# Django Template Inside Template using include **Tag** in Django 5

## 📌 What is the include Tag?

* Django templates allow us to **reuse smaller HTML templates** inside other templates.
* This helps in **avoiding code repetition** (DRY principle – Don’t Repeat Yourself).
* Example: If you have the same **navbar** or **footer** across multiple pages, instead of writing it again and again, you can put it in one file and include it in all other templates.

## 📌 How to Use the include Tag?

* Syntax:

{**%** include 'template\_name.html' **%**}

👉 This will insert the content of that file inside your current template.

## 📌 Example Project

We’ll build a **Student Management** example where we include a **navbar** in all templates.

### 1. ****views.py****

*# student/views.py*

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

*# Home Page View*

**def** home(**request**)**:**

**return** render(request**,** 'student/home.html')

*# About Page View*

**def** about(**request**)**:**

**return** render(request**,** 'student/about.html')

*# Contact Page View*

**def** contact(**request**)**:**

**return** render(request**,** 'student/contact.html')

### 2. ****urls.py****

*# student/urls.py*

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

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

urlpatterns **=** [

    path(''**,** views**.**home**,** **name=**'home')**,**          *# http://127.0.0.1:8000/*

    path('about/'**,** views**.**about**,** **name=**'about')**,**  *# http://127.0.0.1:8000/about/*

    path('contact/'**,** views**.**contact**,** **name=**'contact')**,**  *# http://127.0.0.1:8000/contact/*

]

### 3. ****Templates Structure****

student**/**

 └── templates**/**

      └── student**/**

           ├── base**.**html

           ├── navbar**.**html   (included template)

           ├── home**.**html

           ├── about**.**html

           └── contact**.**html

### 4. ****navbar.html (included file)****

**<**!-- student**/**navbar**.**html --**>**

**<**nav style**=**"background: lightblue; padding: 10px;"**>**

**<**a href**=**"{% url 'home' %}"**>**Home**</**a**>** **|**

**<**a href**=**"{% url 'about' %}"**>**About**</**a**>** **|**

**<**a href**=**"{% url 'contact' %}"**>**Contact**</**a**>**

**</**nav**>**

✅ This file will be reused on every page.

### 5. ****base.html****

**<**!-- student**/**base**.**html --**>**

**<**!DOCTYPE html**>**

**<**html**>**

**<**head**>**

**<**title**>**{**%** block title **%**} My Website {**%** endblock **%**}**</**title**>**

**</**head**>**

**<**body**>**

**<**!-- Include Navbar --**>**

    {**%** include 'student/navbar.html' **%**}

**<**!-- Page Content Goes Here --**>**

**<**div style**=**"margin:20px;"**>**

        {**%** block content **%**}

        {**%** endblock **%**}

**</**div**>**

**</**body**>**

**</**html**>**

✅ This is the **parent template** → it includes navbar.html and defines block title & block content.

### 6. ****home.html****

**<**!-- student**/**home**.**html --**>**

{**%** extends 'student/base.html' **%**}

{**%** block title **%**} Home Page {**%** endblock **%**}

{**%** block content **%**}

**<**h1**>**Welcome to Home Page**</**h1**>**

**<**p**>**This **is** the home page content**.</**p**>**

{**%** endblock **%**}

### 7. ****about.html****

**<**!-- student**/**about**.**html --**>**

{**%** extends 'student/base.html' **%**}

{**%** block title **%**} About Us {**%** endblock **%**}

{**%** block content **%**}

**<**h1**>**About Us**</**h1**>**

**<**p**>**This **is** the about page**.</**p**>**

{**%** endblock **%**}

### 8. ****contact.html****

**<**!-- student**/**contact**.**html --**>**

{**%** extends 'student/base.html' **%**}

{**%** block title **%**} Contact Us {**%** endblock **%**}

{**%** block content **%**}

**<**h1**>**Contact Us**</**h1**>**

**<**p**>**Email**:** info**@**example**.**com**</**p**>**

{**%** endblock **%**}

## 📌 Explanation of What We Did

1. **Created reusable file → navbar.html**
   * Contains menu links.
   * Included inside base.html.
2. **base.html = Parent template**
   * Defines the structure of website.
   * Uses {% include %} to reuse navbar.html.
   * Defines {% block %} placeholders for child templates.
3. **Child templates (home, about, contact)**
   * Use {% extends 'base.html' %} to inherit layout.
   * Replace {% block %} with their own content.

