

# INSTALACIÓN E IMPORTACIÓN DE MÓDULOS O PAQUETES EN PYTHON

---

Cómo mejorar la funcionalidad y eficiencia de nuestro código

---

# CONTENIDO



## **Qué son los módulos y paquetes**

Los módulos y paquetes son colecciones de funciones y código que se pueden utilizar en múltiples programas. Aprenderás qué son los módulos y paquetes y por qué son importantes para la programación.

## **Cómo instalar módulos**

Instalar módulos es fácil cuando se utiliza un administrador de paquetes como pip. Aprenderás cómo instalar módulos y cómo asegurarte de tener la última versión instalada.

## **Cómo importar módulos**

Una vez que tengas instalado un módulo, necesitas saber cómo importarlo en un programa. Aprenderás cómo importar módulos y cómo utilizar las funciones que contienen.

## **Cómo instalar e importar paquetes**

Los paquetes contienen múltiples módulos y pueden ser más complicados de instalar e importar que los módulos individuales. Aprenderás cómo instalar e importar paquetes y cómo encontrar paquetes útiles en línea.

---



---

# INTRODUCCIÓN

## Lenguaje de programación versátil

Python es un lenguaje de programación versátil que se puede utilizar para una amplia variedad de aplicaciones, incluyendo el desarrollo web, la ciencia de datos y la automatización de tareas.

## Módulos y paquetes

Python viene con una amplia gama de módulos y paquetes que se pueden utilizar para mejorar la funcionalidad de nuestros programas y proyectos. Algunos de los módulos populares incluyen Matplotlib para visualización de datos y Numpy para cálculo numérico.

---

# ¿QUÉ SON LOS MÓDULOS Y PAQUETES?

## **Módulos de Python**

Los módulos son archivos de Python que contienen funciones, variables y clases que se pueden utilizar en otros programas. Los módulos pueden ser escritos por el usuario o descargados de un repositorio en línea.

## **Paquetes de Python**

Los paquetes son colecciones de módulos que se utilizan para organizar el código y facilitar su reutilización. Los paquetes pueden contener otros paquetes y subpaquetes, así como módulos individuales.



---

# MÓDULOS EN PYTHON

## Definición de Módulos

- Archivos .py con funciones, clases y código reutilizable

## Encapsulación y Reutilización

- Herramientas para organizar y reutilizar código eficientemente

## Ejemplos Comunes

- Módulos como 'time' y 'math'

