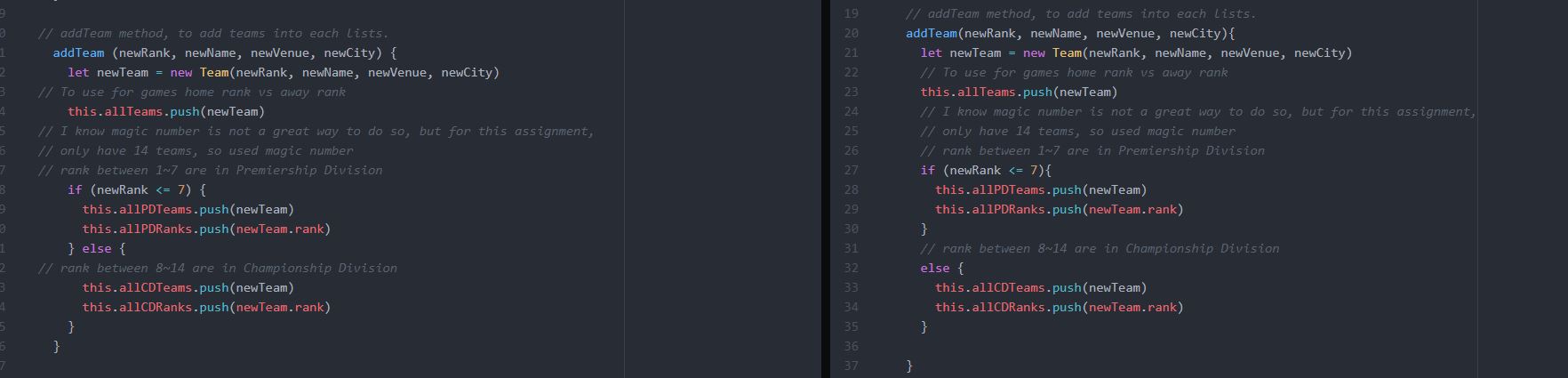
1. Left side is fixed version via JS standard, right side is my original code

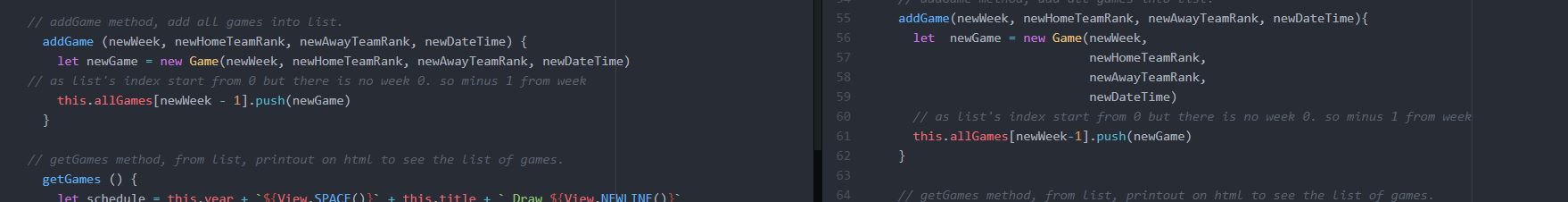
Only errors I am getting from standard site is about indenting comment.

As you can see from image below, only change I had to was location of ‘else’, standards was asking me to put else in the same line with closing curly brace.

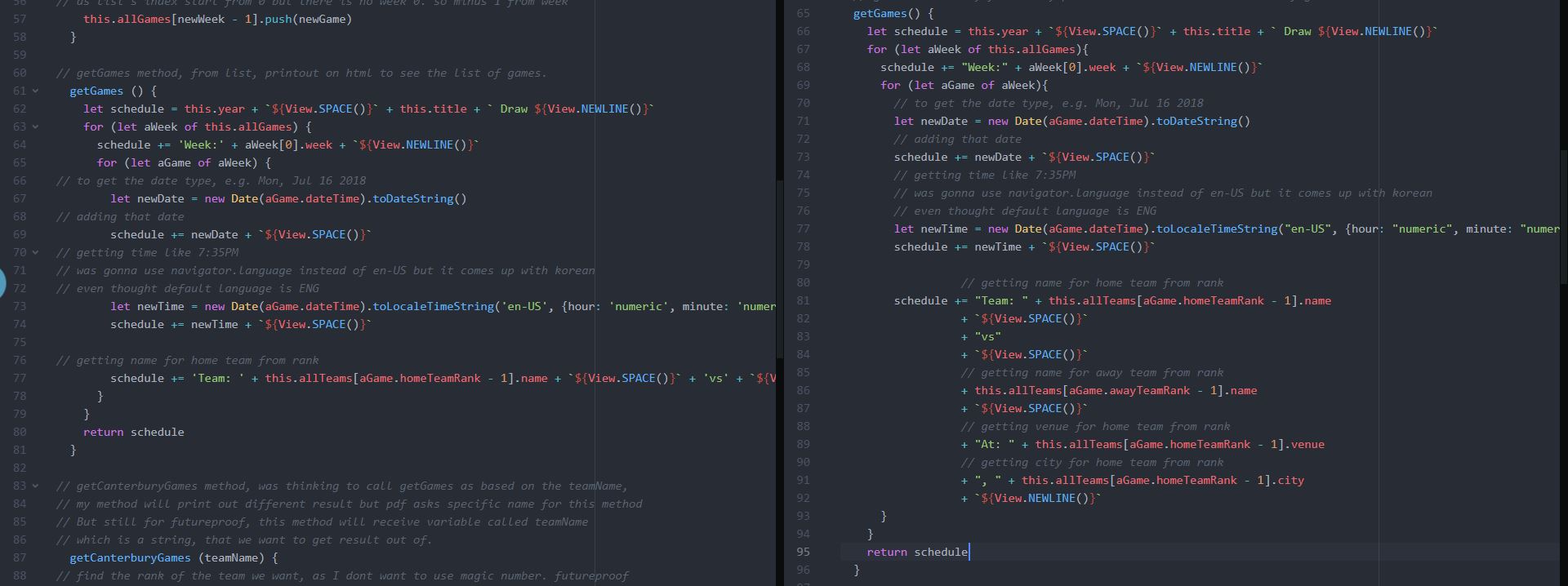


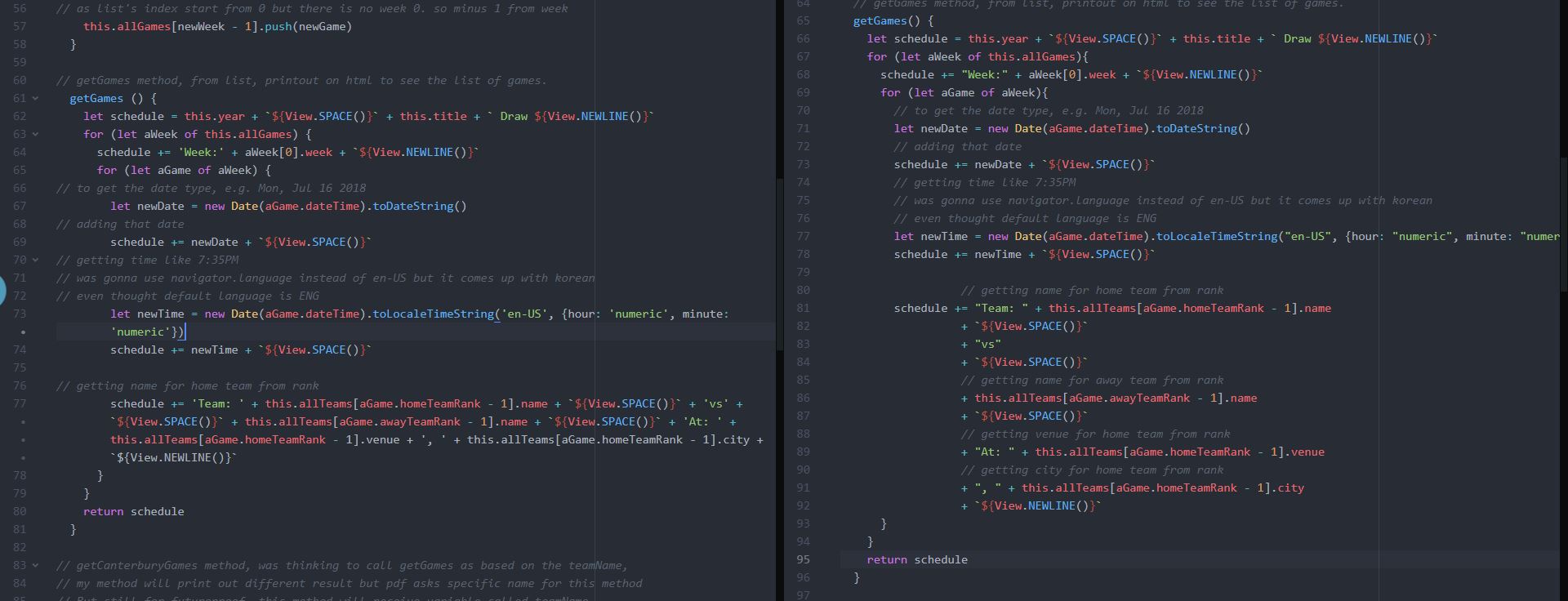
For next six pictures, they all have same issue, so I will just explain once here. Fortunately, our variable names aren’t that long, so I could put them in one line, but I chose to split them (like on right side to make it look better with my personal opinion). However, standards didn’t like it, so I put it back to one line of code. Also, instead of using double quote (“”) asking me to use single quote (‘’)

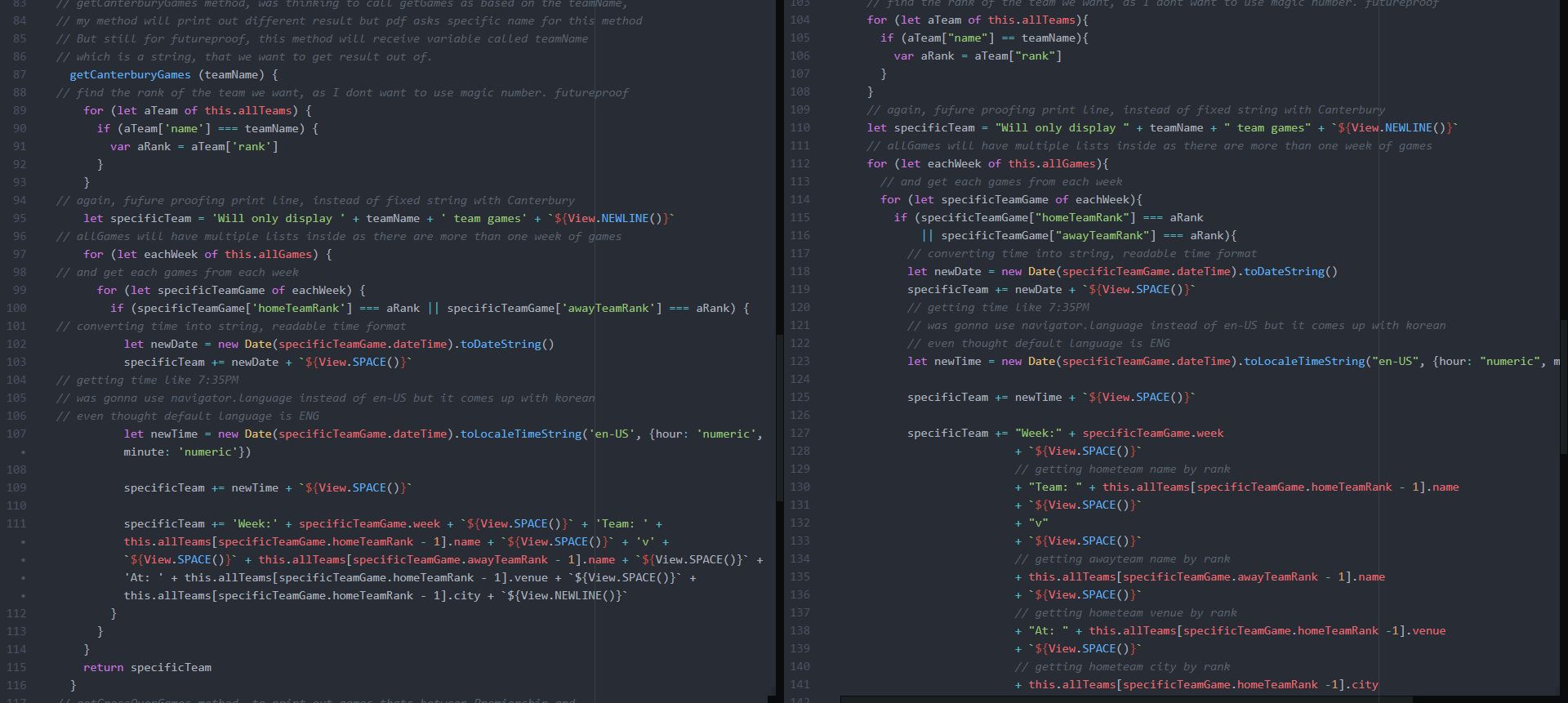
(I couldn’t find the command to add to ignore this on JS Standards)

