List of Django topics in a recommended sequence for learning:

1. **Installation and Project Setup:**
   a. Installing Django
   b. Creating a new Django project
   c. Understanding project structure
2. **Models and Databases:**
   a. Defining models and fields
   b. Creating database tables with migrations
   c. Querying the database with Django ORM
3. **Admin Interface:**
   a. Enabling the admin site
   b. Registering models in the admin
   c. Customizing the admin interface
4. **Views and URLs:**
   a. Creating views and handling requests
   b. URL patterns and routing
   c. Passing data to templates
5. **Templates**:
   a. Template syntax and tags
   b. Rendering dynamic content
   c. Template inheritance and layout
6. **Static Files and Media:**
   a. Managing static files (CSS, JS, images)
   b. Handling media uploads and serving
7. **Forms**:
   a. Creating HTML forms with Django Forms
   b. Handling form submissions and validation
8. **Authentication and User Management:**
   a. User authentication and login
   b. User registration and password reset.
   c. User permissions and authentication decorators

9. **Class-Based Views (CBVs):**
   a. Understanding CBVs and their advantages
   b. Using built-in CBVs for common use cases
10. **Middleware:**
   a. Introduction to middleware
   b. Creating custom middleware for request/response processing
11. **Authentication and Permissions:**
   a. Implementing custom authentication backends
   b. Role-based permissions and access control
12. **Django REST Framework (optional but valuable for API development):**
   a. Overview of Django REST Framework
   b. Creating APIs with DRF serializers and views
   c. Authentication and permissions in DRF
13. **Deployment:**
   a. Preparing the project for deployment
   b. Deploying Django applications on a server (e.g., using Apache, Nginx, Gunicorn, etc.)
14. **Testing**:
   a. Writing unit tests for Django applications
   b. Using Django's testing framework
15. **Advanced Topics (optional and for more complex projects):**
   a. Caching and performance optimization
   b. Signals and asynchronous tasks
   c. Django's built-in middleware and pluggable applications