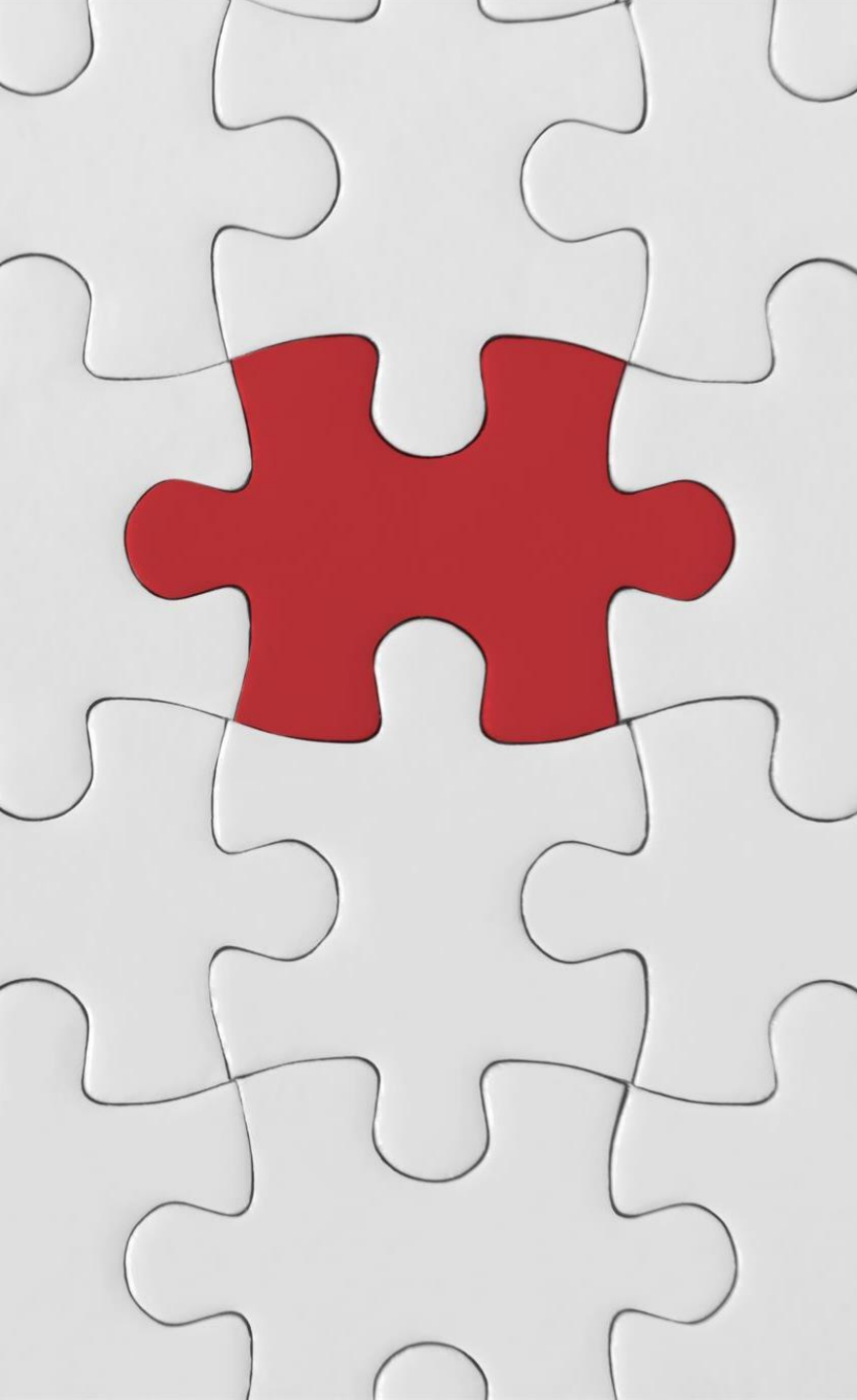
## Instalación de Módulos

- Uso de herramientas como pip para instalar módulos adicionales

## Creación de Módulos Propios

- Posibilidad de invocarlos en otros scripts
-





---

# IMPORTANCIA DE MÓDULOS Y PAQUETES

## **Reciclaje de Código**

Los módulos y paquetes nos permiten reutilizar el código existente, reduciendo así el tiempo y el esfuerzo necesarios para escribir código desde cero.

## **Aprovechar el trabajo de otros programadores**

Los módulos y paquetes existentes nos permiten aprovechar el trabajo de otros programadores, centrándonos en la lógica única de nuestro propio proyecto.

## **Organización y estructuración del código**

Los módulos y paquetes son una forma eficaz de organizar y estructurar nuestro código, lo que facilita el mantenimiento y la escalabilidad del proyecto.

---



Pip es un administrador de paquetes de Python que facilita la instalación y administración de módulos y paquetes. Podemos utilizar el comando `pip install` para instalar un módulo específico de Python.

