

## HINTS

### INTRODUCCIÓN A DJANGO

Python vs. Django es incomparable, ya que el primero es un lenguaje de programación, mientras que el segundo es un marco web. Sin embargo, muchas personas aún se preguntan “¿en qué se diferencia Python de Django?”. Si tiene experiencia trabajando con Python, y no sabe qué es Django, está completamente bien. Django no es algo con lo que todos los desarrolladores de Python deban estar familiarizados. Pero, si está aprendiendo Python para el desarrollo web, Django se vuelve importante, ya que es uno de sus marcos de desarrollo web más populares. Python es un lenguaje de alto nivel, y se encuentra entre los lenguajes más populares hoy, todo gracias a su naturaleza versátil.

Este se utiliza en una amplia variedad de campos, como: la investigación científica, el desarrollo web, el desarrollo de inteligencia artificial, el análisis de datos y el desarrollo de software. Cuando hablamos de desarrollo web usando Python, necesitamos algunos tipos especiales de herramientas llamadas frameworks web para acelerar el proceso. Django es el marco web más famoso para Python que existe. Es por ello que se vuelve irrelevante diferenciarlos, pues uno forma parte del otro.

### PYTHON VS. DJANGO: DESCRIPCIÓN GENERAL, CARACTERÍSTICAS, VENTAJAS, DESVENTAJAS

A continuación, se hará una breve introducción a ambos junto con sus características, ventajas y desventajas.

#### PYTHON

Es un lenguaje interpretado de alto nivel creado por Guido Van Rossum, y se lanzó por primera vez en 1991. Es un lenguaje de programación multiparadigma, y que funciona con el concepto de programación orientada a objetos. Su sintaxis juega un papel importante para diferenciarlo de otros lenguajes de programación de alto nivel, pues la simplicidad de ésta hace que sea bastante fácil de aprender y comprender. El futuro de Python es muy brillante en comparación con otros lenguajes de programación.

### **CARACTERÍSTICAS PRINCIPALES:**

- Es un lenguaje de tipo dinámico.
- Es compatible con la programación orientada a objetos.
- También es un lenguaje de programación imperativo funcional y de alto nivel.

### **VENTAJAS:**

- Fácil de leer y entender.
- No necesita declarar el tipo de variable mientras escribe el código.
- Admite programación orientada a objetos y orientada a procedimientos.
- Se puede extender a otros lenguajes como C o C++ .
- Su código escrito en una plataforma puede ejecutarse en otras plataformas sin necesidad de realizar ningún cambio. Por ejemplo: ejecutar el código de Python escrito en Windows en otro sistema con macOS.
- macOS ya es compatible con Python, no necesita instalarlo por separado.
- Es de código abierto y de uso gratuito.
- Su comunidad de desarrolladores es bastante grande.
- Admite una gran cantidad de bibliotecas estándar.
- Tiene sus propios frameworks web.

### **DESVENTAJAS:**

- Es lento en comparación con otros lenguajes de programación, como C++ o Java.
- No es una opción ideal para desarrollar aplicaciones para teléfonos inteligentes.
- Es un lenguaje dinámico, por lo que el usuario no necesita declarar el tipo de variable mientras escribe el código. Esto puede generar errores, especialmente cuando los programadores confunden un tipo de datos con otro.
- El acceso a su base de datos aún está subdesarrollado.

## **APLICACIONES PYTHON:**

- Python ofrece muchos marcos de desarrollo web, como: Django, Pyramid, entre otros, que permiten a los desarrolladores crear sitios web y aplicaciones web sólidas.
- Es ampliamente utilizado en computación científica y numérica.
- Se puede utilizar para desarrollar un sistema operativo.
- Ayuda en el diseño de aplicaciones de escritorio basadas en GUI.

## **DJANGO**

Es uno de los marcos web más populares de Python. Un framework web está diseñado para soportar el desarrollo de aplicaciones web dinámicas.

### **Características principales:**

- Es bastante rápido.
- Contiene muchos componentes útiles.
- Es altamente seguro.
- Tiene un carácter versátil.

