

Online Tutorials and Guides

1. Django's Official Documentation

- The official Django documentation is a comprehensive resource that includes a tutorial for beginners. It's a great place to start because it not only teaches you how to use Django but also follows best practices recommended by the creators of the framework.
- Start here: [Django Official Tutorial](<https://docs.djangoproject.com/en/stable/intro/tutorial01/>)

2. Django for Beginners by William Vincent

- This book offers a project-based approach to learning Django. It's very beginner-friendly and covers the creation of several projects from a simple "Hello, World" application to more complex web applications involving user authentication, forms, and deployment.
- More details: [Django for Beginners](<https://djangoforbeginners.com/introduction/>)

3. A Complete Beginner's Guide to Django - Simple is Better Than Complex

- This series is perfect for those who prefer learning in a structured format. It breaks down Django into digestible sections, covering everything from setting up Django to understanding its more complex parts like ORM and authentication.
- Access the guide here: [Beginner's Guide to Django](<https://simpleisbetterthancomplex.com/series/beginners-guide/1.11/>)

4. PyCharm's Django Tutorial

- PyCharm, a popular IDE for Python, offers a detailed guide on how to learn Django. It includes setting up Django in PyCharm, which can be very helpful if you prefer using this IDE.
- Check it out: [How to Learn Django: A Comprehensive Guide for Beginners](<https://blog.jetbrains.com/pycharm/2024/01/how-to-learn-django/>)

Video Tutorials

1. Corey Schafer's Django Video Series

- Corey Schafer, a well-known figure in the Python community, has a series of YouTube videos that cover Django for beginners. His tutorials are clear and thorough, making complex concepts easier to understand.
- YouTube Channel: Corey Schafer

2. Django Tutorials - Real Python

- Real Python offers a mix of written and video tutorials that cover not just the basics but also dive into more advanced topics as you progress.
- Explore here: [Real Python Django Tutorials](<https://realpython.com/tutorials/django/>)

Interactive Learning

1. JetBrains Academy

- JetBrains provides an interactive Django track that can be very effective for beginners. It combines small bits of information followed by challenges and quizzes.

- Start learning: [JetBrains Academy Django Track](<https://www.jetbrains.com/academy/>)

2. Django Girls Tutorial

- This tutorial is designed to introduce beginners to web development using Django. It's very detailed and explains concepts in a way that is easy to understand for someone who has never programmed before.

- Tutorial link: [Django Girls Tutorial](<https://tutorial.djangogirls.org/>)

YouTube Links -

1) Django Tutorial for Beginners -

<https://www.youtube.com/watch?v=SlyxjRJ8VNY&list=PLsyeobzWxl7r2ukVgTqIQcl-1T0C2mzau&index=1>

2) Python Django Tutorial for Beginners -

<https://www.youtube.com/watch?v=rHux0gMZ3Eg>

Multiple-choice questions (MCQs) on Django :-

Question 1

What is Django?

- A) A Python library
- B) A Python framework
- C) A Python compiler
- D) A Python function

Answer: B) A Python framework

Question 2

Which architectural pattern does Django follow?

- A) MVC (Model-View-Controller)
- B) MVP (Model-View-Presenter)
- C) MVT (Model-View-Template)
- D) MVVM (Model-View-ViewModel)

Answer: C) MVT (Model-View-Template)

Question 3

What is the command to create a new Django project?

- A) django-startproject projectname
- B) django-admin startproject projectname
- C) django-create projectname
- D) start-django projectname

Answer: B) django-admin startproject projectname

Question 4

Which file defines the settings and configuration of a Django project?

- A) django.py
- B) settings.py
- C) config.py
- D) setup.py

Answer: B) settings.py

Question 5

What does ORM stand for in Django?

- A) Object-Relational Mapping
- B) Object-Repeated Model
- C) Object-Relation Model
- D) Operational Relational Model

Answer: A) Object-Relational Mapping

Question 6

Which of the following is NOT a default app provided by Django?

