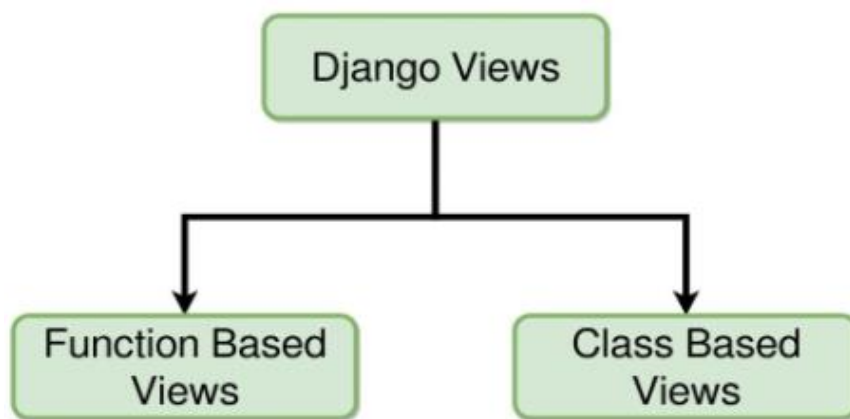


HINTS

TIPOS DE VISTAS

Las vistas de Django se dividen en dos categorías principales:

- Vistas basadas en funciones.
- Vistas basadas en clases.



VISTAS BASADAS EN FUNCIONES

Estas se escriben usando una función en Python que recibe como argumento el objeto `HttpRequest`, y devuelve un objeto `HttpResponse`. Generalmente se dividen en 4 estrategias básicas, es decir, CRUD (Crear, Recuperar, Actualizar, Eliminar), que es la base de cualquier marco que se esté utilizando para el desarrollo.

VISTAS BASADAS EN CLASES

Proporcionan una forma alternativa de implementar vistas como objetos de Python, en lugar de funciones. No reemplazan las vistas basadas en funciones, pero tienen ciertas diferencias y ventajas en comparación con ellas:

- La organización del código relacionado con métodos HTTP específicos (GET, POST, etc.) se puede abordar mediante métodos separados, en lugar de bifurcaciones condicionales.
- Las técnicas orientadas a objetos, como mixins (herencia múltiple), se pueden usar para factorizar el código en componentes reutilizables.

- Las vistas basadas en clases son más simples y eficientes de administrar que las basadas en funciones. Una vista basada en funciones con toneladas de líneas de código se puede convertir en vistas basadas en clases con solo unas pocas líneas. Aquí es donde entra en juego la Programación Orientada a Objetos.

El uso de Django de vistas basadas en funciones (FBV), y vistas basadas en clases (CBV), causa mucha confusión a los recién llegados.

En un proyecto de Django, el orden de las operaciones generalmente es:

urls.py → views.py → models.py + template.html.

Se realiza una solicitud HTTP a una URL como example.com. Dentro de Django, un archivo a nivel de proyecto `urls.py` redirige la solicitud a un archivo `views.py` apropiado que incluye un archivo `template.html`, y un archivo `models.py` (si es necesario), junto con lógica adicional en la vista misma. Luego, la vista devuelve una respuesta HTTP al usuario.

Este proceso ocurre una y otra vez para cada página web dentro de un sitio de Django.