---

# PIP

Es un sistema de **gestión de paquetes**



Muchos paquetes puedes encontrarlos en PyPI (Python Package Index) <https://pypi.org/>



Para instalar un paquete que no existe en su equipo debe usar el comando en una ventana de cmd

- `pip install <nombre del paquete>`



---

# PROBEMOS

01

Abre la terminal  
de tu SO (como  
todo un hacker)

02

Comprueba que  
PIP esté instalado

- `pip --version`

03

Lista los paquetes  
instalados

- `pip list`

---

# PROBEMOS

- Escribe la siguiente instrucción y da enter

```
pip install beautifulsoup4
```

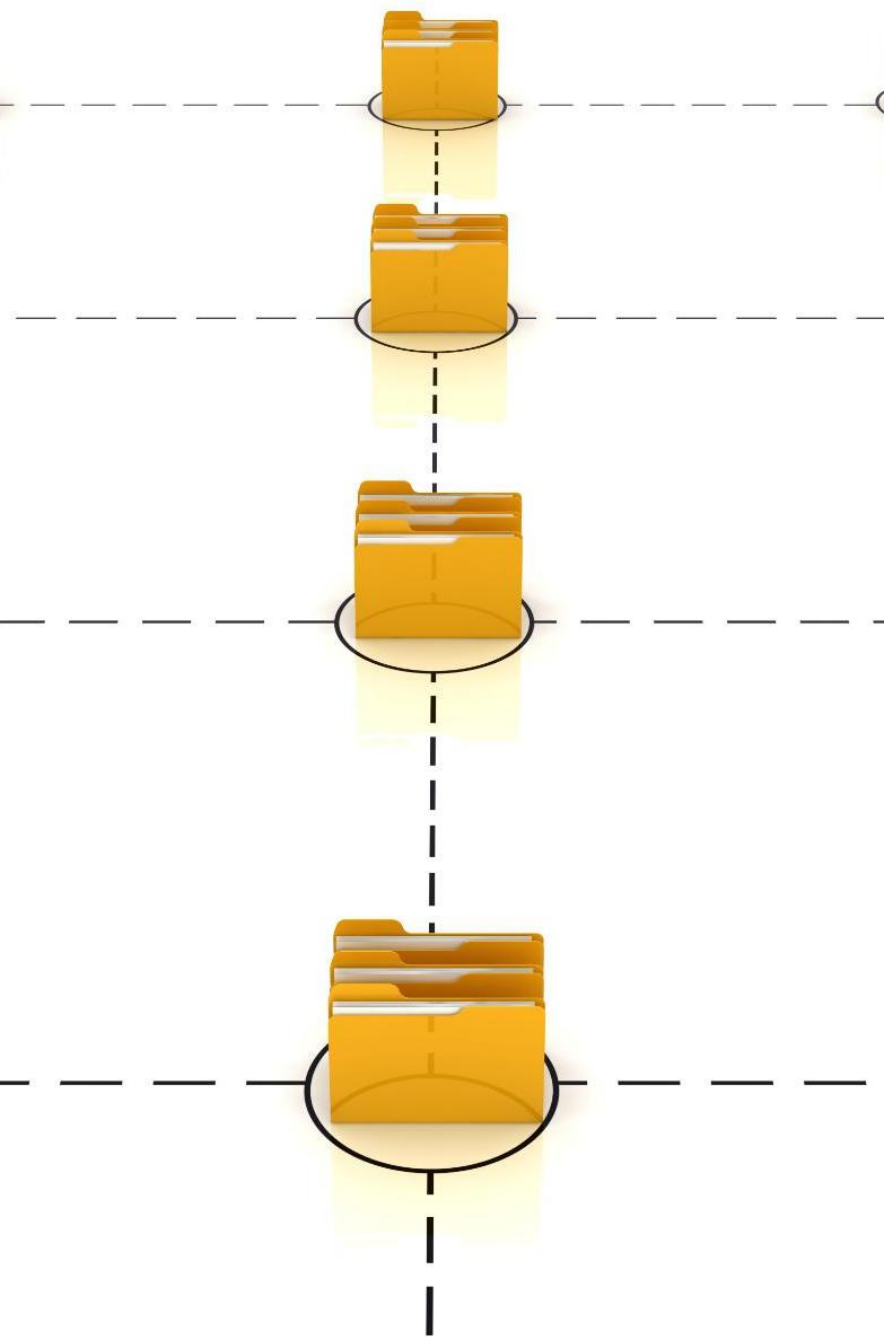
- Podemos desinstalar el paquete

```
pip uninstall beautifulsoup4
```

---

Para usar  
el paquete  
instalado  
en nuestro  
script  
agregamos

```
import <nombre del  
paquete>
```



---

# IMPORTANDO MÓDULOS

## **Declaración de importación**

Podemos importar un módulo entero o partes específicas del mismo utilizando la declaración `import` en nuestro proyecto de Python.