## 📌 Benefits of include Tag

✔ Code Reusability (no repeated navbar/footer)  
✔ Cleaner & more structured templates  
✔ Easy to maintain – change navbar once, it updates everywhere  
✔ Can pass variables into included templates like:

{**%** include 'student/navbar.html' **with** username="Talha" **%**}

✅ So, in **Django 5**, the include tag works the same as before, but with more flexibility:

* You can include small templates.
* You can pass context (variables).
* You can also handle missing templates safely using:

{**%** include "missing.html" ignore missing **%**}

More concept

## 🔹 Concept

* include tag allows us to reuse small parts of HTML (components) across multiple templates.
* We usually keep reusable files inside a **components/ folder** inside the templates/.
* Any variable passed from the **view** is automatically available inside the included template.
* We can **override/update values** while including using with keyword.
* If we want some default values in the included file, we can use {% block %} or conditional checks.
* We can update things using with and only keywork in include to pass the values from view to html and than into **components/ folder**

## 🔹 Example Project

### 1. ****views.py****

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

**def** home(**request**)**:**

**return** render(request**,** 'core/home.html'**,** {

        'name'**:** 'Talha'**,**

        'page\_title'**:** 'Home Page'

    })

**def** about(**request**)**:**

**return** render(request**,** 'core/about.html'**,** {

        'name'**:** 'Ali'**,**

        'page\_title'**:** 'About Page'

    })

### 2. ****urls.py****

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

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

urlpatterns **=** [

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

    path('about/'**,** views**.**about**,** **name=**'about')**,**

]

### 3. ****Base Layout → core/base.html****

**<**!DOCTYPE html**>**

**<**html**>**

**<**head**>**

**<**title**>**{{ page\_title }}**</**title**>**

**</**head**>**

**<**body**>**

**<**header**>**

        {**%** include 'components/navbar.html' **%**}

**</**header**>**

**<**main**>**

        {**%** block content **%**}{**%** endblock **%**}

**</**main**>**

**<**footer**>**

        {**%** include 'components/footer.html' **%**}

**</**footer**>**

**</**body**>**

**</**html**>**

### 4. ****Component Example → components/navbar.html****

**<**nav**>**

**<**ul**>**

**<**li**><**a href**=**"/"**>**Home**</**a**></**li**>**

**<**li**><**a href**=**"/about/"**>**About**</**a**></**li**>**

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

**<**p**>**Welcome**,** {{ name }}!**</**p**>**

**</**nav**>**

➡️ Notice: {{ name }} will come **automatically** from the view.

### 5. ****Page Example → core/home.html****

{**%** extends 'core/base.html' **%**}

{**%** block content **%**}

**<**h1**>**This **is** Home Page**</**h1**>**

**<**!-- include component (**with** same name passed **from** view) --**>**

    {**%** include 'components/welcome.html' **%**}

**<**!-- include component but overwrite value --**>**

    {**%** include 'components/welcome.html' **with** name="Changed in Home" **%**}

{**%** endblock **%**}

### 6. ****Page Example → core/about.html****

{**%** extends 'core/base.html' **%**}

{**%** block content **%**}

**<**h1**>**This **is** About Page**</**h1**>**

**<**!-- reuse same component --**>**

    {**%** include 'components/welcome.html' **%**}

**<**!-- **pass** new value only **for** About page --**>**

    {**%** include 'components/welcome.html' **with** name="Special Name for About" **%**}

{**%** endblock **%**}

### 7. ****Reusable Component → components/welcome.html****

**<**div**>**

**<**h2**>**Hello {{ name }}**</**h2**>**

**<**p**>**This section **is** reused **in** multiple pages**.</**p**>**

**</**div**>**

## 🔹 Keywords to Remember

* **{% include "path/to/file.html" %}** → include file as it is.
* **{% include "file.html" with key="value" %}** → pass/overwrite value.
* **Variables from view are automatically available** in included file.
* **Reusable Components** → put in templates/components/.
* **Use for**: navbar, footer, sidebar, small repeated sections.

✅ Now you have:

* Full working **views.py**, **urls.py**, **base.html**, **home.html**, **about.html**, **components folder**.
* Example of **default values** (from view) and **overwriting** values (with).

Only exmple:

### Updated Example with only

{**%** extends 'core/base.html' **%**}

{**%** block content **%**}

**<**h1**>**This **is** About Page**</**h1**>**

**<**!-- reuse same component **with** all context --**>**

    {**%** include 'components/welcome.html' **%**}

**<**!-- **pass** new value only **for** About page**,** ignore parent context --**>**

    {**%** include 'components/welcome.html' **with** name="Special Name for About" only **%**}

{**%** endblock **%**}

### 🔹 What does only mean?

* Normally, when you use {% include %}, Django **automatically passes all variables from the parent template’s context** into the included template.
* But sometimes you don’t want that — you want the included template to **only use the variables you explicitly pass in**.

That’s when you use **only**.

### 🔹 Without only (default)

{**%** include 'components/welcome.html' **with** name="Special Name for About" **%**}

👉 Here, components/welcome.html can access:

* The new variable name="Special Name for About"
* Plus **all other variables** from the parent (like title, about, etc.)

### 🔹 With only

{**%** include 'components/welcome.html' **with** name="Special Name for About" only **%**}

👉 Here, components/welcome.html can **only access**:

* The variable name="Special Name for About"

❌ It won’t see other variables (like title, about, etc.) from the parent.

### 🔹 Why use only?

* To **isolate** the included template.
* To **avoid conflicts** (maybe the parent also has a name variable, and you don’t want it to leak in).
* To make the component **reusable** — it only depends on what you pass.

✅ In simple words:

* **No only** → Pass everything + new variable.
* **With only** → Pass **only** the new variable(s), nothing else.