



## التكليف الثالث مقرر هندسة برمجيات (عملي)

مقدم من الطالب :  
محمد نشوان محمد مصلح الفقيه

كجزء من متطلبات المقرر

مدرس المقرر  
أ. مالك المصنف

## ■ أولاً : قوالب Django الخارجية

○ تعريف ؟ هي مكتبات خارجية يمكن استخدامها بدلاً من نظام القوالب الرسمي في (DTL) Django لكتابة صفحات HTML باستخدام وسوم وقواعد مخصصة، وتتيح مرونة أكبر أو أداء أعلى.

توجد العديد من القوالب ولاكن فيما يلي مقارنة لأشهر القوالب:

المعيار	DTL	Jinja2	Mako	Chameleon	Genshi
النوع	مدمج مع Django	خارجي (يستخدم مع Flask)	خارجي	خارجي	خارجي
الأداء	متوسط	عالي جداً	عالي جداً	عالي	جيد
الدعم الرسمي في Django	مدعوم رسمياً	مدعوم رسمياً	مدعوم رسمياً	مدعوم رسمياً	غير مدعوم رسمياً
سهولة الاستخدام	سهل جداً	سهل نسبياً	متوسط	سهل	متوسط
إمكانية كتابة كود Python داخل القالب	غير ممكن	غير ممكن	ممكن	غير ممكن	غير ممكن
الأمان التلقائي ضد XSS	نعم	نعم	لا (يحتاج إعداد يدوي)	نعم	نعم
قوة التعامل مع المتغيرات	محدودة	قوية	قوية جداً	جيدة	جيدة
دعم وراثة القوالب (Template Inheritance)	نعم	نعم	نعم	نعم	نعم
دعم HTML5 و XML	نعم	نعم	نعم	نعم	نعم (قوي جداً)
إمكانية التخصيص بفلاتر خاصة	نعم	نعم	نعم	نعم	نعم
التوثيق والدعم المجتمعي	ممتاز	ممتاز	متوسط	محدود	محدود

## ■ ثانياً : أنواع الفاتر في Django

○ الفلاتر (Template Filters) : هي أدوات تُستخدم داخل ملفات القوالب (templates) لمعالجة البيانات وعرضها بطريقة منسقة .

ويمكن تقسيمها الى ثلاثة أنواع :

1- الفلاتر الافتراضية: وهي الفلاتر التي يوفرها Django تلقائياً، ويمكنك استخدامها دون الحاجة إلى تعريفها.

الفلتر	الاستخدام
date	تنسيق التاريخ
length	يحسب عدد العناصر في القائمة أو طول النص
lower	يحوّل النص إلى أحرف صغيرة
upper	يحوّل النص إلى أحرف كبيرة
truncatewords	يقطع النص بعد عدد معين من الكلمات
default	يعطي قيمة افتراضية إذا كان المتغير فارغاً أو None
safe	يعرض النص كـ HTML بدون تطهير

```
<!DOCTYPE html>
<html lang="ar">
<head>
  <meta charset="UTF-8">
  <title>اختبار فلاتر Django</title>
</head>
<body dir="rtl">
  <h1>اختبار الفلاتر</h1>

  <p>• تاريخ الانضمام (date): {{ user.date_joined|date:"Y-m-d" }}</p>

  <p>• عدد الرسائل (length): {{ message_list|length }}</p>

  <p>• الاسم بأحرف صغيرة (lower): {{ name|lower }}</p>

  <p>• الاسم بأحرف كبيرة (upper): {{ name|upper }}</p>

  <p>• وصف مختصر (truncatewords): {{ desc|truncatewords:10 }}</p>

  <p>• اسم المستخدم الافتراضي (default): {{ username|default:"زائر" }}</p>

  <p>• غير مصفى HTML نص (safe): {{ html_text|safe }}</p>

</body>
</html>
```

## اختبار الفلاتر

- تاريخ الانضمام (date): 2025-08-06
- عدد الرسائل (length): 3
- الاسم بأحرف صغيرة (lower): mohamed
- الاسم بأحرف كبيرة (upper): MOHAMED
- وصف مختصر (truncatewords): هذا مثال على وصف طويل نريد قطعه بعد عدد معين ...
- اسم المستخدم الافتراضي (default): زائر
- نص HTML غير مصفًى (safe): مرحبا بك! هذا نص منسق بـ HTML

## 2- الفلاتر المخصصة (Custom Filters)

يمكننا انشاء فلاتر تتناسب مع الحاجة اذا لم تكن موجوده .

### خطوات إنشاء فلتر مخصص:

- أنشئ ملفاً باسم `templatetags` داخل تطبيقك وتأكد من إضافة ملف باسم `__init__.py` حتى يتم التعرف على المجلد ك حزمة (Package) .
- قم بإنشاء ملف بامتداد `.py` وإنشاء بداخله دالة تحمل اسم الفلتر وعمله .
- ثم في المكان الذي تريد ان تستخدمه فيه يجب عليك ان تضمنه أولاً باستخدام هذا الامر `{% load اسم الملف %}`

## 3- الفلاتر من مكتبات خارجية (Third-party filters)

بعض الحزم الإضافية توفر فلاتر جديدة مثل:

الْحِزْمَة	فلاتر توفرها
django-widget-tweaks	فلاتر لتخصيص الـ form fields بسهولة
django-humanize	فلاتر لجعل الأرقام والتواريخ "إنسانية" مثل "منذ 3 أيام"
django-mathfilters	فلاتر لإجراء عمليات رياضية مثل add, subtract

## ■ ثالثاً: انشاء فلتر خاص بي

```
from django import template
from datetime import datetime

register = template.Library()

# فلتر لعكس محتوى مصفوفة نصية
@register.filter(name='reverse_string')
def reverse_string(value):
    return value[::-1]

# * فلتر ليخفي جزء من اليمين بعلامة النجمة
@register.filter(name='mask_middle')
def mask_middle(value):
    if not isinstance(value, str) or len(value) < 5:
        return value
    middle = len(value) // 2
    return value[:2] + '*' * (middle - 2) + value[middle:]

# يميز كلمة بداخل النص باللون الاحمر
@register.filter(name='highlight_word')
def highlight_word(text, word):
    if not isinstance(text, str) or not isinstance(word, str):
        return text
    return text.replace(word, f'<span style="color:red">{word}</span>')
```

## صورة التنفيذ للفلاتر الخاصة

### اختبار الفلاتر المخصصة

- عكس النص: sretliF ognajD
- إخفاء الإيميل: mo\*\*\*\*\*xample.com
- تمييز كلمة داخل النص:
- مرحباً بك يا **صالح** في عالم Django