# Coding Style for IFB299 Project – Official release

Django uses a variety of different objects to create the whole project, as such it requires a layout which is easily read and can be interpreted efficiently. This official release of the coding manual outlines the care taken when creating this project.

## Template Styling:

1. All templates with only special exceptions should extend from the base.html as this allows greater quality control
2. Always indent properly to make the document read easily
3. Make template tag easy to read by making the names relevant and put spacings between the tag and the brackets
   1. {{ tag }} or {% tag %} instead of {{tag}}

## General:

1. Variable names will depend on their context, if its to do with a model or form ir will have camel casing, if not it will be all lowercase.

## Views:

1. All view definitions will be in all lowercase
2. The first argument in a return statement will be request in its full spelling
3. The cleanest amount of arguments should be used in the return statement, often corresponds with the fewest
   1. Return render ( request, ‘template’, context) instead of ( request, ‘template’, ‘form’:form)

## Models and Forms:

These two are grouped together as most of their stylings are similar

1. All models and forms will use all uppercase for their class definition, for each word, and no spacings.
   1. UserProfileForm instead of Userprofile\_form
2. Field types for both models and forms should be all lower case and have an underscore between words
3. The structure of both objects will be as follows
   1. Class definition
   2. Class meta
   3. Save
   4. Def \_\_str\_\_ or def \_\_unicode\_\_