## **Partes específicas del módulo**

Además de importar un módulo entero, también podemos importar partes específicas del mismo, como funciones o clases, utilizando la declaración `import`.

## **Renombrar el módulo**

Podemos renombrar el módulo que importamos para hacer que nuestro código sea más legible y evitar conflictos de nombre.

---



# INSTALANDO PAQUETES

## **Administrador de paquetes de Python**

Pip es un administrador de paquetes de Python que facilita la instalación y la gestión de módulos y paquetes de Python. Es una herramienta esencial para cualquier desarrollador de Python.

## **Instalando paquetes con pip**

El comando 'pip install' se utiliza para instalar paquetes de Python. Es fácil de usar y hace que la instalación de paquetes sea una tarea sencilla.

## **Organización del código**

Los paquetes suelen contener uno o más módulos, que se utilizan para organizar el código y facilitar su reutilización. La organización del código es importante para mantener el código limpio y fácil de entender.

---

# GESTIÓN DE PAQUETES CON PIP



## Instalación de Paquetes con pip

Usar 'pip install' seguido del nombre del paquete

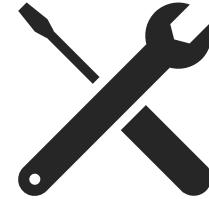
Desinstalar con 'pip uninstall'



## Instalación Masiva de Paquetes

Crear archivo txt con lista de paquetes

Ejecutar 'pip install -r requirements.txt'



## Instalación Específica de Versión

Usar 'pip3 install' para Python3

---

# IMPORTANDO PAQUETES

## **Declaración Import**

La declaración import es una función clave en Python para incorporar paquetes en nuestros proyectos. Podemos importar todo el paquete o solo partes específicas del mismo.

