



REGIONAL ANTIOQUIA  
CENTRO DE SERVICIOS Y GESTION EMPRESARIAL  
TECNOLOGIA EN ANALISIS Y DESARROLLO DE SOFTWARE  
**Programación Orientada Objetos (Python + Django)**

## **INCRUSTAR FUNCIONALIDAD PYTHON EN HTML**

Incrustar Python directamente en HTML no es una buena práctica ni está permitido en las plantillas Django por razones de seguridad y separación de responsabilidades. En lugar de eso, Django utiliza su sistema de templates (plantillas) que permite insertar lógica de presentación limitada (como bucles, condicionales, y filtros), pero no código Python puro.

Aquí tienes algunos tips clave para incrustar funcionalidad Python en HTML usando Django Templates correctamente:

### **1. Mostrar variables con {{ }}**

Las dobles llaves imprimen datos del contexto que la vista envía.

```
<p>Hola, {{ usuario.nombre }}!</p>
```

### **2. Control de flujo con {% %}**

Puedes usar bucles y condicionales para controlar la lógica del HTML.

```
<ul>
  {% for empleado in empleados %}
    <li>{{ empleado.nombre }}</li>
  {% endfor %}
</ul>
```

### **3. Condicionales {% if %}**

Ejecuta secciones según condiciones básicas como en Python.

```
{% if empleados %}
  <p>Hay empleados registrados.</p>
{% else %}
  <p>No hay empleados.</p>
{% endif %}
```



REGIONAL ANTIOQUIA  
CENTRO DE SERVICIOS Y GESTION EMPRESARIAL  
TECNOLOGIA EN ANALISIS Y DESARROLLO DE SOFTWARE  
**Programación Orientada Objetos (Python + Django)**

**INCRUSTAR FUNCIONALIDAD PYTHON EN HTML**

**4. Cargar etiquetas o filtros**

Antes de usar archivos estáticos o funciones especiales, debes cargarlas.

```
{% load static %}  

```

**5. Buena práctica: dejar la lógica en la vista**

Evita lógica compleja en los templates. Prepara los datos desde views.py.

```
# views.py  
context = {'es_admin': usuario.rol == 'admin'}
```

```
<!-- HTML -->  
{% if es_admin %}  
    <p>Bienvenido, administrador.</p>  
{% endif %}
```

**✗ Ejemplo de mala práctica**

Aunque posible, evita escribir lógica compleja directamente en la plantilla.

```
{% if usuario.rol == 'admin' %}  
    <p>Administrador</p>  
{% endif %}
```

**6. Comentarios en templates**

Los comentarios no se renderizan en el HTML.

```
{# Esto es un comentario #}
```



REGIONAL ANTIOQUIA  
CENTRO DE SERVICIOS Y GESTION EMPRESARIAL  
TECNOLOGIA EN ANALISIS Y DESARROLLO DE SOFTWARE  
**Programación Orientada Objetos (Python + Django)**

**INCRUSTAR FUNCIONALIDAD PYTHON EN HTML**

## 7. Aplicar filtros

Usa filtros como `|upper`, `|date`, `|length` para modificar datos.

```
<p>{{ nombre|upper }}</p>
<p>{{ fecha|date:"d M Y" }}</p>
<p>Total: {{ lista|length }}</p>
```

## 8. Crear tus propios filtros

Puedes definir funciones personalizadas para extender los templates.

```
# app/templatetags/mis_filtros.py
from django import template
register = template.Library()

@register.filter
def duplicar(valor):
    return valor * 2
```

## Referentes bibliográficos

1. Django Software Foundation. (2023). Django Template Language: For Python Developers. Documentación oficial de Django. <https://docs.djangoproject.com/en/stable/ref/templates/language/>
2. Forcier, J., Bissex, P., & Chun, W. (2008). Python Web Development with Django. Addison-Wesley. ISBN: 9780321670219
3. Holovaty, A., & Kaplan-Moss, J. (2009). The Definitive Guide to Django: Web Development Done Right (2ª ed.). Apress. <https://doi.org/10.1007/978-1-4302-1944-1>
4. Subramaniam, S. (2022). Django 4 By Example: Build Powerful and Reliable Python Web Applications From Scratch. Packt Publishing. ISBN: 9781801813055
5. Wieruch, R. (2021). The Road to Django: Your Journey to Mastering Django Web Development. Independently published. <https://www.roadtodjango.com/>

**Documento elaborado por:** Alvaro Pérez Niño - Instructor



REGIONAL ANTIOQUIA  
CENTRO DE SERVICIOS Y GESTION EMPRESARIAL  
TECNOLOGIA EN ANALISIS Y DESARROLLO DE SOFTWARE  
**Programación Orientada Objetos (Python + Django)**

**INCRUSTAR FUNCIONALIDAD PYTHON EN HTML**

6. Mozilla Developer Network (MDN). (n.d.). HTML: HyperText Markup Language. <https://developer.mozilla.org/es/docs/Web/HTML>
7. Grinberg, M. (2018). Flask Web Development: Developing Web Applications with Python (2ª ed.). O'Reilly Media.
8. (Aunque enfocado en Flask, es útil para contrastar la filosofía de plantillas en frameworks web de Python.)