



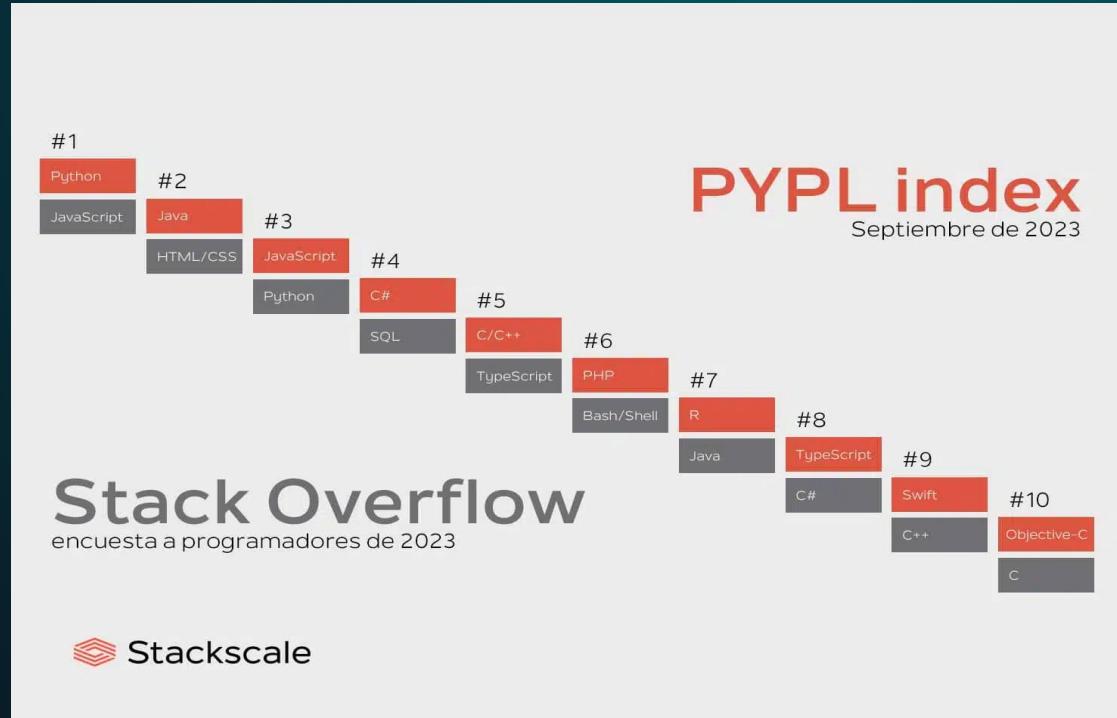
# Taller de Instalación Python, Visual Studio, jupyter y IA





El PYPL (PopularitY of Programming Language Index) se crea a partir de datos extraídos de Google Trends; analizando la frecuencia de búsqueda de tutoriales de lenguajes en Google.

# ¿Porqué Python?





## Lo elegimos por :

- ❖ Legibilidad del código:

Se destaca por su sintaxis clara y legible, lo que facilita la escritura y comprensión del código

- ❖ Amplia comunidad y soporte

Tiene una comunidad activa y grande que contribuye con bibliotecas, frameworks y soluciones a problemas comunes.

- ❖ Multifuncionalidad

Es utilizado en una variedad de campos, como desarrollo web, ciencia de datos, inteligencia artificial, automatización, scripting, entre otros.

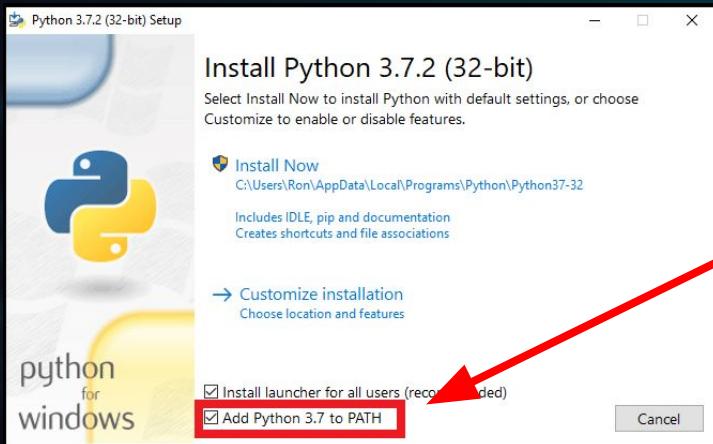
- ❖ Librerías y frameworks:

Tiene una gran cantidad de librerías y frameworks que facilitan el desarrollo en áreas específicas, como Django para desarrollo web, TensorFlow y PyTorch para aprendizaje profundo, y pandas para manipulación de datos, por nombrar solo algunos.



