Python 编程规范（遵循 PEP 8）：

1. 命名规范：

- 变量名、函数名：小写字母，用下划线分隔单词（‘lowercase\_with\_underscores’）。

- 类名：驼峰式命名（‘CapitalizedWords’）。

- 常量：全大写字母，用下划线分隔单词（‘ALL\_CAPS\_WITH\_UNDERSCORES’）。

2. 缩进和空格：

- 使用 4 个空格作为缩进。

- 在运算符周围、函数参数和逗号之后等添加空格以提高可读性。

3. 注释：

- 注释应该清晰、简洁，解释代码的意图或复杂部分的功能。

- 使用文档字符串（docstrings）描述模块、函数、类等的用途、参数和返回值。

4. 代码结构：

- 尽量保持每行字符数在 79 个字符以内。

- 避免过度复杂的表达式，鼓励拆分成多行以提高可读性。

5. 导入模块：

- 按照一定的顺序分组导入模块：标准库模块、第三方库模块、应用程序模块。

- 避免使用通配符导入。

Django 编程规范：

1. 模型命名：

- 模型类名应该是单数形式，并且尽量具有描述性。

2. 视图函数：

- 视图函数应该有描述性的名称，表达清晰的功能。

3. URL 配置：

- 在 URL 配置中使用命名 URL 模式，以便于在模板或视图中引用。

4. 模板命名：

- 模板文件应该有描述性的名称，能够反映其功能。

5. 静态文件：

- 组织静态文件目录结构，按照功能或模块进行分类。

6. 设置和配置：

- 使用合理的设置和配置，避免敏感信息直接暴露在代码中。

7. 中间件和装饰器：

- 合理使用中间件和装饰器，以实现通用的功能或实现权限控制等。

8. 模型设计：

- 模型应该符合数据库设计的最佳实践，避免冗余或不必要的字段。

9. 表单和验证：

- 使用 Django 的表单类来简化表单的处理和验证。