Lab: Classes

Part 1 Player class
Create a `Player` class that we could be used with our number guess game.

The player class should have the following attributes:

- Name for the player
- How many guesses the player has made
- The history of their guesses

The Player class should also have a __repr__(self) method that can be used to convert the Player into a string for printing purposes. You might also consider giving it a __str__() method.

It should also have an __init__ that will be used to initialise the attributes of the object.

You could also give the class *docstring* that describes its purpose.

In terms of behaviour the class might have

- A method to add a guess to the history of guesses
- A method to print out all the guesses
- A method to increment the count guess

You should now be able to run the following code

```
player1 = Player('John')
print(player1)

player1.increment_count()
player1.add_guess(4)

player1.increment_count()
player1.add_guess(2)

print(player1)

print('History:')
player1.print_history()

player2 = Player('Denise')
players = [player1, player2]
print(players)
```

The output from this might look like:

```
Player John - history [] - guess_count 0
Player John - history [4, 2] - guess_count 2
```

```
History:
  guess: 4
  guess: 2
[Player(John), Player(Denise)]
```

(if time allows)
Part 2: Extension Point

Once you have created the class update the number_guess_game.

When the game starts prompt the user to provide their name. Using this information initialise an instance of the player class with the players name.

Next modify the game such that all guesses are recorded with the player rather than with the list you were previously using.

Also use the player `count` attribute to keep a record of the number of goes that the player has made.

Ensure that the game play for the user has not changed.

Part 3: Extension Point Game Class

Create a Game class to hold all game related data.

Also change all relevant functions into methods – note not all functions may be come methods.
Also remember that a method has a special first parameter 'self'.