# Instalación

Descarga e instala la última versión de Python desde el sitio web oficial <https://www.python.org>



Durante la instalación, asegúrate de marcar la opción para agregar Python al PATH, lo que facilitará la ejecución de scripts desde la línea de comandos.



# Instalación de un IDE o editor de texto

Puedes elegir entre diferentes entornos de desarrollo integrados (IDE) como PyCharm, **Visual Studio Code**, o utilizar editores de texto como Atom o Sublime Text.



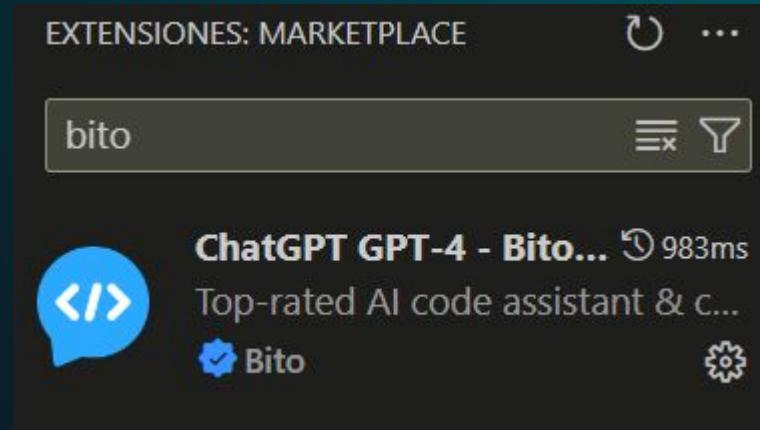
- Gratuito y de Código Abierto
- Multiplataforma (Windows, macOS, Linux)
- Extensible mediante extensiones
- Y mas ...



<https://code.visualstudio.com/>



# Instalación de Chat-Gpt en Visual





# Crear un Entorno Virtual

El concepto de entorno virtual en Python se refiere a la creación de un ambiente aislado en el cual se pueden instalar y gestionar paquetes y librerías de forma independiente de otros proyectos en el mismo equipo.



# Crear un Entorno Virtual

1. Para crear un nuevo entorno virtual:

```
bash
```

```
</> □
```

```
python3 -m venv nombre_del_entorno
```

2. Para activar el entorno virtual (en Windows):

```
bash
```

```
</> □
```

```
nombre_del_entorno\Scripts\activate
```

3. Para activar el entorno virtual (en MacOS/Linux):

```
bash
```

```
</> □
```

```
source nombre_del_entorno/bin/activate
```



# Activar y Desactivar en Powershell

```
Windows PowerShell
Copyright (C) Microsoft Corporation. Todos los derechos reservados.

Prueba la nueva tecnología PowerShell multiplataforma https://aka.ms/pscore6

PS C:\Users\Usuario> & .\Desktop\Taller-bot-\env\Scripts\Activate.ps1
(env) PS C:\Users\Usuario> python
Python 3.12.6 (tags/v3.12.6:a4a2d2b, Sep 6 2024, 20:11:23) [MSC v.1940 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> exit()
(env) PS C:\Users\Usuario> deactivate
PS C:\Users\Usuario> ■
```



# Requisitos previos en Powershell

```
Get-ExecutionPolicy
```

Copiar Editar

Esto mostrará la política de ejecución actual.

## Políticas posibles:

- **Restricted:** No se permite ejecutar scripts. Solo se pueden ejecutar comandos interactivos.
- **AllSigned:** Solo se ejecutan scripts firmados por un editor de confianza.
- **RemoteSigned:** Se pueden ejecutar scripts locales sin restricciones, pero los scripts descargados de internet deben estar firmados por un editor de confianza.
- **Unrestricted:** Se pueden ejecutar todos los scripts, aunque los descargados pueden mostrar una advertencia.
- **Bypass:** No hay restricciones, los scripts se ejecutan sin advertencias.
- **Undefined:** No hay ninguna política definida explícitamente.



# Requisitos previos en Powershell

Para cambiar la política de ejecución:

```
powershell
```

Copiar Editar

```
Set-ExecutionPolicy RemoteSigned
```

Es posible que necesites permisos de administrador para cambiar la política.



# Crear un kernel del entorno

1. Requisito tener jupyter instalado en el global
2. Tener el entorno activado
3. instalar con pip ipykernel (instala júpiter en el entorno)
4. Instalar el kernel

```
(entorno) C:\carpeta> python -m ipykernel install --user --name <nombre del kernel>  
--display-name <nombre del kernel>
```



# Comandos jupyter kernelspec

- Muestra todos los kernels disponibles en tu sistem

jupyter kernelspec list

- Eliminar un kernel

jupyter kernelspec remove nombre\_del\_kernel



# Comandos jupyter kernelspec

- Muestra todos los kernels disponibles en tu sistem

jupyter kernelspec list

- Eliminar un kernel

jupyter kernelspec remove nombre\_del\_kernel