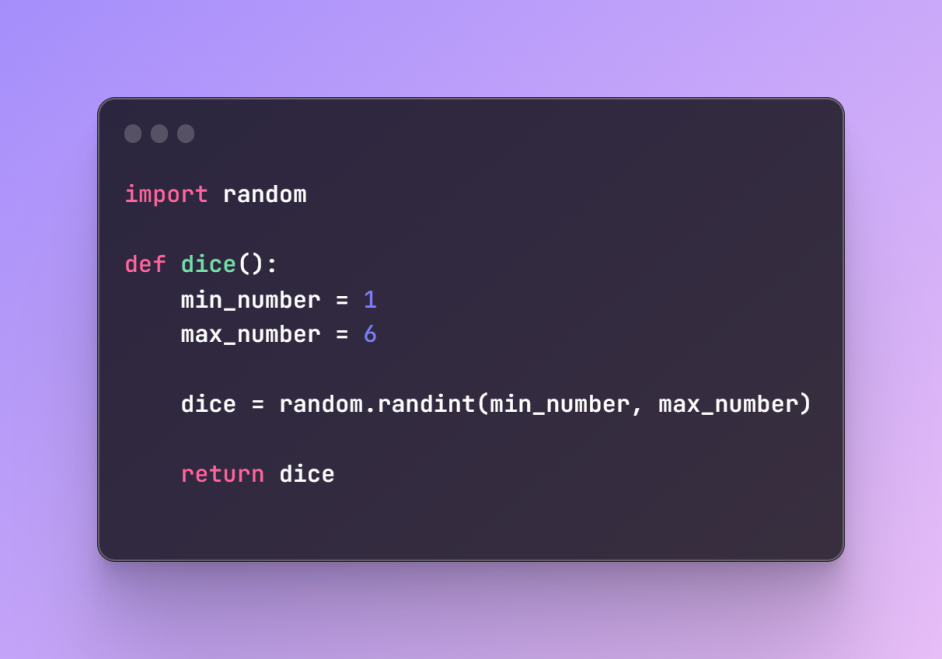
Pig Game

Definitions

Pig is a simple dice game first described in print by John Scarne in 1945. Players take turns to roll a single dice as many times as they wish, adding all roll results to a running total, but losing their granted score for the turn if they roll a 1.

Code and Explanation

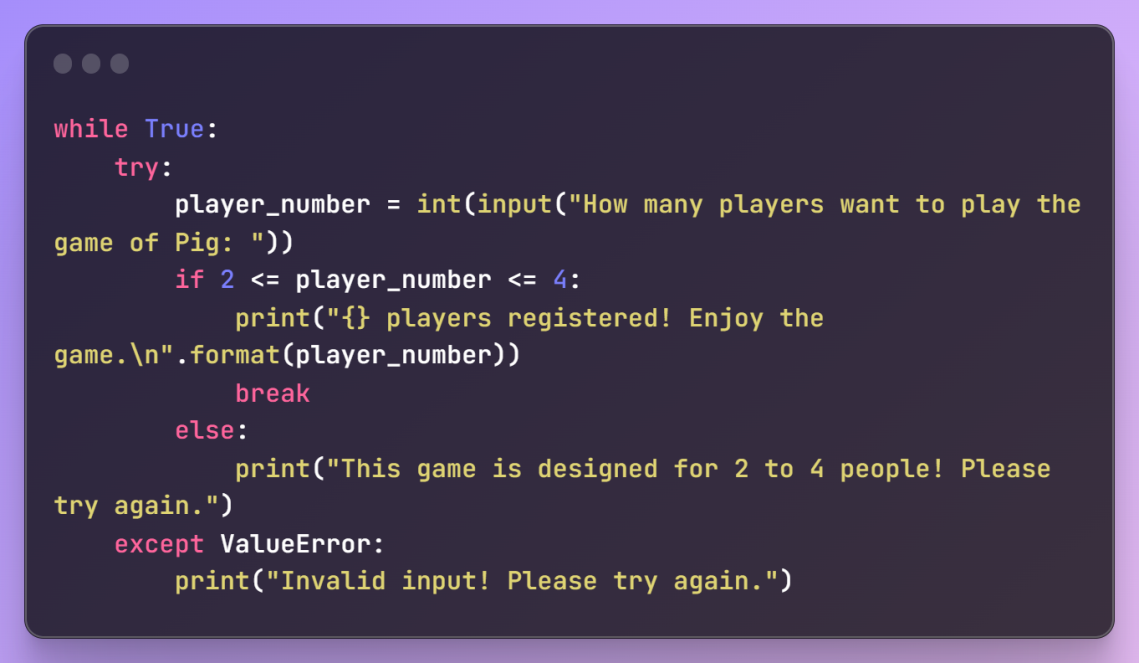
The dice() function is the function that will simulate the roll of a six-sided dice. This function utilizes the built-in **random** module to generate random numbers and the **random.randint()** method to simulate the outcome of rolling a physical dice.



Player Registration Loop

This part of the code initiates a loop that prompts the user to enter the number of players for the game. The **try** block attempts to convert the user’s input into an integer using **int(input())**. If the input is a valid integer and within the range of 2 to 4 players (inclusive), the loop breaks and the game proceeds. If the input is invalid or outside the specified range, an appropriate message is displayed and the user is prompted to try again.

Code:



Player Names Input and Welcome Message

After determining the number of players, the code enters a loop that iterates for each player, prompting them to enter their name. The entered names are then added to the **player\_names** list. Finally, a welcome message is displayed, addressing each player by their chosen names. The **‘, ’.join(player\_names)** part concatenates the player names with commas for a friendly display.

Code:



Winning Score and Scoreboard Initialization

These two lines establish the winning score for the game, **winning\_score**, and initialize the **scoreboard** as a list with zeros, ensuring that each player starts with a score of zero. The **scoreboard** will be updated and maintained as the game progresses allowing the program to keep track of each player’s score until a winner is determined.

Code:



Game Loop

The **while max(scoreboard) < winning\_score** loop ensures that the game continues as long as no player has reached the winning score.

The **for player\_list in range(len(player\_names))** loop iterates through each player’s turn. Player-specific information, such as the player’s name and current score, is displayed.

The inner **while True** loop prompts the player to continue rolling or stop. If the player chooses to stop, the loop breaks, and the turn ends.

The **dice()** function is called to simulate rolling a dice, and the result is stored in **dice\_number.**

If the **dice\_number** is 1, the player’s current score for that turn resets to 0, and the turn ends. Otherwise, the **dice\_number** is added to the player’s current score for that turn, and the total score for the turn is displayed.

The **current\_score** is added to the player’s overall score. (**scoreboard[player\_list])**, and the player’s total score for the game is displayed.



Winner

**Game\_result = max(scoreboard)**, finds the maximum score in the **scoreboard** list, representing the highest total score achieved by any player throughout the game. **Winning\_player = scoreboard.index(game\_result)** is then used to find the index or the position of the maximum score in the **scoreboard** list. This corresponds to the player who achieved the highest score and is determined as the winner.

Code:

