Python: Best Practices

When you work on paid projects, you'll not only have to produce great algorithms, but you'll also have to write good code. So it's essential that you apply best practices from the get-go. Here are some best practices that your skill test code absolutely MUST respect:

1) Every line of code must be contained within a function, or an object.

2) The only exception to rule 1) is if you want your script to do something automatically, as soon as it's called in the Terminal. In that case, move any code that needs to be auto-executed inside an **`if __name__ == '__main__':`** code block. See how these work here: https://stackoverflow.com/questions/419163/what-does-if-name-main-do

3) Every function must contain a clear docstring that explains what the function does, and what each of the input and returned values are.

4) If a line of code runs over more than 1/3 of your screen's size, split it into multiple lines if at all possible. This is to ensure that you don't have to scroll around to read your code.

5) No function should be longer than 30 lines of code (not including docstrings). This ensures that each function is simple, and carries out a well-defined task. There can very rarely be exceptions to this, but you should start feeling very guilty the moment your code runs over 30 lines.

6) No module should be longer than ~400 lines of code. If you have a script that's longer, split it into two modules, and try to make sure that the split is as logical as possible. Having one module that does your data preprocessing (e.g. **`data_preprocessing.py`**) and another that trains and saves your models (e.g. **`train.py`**) and possibly a third that runs it on unseen data in production (e.g. **`predict.py`**) usually works quite well.