- A) Auth
- B) Admin
- C) Sessions
- D) Widgets

Answer: D) Widgets

Question 7

What is the purpose of the `manage.py` file in a Django project?

- A) To manage database settings
- B) To interact with the project through a command line
- C) To handle security settings
- D) To manage application URLs

Answer: B) To interact with the project through a command line

Question 8

Which command is used to start the Django development server?

- A) django-admin runserver
- B) manage.py startserver
- C) manage.py runserver
- D) django-admin startserver

Answer: C) manage.py runserver

Question 9

In Django, what is a "view"?

- A) A Python function or class that takes a web request and returns a web response
- B) A database model that specifically handles user authentication
- C) A static HTML file
- D) A configuration file for URL dispatching

Answer: A) A Python function or class that takes a web request and returns a web response

Question 10

Which file is used to define the URLs for a Django project?

- A) urls.py
- B) views.py
- C) models.py
- D) scripts.py

Answer: A) urls.py

Multiple-choice coding-related questions on Django :-

Question 1

What is the correct syntax to import Django's models module in your application?

- A) ``import models from django.db``
- B) ``from django.db import models``
- C) ``import django.db.models``
- D) ``from models import django.db``

Answer: B) ``from django.db import models``

Question 2

How do you create a new model called `Book` with a field `title` of type `CharField`?

- A)

```
class Book(models.Model):  
    title = models.CharField()
```

- B)

```
class Book(models.Model):  
    title = models.CharField(max_length=100)
```

- C)

```
class Book(models.Model):  
    title = models.StringField(max_length=100)
```

- D)

```
class Book(models.Model):  
    title = models.Text(max_length=100)
```

Answer: B)

```
class Book(models.Model):  
    title = models.CharField(max_length=100)
```

Question 3

Which command is used to apply migrations in Django?

- A) `python manage.py migrate`
- B) `python manage.py applymigrations`
- C) `python manage.py makemigrations`
- D) `python manage.py update_database`

Answer: A) `python manage.py migrate`

Question 4

How do you start an interactive shell with your Django project environment?

- A) `python manage.py shell`
- B) `python manage.py console`
- C) `django-admin shell`
- D) `django-admin console`

Answer: A) ``python manage.py shell``

Question 5

What is the correct way to define a URL pattern in Django that calls the ``index`` view function in ``views.py``?

- A)

```
path('home/', 'views.index')
```

- B)

```
path('home/', views.index, name='home')
```

- C)

```
url('home/', views.index, name='home')
```

- D)

```
link('home/', views.index)
```

Answer: B)

```
path('home/', views.index, name='home')
```

Question 6

Which file is typically used to store database settings in a Django project?

- A) ``database.py``

- B) ``settings.py``

- C) ``config.py``

- D) ``db.py``

Answer: B) ``settings.py``

Question 7

How do you define a ForeignKey relationship in Django?

- A)

```
publisher = models.ForeignKey('Publisher')
```

- B)

`publisher = models.ForeignKey('Publisher', on_delete=models.CASCADE)`

- C)

`publisher = models.OneToOneField('Publisher', on_delete=models.CASCADE)`

- D)

`publisher = models.ManyToOne('Publisher', on_delete=models.CASCADE)`

Answer: B)

`publisher = models.ForeignKey('Publisher', on_delete=models.CASCADE)`

Question 8

What is the purpose of the `makemigrations` command in Django?

- A) To create new migrations based on changes to your models
- B) To apply migrations to the database
- C) To revert changes made by the last migration
- D) To create a new Django project

Answer: A) To create new migrations based on changes to your models

Question 9

How do you create a superuser for a Django project?

- A) `python manage.py createsuperuser`
- B) `python manage.py superuser`
- C) `django-admin createsuperuser`
- D) `django-admin superuser`

Answer: A) `python manage.py createsuperuser`

Question 10

Which method in a Django model is used to customize how an instance of the model is displayed?

- A) `__str__(self)`
- B) `__repr__(self)`
- C) `__unicode__(self)`
- D) `__display__(self)`

Answer: A) `__str__(self)`