

مقارنة شاملة لمحرركات القوالب في Django

استكشاف الخيارات الأمثل لمطوري الويب ومهندسي البرمجيات

إعداد الطالب:

م/ محمد فهد علي محمد

جدول مقارنة DTL vs Jinja2 vs Mako vs Cheetah :

الخاصية	Django Template (DTL)	Jinja2	Mako	Cheetah
التكامل مع Django	مدمج بالكامل - سهل الضبط	مدعوم مع إعداد بسيطة	قابل للاستخدام لكن أقل شيوعاً	غير مدمج عادةً - أكثر تعقيداً
الأداء	أبطأ نسبياً	أسرع بـ 5x-2 بحسب الاختبارات	أداء سريع بدرجة مشابهة لـ Jinja2	لا توجد بيانات حديثة واضحة على الأداء
السلوك بالأخطاء والمتغيرات غير الموجودة	يتجاهل المتغيرات غير المعرفة - بدون أخطاء	أحكام واضحة وسلوكية أكثر برمجية	يسمح بتنفيذ كود Python داخلي - أكثر مرونة	يحول الـ templates إلى كلاس Python - صعب التصحيح
سهولة القواعد والتركيب	بسيط وواضح، صُمم للمطور والمصمم	مرن، يدعم الـ filters والإنجكشن بسهولة	مرن جداً، لكنه معقد التعلّم حسب بعض المصادر	بناءً على الكود المولد - قد يؤدي إلى مشاكل تصحيحية
الدعم المجتمعي	مستخدم من قبل 94% من المطورين في Django	شائع جداً ومستخدم في Flask وغيره	مستقر، مدعوم من Pyramid وReddit	قديم ومعقد، موثق أقل
قابلية الاستخدام	جيد لبساطة الأعمال ووضوح الإخراج	رائع للمشاريع الكبيرة أو حيث الأداء مهم	مثالي لو أردت تحكماً كاملاً بلغة القالب	أقل استخداماً - قد لا يوصى به للمشاريع الحديثة

مقتطفات من المجتمع

"أنا شخصياً أفضل
Jinja2 على Mako، أمثلة
Jinja2 أكثر
وضوحاً بكثير".

—مستخدم من StackOverflow

"قوالب Django
مثل Django
نفسه... أنا ألتزم
بمحرك القوالب
الافتراضي لأنه يقلل
من التعقيدات".

—مستخدم من Reddit

التوصيات بناءً على احتياجاتك

1

للمشاريع التقليدية في Django

استخدم **DTL** - لسهولة، دعمه القوي، وموثوقيته. إنه الخيار الأمثل للمشاريع القياسية.

3

للمشاريع متقدمة جداً أو تحتاج تحكم كبير داخل الـ **template**

Mako قد يكون الأفضل، رغم صعوبة التعلم النسبية، لأنه يوفر تحكماً كاملاً بلغة القالب.

2

إذا كنت تحتاج أداء أعلى أو مرونة أكثر

جرب **Jinja2** - فهو أسرع بكثير وأكثر قابلية للبرمجة، مما يجعله مثالياً للتطبيقات الكبيرة.

4

متى تستخدم **Cheetah**؟

Cheetah مناسب فقط إذا كنت تعمل على مشاريع قديمة أو لديك حاجة خاصة لخيار مخصص، لكن بشكل عام لا يُنصح به للمشاريع الحديثة.