**1. What is Rendering Dynamic Templates?**

* **Dynamic templates** allow you to display data from your Django views in HTML.
* Django uses the **render()** function to send data from **views.py** to **templates (.html)**.

**2. Project Setup Example**

Assume we have this structure:

ch10**/**

│

├── djangoproject\_ch10**/**

│   ├── urls**.**py

│   ├── settings**.**py

│

├── course**/**

│   ├── views**.**py

│   ├── urls**.**py

│   ├── templates**/**

│       ├── course**/**

│           ├── index**.**html

**3. render() Syntax**

* **return** render(
* request**,**
* template\_name**,**
* **context=**None**,**
* **content\_type=**None**,**
* **status=**None**,**
* **using=**None
* )
* **template\_name**: Path of template ('course/index.html')
* **context**: Dictionary containing variables to send to the template
* **content\_type**: (optional) MIME type, default is text/html
* **status**: (optional) HTTP status code
* **using**: (optional) Template engine to use

**4. Three Methods to Pass Data Dynamically**

**Method 1: Pass a Dictionary Directly in render()**

In **views.py**:

**from** django**.**shortcuts **import** render

**def** index(**request**)**:**

**return** render(request**,** 'course/index.html'**,** {

        'title'**:** 'Welcome to Django'**,**

        'author'**:** 'Talha'

    })

In **index.html**:

**<**h1**>**{{ title }}**</**h1**>**

**<**p**>**Author**:** {{ author }}**</**p**>**

✅ This is a quick method when you only have a few variables.

**Method 2: Create Variables in View, Then Pass as Dictionary**

In **views.py**:

**def** index(**request**)**:**

    title **=** "Welcome to Django"

    author **=** "Talha"

    courses **=** ['Python'**,** 'Django'**,** 'React']

    context **=** {

        'title'**:** title**,**

        'author'**:** author**,**

        'courses'**:** courses

    }

**return** render(request**,** 'course/index.html'**,** context)

In **index.html**:

**<**h1**>**{{ title }}**</**h1**>**

**<**p**>**Author**:** {{ author }}**</**p**>**

**<**h3**>**Courses**:</**h3**>**

**<**ul**>**

    {**%** **for** c **in** courses **%**}

**<**li**>**{{ c }}**</**li**>**

    {**%** endfor **%**}

**</**ul**>**

✅ This method is cleaner when passing multiple variables.

**Method 3: Pass a Predefined Dictionary (Global or from Database)**

In **views.py**:

**def** index(**request**)**:**

    context **=** {

        'title'**:** 'Django Dynamic Templates'**,**

        'author'**:** 'Talha'**,**

        'details'**:** {'email'**:** 'talha@example.com'**,** 'role'**:** 'Developer'}

    }

**return** render(request**,** 'course/index.html'**,** context)

In **index.html**:

**<**h1**>**{{ title }}**</**h1**>**

**<**p**>**Author**:** {{ author }}**</**p**>**

**<**p**>**Email**:** {{ details**.**email }} **|** Role**:** {{ details**.**role }}**</**p**>**

✅ This method is used when data is **fetched from the database** or **comes from another source**.

**5. Using Dynamic Templates with Loops & Conditions**

In **index.html**:

* **<**h3**>**Courses**:</**h3**>**
* **<**ul**>**
* {**%** **if** courses **%**}
* {**%** **for** c **in** courses **%**}
* **<**li**>**{{ c**|**upper }}**</**li**>**
* {**%** endfor **%**}
* {**%** **else** **%**}
* **<**li**>**No courses available**</**li**>**
* {**%** endif **%**}
* **</**ul**>**
* **{% if %}**: Check if variable exists or has data
* **{% for %}**: Loop through list
* **|upper**: Template filter to convert text to uppercase

**6. Best Practices**

1. Always keep templates inside templates/app\_name/.
2. Pass a single context dictionary rather than too many variables directly.
3. Use **loops** and **filters** to avoid complex logic inside templates.
4. Use base.html with {% extends %} and {% block %} for DRY code.

**7. URL Configuration**

In **course/urls.py**:

**from** django**.**urls **import** path

**from** **.** **import** views

urlpatterns **=** [

    path(''**,** views**.**index**,** **name=**'index')**,**

]

In **djangoproject\_ch10/urls.py**:

**from** django**.**contrib **import** admin

**from** django**.**urls **import** path**,** include

urlpatterns **=** [

    path('admin/'**,** admin**.**site**.**urls)**,**

    path(''**,** include('course.urls'))**,**

]

**8. Flow of Dynamic Template Rendering**

1. **User** visits / URL
2. **urls.py** routes the request to index view
3. **View** prepares data and calls render()
4. **Template** (index.html) displays data dynamically
5. **Response** goes back to browser