

Exercise: Marathon

Table of Contents

Background	1
Instructions	1
Note	2

Background

For this exercise we are going to create a simple game. This game will have players that will race against each other. However, in order to make this game more interesting, we will create two kinds of players which will both move differently based on the functionality of its class.

Instructions

- Write a script called `players.py`. This script should contain the following:
 - Write a class called **Player**
 - Write an `__init__()` method. This method should have two parameters, **self** and **name**. This method should do the following:
 - Create **name** and a position attribute. Position will be represented by a number. The position represents how many miles they are along a trail
 - Write a **RedPlayer** class and a **BluePlayer** class. Both of these classes should be subclasses of **Player**. These classes should do the following:
 - Write a **walk** method that will change the position attribute by a certain amount given the subclass.
 - When a **RedPlayer** walks, they should move forward between 1 and 10 steps. A **BluePlayer** should move forward between 4 and 8 steps. The actual number of steps should be randomly generated within the given range. (hint. You want to use the random library which contains a `randrange` function. This function generates random numbers based on a the range that you define)
 - Write a **play_game** function. This function should not have any parameters but it should do the following:
 - It should create a list that will contain 3 instances of **BluePlayer** and 3 instances of **RedPlayer**. This function should not use inputs to assign names, instead, it should dynamically create names while it creates instances.
 - Ex. `BluePlayer1`, `BluePlayer2`, etc.
 - Hint: Use a while loop to handle this kind of problem.
 - Iterate over the list, for each instance in the list invoke the **walk** method. This will cause the player to walk a certain amount of steps. Count how many times you iterate over the

whole list.

- Check the **position** of the instance and if the **position** is over 100 on any player return the **name** of the instance and the amount of times that you iterated over the list as a tuple.
- Write docstrings for each method. (`__init__()` doesn't need a docstring.)
- Make an `if __name__ == "__main__":` statement in which you call the `play_game()` function.
 - Unpack the returned value into two variables and print them both for the user to see the winner and how many turns it took for the game to finish.
- Write a script called `marathon.py`. This script should contain the following:
 - `import players.py`
 - Re-write the `play_game` function. The difference here is that instead of checking if the position of any instance is over 100, we will check if the position of any instance is over 1000.
 - Make an `if __name__ == "__main__":` statement in which:
 - You call the `play_game()` function that is in this script and print the result.
 - You call the `play_game()` function that is in the namespace of the module that you imported (`players.py`) and print the result.

Note

Run the marathon script multiple times. Observe who the winners are when you run the simulation multiple times under the different conditions (`pos >= 100` vs `pos >= 1000`).