



م ٤-ت ١ مجموعة (B2)

الاسم/ ضرار علي محمد عقلان

م/مالك المصنف

تكليف رقم (٣) مقرر: هندسة برمجيات

## 1. مقارنة بين أربعة django template :

### المقدمة

أنظمة القوالب هي أدوات حيوية في أي إطار عمل ويب مثل Django. وظيفتها الأساسية هي فصل منطق التطبيق (البيانات) عن العرض (HTML) ، مما يسهل على المطورين والمصممين العمل بشكل متزامن وفعال. على الرغم من أن Django يأتي مع نظام قوالب خاص به (DTL) ، إلا أن مرونته تسمح بدمج أنظمة قوالب أخرى لتلبية احتياجات معينة.

## 1.1 Django Template Language (DTL)

نظرة عامة :هو نظام القوالب الافتراضي الذي يأتي مع Django. تم تصميمه ليكون بسيطًا وآمنًا، مما يجعله خيارًا ممتازًا للمبتدئين وللمشاريع التي لا تتطلب منطقًا معقدًا داخل القوالب. فلسفته الأساسية هي منع تنفيذ شيفرة بايثون عشوائية داخل القوالب.

### المزايا:

- التكامل الأصيل: مدمج بالكامل مع Django ، مما يسهل عملية الإعداد والاستخدام. لا يتطلب تثبيت حزم إضافية.
- البساطة والأمان: لغته بسيطة ومحدودة، مما يقلل من احتمالية الأخطاء الأمنية أو المنطقية داخل القوالب. لا يمكن استدعاء دوال عشوائية.
- سهولة التعلم: قواعده واضحة ومباشرة (مثل `{{ var }}` و `{% tag %}`).
- الميزات الأساسية: يدعم التضمين (include) ، والتوريث (extends) و (block) والفلاتر (filters) ، وكلها كافية لمعظم التطبيقات.

### العيوب:

- محدودية المنطق: يفرض قيودًا صارمة على المنطق الذي يمكن وضعه داخل القوالب. لا يدعم الدوال المعقدة أو العمليات الحسابية المتقدمة.

- عدم المرونة: قد يكون صعبًا في بعض الأحيان عند الحاجة إلى معالجة بيانات معقدة أو استخدام بنى بيانات غير تقليدية.

- الأداء: في بعض الحالات، قد يكون أداؤه أبطأ من Jinja2 ، خاصة في المشاريع الكبيرة التي تحتوي على قوالب معقدة.

متى تستخدمه:

- للمشاريع الصغيرة والمتوسطة.
- للمطورين الجدد على Django.
- عندما تكون البساطة والأمان أولوية قصوى.

---

## Jinja2.2

نظرة عامة Jinja2 : هو نظام قوالب قوي ومشهور جدًا، مستوحى من DTL لكنه أكثر قوة ومرونة. تم تطويره من قبل فريق Flask ، وهو معيار الصناعة في العديد من مشاريع بايثون. يمكن دمجه بسهولة مع Django.

المزايا:

- قوة ومرونة: يدعم تنفيذ شيفرة بايثون داخل القوالب، مما يسمح بمنطق أكثر تعقيدًا (مثل الدوال، الحلقات، واستخدام مكتبات بايثون).
- أداء ممتاز: يعتبر أسرع بشكل ملحوظ من DTL ، حيث يقوم بتجميع القوالب إلى شيفرة بايثون (bytecode).
- إعادة الاستخدام: يدعم الماكرو (macro) ، وهي ميزة قوية تسمح بإنشاء مكونات HTML قابلة لإعادة الاستخدام.
- سهولة التكامل: يمكن دمجه مع Django بسهولة باستخدام حزم مثل django-jinja أو عن طريق الإعدادات المدمجة في Django 1.8 فما فوق.
- التوثيق: لديه توثيق شامل ومجتمع كبير.

العيوب:

- مخاوف أمنية محتملة: مرونته قد تشكل خطرًا إذا لم يتم التعامل معها بحذر. وضع منطق معقد في القوالب قد يؤدي إلى ثغرات أمنية.
- منحنى تعلم أعلى: على الرغم من تشابهه مع DTL ، إلا أن خصائصه المتقدمة تتطلب وقتًا أطول للتعلم.

