Python Number Guessing Game Project

Before You Start:

Download PyCharm Community Edition. We will be utilizing this Python IDE to write, compile, and run your projects. Here is the link: https://www.jetbrains.com/pycharm/download/#section=mac

In this project, we will be creating a Python number guessing game. This project will put together many of the concepts we have learned so far with Codecademy. You will be generating random numbes and the user will guess that number until they get it correct.

This is what you have to do:

1. To generate random numbers, you will need to import the "random" library
2. Create a variable that stores a random number between 1-99
3. Create a variable that stores the user's guess of what the number could be
4. Use a while loop to check if the guessed number matches the randomly generated number
    a. Exit the while loop once the guessed number matches the randomly generated number
5. If the guessed number is less than the randomly generated number, let the user know the number is too low and they should guess higher. Have them try again.
6. If the guessed number is more than the randomly generated number, let the user know the number is too high and they should guess lower. Have them try again.
7. If the guessed number is out of the range of numbers to guess from, let the user know and have them guess again. Example: if the user guesses 100.
8. After the guessed number is equal to the randomly generated number, let the user know the guessed correctly. Remember to exit the loop prior to this.

Once you finish this part of the project, we will build on the game by adding a scoring aspect.

Scoring Aspect:

1. Create a list to track the number of attempts
2. Create a function named show_score.
3. In this function, if the attempts list is empty (the length of it is less than or equal to 0), let the user know that there is not current high score.
4. Otherwise, print the current high score.
    a. Hint: To win the game, the user should take a minimum number of guesses. So, the high score should be the minimum value in the attempts list.
    b. Hint: This is how string formatting works.
        i. print("The current high score is {}".format(VARIABLE NAME))
5. Now we will modify your existing code
    a. Create a string variable that asks the user if they want to play the game. Y/y for yes and N/n for no.

b. Create a function named play_game(). In this function, move in all of your existing code
c. Create an integer variable to keep track of number of attempts it takes the user to guess. Initialize this variable to 0.
d. Call the your function show_score()
e. Modify your while loop. Stay in the while loop if the answer to if they want to play the game is Y/y.
f. If the guessed number is less than the randomly generated number, increment the integer variable that tracks the number of attempts by 1.
g. If the guessed number is more than the randomly generated number, increment the integer variable that tracks the number of attempts by 1.
h. Add another elif statement to check if the guessed number is equal to the randomly generated number
    i. In this elif statement, let the user know they guessed the number correctly
    ii. Increment the integer variable that tracks the number of attempts by 1
    iii. Append the value of this integer variable to the attempts list
        1. Hint: use the list function .append()
    iv. Print the number of attempts it took the user to guess
        1. Hint: use string formatting again
    v. Create a string variable that asks the user if they want to play again. Y/y for yes and N/n for no
    vi. If the user input Y or y
        1. Call your function play_game
        2. Reset the interger variable that tracks the number of attempts to 0
        3. Call your function show_score
    vii. Otherwise,
        1. Print a statement letting the user know they are exiting the game
        2. Break out of your while loop
            a. Hint: “break”
i. Outside of this function, remember that you must call it!