

quiero deinstalar el python de mi ordenador vias cmd



Para desinstalar Python de tu ordenador usando la línea de comandos (CMD) en Windows, sigue estos pasos:

1. Abre el CMD como administrador:

- Haz clic derecho en el menú de Inicio, selecciona "Símbolo del sistema (Administrador)" o "Windows PowerShell (Administrador)".

2. Localiza el nombre exacto de la instalación de Python:

- Escribe el siguiente comando para ver una lista de programas instalados:

```
shell
wmic product get name
```

Copier le code

- Este comando puede tardar unos segundos en cargar todos los programas instalados.
- Busca el nombre exacto de Python en la lista, por ejemplo: Python 3.9.1 o Python 3.8 .

3. Desinstala Python usando el comando `wmic` :

- Una vez que tengas el nombre exacto de la versión de Python que deseas eliminar, usa el siguiente comando:

```
shell
wmic product where name="Python 3.x.x" call uninstall
```

Copier le code

- Reemplaza "Python 3.x.x" con el nombre exacto de la versión que encontraste en el paso anterior. Ejemplo:

```
shell
wmic product where name="Python 3.9.1" call uninstall
```

Copier le code

- Confirma la acción cuando te lo pida.

4. Verifica la desinstalación:

- Una vez completada la desinstalación, puedes verificar si Python fue eliminado escribiendo:

```
shell
python --version
```

Copier le code

- Si se muestra un error o indica que el comando no se reconoce, la desinstalación fue exitosa.

Recuerda que, aunque se desinstale Python, algunas variables de entorno o archivos residuales pueden quedar en tu sistema.



esta tardando mucho es normal?



Sí, puede ser normal que el comando `wmic` tarde un poco en responder, especialmente cuando está listando o desinstalando programas. La herramienta `wmic` suele ser un poco lenta en sistemas Windows porque consulta información del sistema en detalle.

Si el proceso parece estar **demasiado lento** o se queda sin responder, aquí tienes algunas alternativas:

1. Desinstala Python desde el Panel de control:

- Abre el menú de Inicio, busca "Panel de control" y accede a "Programas y características".
- Busca "Python" en la lista de programas instalados, selecciónalo y haz clic en "Desinstalar".

2. Usa PowerShell con un comando de desinstalación alternativo:

- Abre PowerShell como administrador (clic derecho en el menú de Inicio > "Windows PowerShell (Administrador)").
- Ejecuta este comando para ver una lista de aplicaciones instaladas y buscar Python:

```
shell
```

Copier le code

```
Get-WmiObject -Class Win32_Product | Select-Object -Property Name
```

- Una vez que encuentres el nombre exacto, puedes ejecutar:

shell

 Copier le code

```
(Get-WmiObject -Query "SELECT * FROM Win32_Product WHERE Name = 'Python 3.x.x').Uninstall()
```

- Sustituye 'Python 3.x.x' por el nombre exacto que viste en la lista.

como puedo saber el path donde python esta instalado

 Para averiguar la ruta de instalación de Python en tu sistema, puedes hacer lo siguiente:

1. Desde CMD:

- Abre el CMD y escribe:

shell

 Copier le code

```
where python
```

- Esto mostrará el path de la(s) instalación(es) de Python en tu sistema, si está configurada en las variables de entorno. Si tienes varias versiones de Python instaladas, se listarán todas sus rutas.

2. Desde PowerShell:

- Abre PowerShell y ejecuta:

shell

 Copier le code

```
Get-Command python
```

- Esto también mostrará la ruta de instalación de Python que esté en las variables de entorno.

3. Desde el propio intérprete de Python:

- Si puedes abrir Python, ejecuta estos comandos:

python

 Copier le code


```
import sys print(sys.executable)
```

- Este comando imprimirá el path completo de la instalación de Python que estás usando en ese momento.

Estas son las formas más rápidas de obtener la ruta de instalación de Python en tu ordenador.

como puedo saber si tengo el pyhon por win 32 o 64

 Para verificar si tienes Python en la versión de 32 o 64 bits, puedes hacerlo fácilmente desde la línea de comandos o desde el propio intérprete de Python.