متى تستخدمه:

- للمشاريع الكبيرة والمعقدة.
- عندما يكون الأداء أمرًا حاسمًا.
- عندما تحتاج إلى منطق قوي داخل القوالب (مثل الماكرو).

---

### Mako.3

نظرة عامة Mako: هو نظام قوالب آخر قوي ومشهور في عالم بايثون، يتميز بسرعته وقوته الفائقة. فلسفته الرئيسية هي إتاحة تنفيذ شيفرة بايثون غير مقيدة داخل القوالب، مما يجعله أكثر مرونة من Jinja2.

المزايا:

- أقصى قدر من القوة: يسمح بتنفيذ شيفرة بايثون عشوائية بالكامل (باستخدام `<% ... %>` و `<%doc ... %>`)، مما يجعله أقرب إلى البرمجة التقليدية.
- أداء فائق: يعتبر من أسرع أنظمة القوالب المتاحة لبايثون، حيث يقوم بتجميع القوالب إلى وحدات بايثون.
- مرونة في الأسلوب: يدعم أساليب مختلفة لكتابة القوالب، مثل استخدام الكتل، أو ببساطة استخدام شيفرة بايثون.

العيوب:

- مخاطر أمنية كبيرة: مرونته المطلقة تجعله عرضة بشكل كبير للثغرات الأمنية إذا لم يتم استخدامه بحذر شديد. وضع منطق معقد في القوالب قد يكون فحًا للمطورين.
- صعوبة في الصيانة: قد يصبح من الصعب إدارة وصيانة القوالب المكتوبة باستخدام Mako إذا كانت تحتوي على الكثير من شيفرة بايثون.
- منحني تعلم حاد: يتطلب فهمًا عميقًا لأسلوب Mako وقيوده.

متى تستخدمه:

- للمشاريع التي تتطلب أداءً فائقًا ومنطقًا معقدًا جدًا داخل القوالب.
- عندما تكون على دراية كاملة بالمخاطر الأمنية ولديك القدرة على إدارتها.
- في بيئة موثوقة بالكامل.

---

### Mustache.4

نظرة عامة **Mustache**: يختلف عن الأنظمة السابقة. هو نظام قوالب "logic-less" ، بمعنى أنه لا يسمح بأي منطق برمجي (مثل if أو for) داخل القوالب. وظيفته الوحيدة هي عرض البيانات. يتميز بتعددته (يمكن استخدامه مع لغات برمجة مختلفة).

#### المزايا:

- فصل كامل: يفرض فصلاً صارماً بين العرض والمنطق، مما يجعله مثالياً لفرق العمل التي تضم مصممين ومطورين.
- محمولية عالية: يمكن استخدام نفس القوالب مع لغات برمجة مختلفة (بايثون، جافا سكريبت، روبي، إلخ)، مما يسهل بناء تطبيقات متعددة المنصات.
- بساطة وأمان: لا يحتوي على منطق برمجي، مما يجعله آمناً جداً وسهل الفهم.

#### العيوب:

- محدودية المنطق: افتقاره إلى المنطق يجعله غير مناسب للحالات التي تتطلب عرضاً مشروطاً أو حلقات.
- صعوبة في التعبير: قد تحتاج إلى تحضير البيانات بشكل مسبق في Python قبل تمريرها إلى القالب، مما قد يزيد من تعقيد المنطق في جزء العرض.
- غير مناسب للحالات المعقدة: ليس الخيار الأفضل لتطبيقات الويب الديناميكية التي تتطلب تفاعلاً كبيراً مع البيانات.

#### متى تستخدمه:

- لإنشاء واجهات مستخدم ثابتة أو بسيطة.
- في بيئة متعددة المنصات حيث تحتاج إلى مشاركة القوالب.
- عندما يكون الأمان والفصل بين العرض والمنطق أولوية قصوى.

