

Investigación sobre los sistemas de plantillas en Django

Sistema básico de Plantillas

Una plantilla de Django es una cadena de texto que pretende separar la presentación de un documento de sus datos. Una plantilla define rellenos y diversos bits de lógica básica (esto es, etiquetas de plantillas) que regulan cómo debe ser mostrado el documento. Normalmente, las plantillas son usadas para producir HTML, pero las plantillas de Django son igualmente capaces de generar cualquier formato basado en texto.

Esta es la forma básica, en la que podemos usar el sistema de plantillas de Django en código Python.

```
>>> from __future__ import print_function
>>> from django import template
>>> t = template.Template('Mi nombre es {{ nombre }}.')
>>> c = template.Context({'nombre': 'Adrian'})
>>> print (t.render(c))
Mi nombre es Adrian.
```

1. Crea un objeto Template pasándole el código en crudo de la plantilla como una cadena.
2. Llama al método render() del objeto Template con un conjunto de variables (o sea, el contexto). Este retorna una plantilla totalmente renderizada como una cadena de caracteres, con todas las variables y etiquetas de bloques evaluadas de acuerdo al contexto.

```
<html>
<head><title>Orden de pedido</title></head>
<body>

<h1>Orden de pedido</h1>

<p>Estimado: {{ nombre }},</p>

<p>Gracias por el pedido que ordeno de la {{ empresa }}.
El pedido junto con la mercancía se enviarán el
{{ fecha|date:"F j, Y" }}.</p>

<p>Esta es la lista de productos que usted ordeno:</p>

<ul>
{% for pedido in lista_pedido %}
  <li>{{ pedido }}</li>
{% endfor %}
</ul>

{% if garantía %}
  <p>La garantía será incluida en el paquete.</p>
{% else %}
  <p>Lamentablemente no ordeno una garantía, por lo
que los daños al producto corren por su cuenta.</p>
{% endif %}

<p>Sinceramente <br /> {{ empresa }}</p>
</body>
</html>
```

Esta plantilla es un archivo HTML básico con algunas variables y etiquetas de plantillas agregadas.

- ✓ Cualquier texto encerrado por un par de llaves (por ej. {{ nombre }}) es una variable. Esto significa “insertar el valor de la variable a la que se dio ese nombre”.
- ✓ Cualquier texto que esté rodeado por llaves y signos de porcentaje (por ej. {% if garantía %}) es una etiqueta de plantilla. La definición de etiqueta es bastante amplia: una etiqueta sólo le indica al sistema de plantilla “haz algo”.
- ✓ Este ejemplo de plantilla contiene dos etiquetas: la etiqueta {% for pedido in lista_pedido %} (una etiqueta for) y la etiqueta {% if garantía %} (una etiqueta if).
- ✓ Finalmente, el segundo párrafo de esta plantilla, muestra un ejemplo de un filtro, con el cual puedes alterar la exposición de una variable. En este ejemplo, {{ fecha|date:"F j, Y" }}, estamos pasando la variable fecha por el filtro date, pasando los argumentos "F j, Y" al filtro. El filtro date formatea fechas en el formato dado, especificado por ese argumento.

Creación de Objetos Template

La manera sencilla de crear objetos Template es instanciarlos directamente. La clase Template se encuentra en el módulo django.template, y el constructor toma un argumento, el código en crudo de la plantilla.

Renderizar una plantilla

Una vez que tienes un objeto Template, le puedes pasar datos brindándole un contexto. Un contexto es simplemente un conjunto de variables y sus valores asociados. Una plantilla usa estas variables para llenar y evaluar estas etiquetas de bloque.

Un contexto es representado en Django por la clase Context, ésta se encuentra en el módulo django.template. Su constructor toma un argumento opcional: un diccionario que mapea nombres de variables con valores. Llama al método render() del objeto Template con el contexto para “llenar” la plantilla:

```
>>> from __future__ import print_function
>>> from django.template import Context, Template
>>> t = Template("Mi nombre es {{ nombre }}.")
>>> c = Context({"nombre": "Estefanía"})
>>> t.render(c)
'Mi nombre es Estefanía.'
```

Bibliografía

García M., S. (2015). *La Guía definitiva de Django*. Celayita México: Django Software Corporation.