# FizzBuzz Game - Code Explanation

**CODE:**

## Step-by-step Explanation

### Print game title:

print("FizzBuzz Game") prints the game title to the console so the player knows what they started.

### Initialize score:

points = 0 creates a variable to track how many correct answers the player has given.

### Set rounds:

rounds = 10 sets how many rounds the game will run.

### Start values for sequence

pre = 0 and curr = 1 are the two starting values used to create the next number in each round by adding them (Fibonacci-like).

### Loop over rounds

for num in range(1, rounds + 1): loops from 1 up to and including the number of rounds, handling each round of the game.

### Generate current number

total\_num = pre + curr computes the number used for this round by adding the previous and current values.

### Show round and number

print(f"Round {num}: The number is {total\_num}") displays the current round and the generated number to the player.

### Get player's answer

answer = input("Your answer: ") reads the player's guess. The player should enter 'Fizz', 'Buzz', 'FizzBuzz', or the number itself.

### Update sequence values

pre = curr and curr = total\_num shift the sequence so the next loop will produce the next sum (previous becomes the old current, current becomes the new total).

### FizzBuzz logic

The if/elif/else block checks divisibility:  
- if total\_num % 3 == 0 and total\_num % 5 == 0 -> correct = "FizzBuzz"  
- elif total\_num % 3 == 0 -> correct = "Fizz"  
- elif total\_num % 5 == 0 -> correct = "Buzz"  
- else -> correct = str(total\_num)

### Check answer and update score

if answer == correct: increments points and prints a success message; otherwise shows the correct answer.

### Show current score and separator

print(f"Current Score: {points}") and print("-" \* 40) give feedback and a visual separator between rounds.

## Example Sequence Produced

Starting from pre=0 and curr=1, the sequence produced each round (total\_num = pre + curr) will be: 1, 2, 3, 5, 8, 13, 21, 34, 55, 89 (for 10 rounds).

Note: Player input must exactly match the string used for checking (e.g., "Fizz", "Buzz", "FizzBuzz", or the numeric string like "8"). Consider normalizing input (e.g., answer.strip().capitalize() or answer.strip().lower()) if you want case-insensitive checking.