

**Template inheritance** in Django is a feature that allows you to build a base HTML structure (a base template) and extend it in other templates. It promotes reusability and consistency across your web application.

**For example:**

1. **Base Template (base.html):** Contains the common structure (e.g., header, footer).

html

Copy code

```
<!DOCTYPE html>
<html>
<head>
  <title>{% block title %}My Site{% endblock %}</title>
</head>
<body>
  <header>Site Header</header>
  {% block content %}{% endblock %}
  <footer>Site Footer</footer>
</body>
</html>
```

2. **Child Template (child.html):** Extends the base and overrides specific blocks.

html

Copy code

```
{% extends 'base.html' %}

{% block title %}Blog Page{% endblock %}

{% block content %}
  <h1>Welcome to the Blog</h1>
{% endblock %}
```

When rendering `child.html`, Django combines the base template and overridden blocks, resulting in a complete page.

In Django, the syntax for template inheritance involves the following:

1. **Base Template:** Define blocks with `{% block block_name %}{% endblock %}` in the base template:

html

Copy code

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<title>{% block title %}Default Title{% endblock %}</title>
```

```
</head>
```

```
<body>
```

```
<header>Site Header</header>
```

```
{% block content %}{% endblock %}
```

```
<footer>Site Footer</footer>
```

```
</body>
```

```
</html>
```

2. **Child Template:** Extend the base template and override blocks using `{% extends %}` and `{% block %}`:

html

Copy code

```
{% extends 'base.html' %}
```

```
{% block title %}Child Page Title{% endblock %}
```

```
{% block content %}
```

```
<p>This is the child page content.</p>
```

```
{% endblock %}
```

This ensures reusability and consistent layout across your application.

In Django, the syntax for template inheritance involves the following:

1. **Base Template:** Define blocks with `{% block block_name %}{% endblock %}` in the base template:

html

Copy code

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<title>{% block title %}Default Title{% endblock %}</title>
```

```
</head>
```

```
<body>
```

```
<header>Site Header</header>
```

```
{% block content %}{% endblock %}
```

```
<footer>Site Footer</footer>
```

```
</body>
```

```
</html>
```

2. **Child Template:** Extend the base template and override blocks using `{% extends %}` and `{% block %}`:

html

Copy code

```
{% extends 'base.html' %}
```

```
{% block title %}Child Page Title{% endblock %}
```

```
{% block content %}
```

```
<p>This is the child page content.</p>
```

```
{% endblock %}
```

1. This ensures reusability and consistent layout across your application.

## types of template inheritance:

In Django, there are two primary types of template inheritance:

### 1. Single-level Inheritance

This is the simplest form where a child template directly inherits from a single base template.

Base template (**base.html**) defines common blocks:

html

Copy code

```
<!DOCTYPE html>
<html>
<head>
  <title>{% block title %}Default Title{% endblock %}</title>
</head>
<body>
  {% block content %}{% endblock %}
</body>
</html>
```

Child template (**child.html**) overrides those blocks:

html

Copy code

```
{% extends 'base.html' %}

{% block title %}Child Page Title{% endblock %}
{% block content %}
  <p>This is child content.</p>
{% endblock %}
```

---

### 2. Multi-level Inheritance

In this approach, multiple layers of templates extend from one another. For example:

Base template (**base.html**):

html

Copy code

```
<!DOCTYPE html>
<html>
<body>
  <header>Common Header</header>
  {% block body_content %}{% endblock %}
  <footer>Common Footer</footer>
</body>
</html>
```

Intermediate template (**section\_base.html**):

html

Copy code

```
{% extends 'base.html' %}

{% block body_content %}
  <div class="section">
    {% block section_content %}{% endblock %}
  </div>
{% endblock %}
```

Final template (**specific\_page.html**):

html

Copy code

```
{% extends 'section_base.html' %}

{% block section_content %}
  <h1>Specific Page Content</h1>
{% endblock %}
```

This allows for layered reusability, making the design more modular and maintainable.

## Advantages of Template Inheritance in Django

1. **Reusability:** Common components like headers, footers, and navigation bars are defined once in a base template and reused across all child templates.
  2. **Consistency:** Ensures a uniform layout and design throughout the application.
  3. **Modularity:** Changes in the base template automatically reflect across all extending templates.
  4. **Efficiency:** Reduces duplication of HTML code, making templates easier to manage and maintain.
  5. **Scalability:** Supports multi-level inheritance for large and complex projects.
- 

## Disadvantages of Template Inheritance

1. **Complexity:** Multi-level inheritance can become difficult to track and debug.
2. **Dependency:** If the base template changes, all child templates might require adjustments.
3. **Learning Curve:** New developers might find it challenging to understand nested template structures.
4. **Overhead:** Too many nested blocks can make templates harder to read and maintain.

By using template inheritance wisely, you can minimize these drawbacks while leveraging its benefits.