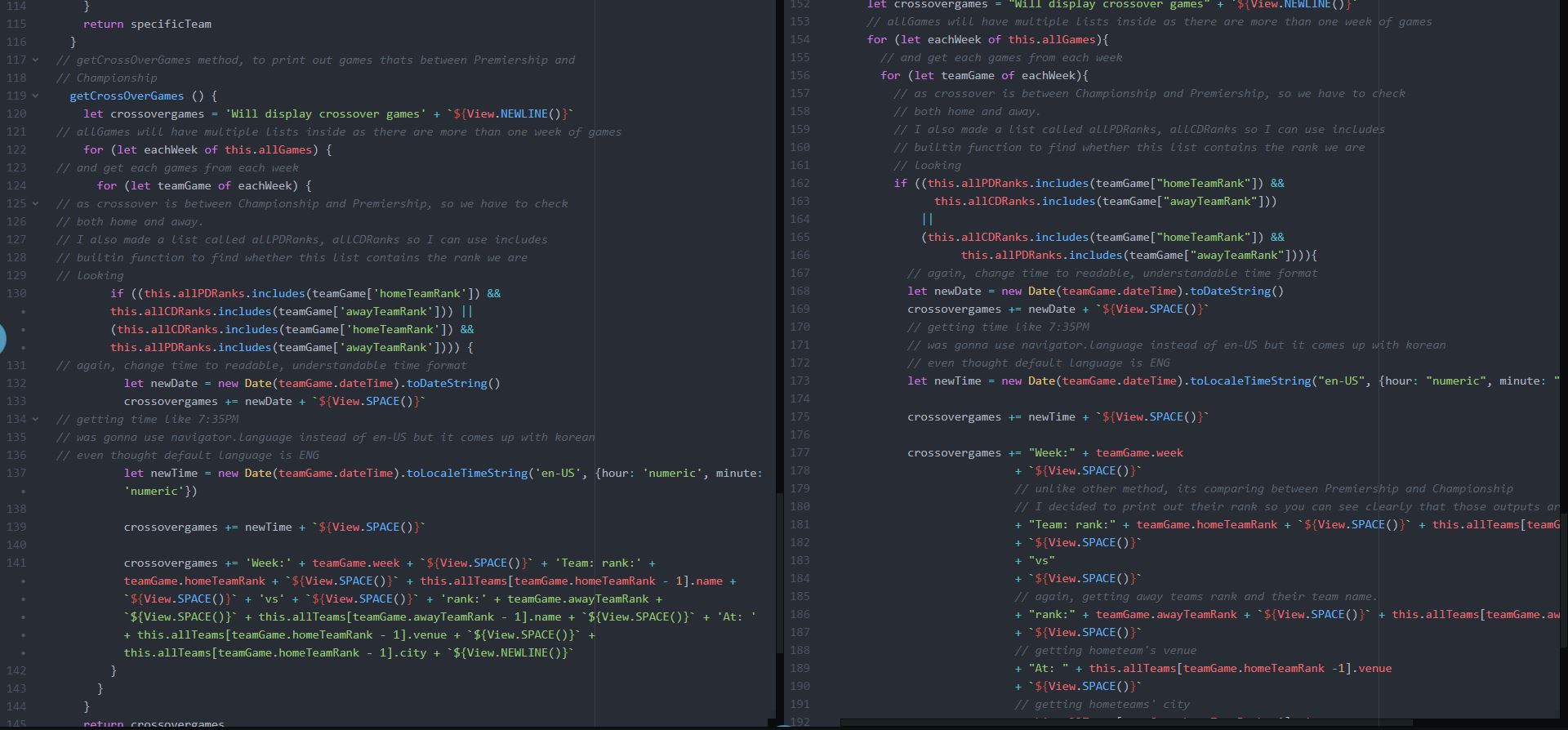


Those two photos are for getGames() method, but top one is one without linewrap ( Soft Wrap from Atom) and bottom one is the one with linewrap. As you can see bottom looks way better than top one but I still prefer the way I wrote as we can see line by line for this case.



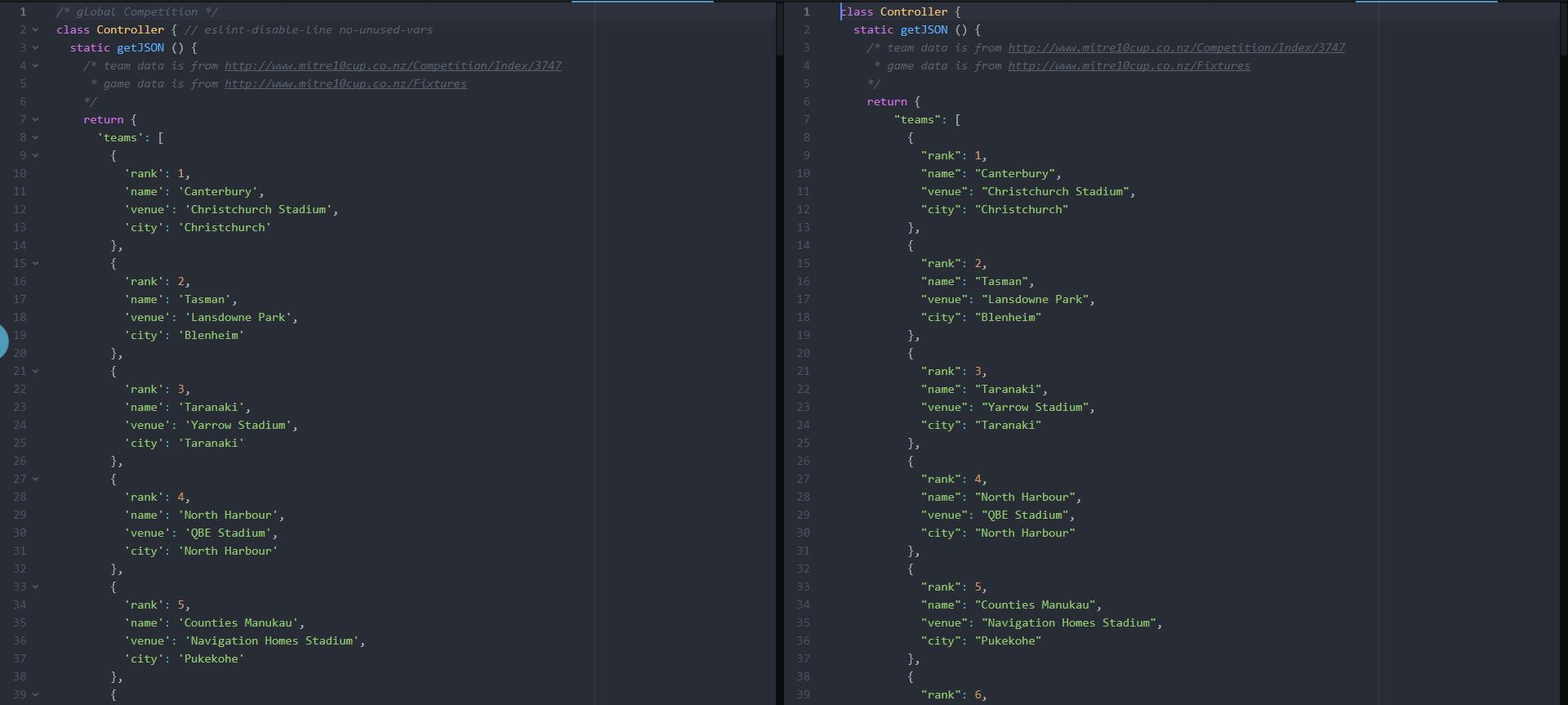






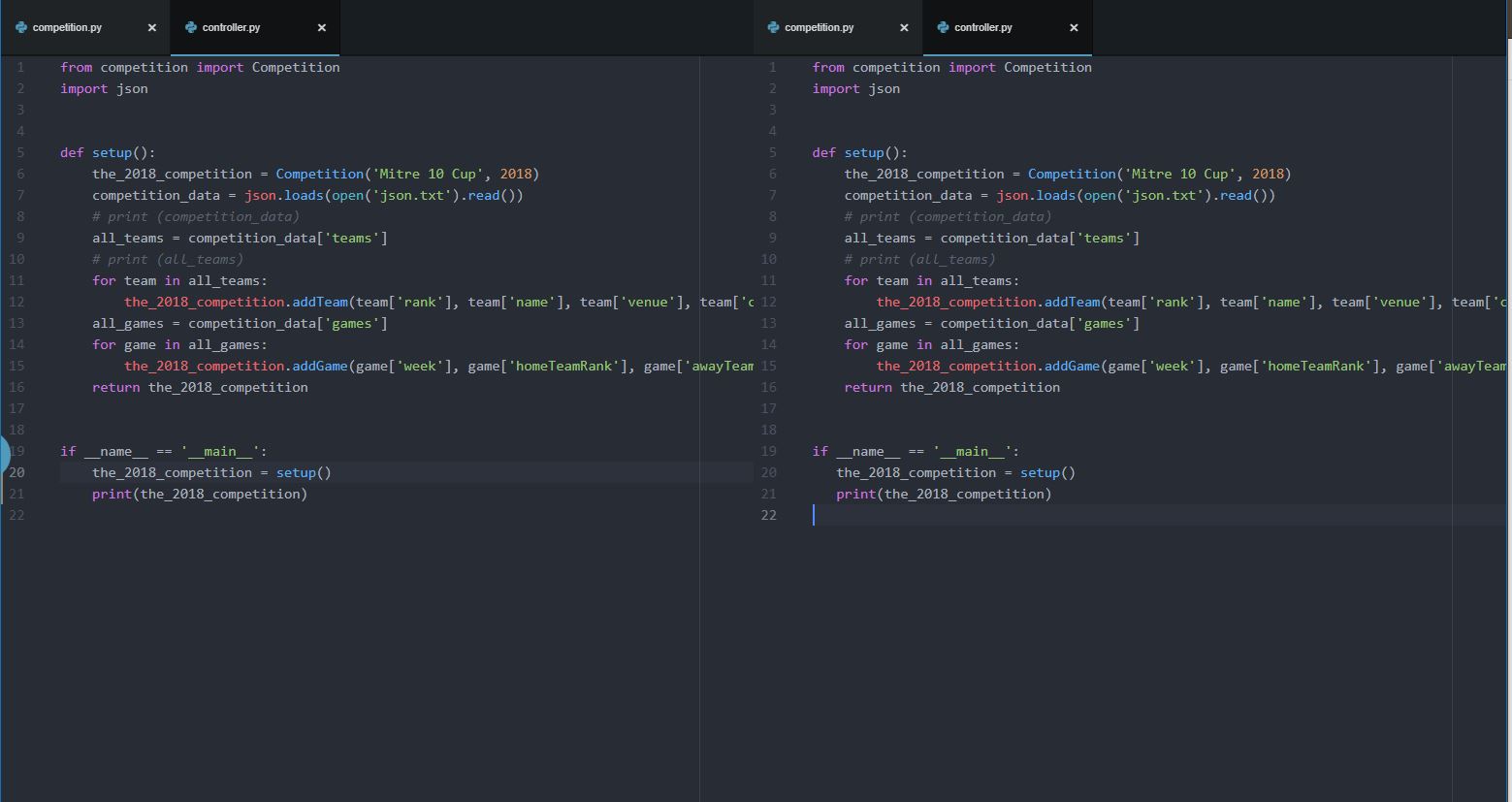


Standards like to have single quote not double quotes.

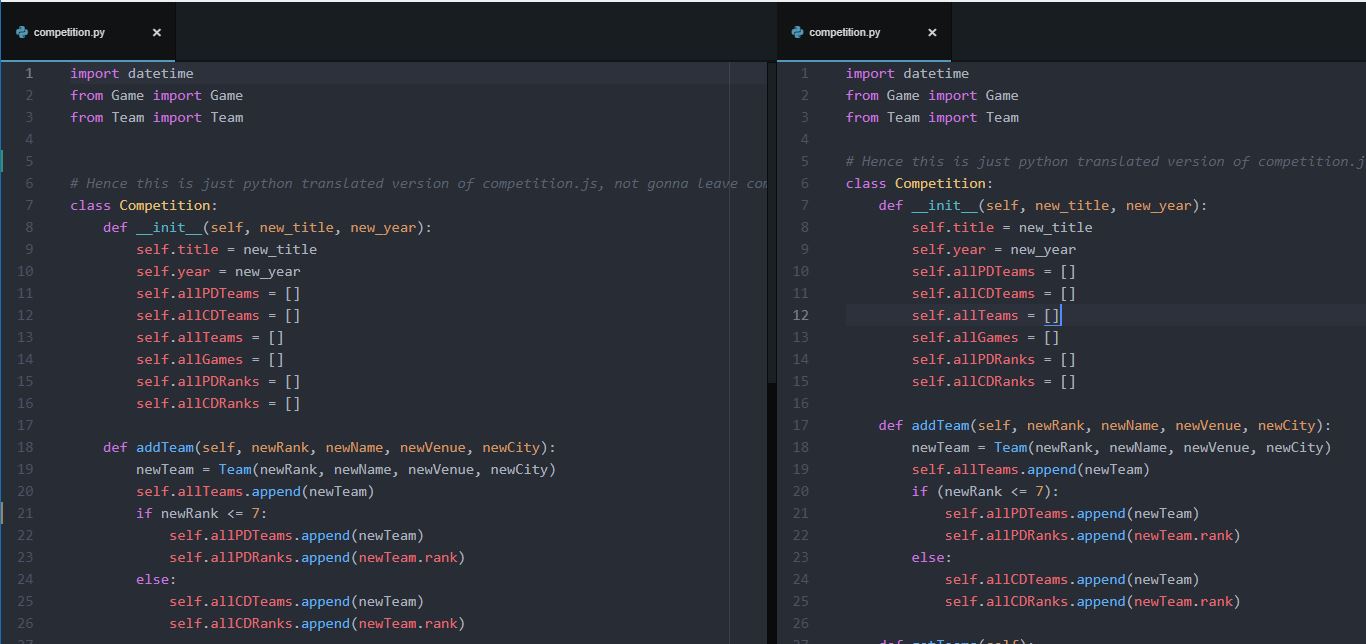


2. Left side is fixed version via PyCharm, right side is my original code

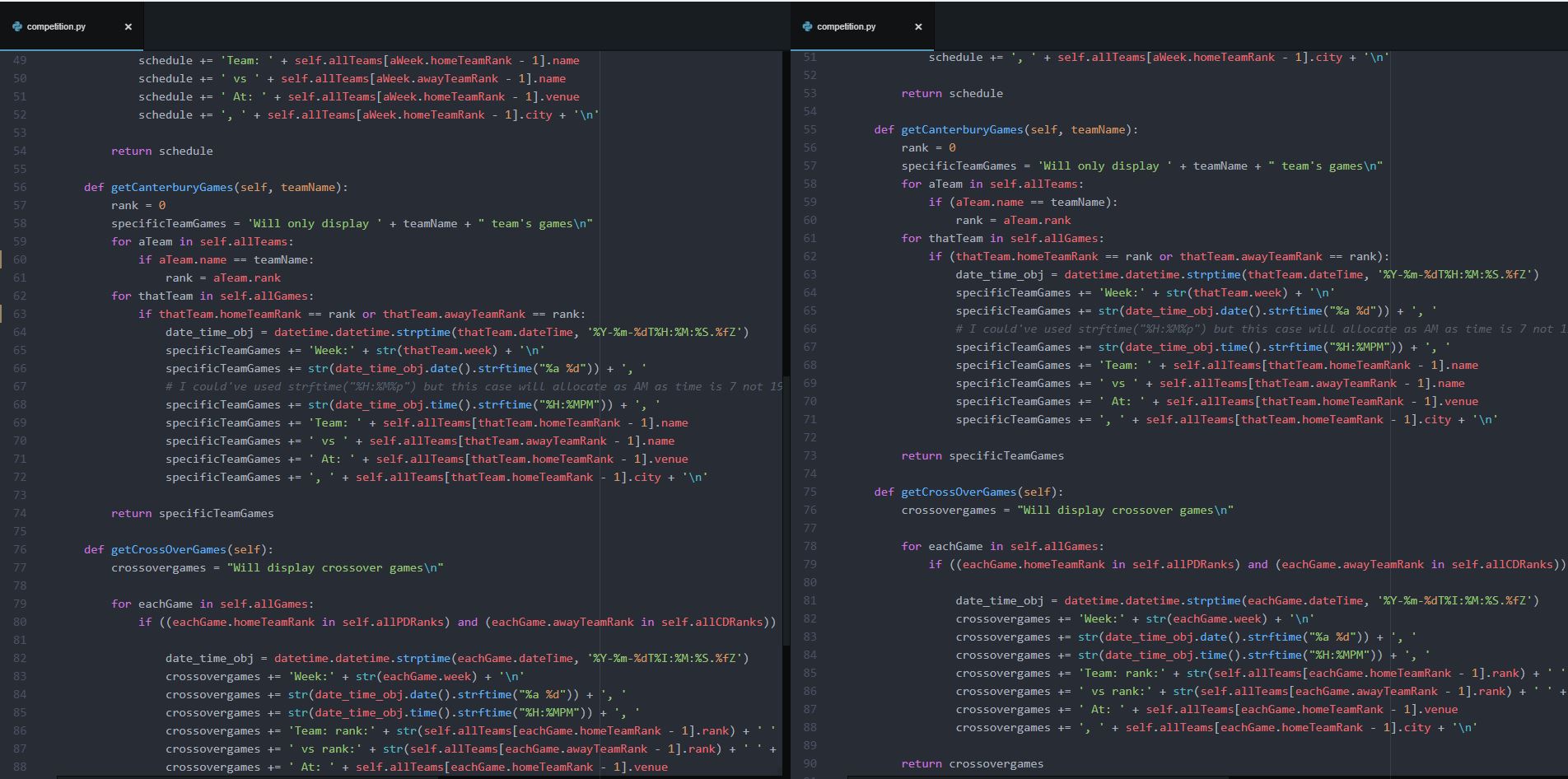
Number of tabs on line 20, 21 were 3 with Atom and fixed to 4 as PyCharm was expecting number of tabs = x \* 4.



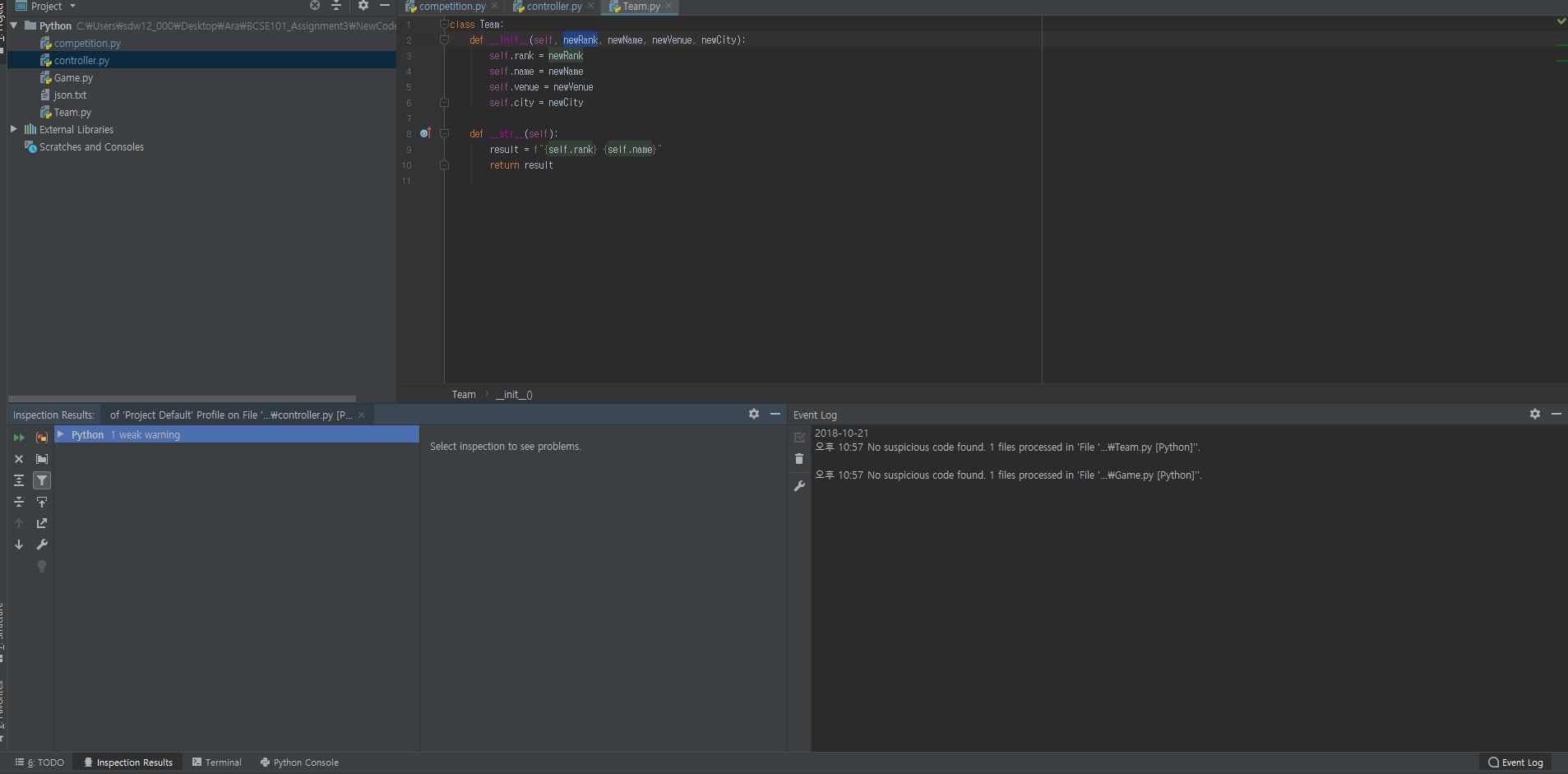
Line number 21, PyCharm didn’t want me to have () around newRank <= 7 which kind of make sense as we are only trying to compare one variable. Also, asked me to have two spaces before class Competition whereas I had one space before.



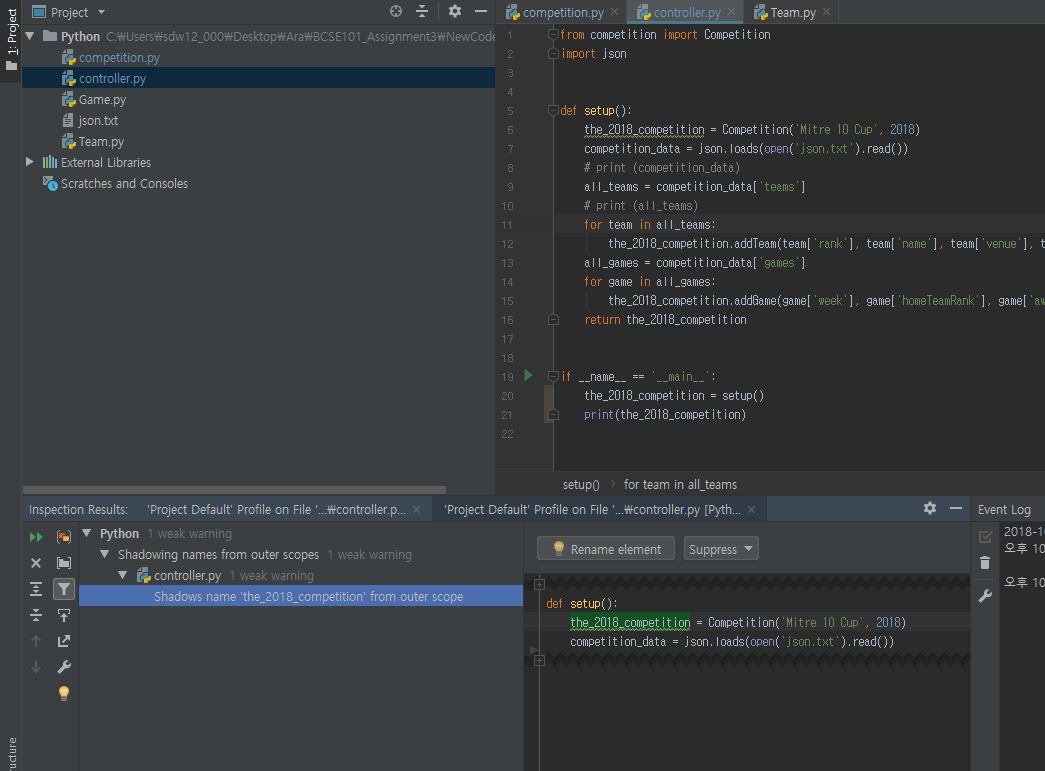
Same as previous one, line number 60, 63, didn’t want me to have () around compare statement.



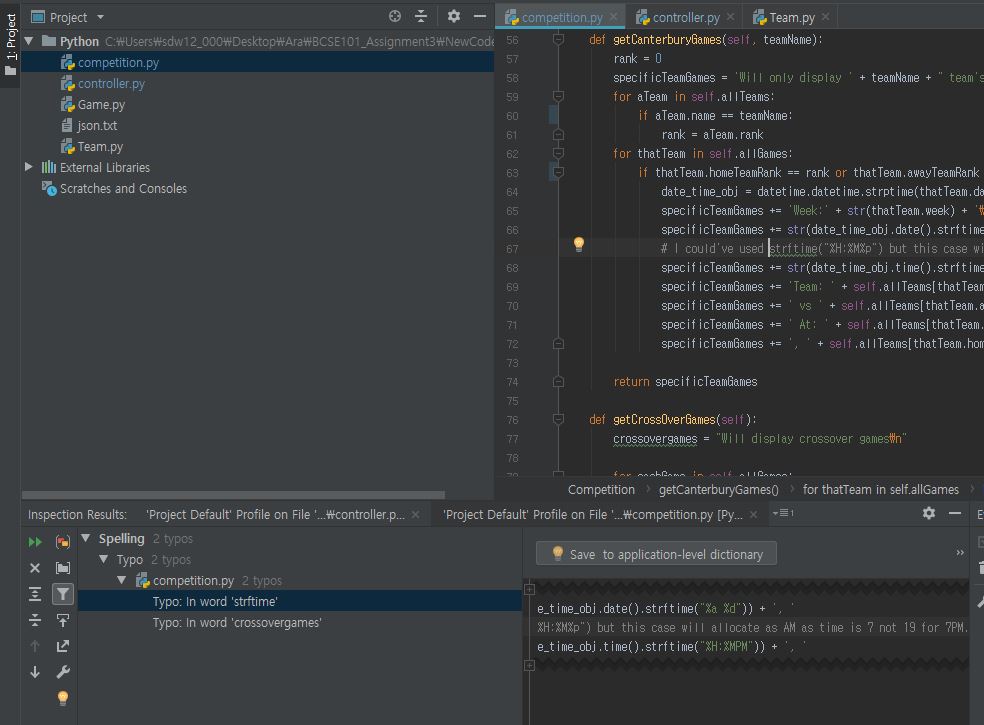
Fortunately no issues were found for Team.py and Game.Py



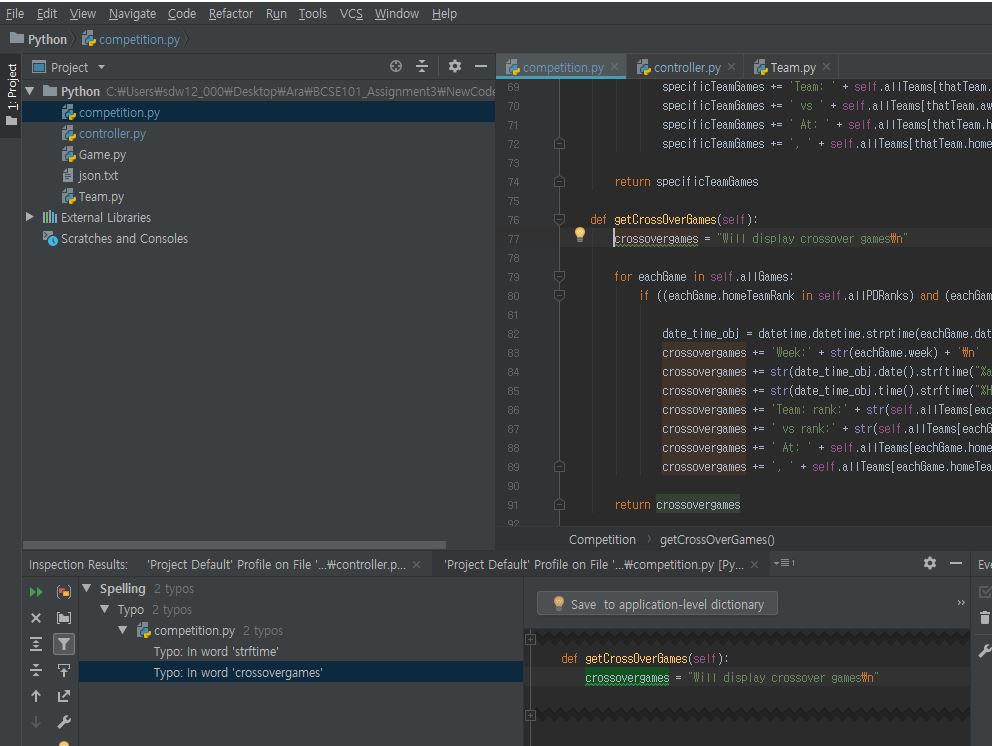
The\_2018\_competition, as I am trying to use variable name function inside the function, but this function is also a global variable. This warning is there to help me prevent bugs that might be caused by me trying to access the global variable but getting the local variable instead.



Issues for Competition are typo, but strftime is actually a built-in function with python so it’s not a typo, also it’s a comment.

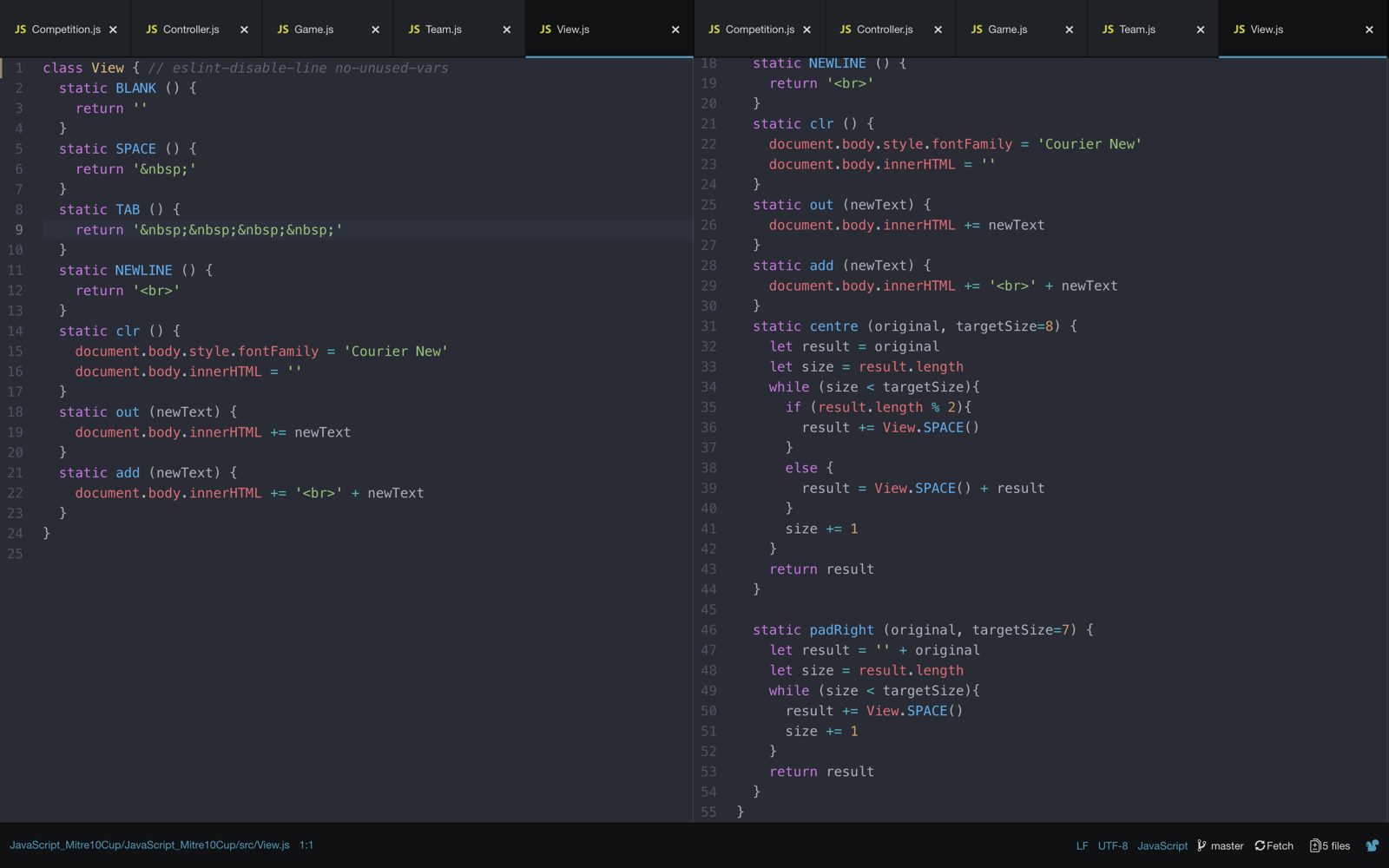


Crossovergames is just a variable name I made, so don’t really need to worry about as typo.

