



# give me 50 important questions on the django

Here are **50 important questions on Django** covering fundamental to advanced concepts:

## Basic Django Questions

1. **What is Django?** - Explain Django as a Python web framework
2. **What are the advantages of using Django?** - Security, scalability, rapid development
3. **Explain Django's MVT architecture** - Model-View-Template pattern
4. **What is the difference between Django and Flask?** - Framework comparison
5. **Name some companies that use Django** - Instagram, Mozilla, Pinterest, etc.
6. **What is the difference between a Project and an App in Django?** - Project structure concepts
7. **How do you check the Django version installed?** - Using `django.VERSION` or command line
8. **What are Django's key features?** - ORM, admin interface, templating, etc.
9. **How did Django come into existence?** - Origin story and development history
10. **What databases does Django support?** - PostgreSQL, MySQL, SQLite, Oracle, MariaDB<sup>[1]</sup>  
<sup>[2]</sup>

## Django Models and Database

11. **What is a Django model?** - Python class defining data structure
12. **What is Django ORM?** - Object-Relational Mapping system
13. **How do you create a model in Django?** - Class inheritance from `models.Model`
14. **What are Django migrations?** - Database schema version control
15. **Explain the use of 'migrate' command** - Applying database changes
16. **How do you query all items from a model?** - Using `Model.objects.all()`
17. **How do you filter items from the database?** - Using `filter()` and `get()` methods
18. **What are Django model fields?** - CharField, IntegerField, DateField, etc.
19. **What is a ForeignKey in Django?** - Relationship between models
20. **What are Django model managers?** - Interface for database query operations<sup>[2]</sup> <sup>[1]</sup>

## Django Views and URLs

21. **What are Django views?** - Python functions handling HTTP requests
22. **What are Django URLs?** - URL routing and pattern matching
23. **What is URL dispatching in Django?** - How Django matches URLs to views
24. **What are function-based views vs class-based views?** - Different view implementations
25. **How do you handle HTTP methods in views?** - GET, POST, PUT, DELETE handling
26. **What is URL namespacing?** - Organizing URLs with namespaces
27. **How do you pass parameters through URLs?** - URL parameter capturing
28. **What are generic views in Django?** - Pre-built view classes
29. **How do you handle form data in views?** - Processing form submissions
30. **What is the purpose of `reverse()` function?** - URL reversal from view names<sup>[1]</sup> <sup>[2]</sup>

## Django Templates

31. **What is Django templating system?** - Template engine for dynamic HTML
32. **What is Jinja templating?** - Alternative templating engine
33. **What is template inheritance?** - Extending base templates
34. **What are template filters?** - Data transformation in templates
35. **What are template tags?** - Control flow and logic in templates
36. **What is context in Django?** - Data dictionary passed to templates
37. **How do you handle static files in Django?** - CSS, JavaScript, images management
38. **What is the difference between static and media files?** - Static vs user-uploaded content
39. **How do you include CSS and JavaScript in templates?** - Static file loading
40. **What are custom template tags and filters?** - Extending template functionality<sup>[2]</sup> <sup>[1]</sup>

## Advanced Django Concepts

41. **What is Django middleware?** - Request/response processing layers
42. **Explain Django's authentication system** - User authentication and authorization
43. **What is Django Rest Framework (DRF)?** - API development toolkit
44. **What are Django signals?** - Event-driven programming in Django
45. **What is Django admin interface?** - Built-in administrative interface
46. **What are Django forms?** - HTML form handling and validation
47. **What is CSRF protection in Django?** - Cross-Site Request Forgery prevention
48. **What is the Django session framework?** - Session management system
49. **What is Django caching?** - Performance optimization through caching
50. **What are Django management commands?** - Custom command-line utilities<sup>[1]</sup> <sup>[2]</sup>

## Additional Important Topics

These questions cover the **core concepts** every Django developer should understand, from basic framework knowledge to advanced features like **authentication, middleware, and API development**. The questions progress from foundational concepts to more complex topics, making them suitable for developers at different skill levels.



1. <https://www.geeksforgeeks.org/python/django-interview-questions/>
2. <https://www.simplilearn.com/django-interview-questions-article>
3. <https://codefinity.com/blog/Top-50-Technical-Interview-Questions-for-Django-Developers>
4. <https://www.interviewbit.com/django-interview-questions/>
5. <https://www.turing.com/interview-questions/django>
6. <https://www.linkedin.com/pulse/top-50-django-interview-questions-answers-you-need-know-nazir>
7. <https://www.index.dev/interview-questions/django>
8. <https://www.youtube.com/watch?v=F8Ow2tdle9A>
9. <https://flexiple.com/django/interview-questions>