1. How to Create a Class-Based View in Django?

A **Class-Based View (CBV)** is a way to define views using Python classes instead of functions.

Example: Simple Class-Based View

In views.py:

```
from django.views.generic import TemplateView


class HomeView(TemplateView):
    template_name = "home.html"  # Specifies the template to display
```

Connect It in urls.py

```
from django.urls import path
from .views import HomeView


urlpatterns = [
    path('', HomeView.as_view(), name='home'),  # Calls the view
]
```

👉 Now, when a user visits /, the **home.html** template is shown.

---

2. Role of urls.py in Connecting a View to a URL

- urls.py **maps** a URL to a specific view so Django knows which page to load.
- It connects **URLs (links)** with **views (functions/classes)**.

Example (urls.py)

```
from django.urls import path
from .views import about_view


urlpatterns = [
    path('about/', about_view, name='about'),
]
```

📌 Now, visiting /about/ will trigger about_view() from views.py.

---

3. Purpose of the Context Dictionary in Django Views

- The **context dictionary** is used to **pass data** from a view to a template.
- It makes templates **dynamic** instead of hardcoded.

Example: Using Context in a View

```
from django.shortcuts import render


def profile_view(request):
   context = {'name': 'Alice', 'age': 30}
   return render(request, 'profile.html', context)
```

Example: Using Context in a Template (profile.html)

```
<h1>Hello, {{ name }}!</h1>
<p>Age: {{ age }}</p>
```

👉 **Output:**

makefile

```
Hello, Alice!
Age: 30
```