

# Templates

---

## ¿Cómo usar un template en Django?

Anteriormente habíamos hablado del concepto de plantillas en Django. Ahora veamos paso a paso cómo utilizarlas. Para empezar deberás importar un modelo de Django en tu vista con `render_to_response` que será el encargado de recibir el *template* y los datos que debe enviarle al mismo.

Pasemos a la forma en que debes especificarle a tu vista cuál *template* usar y cuáles datos *renderizar*. Supon que tienes una vista “hola” :

```
from django.shortcuts import render_to_response

def hola(request):
    mejor= "Diseno Web"
    return render_to_response (template.html, {'nombre':'Daniel
Zavala', 'edad':22, 'cursos': [mejor, 'frontend', 'backend',
'estrategia']})
```

## Agrega los datos a tu plantilla

Cuando deseas agregar un dato de tu *view* a tu *template* (que es un archivo html normal) lo pones entre llaves:

```
{{ nombre }}
```

“Nombre” es el identificador del dato que manda la vista. En este caso en la vista te aparecería “Daniel Zavala” en vez del texto `{{ nombre }}`; igual pasaría si pones `{{ edad }}`. Si quieres hacer un condicional basta poner:

```
{% if edad > 18 %}
    <span> {{ edad }} años <span>
{% elif edad == 18 %}
    <span> Tienes 18 años! </span>
{% else %}
    <span> eres menor de edad según la ley europea </span>
{% endif %}
```

Este código verificará si la edad es mayor a 18 y te dirá la edad. En caso de ser igual a 18 te dará un mensaje, y en caso de ser negativo te dará otro diferente. Por último, si quisieras mostrar los cursos que tiene debes tener en cuenta que es una lista, así que crearemos en ella como en un **for**.

```
{% for i in cursos %}
    <span> {{ i }} </span>
{% endfor %}
```

En este caso i tomará el valor de cada curso y lo mostrará en la plantilla hasta recorrer todos los cursos.

## Filtros

En Django existen múltiples filtros por defecto que automáticamente se da formato a las variables dentro de los *templates*:

**Convierte el texto en minúsculas.**

```
{{ nombre | lower }}
```

**Les da formato día/mes/año a tus fechas.**

```
{{ fecha | date:"d M Y" }}
```

**Retorna el tiempo que ha pasado desde.**

```
{{ timestamp | timesince }}
```

**Retorna el tamaño del valor.**

```
{{ value | length }}
```

**Retorna el plural de un sufijo si el valor no es 1. Por defecto el sufijo es s', pero este se puede alterar.**

```
{{ value | pluralize }}
```

**Ejemplo:**

```
{{ value | pluralize: "es" }}
```

Django tiene muchos *tags* y *filters* más.

## Plantilla base

En Django puedes usar plantillas base a las que insertar bloques personalizados de código. Primero debes crear una plantilla (no influye el nombre) **base.html** en donde pondrás todo el html. Si quieres poner partes en donde el código puede variar, como en el *body*, usarás bloques de esta forma:

```
{% block cuerpo %}  
{% endblock %}
```

Luego en tu plantilla **nueva.html** debes importar la base así:

```
{% extends 'base.html' %}
```

Después en **nueva.html** llamas al bloque **cuerpo** y dentro de él pones el html que quieras:

```
{% block cuerpo %}  
<h1> mucho html </h1>  
{% endblock %}
```