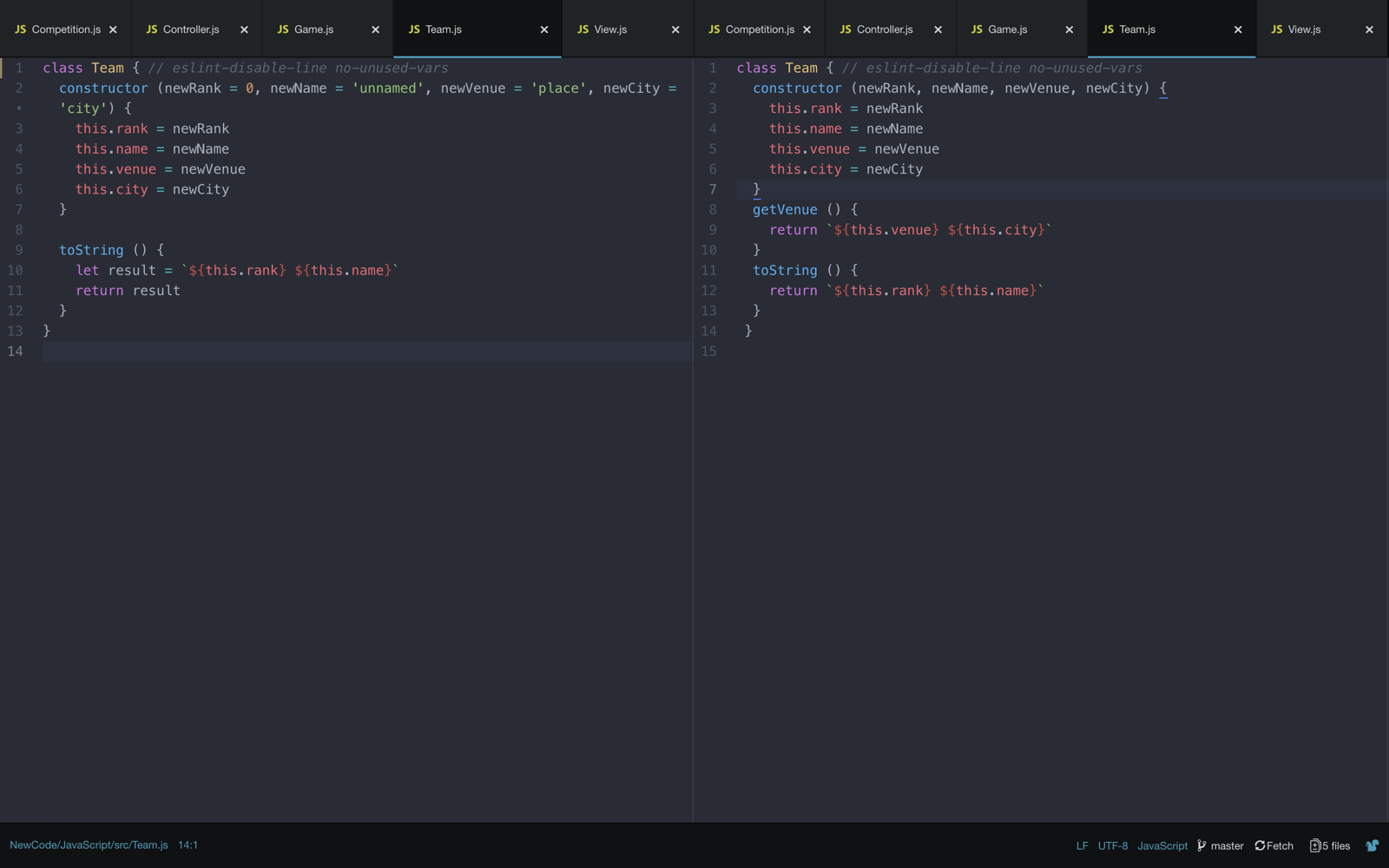


3. Left side is fixed version via JS Standard, right side is model answer

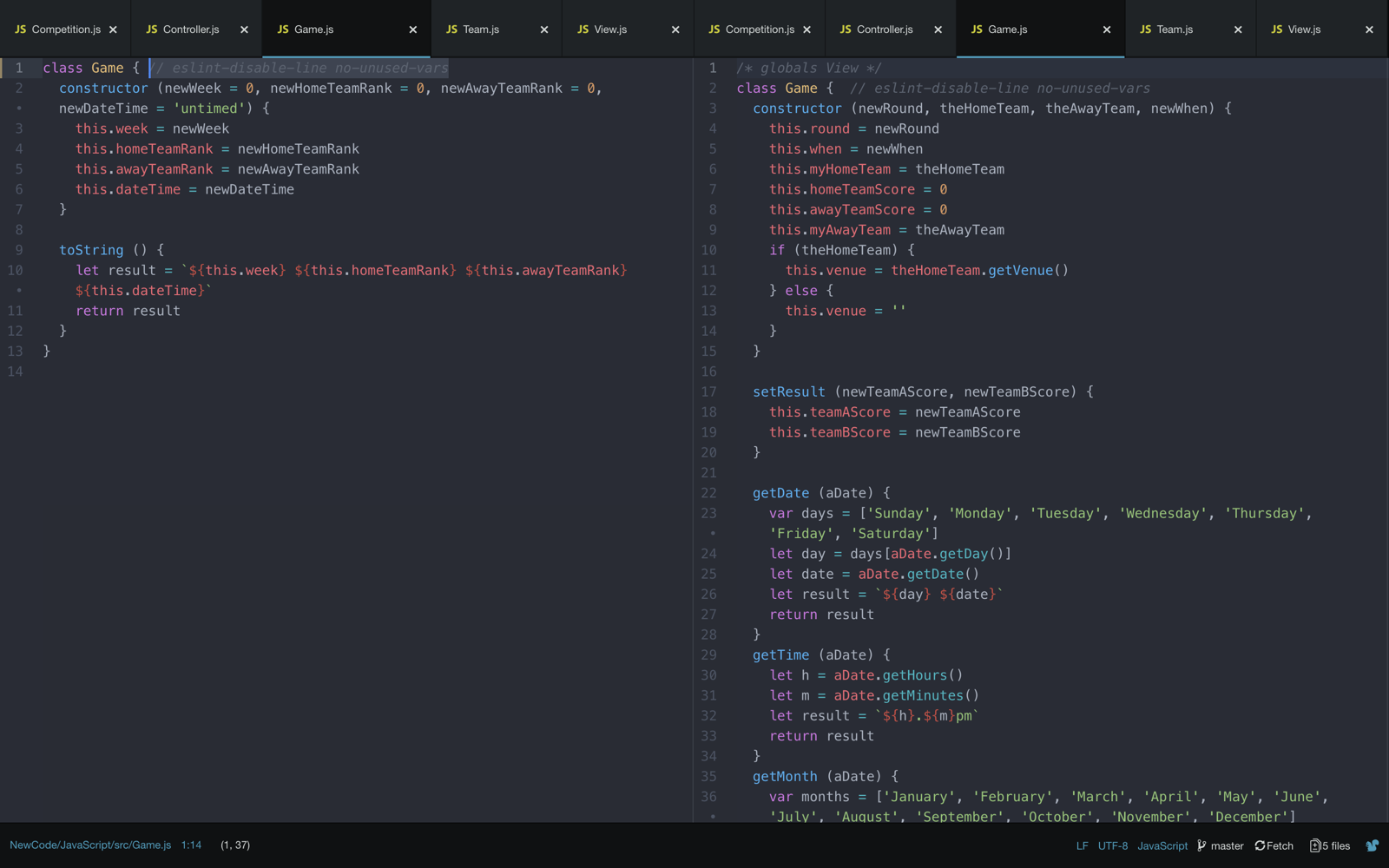
Mike put more methods than I used and based on method called ‘cetnre’ and ‘padRight’, I assume those will be used to make printout looks better.

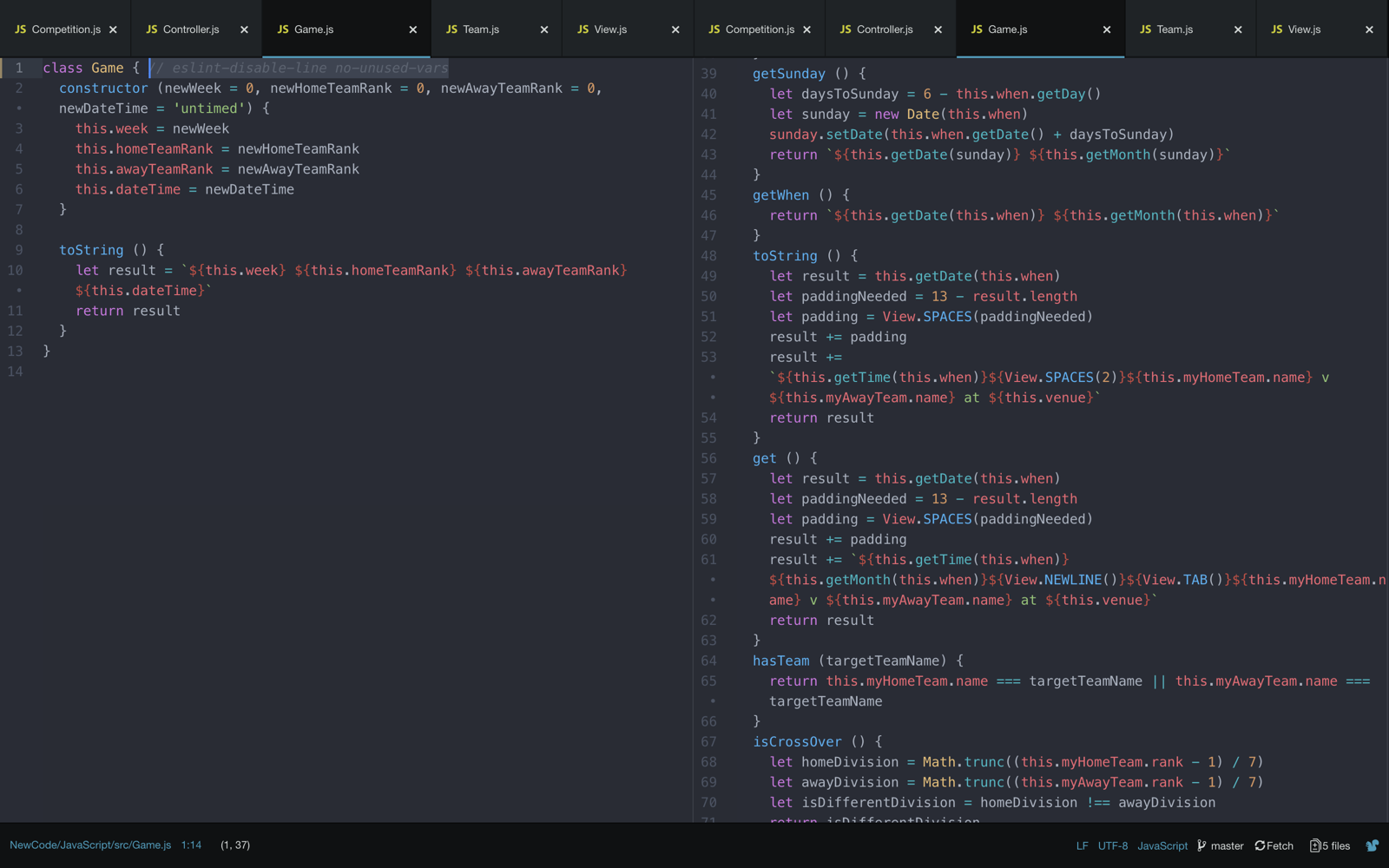


Instead of setting default variable like I did, model answer didn’t really set one. Also, I printed Venue via using this.venue where as Model answers has extra method to print just a venue.