1. Desde CMD o PowerShell

- Abre el CMD o PowerShell y escribe:

shell

 Copier le code

```
python
```

- Esto abrirá el intérprete de Python. La primera línea que aparece suele mostrar información sobre la versión y arquitectura. Por ejemplo, verás algo como:

scssCopier le code

```
Python 3.9.1 (tags/v3.9.1:1a84f5, Dec 8 2020, 07:51:42) [MSC v.1916 64 bit (AMD64)] on win32
```

- Si ves `64 bit` en la línea, tienes la versión de 64 bits. Si dice `32 bit`, entonces es la versión de 32 bits.

2. Usando el intérprete de Python

- Si ya estás en el intérprete de Python, puedes verificarlo con este código:

pythonCopier le code

```
import platform print(platform.architecture()[0])
```

- Este comando imprimirá `64bit` si estás usando la versión de 64 bits, o `32bit` si tienes la versión de 32 bits.

3. Alternativa desde CMD

- También puedes usar este comando en CMD o PowerShell sin abrir Python directamente:

shellCopier le code

```
python -c "import platform; print(platform.architecture()[0])"
```

- Esto te dará la misma información sobre la arquitectura en un solo paso.

Estas opciones te permitirán saber si tu instalación de Python es de 32 o 64 bits.



donde ves la informacion 64bis en la respuesta Python 3.9.1 (tags/v3.9.1:1a84f5, Dec 8 2020, 07:51:42) [MSC v.1916 64 bit (AMD64)] on win32

En la respuesta de Python, la información sobre la arquitectura (32 o 64 bits) se encuentra entre corchetes, específicamente en esta sección:

csharpCopier le code

```
[MSC v.1916 64 bit (AMD64)]
```

Aquí:

- `64 bit` indica que esta es una instalación de Python de 64 bits.
- `AMD64` es la arquitectura que también representa 64 bits en sistemas Windows (incluso si tu procesador no es AMD, este es solo un estándar para identificar 64 bits).

Si fuera una instalación de 32 bits, verías `32 bit` en lugar de `64 bit` en ese mismo lugar.



ahora quiero crear un directory donde pongo mis archivos .py y otros datos del codigo tambien instalar las librerias que va a necesitar mi codigo?

Para crear un directorio de trabajo donde almacenar tus archivos `.py`, otros datos del código y las librerías necesarias, sigue estos pasos:

1. Crear un Directorio para tu Proyecto

- Abre CMD o PowerShell y navega hasta donde quieres crear el directorio del proyecto. Usa `cd` para cambiar de directorio, por ejemplo:

shellCopier le code

```
cd C:\ruta\de\tu\proyecto
```

- Crea una carpeta para tu proyecto. Supongamos que quieres llamarla `mi_proyecto` :

```
shell
```

 Copier le code

```
mkdir mi_proyecto cd mi_proyecto
```

- Ahora, el directorio `mi_proyecto` es donde puedes almacenar todos tus archivos `.py` y otros recursos.

2. Crear y Activar un Entorno Virtual

Un entorno virtual permite instalar librerías de Python solo para este proyecto, sin afectar otras instalaciones de Python en tu sistema.

- Dentro de la carpeta del proyecto, crea un entorno virtual ejecutando:

```
shell
```

 Copier le code

```
python -m venv env
```

- Esto creará una carpeta llamada `env` dentro de tu proyecto, que contendrá una copia aislada de Python.
- Activa el entorno virtual:

- En Windows:

```
shell
```

 Copier le code

```
.\env\Scripts\activate
```

- En macOS/Linux:

```
shell
```

 Copier le code

```
source env/bin/activate
```

- Cuando el entorno esté activo, verás algo como `(env)` al principio de la línea de comando. Esto indica que cualquier librería que instales solo se aplicará a este entorno.