#### جدول المقارنة:

الخاصية	DTL	Jinja2	Mako	Mustache
المرونة والمنطق	محدودة (منطق أساسي)	عالية (منطق بايثون)	فائقة (بايثون غير مقيد)	لا يوجد منطق (Logic-less)
الأداء	جيد	ممتاز	ممتاز جداً	جيد
سهولة التعلم	عالي	متوسط	منخفض	عالي
الأمان	عالي جداً	جيد (يعتمد على الاستخدام)	منخفض (خطير)	عالي جداً

التكامل مع Django	أصيل ومباشر	سهل (عبر django-jinja)	يتطلب حزم إضافية	يتطلب حزم إضافية
أفضل استخدام	مشاريع بسيطة ومتوسطة	مشاريع كبيرة ومعقدة	تطبيقات عالية الأداء	قوالب بسيطة ومتعددة المنصات

## 2. أنواع الفلاتر dtl من حيث الاسم الوظيفة طريقة الكتابة:

### 1. فلاتر التنسيق (Formatting Filters)

اسم الفلتر	الوظيفة	طريقة الكتابة	مثال
Date	تنسيق كائن datetime أو date إلى سلسلة نصية بالشكل المحدد.	{{ value date:"F j, Y" }}	إذا كان value هو datetime.datetime(2025, 8, 5), سيكون الناتج : August 5, 2025
Time	تنسيق كائن datetime أو time إلى سلسلة نصية بالشكل المحدد.	{{ value time:"P" }}	إذا كان value هو datetime.datetime(2025, 8, 5, 13, 11), سيكون الناتج : 1:11 p.m.
Floatformat	تنسيق رقم عشري مع تحديد عدد الأرقام العشرية.	{{ value floatformat:2 }}	إذا كان value هو 3.1415926, سيكون الناتج : 3.14
default	يوفر قيمة افتراضية إذا كان المتغير فارغاً (None, "", []).	{{ value default:" " }}	إذا كان value فارغاً، سيكون الناتج : لم يتم التحديد.

### 2. فلاتر القوائم (List Filters)

اسم الفلتر	الوظيفة	طريقة الكتابة	مثال
Join	ضم عناصر القائمة إلى سلسلة نصية واحدة باستخدام فاصل محدد.	{{ value join:"" }}	إذا كان value هو ['apple', 'orange', 'banana'], سيكون الناتج : "apple, orange, banana".
First	إرجاع العنصر الأول من القائمة.	{{ value first }}	إذا كان value هو ['a', 'b', 'c'], سيكون الناتج : 'a'.
Last	إرجاع العنصر الأخير من القائمة.	{{ value last }}	إذا كان value هو ['a', 'b', 'c'], سيكون الناتج : 'c'.
random	إرجاع عنصر عشوائي من القائمة.	{{ value random }}	إذا كان value هو ['a', 'b', 'c'], قد يكون الناتج 'a', 'b', أو 'c'.

### 3. فلاتر HTML (HTML Filters)

اسم الفلتر	الوظيفة	طريقة الكتابة	مثال
Safe	يخبر Django أن المتغير آمن للعرض ولا	{{ value safe }}	إذا كان value هو "عنوان", </h1> سيتم عرض <h1>عنوان </h1> ك

يجب إجراء Escaping لشفرة HTML فيه . استخدم بحذر!		HTML بدون الفلتر، سيتم عرضه كنص عادي.
Escape	يقوم بتحويل أحرف HTML الخاصة (مثل <, >, &) إلى رموز HTML (entities).	إذا كان value هو <h1>"عنوان", </h1> سيكون الناتج : &lt;/h1&gt;عنوان&lt;h1&gt;
striptags	يزيل جميع علامات HTML من النص.	إذا كان value هو <h1>"عنوان", </h1> سيكون الناتج : "عنوان."

#### 4. فلتر رقمية (Numeric Filters)

اسم الفلتر	الوظيفة	طريقة الكتابة	مثال
Add	يضيف قيمة إلى المتغير. يمكن أن تكون القيمة رقمًا أو سلسلة نصية.	{{ value add:5 }}	إذا كان value هو 10، سيكون الناتج 15.
Divisibleby	يعيد True إذا كان المتغير قابلاً للقسمة على القيمة المحددة.	{% if value divisibleby:3 %}	إذا كان value هو 9، فسيكون الشرط صحيحًا.
filesizeformat	تنسيق حجم الملف بشكل مقروء (مثل "١٣ KB" أو "٤ MB").	{{ value filesizeformat }}	إذا كان value هو 1024، سيكون الناتج 1 KB.

