SoftWare

Engineering

Eng: Malek Almosanif

Eng: Fadwa Fuad Ali

Django Template Engine :

هي نظام يُستخدم لعرض البيانات بطريقة منظمة ومنسقة في واجهات HTML ويفصل منطق البرمجة (Backend) وتصميم العرض (frontend).

كـــ تعريف ادق:

هو محرك قوالب فب Django يُستخدم لأنشاء صفحات HTML ديناميكية عن طريق دمج ملفات HTML مع البيانات التي يرسلها ال View.

استخداماتة:

* لعرض البيانات القادمة من الـ database على شكل صفحات HTML.
* لفصل منطق البرمجة (Python) عن تنسيق الواجهة (HTML).
* لتكرار عناصر الصفحة (مثل الجداول، القوائم) بطريقة ديناميكية دون كتابة نفس الكود مرارًا.
* لتوفير وسائل تحكم مثل **if**, **for**, **filters**, وغيرها.

كيفية عملة:

* **إنشاء ملف HTML** يحتوي على تعليمات Django (وليس HTML فقط)..
* **View** يرسل البيانات إلى هذا الملف.
* **Template Engine** يدمج البيانات مع ملف HTML ويعرضها للمستخدم.

DTL

* **النوع الافتراضي** في . Django
* خفيف، بسيط، ومصمم ليتكامل مباشرة مع. Django
* يستخدم {% ... %} للتعليمات، و {{ ... }} لعرض البيانات
* يدعم الوراثة بين القوالب، الفلاتر، التكرار، الشروط، وغيرها.
* المفضل للمشاريع اللي تبغين تسويها بسرعة وببساطة..

Jinja2

* محرك قوالب خارجي، لكنه مدعوم رسميًا من Django
* أسرع من DTL في الأداء.
* صُمم ليكون أكثر مرونة ويشبه Python أكثر.
* تنفيذ عمليات أكثر تعقيدًا.
* استخدام هياكل بايثونية داخل القوالب.

Mako

* سريع جدًا ويعتمد على بايثون.
* مرن لكن أقل تكاملًا مع. Django

Chameleon

* يعتمد على XML.
* ممتاز للأداء لكن نادر الاستخدام.

أبرز الاختلافات

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| نوع المحرك | سرعته | مدى السهولة | تكاملة مع Django | مميزاتة | |  |
| DTL | جيدة | سهل للغاية | مدمج معه | جيد للفصل بين الواجهة والمنطق | الافتراضي. | |
| Jinja2 | اسرع من Dtl | متوسط السهولة. | مدعوم | أقرب لبايثون، فلاتر قوية | يستخدم في Flask | |
| Mako | سريع جداأ | معقد. | يحتاج تكامال يدوي | مرن جدًا وقوي | مناسب لمشاريع الأداء العالي | |
| Chameleon | سريع جداً | لايعتبر بسيط لانة يعتمد على XML | تكاملة متوسط | ممتاز في الأداء | يُستخدم في بعض أنظمة CMS | |
| Cheetah | بطيء | صعب وقديم . | لايدعمة رسمياً | يشبه أسلوب البرمجة الكلاسيكية | غير مفضل حالياً | |

فلاتر القوالب في

**Django Template Language (DTL)**

هي أدوات قوية تساعد على **تنسيق البيانات أو تعديلها داخل القالب** قبل عرضها للمستخدم، وكل هذا بدون كتابة كود بايثون في الواجهة.

أنواع الفلاتر في DTL

فلاتر التنسيق

**تنسيق النصوص أو القيم.**

فلاتر القوائم والسلاسل

**تعديل أو معالجة القوائم أو السلاسل النصية.**

فلاتر الشرط والتعامل مع القيم

**إعطاء قيم بديلة أو فحص القيم.**

فلاتر التاريخ والزمن

**تنسيق التواريخ والأوقات**

فلاتر الأمان والعرض

**التعامل مع HTML والنصوص البرمجية.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| |  | | --- | | نوع الفلتر |  |  | | --- | |  | | **يفيد في** | ــــ |
| التنسيق | جعل المخرجات مرتبة للمستخدم | سريع وبسيط |
| القوائم والسلاسل | التعامل مع بيانات متكررة | |  | | --- | |  |   ممتاز للجداول والقوائم |
| القيم الشرطية | تقديم بدائل ذكية عند نقص البيانات. | مفيد في التعامل مع البيانات غير المكتملة |
| التاريخ والزمن | |  | | --- | | تنسيق مخرجات التواريخ بشكل أنيق |  |  | | --- | |  | | |  | | --- | |  |   مفيد للمدونات والمواقع الزمنية |
| |  | | --- | | الأمان والعرض |  |  | | --- | |  | | حماية او تزيين النصوص البرمجية.   |  | | --- | |  | | مهم جدًا في تطبيقات HTML وMarkdown |

فلتري الخاص

* فلتر يقوم بعمل "\_" ممكان كل فراغ.
* فلتر يقوم بتلوين اسم " الله " باللون الأحمر.

كما درسنا سابقا في انشاء المشاريع

* انشاء مشروع جديد"fadwaspro2 " فية تطبيق جديد.
* اضافه او تسجيل التطبيق fadwa فيsettings.py
* عمل url باسم urls.py
* عمل ملف للتطبيق با اسم urls.py واضافه اول رابط
* ضبط ال views

انشاء الفلتر

* انشاء ملف جديد في التطبيق ب اسم templatetags
* اضافه الفلاتر الي نصنعها داخل مجلد templatetagsب اسم "fadwa\_filter1" "fadwa\_filter2"

ب داخل fadwa\_filter1

from django import template

import re

()register = template.Library

@register.filter(name='fadwa\_filter1'))

Def f1(value):

if isinstance(value, str):

return value.replace(' ','\_')

return value

ب داخل fadwa\_filter2

from django import template

import re

()register = template.Library

@register.filter(name='fadwa\_filter1'))

Def f1(value):

if isinstance(value, str):

return re.sub(r'\d', '\*',value)

return value

ب داخل fadwa\_filter3

from django import template

import re

()register = template.Library

@register.filter(name='fadwa\_filter1'))

Def f1(value):

if isinstance(value, str):

return re.sub(r'()',r'<span style=color:red;">\1</span>',value)

return value

* نحمل الفلتر بداخل القالب
* {% load fadwa\_filter1 %}

استخدام الفلاتر

استخدام الفلاتر على البيانات

<p>{["nice filter"|fadwa\_filter1</p>

<p>{["nice filter1 and2"|fadwa\_filter2</p>

و لله الحمد والشكر..