3. Instalar las Librerías Necesarias

- Ahora, con el entorno virtual activo, puedes instalar las librerías necesarias usando `pip` . Por ejemplo:

```
shell
```

 Copier le code

```
pip install nombre_libreria
```

- Si tienes un archivo `requirements.txt` con todas las librerías que necesitas, puedes instalar todas de una vez:

```
shell
```

 Copier le code

```
pip install -r requirements.txt
```

4. Organizar Archivos en el Directorio de Proyecto

- Guarda tus archivos `.py` en el directorio de proyecto (`mi_proyecto`).
- Si tienes datos o recursos, organiza tus archivos en carpetas, por ejemplo:

```
bash
```

 Copier le code

```
mi_proyecto/ ├── env/ # Entorno virtual (no necesitas tocar esta carpeta) ├── main.py # Tu archivo principal de Python  
├── datos/ # Carpeta para datos o archivos auxiliares ├── utils/ # Código de soporte, como funciones auxiliares └──  
requirements.txt # Lista de librerías necesarias
```

5. Desactivar el Entorno Virtual cuando termines

- Para salir del entorno virtual, simplemente escribe:


```
shell
```

`deactivate` Copier le code

Este proceso te permite mantener el entorno y las librerías organizadas para el proyecto, además de facilitar el desarrollo y despliegue del código.

y si se me olvida desactivar el entorno virtual, que pasa?

-  Si se te olvida desactivar el entorno virtual, no pasa nada grave. Sin embargo, aquí tienes algunos efectos y posibles inconvenientes:
1. **Uso Continuo del Entorno:** Si mantienes el entorno virtual activado, cualquier comando de Python o instalación de librerías que hagas seguirá aplicándose solo a ese entorno virtual. Esto puede ser útil mientras trabajas en el proyecto, pero puede generar confusión si cambias a otro proyecto o directorio y sigues teniendo el entorno activo sin darte cuenta.
 2. **Riesgo de Instalaciones No Intencionadas:** Si olvidas que estás en el entorno virtual y decides instalar una librería que realmente no necesitas para el proyecto, esta librería quedará en el entorno virtual en lugar de la instalación global de Python. Esto no afectará a otros proyectos, pero puede llenar el entorno virtual con librerías innecesarias.
 3. **Pérdida de Consistencia:** Si trabajas en varios proyectos, tener un entorno virtual activado en el momento equivocado puede hacer que instales o modifiques dependencias en el proyecto incorrecto.

Cómo Saber si el Entorno Virtual Sigue Activo

Cuando un entorno virtual está activo, deberías ver su nombre entre paréntesis al inicio de tu línea de comandos, por ejemplo:



`bash` Copier le code`(env) C:\ruta\de\tu\proyecto>`

Cómo Desactivar el Entorno Virtual


Si te das cuenta de que aún está activo, simplemente escribe:

`shell` Copier le code`deactivate`

Es un proceso sencillo y seguro, y no hay consecuencias importantes si olvidas desactivarlo, más allá de esos pequeños inconvenientes.

voy a guardar esta conversacion en caso que la uso en el futuro para otros proyectos. como hago

-  Guardar esta conversación es una gran idea para tener estos pasos a mano para futuros proyectos. Aquí tienes algunas formas de hacerlo:
1. **Copiar y Pegar en un Documento:**
 - Copia toda la conversación y pégala en un archivo de texto o en un documento de Word o Google Docs.
 - Guarda el archivo con un nombre descriptivo, como "Guía de Configuración de Proyectos en Python".
 2. **Guardar en PDF:**
 - Abre la conversación en el navegador y usa la función de "Imprimir" (normalmente con `Ctrl + P` o `Cmd + P`).
 - En el menú de impresión, selecciona "Guardar como PDF" o "Imprimir en PDF" como destino.
 - Esto guardará la conversación en un PDF que puedes consultar cuando quieras.
 3. **Capturas de Pantalla:**
 - Toma capturas de pantalla de la conversación y guárdalas en una carpeta de recursos para tu proyecto.

- Organiza las capturas con nombres específicos para facilitar su uso posterior.

4. Usar Herramientas de Notas:

- Si usas una herramienta de notas como Notion, Evernote o OneNote, puedes copiar y pegar la conversación allí. Estas aplicaciones te permitirán organizar la conversación dentro de una página específica para tu proyecto.

Elige la opción que mejor se adapte a tu flujo de trabajo, y tendrás todo listo para usarlo fácilmente en futuros proyectos!