#### 5. فلتر المنطق والترتيب (Logic & Ordering Filters)

اسم الفلتر	الوظيفة	طريقة الكتابة	مثال
Dictsor	فرز قائمة من القواميس بناءً على مفتاح محدد.	{% for item in list_of_dicts dictsort:"name" %}	إذا كانت القائمة تحتوي على قواميس بها مفتاح name، سيتم ترتيبها أبجديًا بناءً عليه.
default_if_none	يوفر قيمة افتراضية إذا كان المتغير None على وجه التحديد.	{{ value default_if_none:"لا يوجد" }}	إذا كان value هو None، سيكون الناتج : default (لا يوجد). يختلف عن default الذي يعمل على القيم الفارغة أيضًا).
pluralize	إضافة حرف الجمع (مثل 's' أو سلسلة نصية أخرى إلى نهاية الكلمة بناءً على قيمة رقمية).	{{ number pluralize }}	إذا كان number هو 1، سيكون الناتج : 1 كتاب. إذا كان number هو 2، سيكون الناتج : 2 كتابs.

## 6. فلاتر متفرقة (Miscellaneous Filters)

اسم الفلتر	الوظيفة	طريقة الكتابة	مثال
Linebreaks	يحول فواصل الأسطر في النص إلى وسم <code>&lt;br /&gt;</code> في HTML.	<code>{{ value   linebreaks }}</code>	إذا كان <code>value</code> يحتوي على أسطر جديدة، سيتم تنسيقه في فقرات HTML.
Urlize	يحول الروابط النصية إلى روابط HTML قابلة للنقر.	<code>{{ value   urlize }}</code>	إذا كان <code>value</code> هو "زوروا <code>https://www.example.com</code> ", سيتم تحويله إلى وسم <code>&lt;a&gt;</code> .
slice	يقطع (ياخذ جزءاً) من قائمة أو سلسلة نصية باستخدام نفس قواعد slicing في بايثون.	<code>{{ my_list   slice:" :2" }}</code>	يأخذ أول عنصرين من <code>my_list</code> .

## 3. إضافة فلتر الى تنفيذ المحاضرة:

صورة الكود

```
File Edit Selection View Go ... < > SchoolManagement
```

EXPLORER

- SCHOOLMANAGEMENT
  - SchoolManagement
  - Students
    - \_\_pycache\_\_
    - migrations
    - templates
      - deletestudents.html
      - editstudents.html
      - home.html
      - index.html
      - showstudents.html
    - \_\_init\_\_.py
    - admin.py
    - apps.py
    - models.py
    - tests.py
    - urls.py
    - views.py
    - db.sqlite3
    - manage.py

Students > templates > index.html > html > body > h3

```
2 <html lang="en">
7 </head>
8 <body>
9   <h1>Hello For All Student</h1>
10  <h3>Your Name {{fname | default:'aqlan'}}</h3>
11  <h3>Your Name {{fname | upper}}</h3>
12  <h3>Your Name {{fname | lower}}</h3>
13  <h3>Your Name {{fname | capfirst}}</h3>
14  <h3>Your Name {{fname | length}}</h3>
15  <h3>Your Name {{fname | slice:2}}</h3>
16  <h3>Your Name {{fname | add:' aqlan'}}</h3>
17  # الفلاتر المضافة
18  # يزيل المسافة
19  <h3>Your Name {{fname | cut:" "}}</h3>
20  <h3>Your Name {{fname | join:"-"}}</h3>
21  <h3>Your Name {{fname | random}}</h3>
22  <h3>Your Name {{fname | last}}</h3>
23
24 </body>
25 </html>
```

صفحة التنفيذ:



# **Hello For All Student**

**Your Name Dherar Aqlan**

**Your Name DHERAR AQLAN**

**Your Name dherar aqlan**

**Your Name Dherar Aqlan**

**Your Name 12**

**Your Name Dh**

**Your Name Dherar Aqlan aqlan**

الفلانتر المضافة # يزبل المسافة #

**Your Name DherarAqlan**

**Your Name D-h-e-r-a-r- -A-q-l-a-n**

**Your Name e**

**Your Name n**

تم بحمد الله تعالى