JavaScript Team Sports Instructions

Create, retrieve, and add information about your favourite sports team. Rugby, football, tennis, or cricket, you pick it. It’s your job to create data using the JavaScript data structures at your disposal: arrays and objects.

After you create these data structures in this project, feel free to challenge yourself to gain insights from them. For example, you might want to get the total number of games your team has played, or the average score of all of their games.

1	**Populating Data** Let’s make a data structure to store the information about our team. Declare a `const` variable called `team` and set it to an empty object.
Hint	Remember that you can initialize objects using curly braces { }.
2	Our team has players and the games that they have played. Let’s represent both of these by adding two properties to your `team` object. Add a `_players` property and a `_games` property and initialize both to empty arrays.
Hint	Inside the curly braces of the `team` object, add `key: value` pairs for `_players` and `_games` properties. Remember to separate each `key: value` pair with a comma. You can create empty arrays using square brackets [ ].
3	Next, populate the empty`_players` array with three players. Each player should be an object containing three properties: `firstName`, `lastName`, and `age`. Put each player on a new line to prevent the line from getting too long.
Hint	If we want to add a player named Steve Bull, age 52, your object will look like the following: `{firstName: 'Steve', lastName: 'Bull', age: 52}` Remember to separate each object with a comma.
4	Let’s do the same for our `_games` array. Populate the empty array with three games. Each game should be an object containing three properties: `opponent`, `teamPoints`, `opponentPoints`.
Hint	Make sure to add three new objects inside the `_games` array. Separate each object with a comma.
5	**Getting Data** Create a getter method called `players` to retrieve the `_players` property. Inside the getter method, `return` the `_players` property. For the scope of this project, we won’t need to create setter methods, because we don’t want anyone to change the data saved to the properties.
Hint	Remember that we can access the calling object’s internal properties using the `this` keyword. Make sure that you add a comma after the closing square bracket (]) of the `_games` array. Without this comma, our program will throw a syntax error!
6	Create another getter method called `games` to retrieve the `_games` property. Inside the getter method, `return` the `_games` property.
Hint	Make sure to add a comma after the closing curly bracket (}) of the `players()` getter method. Without this comma, our program will throw a syntax error!

7	**Adding Data** We want to add a new player to our team. Add a .addPlayer() method to the team object. This method should take in three parameters: newFirstName, newLastName, and newAge. Inside the method, create a player object by setting the three parameters to be the values for the object’s three properties: firstName, lastName, age. Finally, add the player object to the team‘s _players array.
Hint	You can add objects to an array using the .push() method. Make sure to add a comma after the closing curly bracket (}) of the games() getter method. Without this comma, our program will throw a syntax error!
8	Below the team object, let’s try out our new .addPlayer() method to add a new player: Bugs Bunny, age 76. Log the team‘s _players property to check that our new player was added.
Hint	You can use the .players() getter method to console.log() the _players property like below: `console.log(team.players);`
9	The scorekeeper has some new information for us! Create a method for adding game results called addGame that takes three parameters: newOpponent, newTeamPoints, newOpponentPoints. Inside the .addGame() method, create a game object by setting the three parameters to be the values for the object’s three properties: opponent, teamPoints, opponentPoints. Add the game object to the team‘s _games array.
Hint	Make sure to add a comma after the closing curly bracket (}) of the .addPlayer() method. Without this comma, our program will throw a syntax error!
10	Finally, below our team object, use the .addGame() method to add a record of a new game. Our team played against the 'Titans' where we scored 3 points and the opponent scored 2 points. Log the team‘s _games property to check that our new game record was properly added.
Hint	Make sure to call .addGame() outside the team object definition. Remember that you can use the .games() getter method to console.log() the _games property like below: `console.log(team.games);`