

[Latest Django Interview Questions](#)

Python Django interview questions and answers

Django is a free and open-source web framework written in [Python](#) and Based on the model-view-template architectural pattern. Django web framework is a set of components that help you to develop websites earlier and easier.

While building a website, you always need a similar set of components: a way to handle user authentication (signing up, signing in, signing out), a management panel for your website.

Fortunately, other people long ago noticed that the same problem is faced by developers while making a website, so they teamed up and created frameworks that give you ready-made components to use and Django is one of them.

Top Django interview questions and answers are given below.

Q1. [What do you understand by Django?](#)

Django is a free and open source web application framework, written in Python. Django is named after Django Reinhardt, Jazz guitarist from the 1930s to early 1950s who is one of the best guitarists of all time. Django was mainly developed to handle the intensive design of the newsroom. You can even build high-quality web applications using this. It adheres to the DRY principle and focuses completely on automating as much as possible.

Q2. [What does Of Django Field Class types do?](#)

Following points are specified by the Django Field Class type: –

- It specifies the database column type.
- It also specifies the default HTML widget which is availed while we render the form field.
- The requirements of the minimal validation which is used in Django admin is also specified by the field class.

Q3. [Does Django Follow Architectural pattern?](#)

Yes, Django follows Model-View-Controller (MVC) architectural pattern.

Q4. [Clarify the architecture of Django?](#)

Django follows MVC -MVT architecture. MVT stand for Model View Template design Pattern which is little bit different from MVC (Model View Controller) Pattern.

Q5. Is Django stable or not?

Of course, Django is stable. Most of the companies are using it.

Q6. Is Django a high level or low-level framework?

Django is a high-level Python's web framework which was designed for rapid development and clean realistic design.

Q7. How does Django work?

Django can be broken into several components:

- **Models.py file:** This file defines your data model by extending your single code line into full database tables and add a pre-built administration section to manage content.
- **Urls.py file:** It uses a habitual expression to confine URL patterns for processing.
- **Views.py file:** It is the main part of Django. The actual processing happens in view.

Q8. What is the name of the Foundation which manages Django web framework?

Django web framework is managed and maintained by an independent and non-profit organization named Django Software Foundation (DSF).

Q9. Name the features available in Django web framework?

Features available in Django web framework are:

- Admin Interface (CRUD)
- Templating
- Form handling
- Internationalization
- Session, user management, role-based permissions
- Object-relational mapping (ORM)
- Testing Framework
- Fantastic Documentation

Q10. What are the advantages of using Django for web development?

- Auto-generated web admin is provided by Django to make website administration easy.
- Pre-packaged API is also available for common user tasks.
- Business logic can be separated from the HTML using this framework.
- You can even divide the code modules into logical groups so as to make it flexible for changing.
- Template system is being provided so as to define HTML template for your web page to avoid code

duplication.

Q11. What is the process of creating a project in Django?

To start a project in Django, use the command `$django-admin.py` and then use the following command:

`Project`

`_init_.py`

`manage.py`

`settings.py`

Q12. Is Django a content management system (CMS)?

No, Django is not a Content Management System (CMS). Instead, it is a Web framework and a programming tool that helps you in building elegant websites.

Q13. What does the Django templates contain?

A template is a simple text file. It can create any text-based format like XML, CSV, HTML, etc. A template contains variables that get replaced with values when the template is evaluated and tags (`%tag%`) that control the logic of the template.

Q14. What is the use of session framework in Django?

The session framework helps you in storing and retrieving arbitrary data on a per-site visitor basis. The data is stored on the server side and abstracts the receiving and sending of cookies. We can implement sessions through a piece of middleware.

Q15. How can you set up static files in Django?

Basically, we require three main things to set up static files in Django:

- 1) Set `STATIC_ROOT` in `settings.py`
- 2) Run `manage.py collect static`
- 3) Set up a Static Files entry on the PythonAnywhere web tab

Q16. How to use file based sessions?

You have to set the SESSION_ENGINE settings to “django.contrib.sessions.backends.file” to use file-based session.

Q17. What is some typical usage of middlewares in Django?

Middleware is a function that acts on or transforms a request/response before/after it passes through the view layer (e.g. adding the user object to the request)

Some usage of middlewares in Django is:

- It can be used for Session management,
- User authentication can be done with the help of this.
- It helps in Cross-site request forgery protection
- Content Gziping, etc.

Q18. What is the usage of Django-admin.py and manage.py?