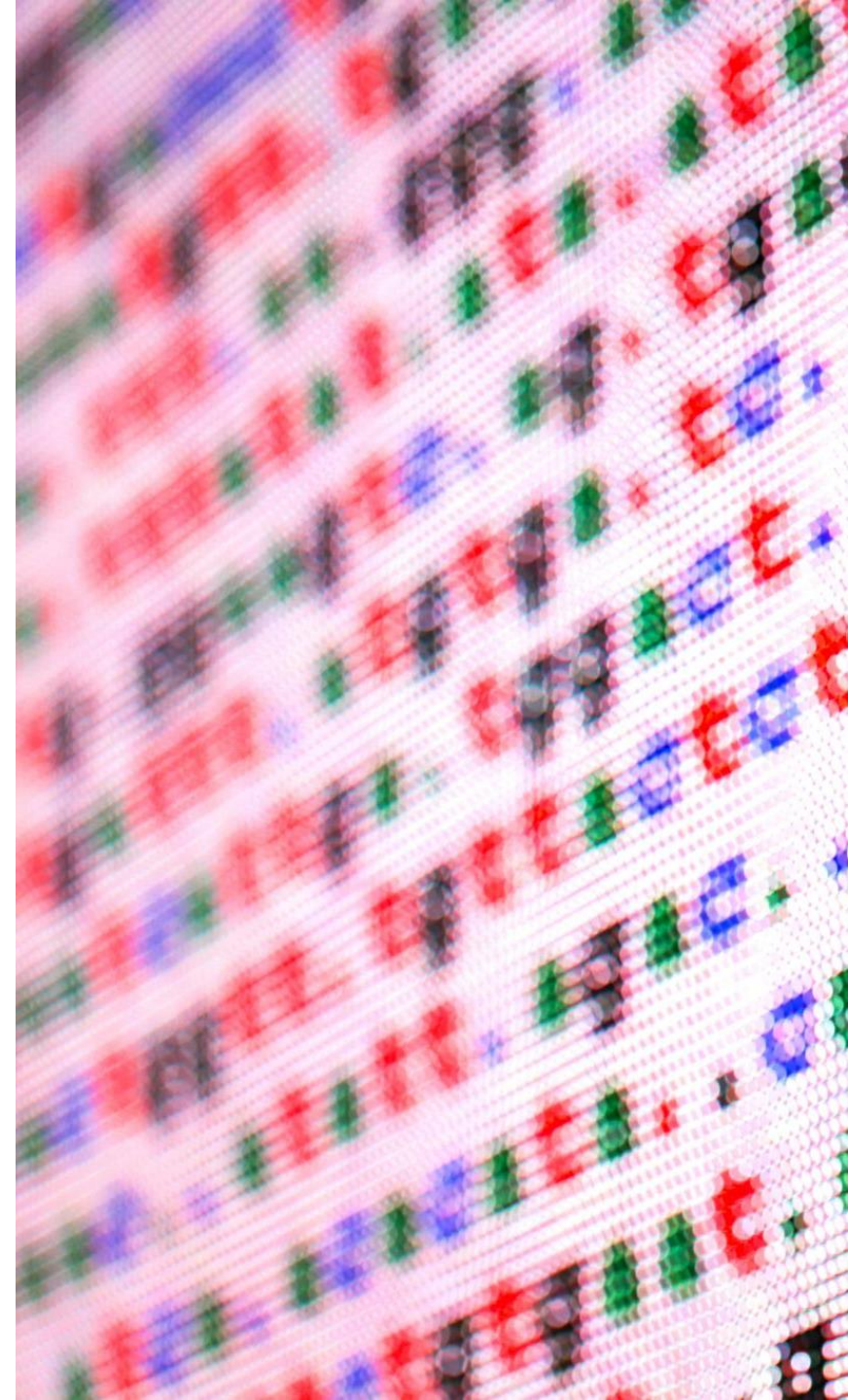
## **Importando partes específicas**




Podemos importar partes específicas de un paquete en lugar de importar el paquete completo. Esto puede ayudar a reducir el tamaño de nuestro código y mejorar su legibilidad.

## **Renombrando paquetes**

También podemos renombrar un paquete para hacer que nuestro código sea más legible y fácil de entender. Podemos asignar un nuevo nombre a un paquete de manera que sea más fácil de recordar y utilizar en nuestro código.

---



 dnspython-2.0.0	7/17/2020 6:32 AM	File folder	
 examples	7/17/2020 6:32 AM	File folder	
 tests	7/17/2020 6:32 AM	File folder	
 LICENSE	6/26/2020 2:20 PM	File	2 KB
 MANIFEST.in	6/28/2020 6:58 PM	IN File	1 KB
 PKG-INFO	7/17/2020 6:32 AM	File	2 KB
 README.md	7/16/2020 7:27 AM	MD File	4 KB
 setup.cfg	7/17/2020 6:32 AM	CFG File	1 KB
 setup	7/16/2020 7:26 AM	Python File	4 KB

## INSTALACIÓN DE MÓDULOS OFFLINE O DE TERCEROS

- Instalación de Módulos Offline o de Terceros
  - Obtener archivos del módulo a instalar
  - Descargar módulo DNSPython como ejemplo
  - Localizar el archivo setup.py en la carpeta descargada
  - Navegar a la carpeta en la terminal como administrador
  - Ejecutar el comando 'python setup.py install'



# CONCLUSIÓN

La instalación y uso de módulos y paquetes en Python es una forma efectiva de reducir tiempo y esfuerzo en la escritura de código. También ayuda a organizar y estructurar mejor los proyectos. Ahora tienes una comprensión sólida de cómo utilizar estos recursos.