Django Template Inheritance: Schritt-für-Schritt-Anleitung

Einführung

Template-Inheritance in Django erlaubt es, eine Basisstruktur für Webseiten zu definieren, die von mehreren Seiten verwendet werden kann. Dies reduziert Redundanz und erleichtert das konsistente Design.

In diesem Skript erstellen wir eine kleine Anwendung, die eine Hauptvorlage (`base.html`) mit Header, Footer und Navigation enthält. Außerdem zeigen wir, wie Seiten diese Vorlage erweitern können, und wie Teile wiederverwendet werden können.

---

1. Projekt einrichten

Erstelle ein Django-Projekt und eine App:

```
django-admin startproject myproject
cd myproject
python manage.py startapp myapp
```

Registriere die App in `settings.py`:

```
INSTALLED_APPS = [
    ...,
    'myapp',
]
```

Erstelle die Ordnerstruktur für Templates und statische Dateien:

```
myproject/
    myapp/
        templates/
            myapp/
                base.html
                header.html
                footer.html
                home.html
                about.html
        static/
            myapp/
                css/
                    styles.css
```

---

2. Basisvorlage erstellen: `base.html`

Die `base.html` enthält den Header, Footer und einen Platzhalter für den Hauptinhalt.

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>{% block title %}My Website{% endblock %}</title>
    <link rel="stylesheet" href="{% static 'myapp/css/styles.css' %}">
</head>
<body>
    {% include "myapp/header.html" %}
```

```
    <main>
        {% block content %}{% endblock %}
    </main>

    {% include "myapp/footer.html" %}
</body>
</html>
```

---

3. Header erstellen: `header.html`

Der Header enthält eine einfache Navigation.

```
<header>
    <h1>My Website</h1>
    <nav>
        <ul>
            <li><a href="/home/">Home</a></li>
            <li><a href="/about/">About</a></li>
        </ul>
    </nav>
    <hr>
</header>
```

---

4. Footer erstellen: `footer.html`

Der Footer wird ebenfalls eingebunden und kann angepasst werden.

```
<footer>
    <hr>
    {% block footer %}
    <p>© 2024 My Website. All rights reserved.</p>
    {% endblock %}
</footer>
```

---

5. Statische Dateien

Erstelle eine Datei `styles.css` im Ordner `static/myapp/css/`:

```
body {
    font-family: Arial, sans-serif;
    margin: 0;
    padding: 0;
}

header, footer {
    background-color: #f4f4f4;
    padding: 10px;
    text-align: center;
}
```

---

6. Child Templates: `home.html` und `about.html`

`home.html`

```
{% extends "myapp/base.html" %}

{% block title %}Home{% endblock %}

{% block content %}
<h2>Welcome to the Home Page!</h2>
<p>This is the homepage of our site.</p>
{% endblock %}
```

about.html

```
{% extends "myapp/base.html" %}

{% block title %}About{% endblock %}

{% block content %}
<h2>About Us</h2>
<p>We are a company dedicated to providing the best services.</p>
{% endblock %}
```

---

7. Views und URLs

Erstelle View-Funktionen in `views.py`:

```
from django.shortcuts import render

def home(request):
    return render(request, 'myapp/home.html', {})

def about(request):
    return render(request, 'myapp/about.html', {})
```

Konfiguriere die URLs in `urls.py`:

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

urlpatterns = [
    path('home/', views.home, name='home'),
    path('about/', views.about, name='about'),
]
```

---

8. Server starten und testen

Starte den Server und besuche die URLs:

```
python manage.py runserver
```

- **Home Page**: http://127.0.0.1:8000/home/
- **About Page**: http://127.0.0.1:8000/about/

---

9. Erweiterungen: Dynamischer Footer

Füge im `about.html` einen spezifischen Footer hinzu:

```
{% block footer %}
{{ block.super }}
<p>Contact us at info@mywebsite.com.</p>
{% endblock %}
```

Zusammenfassung

1. `base.html`: Definiert die Grundstruktur der Seite.
2. `{% include %}`: Bietet wiederverwendbare Teile (z. B. Header, Footer).
3. `{% block %}`: Ermöglicht die Anpassung von spezifischen Teilen.
4. **Statische Dateien**: Fügen Design und Stil hinzu.

Diese Struktur sorgt für ein konsistentes Design und spart Entwicklungszeit.