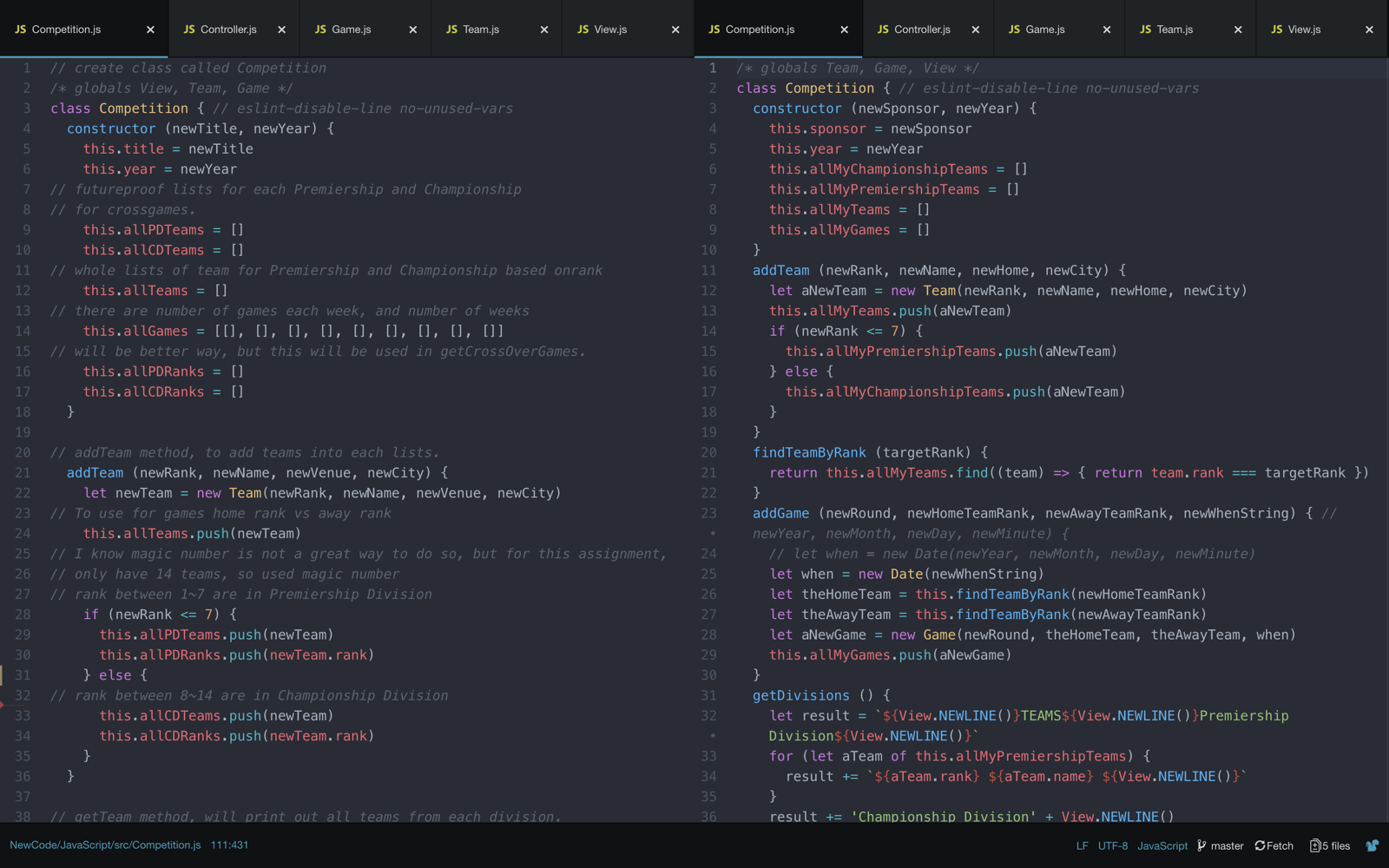
My Game class is pretty much designed for just to store data and print out, whereas model answer does more than just storing data. When I had to do all comparisons in getgames, getTeams, those jobs are done in each class, reason that Model answer for Competition has a lot less lines of code than mine as I didn’t make them as function and I think it is better to have a function to reduce duplication.



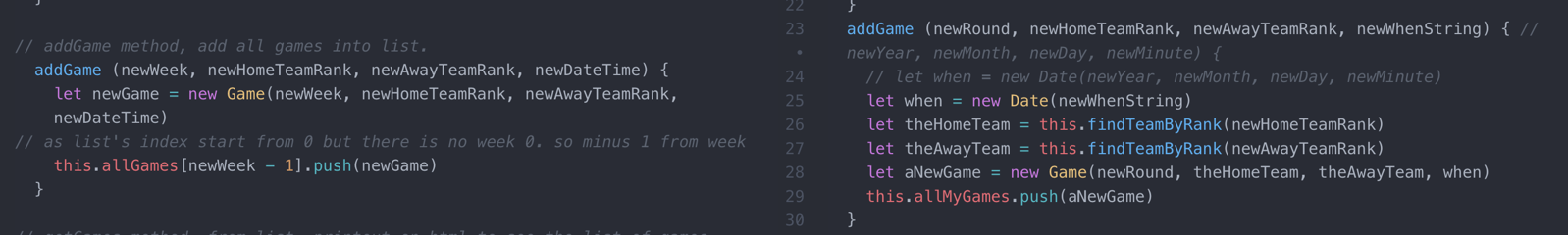


Here is the competition part, I didn’t compare Controller.js as it’s a given file, we didn’t have to fix anything.

The way I construct Competition and addTeam looks identical but model answer does have extra method called findTeamByRank, by naming convention, I assume this will be useful to find out the name with rank as we need to print out the team name by rank, I will explain how I did later in this report.



The way I made addGame was just putting the rank instead of name, whereas model number is putting actual teams name by knowing the rank.



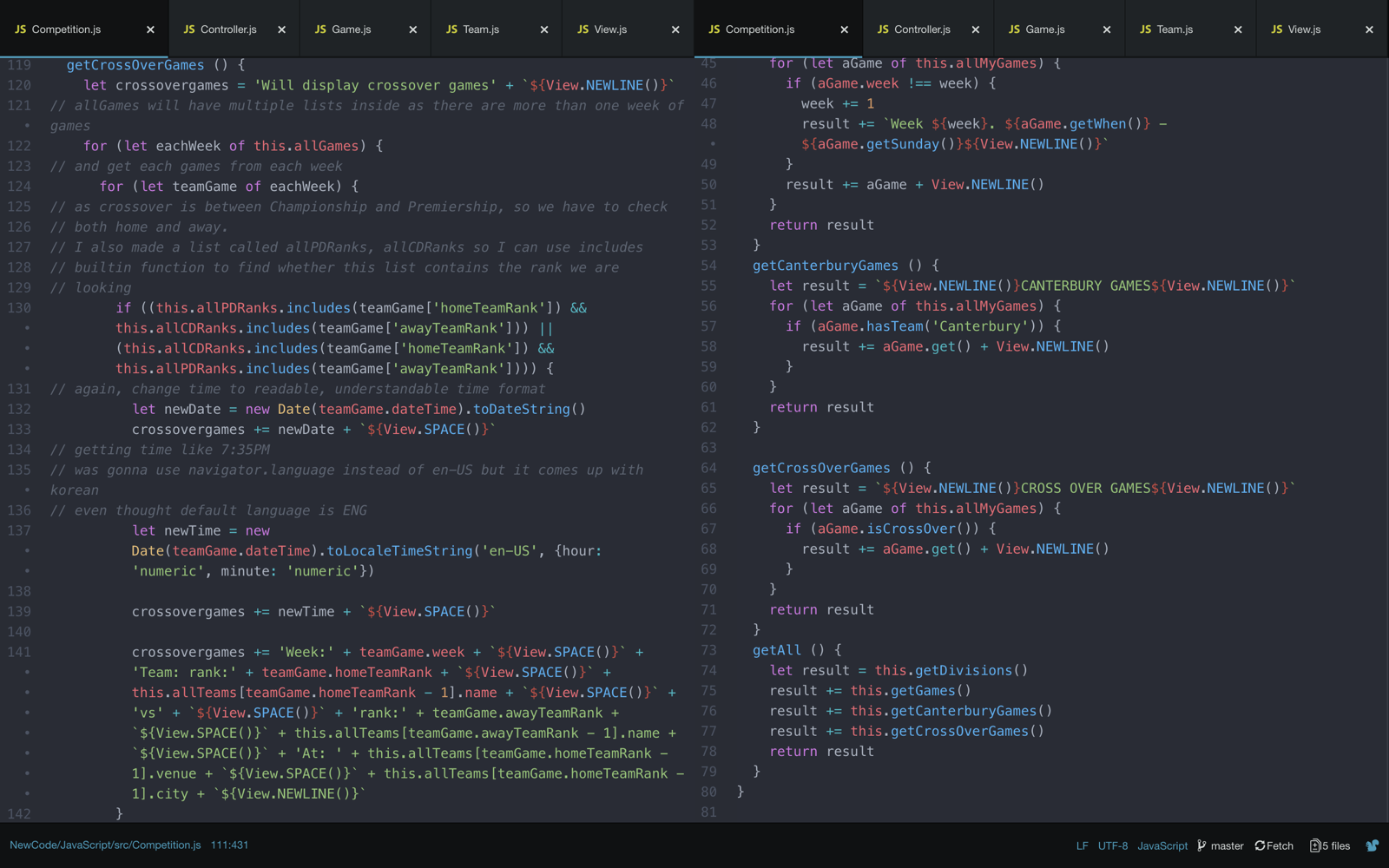
Idea looks okay but Model Answer has mis-located week += 1, so it increases every time the day changes as well.



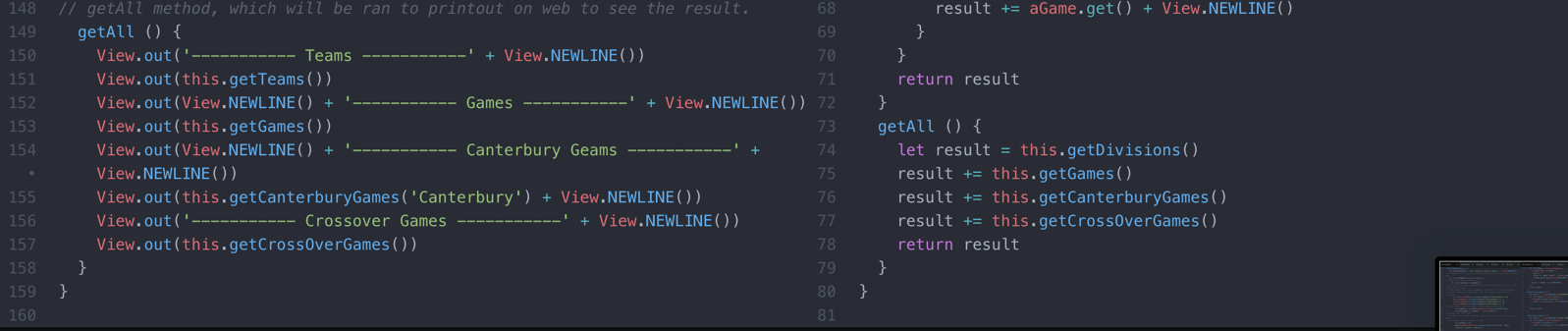
getCanterburyGames(), quite impressed how this get method replace whole my code. My code was slightly longer as I made it with future proofing, such as method name can be changed to getSpecificteamGames(teamName), then it will printout the games with only that given team. Not just a Canterbury like our assignment.



Same as above, model answer shows me the power of get() & isCrossOver() (seems checking they are between divisions).



I used View to printout like cookie jar when model answer just define variable and call them.

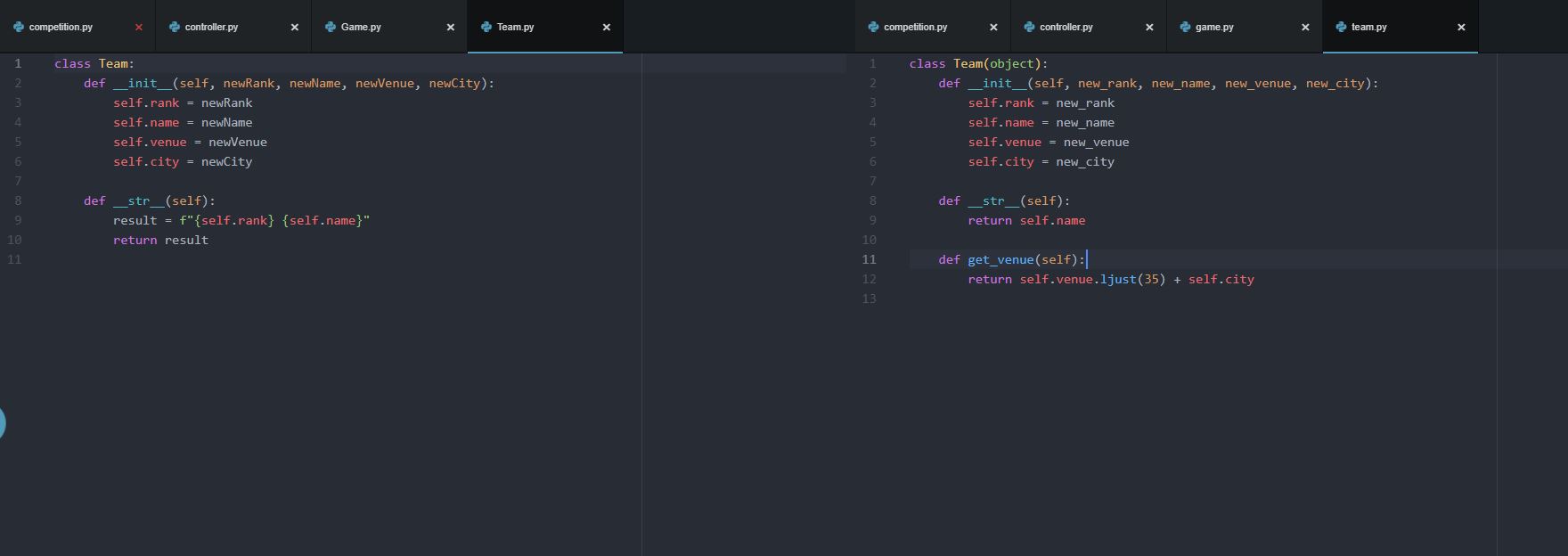


Overall, yes model answer is definitely a better code. By removing duplication like I had whenever try to print out games.

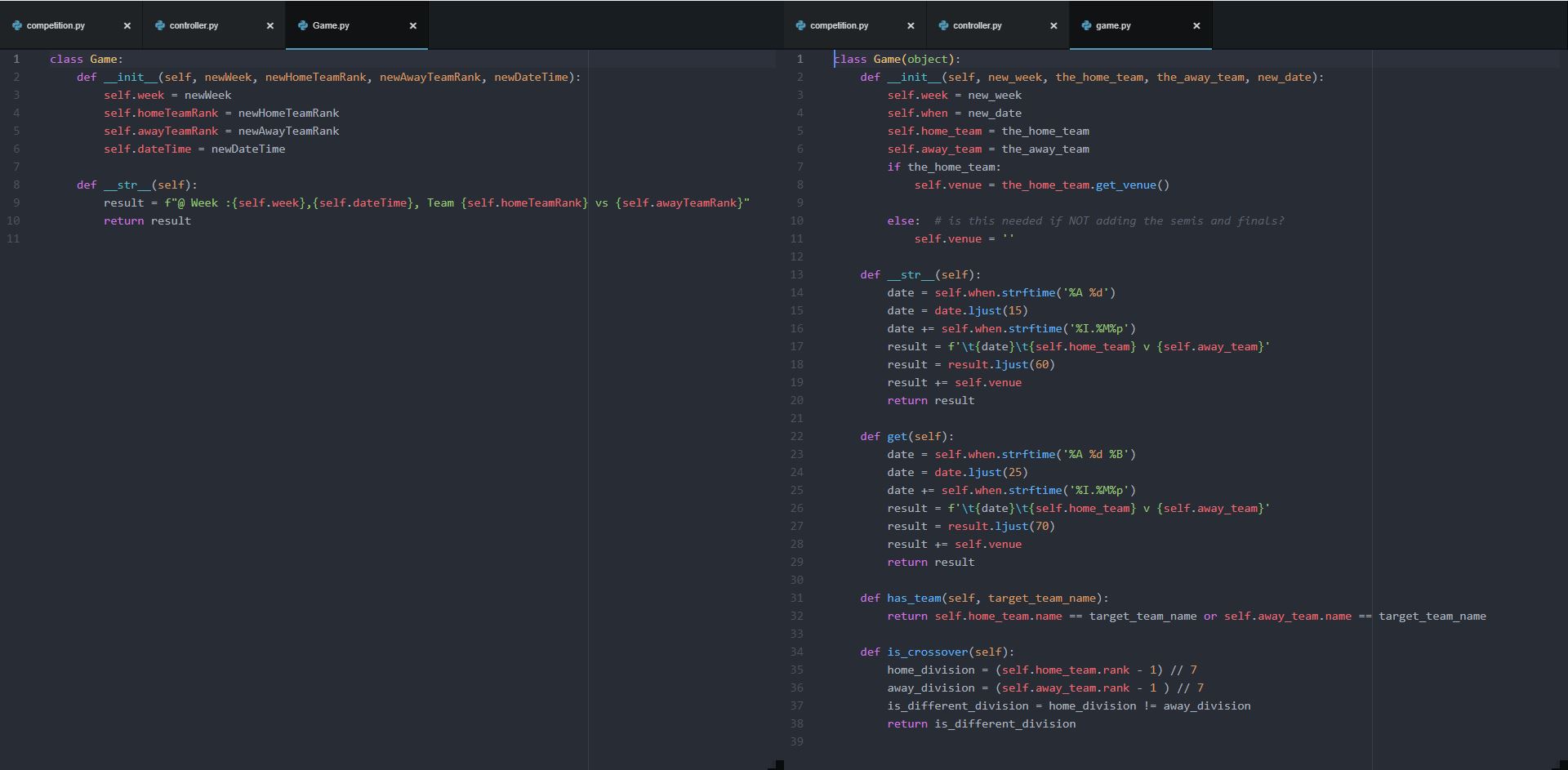
Not only that, model answer is trying to keep each method small as possible whereas some of my methods are bit longer than usual.

4. Left side is fixed version via PyCharm, right side is model answer

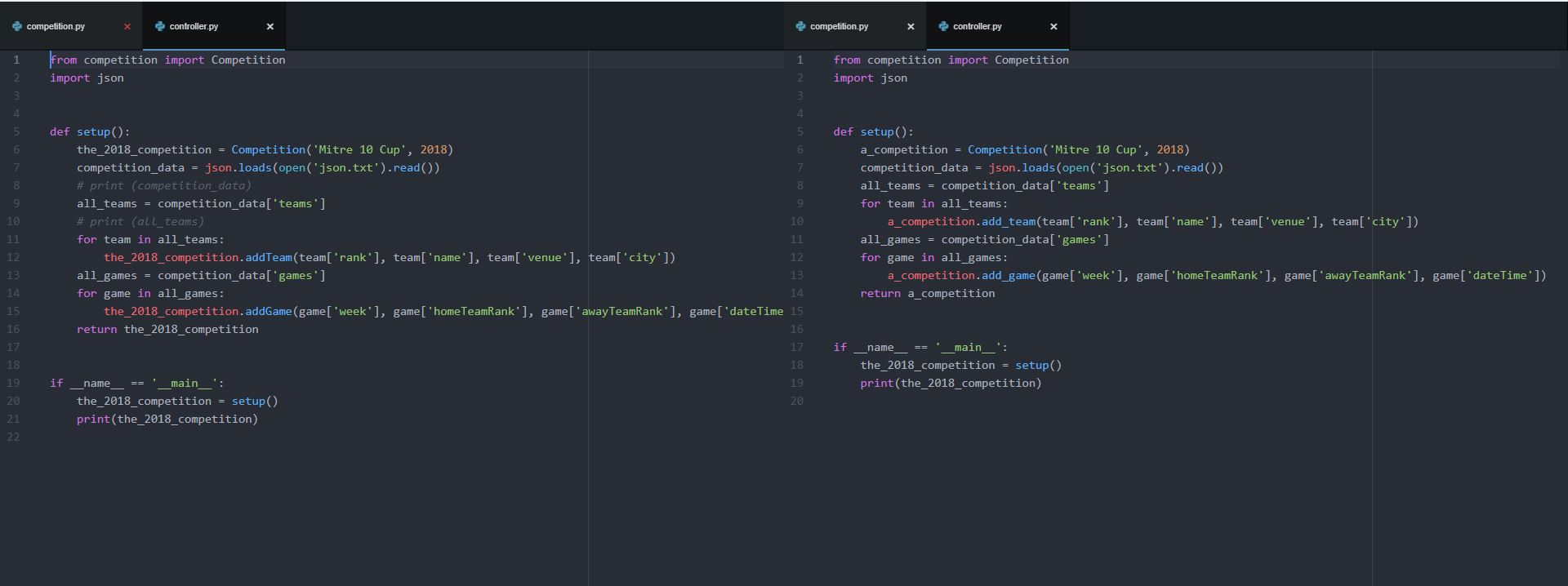
Just like JS, model answer has function for getting venue also, naming convention is different.



Same as JS case, Game class contains more functions to reduce work on competition side, such as get(), has\_team(), is\_crossover()…

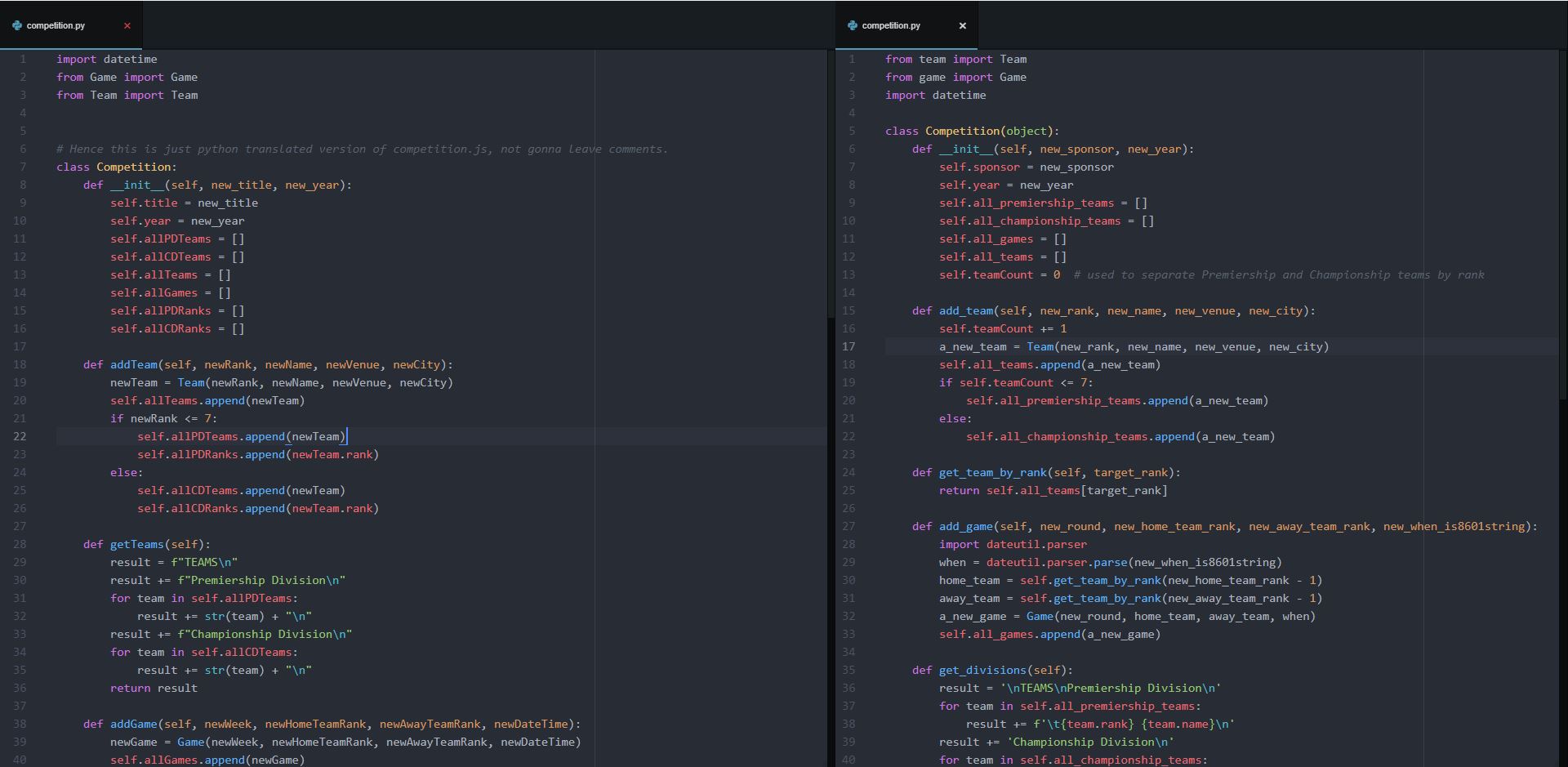


Controller is a given file, so no changes.



addTeam() function is pretty much same, except I am making two different lists, one for team name, one for team’s rank.