Django-admin.py: It is a Django’s command line utility for administrative tasks.**Manage.py:** It is an automatically created file in each Django project. It is a thin wrapper around the Django-admin.py. It has the following usage:

- It puts your project’s package on sys.path.
- It sets the DJANGO_SETTING_MODULE environment variable to points to your project’s setting.py file.

Q19. What are signals in Django?

Signal are inbuilt utility in Django. They allow to execute some piece of code based on some action or event is occurred in framework something like a new user register, on delete of a record.

Below is the list of some inbuilt signal in Django.

- pre_save and post_save.
- pre_delete and post_delete
- pre_request and post_request
- pre_request and post_request

Q20. Is Django free?

Yes, Django is free open source web framework for Python

Q21. Django is based on which design pattern.

Django closely follows the MVC (Model View Controller) design pattern, however, it does use its own logic in the implementation. Because the “C” is handled by the framework itself and most of the excitement in Django

happens in models, templates, and views, Django is often referred to as an MTV framework. In the MTV development pattern:

- M stands for “Model,” the data access layer. This layer contains anything and everything about the data: how to access it, how to validate it, which behaviors it has, and the relationships between the data.
- T stands for “Template,” the presentation layer. This layer contains presentation-related decisions: how something should be displayed on a Web page or other type of document.
- V stands for “View,” the business logic layer. This layer contains the logic that accesses the model and defers to the appropriate template(s). You can think of it as the bridge between models and templates.

Further reading <https://djangobook.com/model-view-controller-design-pattern/>

Q22. When and who create Django?

According to [https://en.wikipedia.org/wiki/Django_\(web_framework\)](https://en.wikipedia.org/wiki/Django_(web_framework)), Django was created in the fall of 2003, when the web programmers at the Lawrence Journal-World newspaper, Adrian Holovaty and Simon Willison, began using Python to build applications. It was released publicly under a BSD license in July 2005. The framework was named after guitarist Django Reinhardt.

Q23. List server requirement to install Django Framework.

As Django is Python Framework, in order to install Django Python is required. Django comes with an inbuilt lightweight web server that you can use for the testing purpose. If you are using Django on production **Apache with mod_wsgi** is required.

Q24. List the database backends supported by Django Framework?

Django officially supports four database backends, they are

- PostgreSQL
- MySQL
- SQLite
- Oracle

In addition to these, you can also use following 3rd parties

- SAP SQL Anywhere
- IBM DB2
- Microsoft SQL Server
- Firebird
- ODBC

Q25. What is recommended way to install Django?

Installing using pip is the recommended way to install Django Framework. Below are the steps to install official release of Django with pip

- Install pip.
- Configure virtualenv and virtualenvwrapper
- Once virtual environment is created and activated, enter the command **pip install Django** to install Django

Q26. How to install the development version of Django

Follow the below steps to Install the development version of Django Framework.

- Check out Django's main development branch

```
$ git clone https://github.com/django/django.git
```

- Make sure that the Python interpreter can load Django's code. The most convenient way to do this is to use virtualenv, virtualenvwrapper, and pip.
- After setting up and activating the virtualenv, run the following command:

```
$ pip install -e django/
```

Source:<https://docs.djangoproject.com/en/2.0/topics/install/>

Q27. Where are Django migrations stored?

You can think Django Migrations as version control system for your database/Model. It keeps track of changes done in your application Models/Table like adding a field, deleting a model, etc. Migrations in Django are stored as an on-disk format, referred to here as “migration files”. These files are actually just normal Python files with an agreed-upon object layout, written in a declarative style. A basic migration file looks like this:

```
from django.db import migrations, models
class Migration(migrations.Migration):
    dependencies = [
        ('migrations', '0001_initial')
    ]
    operations = [
        migrations.DeleteModel('Tribble'),
        migrations.AddField('Author', 'rating', models.IntegerField(default=0)),
    ]
```

Further Reading <https://docs.djangoproject.com/en/2.0/topics/migrations/>

Q28. How a request is processed in Django?

In Django whenever a request is made by a user, it goes through the following steps:

- Django determines the root URLconf module to use. Ordinarily, this is the value of the

ROOT_URLCONF setting, but if the incoming HttpRequest object has a **urlconf** attribute (set by middleware), its value will be used in place of the **ROOT_URLCONF** setting.

- Django loads that Python module and looks for the variable `urlpatterns`. This should be a Python list of **`django.urls.path()`** and/or **`django.urls.re_path()`** instances.
- Django runs through each URL pattern, in order, and stops at the first one that matches the requested URL.
- Once one of the URL patterns matches, Django imports and calls the given view, which is a simple Python function (or a class-based view). The view gets passed the following arguments:
 - An instance of `HttpRequest`.
 - If the matched URL pattern returned no named groups, then the matches from the regular expression are provided as positional arguments.
 - The keyword arguments are made up of any named parts matched by the path expression, overridden by any arguments specified in the optional `kwargs` argument to **`django.urls.path()`** or **`django.urls.re_path()`**.
 - If no URL pattern matches, or if an exception is raised during any point in this process, Django invokes an appropriate error-handling view.

Q29. [When to use the iterator in Django ORM?](#)

Iterators are used for traversing an object in Python which implements iterator protocol. It consists of two methods `__iter__()` and `next()`.

In Django, a good use of iterator is when you are processing results that take up a large amount of available memory (lots of small objects or fewer large objects).

For more clarification please read when to use and when to not use `iterator()` in the Python Django ORM <https://stackoverflow.com/questions/12681653/when-to-use-or-not-use-iterator-in-the-django-orm>

Q30. [When QuerySets are evaluated in Django?](#)

In Django, a `QuerySet` can be evaluated in Iteration, Slicing, Pickling/Caching, `repr()`, `len()`, `list()` and `bool()`.

Q31. [How to check installed version of Django?](#)

By running below command on Terminal. You can check installed version of Django Framework.

```
python -m django --version
```

Q32. [How to set/unset session in Django?](#)

Setting Session in Django

```
request.session['key'] = 'value'
```

Unset Session in Django

```
del request.session['key']
```

Q33. What is a context in Django?

In Django Context is a dictionary with variable names in the form of key and value like {variable1: 101, variable2: 102}, when we pass this context to the template render method, {{ variable1 }} would be replaced with 101 and {{ variable2 }} with 102 in your template.

Q34. Explain mixins in Django.

A mixin is a special kind of multiple inheritances in Python. There are two main situations where mixins are used:

- You want to provide a lot of optional features for a class.
- You want to use one particular feature in a lot of different classes.

Read More from <https://stackoverflow.com/questions/533631/what-is-a-mixin-and-why-are-they-useful>

Q35. How to get current page URI in Django template.

You can use {{ request.path }} and {{ request.get_full_path }} to get current page URI in Django template

Q36. List some popular websites built on Django framework?

Following are the list of top 10 websites built on Django framework.

1. Instagram
2. Disqus
3. Bitbucket
4. Mozilla Firefox
5. Pinterest
6. NASA
7. Onion
8. The Washington Post
9. Eventbrite
10. Mahalo

Q37. How to create an Constant in Django.

To create a constant in Django. Open your settings.py file and add a variable like MY_CONST = "MY_VALUE".

To use this constant in your views simply import setting like "Import settings in views.py" and use it as **settings.MY_CONST**

Q38. How to Configure Database in Django.

Q39. List type of relationships/ asociations supported by Django.

Q40. What django collectstatic command does?

Q41. What are Class-based views in Django

Q42. What constitutes Django templates?

Template can create formats like XML,HTML and CSV(which are text-based formats). In general terms template is a simple text file. It is made up of variables that will later be replaced by values after the template is evaluated and has tags which will control template's logic.

Q43. List some caching strategies that you know in Django!

Few caching strategies that are available in Django are as follows:

- File system caching
- In-memory caching
- Using Memcached
- Database caching

Q44. List type of inheritance supported in Django?

Django supports 3 types of inheritance. They are

- Abstract base classes
- Multi-table Inheritance
- Proxy models

Django is a web framework that helps to build the web application using Python programming language. It follows the MVC (model-view-controller) pattern as its architecture. Django is basically used to create complex

database-driven web pages while also providing features of creating, reading, updating and deleting interfaces.

Django runs on MVC architecture. Following are the components that makeup Django architecture:

- **Models:** Models elaborate back-end stuffs like the database schema. (relationships)
- **Views:** Views control what is to be shown to end-user.
- **Templates:** Templates deal with the formatting of view.
- **Controller:** Takes entire control of Models. An MVC framework can be compared to a Cable TV with remote. A Television set is a View(that interacts with end user), the cable provider is model(that works in the back-end) and Controller is remote that controls which channel to select and display it through the view.