# FizzBuzz Game -

## 1) Original Code:

def fizzbuzz(num):  
 if num % 3 == 0 and num % 5 == 0:  
 return "fizzbuzz"  
 elif num % 3 == 0:  
 return "fizz"  
 elif num % 5 == 0:  
 return "buzz"  
 else:  
 return str(num)  
   
  
def fizzbuzz\_game():  
 print("Welcome to the FizzBuzz Game!")  
 score = 0  
 prev = 0  
 i = 1  
 while True:  
 if prev == 0:  
 print("Game Started, The first number is 1")  
 else:  
 total = prev + i  
 correct\_answer = fizzbuzz(total)  
  
  
 choice = input(f"Enter fizz/buzz/fizzbuzz or {total}, or 'q' to quit: ").strip().lower()  
  
   
 if choice == 'q':  
 print(f"Game ended. Your final score is: {score}")  
 break  
 if choice == correct\_answer:  
 score += 1  
 print(f"Correct! Your score is: {score}\n")  
 else:  
 print(f"Wrong! The correct answer was '{correct\_answer}'")  
 print(f"Final score is: {score}")  
 break  
 prev = i  
 i += 1  
  
  
fizzbuzz\_game()

## 3) Step-by-step Explanation

### fizzbuzz(num) function:

* Input: an integer named `num`.
* Checks if `num` is divisible by both 3 and 5 (i.e., by 15). If so, returns the string 'fizzbuzz'.
* Else checks divisibility by 3 → returns 'fizz' if true.
* Else checks divisibility by 5 → returns 'buzz' if true.
* If none of the above, returns the numeric value as a string (so user input can match).

### fizzbuzz\_game() — overall goal:

* Run an interactive loop where each iteration shows the next number (starting at 1) and asks the user for the correct FizzBuzz response.
* Track a `score` counting consecutive correct answers.
* Allow the player to quit by entering 'q'.

### Variables in the improved version:

* `score` — integer counter for correct answers.
* `current` — the current number to evaluate (starts at 1 and increments by 1 on each correct answer).

### Loop behavior (improved):

* Each loop iteration calculates `correct\_answer = fizzbuzz(current)` — the expected response for the current number.
* Prompt the user with the accepted answers (fizz/buzz/fizzbuzz or the number) and read their input.
* If input is 'q', exit and show final score.
* If the input matches `correct\_answer`, increment `score`, print confirmation, and increment `current` for the next round.
* If the input does not match, print the correct answer and end the game, showing final score.

### Differences & bugs fixed from original code:

* The original used `prev` and `i` and calculated `total = prev + i`. That produces a nonstandard sequence (1, then 1+2=3, then 2+3=5, etc.) — effectively giving Fibonacci-like sums rather than consecutive integers. The improved version uses a single `current` counter that increments by 1 each round.
* Improved clarity: explicit docstring, clearer variable names, and a `if \_\_name\_\_ == '\_\_main\_\_':` guard for script execution.
* Input normalization: `.strip().lower()` so answers are compared case-insensitively and without extra whitespace.

### Example round (how it runs):

* Start: current = 1 → fizzbuzz(1) returns '1' (string). Prompt shows 'or 1'.
* If the user types '1' → correct → score becomes 1, current becomes 2.
* current = 2 → fizzbuzz(2) returns '2'. User types '2' → correct → score 2, current 3.
* current = 3 → fizzbuzz(3) returns 'fizz'. User must type 'fizz' to be correct.
* current = 5 → fizzbuzz(5) returns 'buzz'. User must type 'buzz'.
* current = 15 → fizzbuzz(15) returns 'fizzbuzz'. User must type 'fizzbuzz'.