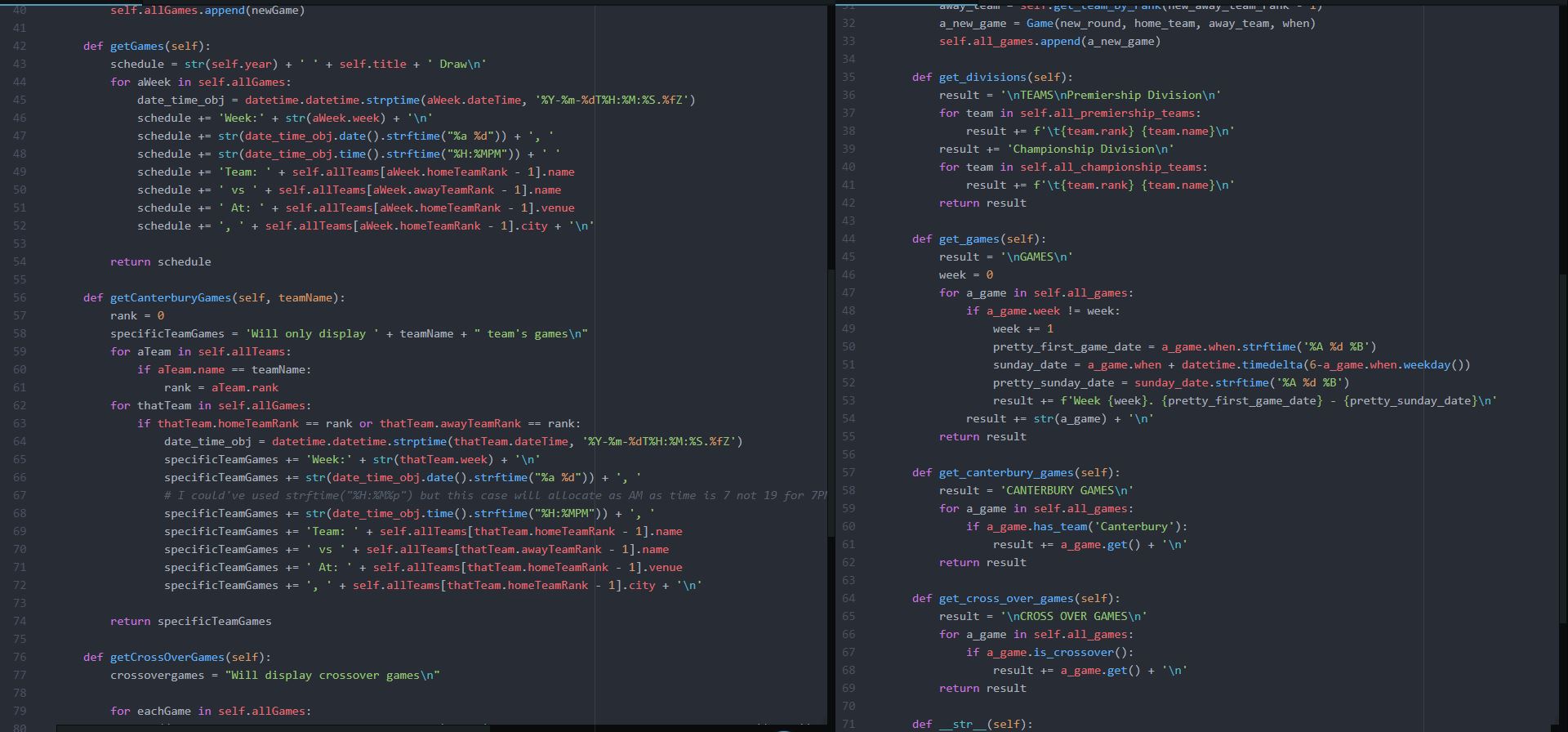
getTeams() is equivalent to get\_divisions() in model answer, and pretty much they are identical.



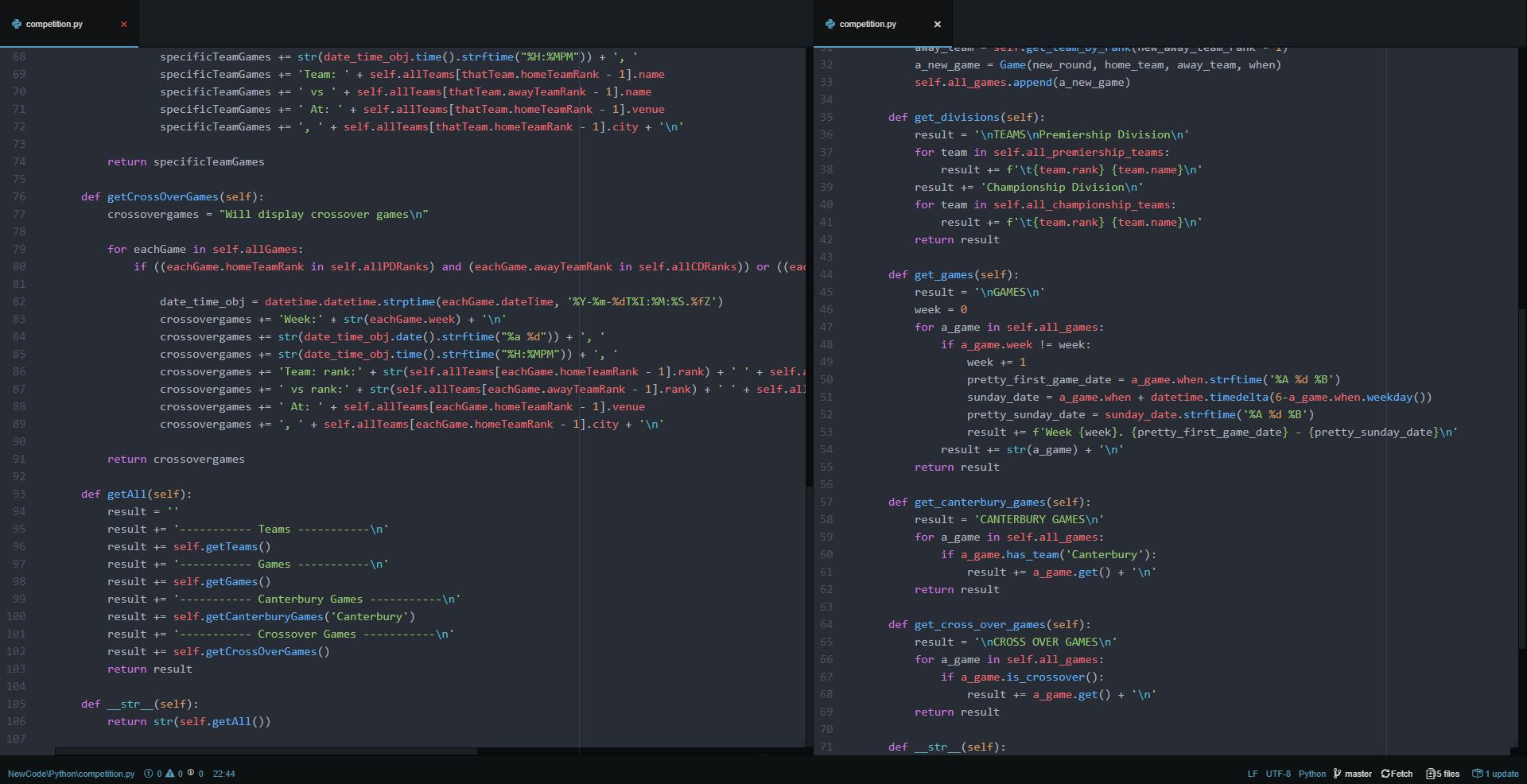
getGames() and get\_games(), I don’t really think that there any differences that I can write, it’s bit clearer to read on model answer, but I think it’s a difference in order to write, I start from A and add B, C, D… when model answer get A,B,C,D first then combine them to get one full line.

getCanterburyGames() and get\_canterbury\_games(), unlike previous function, it is a lot clearer than my code by using has\_team to check that team is Canterbury or not, and by using get(), does all the jobs that my 10 lines of code.

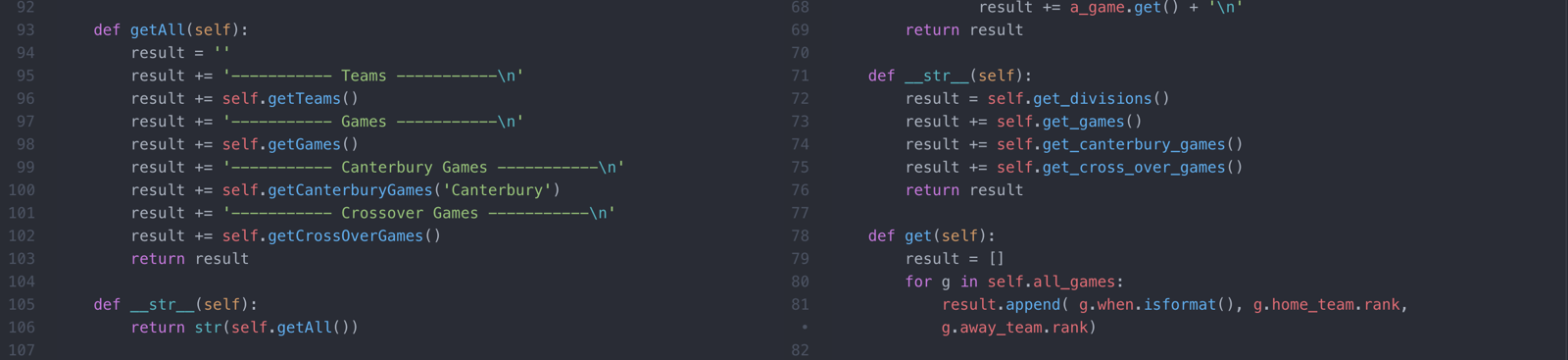
Again, better to make a function to reduce duplication, like my getGames() and getCanterburyGames(), the way I am getting output is really similar.



Same as above, massive difference of getting crossovergames.



I was going to use just \_\_str\_\_(self) like model answer, but part 1 was asking me to call getAll() function so I had to use like in left photo.



Overall, yes model answer is definitely a better code. By removing duplication like I had whenever try to print out games.

Not only that, model answer is trying to keep each method small as possible whereas some of my methods are bit longer than usual.