### **VENTAJAS:**

- Es compatible con Object-Relation Mapper, y puede definir sus modelos de datos.
- Como está escrito en Python, es muy fácil de aprender.
- Tiene una gran comunidad que se extiende por todo el mundo.
- Se utiliza para administrar el contenido de usuario del sitio web, ya sea que desee agregar un usuario o desee crear un grupo.
- Tiene un sistema de autenticación completo que puede manejar cuentas de usuario, contraseñas y cookies.

## DESVENTAJAS:

- El patrón de enrutamiento especifica su URL.
- Es excesivamente monolítico; dirige al desarrollador a escribir en el patrón dado, en lugar de la estructura de programación libre.
- No tiene la capacidad de gestionar múltiples solicitudes.
- Las baterías Incluidas en Django.

## DJANGO CON BATERÍAS INCLUIDAS:

Una de las características más apreciadas de Python, es su filosofía de baterías incluidas. Cuando lo instalamos, éste viene con una amplia biblioteca de paquetes que se pueden utilizar inmediatamente, sin necesidad de descargar nada más. Django trata de seguir esta filosofía, e incluye su propia biblioteca de paquetes comunes al desarrollo web.

## LIBRERÍAS PROPIAS DE CADA PROYECTO

Cada proyecto desarrollado con Django tiene la opción de incluir tantos paquetes o complementos que queramos, estos diferirán según los requisitos de nuestras aplicaciones. Puedes encontrar paquetes que se adapten a tus necesidades en la documentación de Django, pero hay una librería estándar que cubre la mayoría de los paquetes que normalmente usamos en estos proyectos.

## LA LIBRERÍA ESTÁNDAR DE DJANGO:

Esta vive en el paquete `django.contrib`. Dentro de cada sub-paquete hay una pieza aislada de funcionalidad para agregar. Estas piezas no están necesariamente relacionadas, pero algunos sub-paquetes de `django.contrib` pueden requerir a otros.

No hay grandes requerimientos para los tipos de funcionalidad que hay en `django.contrib`. Algunos de los paquetes incluyen modelos (y, por lo tanto, requieren que instales sus tablas en tu base de datos), pero otros consisten solamente de middleware o de etiquetas de plantillas (template tags).

La única característica común a todos los paquetes de `django.contrib` es la siguiente: si borrasas dicho paquete por completo, seguirías pudiendo usar las capacidades fundamentales de Django sin

problemas. Cuando los desarrolladores de Django agregan nueva funcionalidad al framework, emplean esa regla de oro al decidir en dónde va a residir la nueva funcionalidad, si en `django.contrib`, o en algún otro lugar.

**`django.contrib`** cuenta con los siguientes paquetes:

- **`admin`**: el sitio automático de administración.
- **`auth`**: el framework de autenticación de Django.
- **`comments`**: una aplicación para comentarios. Está actualmente bajo un fuerte desarrollo, por lo que es importante chequear constantemente el sitio web de Django para obtener la última información sobre esta aplicación.
- **`contenttypes`**: un framework para conectar "tipos" de contenido, en que cada modelo de Django instalado es un tipo de contenido aislado. Este framework es usado internamente por otras aplicaciones "contrib", y está especialmente enfocada a los desarrolladores de Django muy avanzados. Dichos desarrolladores pueden hallar más información sobre esta aplicación, leyendo el código fuente que está en: `django/contrib/contenttypes/`.
- **`csrf`**: protección ante un ataque de falsificación de petición en sitios cruzados, en inglés: Cross-Site Request Forgery (CSRF).
- **`flatpages`**: un framework para administrar contenido HTML simple, "plano", dentro de la base de datos.
- **`humanize`**: un conjunto de filtros de plantillas Django, útiles para darle un "toque de humanidad" a los datos.
- **`markup`**: un conjunto de filtros de plantillas de Django, que implementan varios lenguajes de marcado conocidos.
- **`redirects`**: un framework para administrar redirecciones.
- **`sessions`**: el framework de sesiones de Django.
- **`sitemaps`**: un framework para generar archivos de mapas de sitio XML.
- **`sites`**: un framework que te permite operar múltiples sitios web desde la misma base de datos, y con una única instalación de Django.
- **`syndication`**: un framework para generar documentos de sindicación (feeds), en RSS y en Atom.