## Project:

Project - django-admin startproject class4

App - django-admin startapp blog

Folder - blogapp - templates folder:

1. about.html
2. base.html
3. contact.html
4. home.html

## 5. service.html

### About.html:

```
{% extends 'base.html' %} {% block title %}

About - MyBlog

{% endblock %}

{% block content %}

<h1>Welcome to About page</h1>

<p>This is the about page of my blog. Lorem Ipsum is simply dummy text of
the printing and typesetting industry. Lorem Ipsum has been the industry's
standard dummy text ever since the 1500s, when an unknown printer took a
galley of type and scrambled it to make a type specimen book. It has
survived not only five centuries, but also the leap into electronic typesetting,
remaining essentially unchanged. It was popularised in the 1960s with the
release of Letraset sheets containing Lorem Ipsum passages, and more
recently with desktop publishing software like Aldus PageMaker including
versions of Lorem Ipsum. </p>

{% endblock %}
```

### Base.html:

```
<!DOCTYPE html>

{% load static %}
```

```
<html lang="en">

<head>

  <meta charset="UTF-8">

  <meta name="viewport" content="width=device-width, initial-scale=1.0">

  <title>Document</title>

  <style>

    body {

      font-family: Arial, sans-serif;

      margin: 0;

      padding: 0;

      background-color: #f4f4f4;

      color: #333;

    }

    header {

      background-color: #333;

      color: white;

      padding: 20px;

      text-align: center;

    }
```



```
nav ul {  
  
    list-style-type: none;  
  
    padding: 0;  
  
    margin: 0;  
  
    margin-top: 10px;  
  
}  
  
nav ul li {  
  
    display: inline;  
  
    margin-right: 15px;  
  
}  
  
nav ul li a {  
  
    color: white;  
  
    text-decoration: none;  
  
}  
  
nav ul li a:hover {  
  
    text-decoration: underline;  
  
}
```

```
div {  
  
    padding: 20px;  
  
}
```

```
footer {  
  
    background-color: #333;  
  
    color: white;  
  
    text-align: center;  
  
    padding: 10px;  
  
    position: fixed;  
  
    width: 100%;  
  
    bottom: 0;  
  
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<header>
```

Welcome to MyBlog

```
<nav>
```

```
<ul>

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

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

  <li><a href="/contact/">Contact</a></li>

  <li><a href="/service/">Service</a></li>

</ul>

</nav>

</header>

<div>

{% block content %}


{% endblock %}

</div>

<footer>

  <p>&copy; 2024 MyBlog </p>

</footer>
```

```
</body>
```

```
</html>
```

## Contact.html:

```
{% extends 'base.html' %} {% block title %}
```

```
Contact - MyBlog
```

```
{% endblock %}
```

```
{% block content %}
```

```
<h1>Welcome to contact page</h1>
```

```
<p>This is the contact page of my blog. Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book. It has survived not only five centuries, but also the leap into electronic typesetting, remaining essentially unchanged. It was popularised in the 1960s with the release of Letraset sheets containing Lorem Ipsum passages, and more recently with desktop publishing software like Aldus PageMaker including versions of Lorem Ipsum. </p>
```

```
<h2>Contact Us</h2>
```

```
<p>Shreya</p> <br>
```

```
<p>Address: Bangalore</p> <br>
```

```
<p>Phone: 1234567890</p> <br>

<p>Email: shreya@gmail.com</p> <br>

{% endblock %}
```

## Home.html:

```
{% extends 'base.html' %} {% block title %}

Home - MyBlog

{% endblock %}

{% block content %}

<h1>Welcome to home page</h1>

<p>This is the home page of my blog. Lorem Ipsum is simply dummy text of the
printing and typesetting industry. Lorem Ipsum has been the industry's standard
dummy text ever since the 1500s, when an unknown printer took a galley of type
and scrambled it to make a type specimen book. It has survived not only five
centuries, but also the leap into electronic typesetting, remaining essentially
unchanged. It was popularised in the 1960s with the release of Letraset sheets
containing Lorem Ipsum passages, and more recently with desktop publishing
software like Aldus PageMaker including versions of Lorem Ipsum. </p>

{% endblock %}
```

### Service.html:

```
{% extends 'base.html' %} {% block title %}

Service - MyBlog

{% endblock %}

{% block content %}

<h1>Welcome to Service page</h1>

<p>This is the service page of my blog. Lorem Ipsum is simply dummy text of the
printing and typesetting industry. Lorem Ipsum has been the industry's standard
dummy text ever since the 1500s, when an unknown printer took a galley of type
and scrambled it to make a type specimen book. It has survived not only five
centuries, but also the leap into electronic typesetting, remaining essentially
unchanged. It was popularised in the 1960s with the release of Letraset sheets
containing Lorem Ipsum passages, and more recently with desktop publishing
software like Aldus PageMaker including versions of Lorem Ipsum. </p>

{% endblock %}
```

### App - urls.py file:

```
from django.urls import path

from . import views

urlpatterns = [
```

```
path("", views.home, name='home'),

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

path('contact/', views.contact, name='contact'),

path('service/', views.service, name='service'),

]
```

#### App - views.py file:

```
from django.shortcuts import render

# Create your views here.

def home(request):

    return render(request, 'home.html')


def about(request):

    return render(request, 'about.html')


def contact(request):

    return render(request, 'contact.html')


def service(request):

    return render(request, 'service.html')
```

settings.py file:

```
INSTALLED_APPS = [  
    "django.contrib.admin",  
    "django.contrib.auth",  
    "django.contrib.contenttypes",  
    "django.contrib.sessions",  
    "django.contrib.messages",  
    "django.contrib.staticfiles",  
    'blog',  
]
```

```
TEMPLATES = [  
    {  
        "BACKEND": "django.template.backends.django.DjangoTemplates",  
        "DIRS": [os.path.join(BASE_DIR, 'templates')],  
        "APP_DIRS": True,  
        "OPTIONS": {  
            "context_processors": [  
                "django.template.context_processors.debug",  
                "django.template.context_processors.request",
```



```
        "django.contrib.auth.context_processors.auth",  
        "django.contrib.messages.context_processors.messages",  
    ],  
    },  
    },  
]
```

Project - urls.py file:

```
from django.contrib import admin  
  
from django.urls import path, include  
  
urlpatterns = [  
    path("admin/", admin.site.urls),  
    path("", include("blog.urls")),  
]
```

Output:



