

Introducción a Python y Configuración del Entorno

PRESENTED BY

Aitor Pérez

Grecia Landazuri Herrera

Jose Rodriguez

María Rosa

Vittoria De Novellis

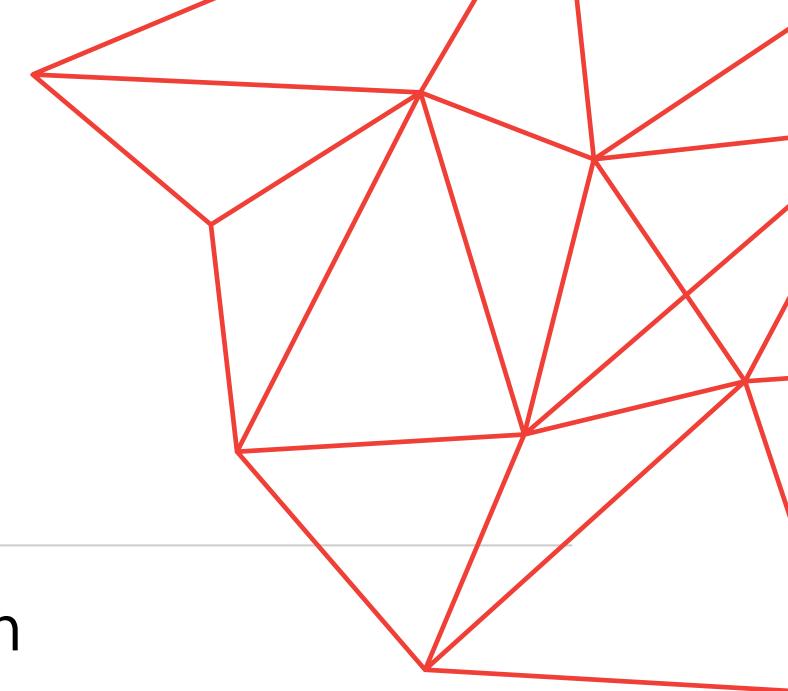


Índice



python™

3	La historia de Python
5	Usos y beneficios de Python
11	Instalación
17	IDEs para Python
23	<u>Primeros pasos con Python: Ejecución de scripts simples</u>



La Historia de PYTHON

¿Quien es Guido Van Rossum

- Guido es un programador de computación de los Paises Bajos
- Creador del lenguaje Python
- Trabajaba en el Stichting Mathematisch Centrum donde empezó el proyecto en 1989
- Publica la primera versión en 1991, la versión 0.9.0



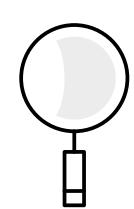
"Se llamó Python por el programa de la BBC
"Monty Python's Flying Circus"



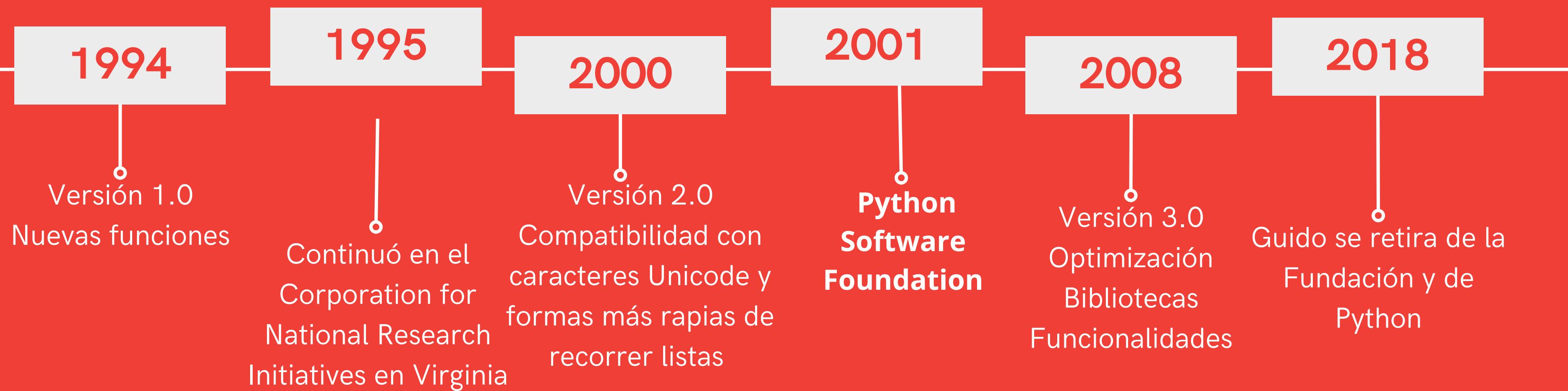
python™

Python Timeline

Primera versión 0.9.0 en 1991



"Me gustaría pensar en Python como un lenguaje de programación en el que es posible escribir código que es tan fácil de leer como el lenguaje en sí mismo"



Usos y beneficios de Python.



Beneficios que aporta Python

- **Fácil de aprender y usar**

- Sintaxis sencilla y legible:
 - Python requiere menos líneas de código para realizar la misma tarea.
 - La sintaxis de Python es más parecida al lenguaje natural, facilitando su comprensión.
 - Sin necesidad de declarar clases y métodos: para tareas simples, Python no requiere estructura compleja.
- Ideal para principiantes y profesionales:
 - Interpretado: Permite probar y ejecutar código rápidamente sin compilación.



```
# Python
print("Hola, Mundo!")
```



```
// Java
public class HolaMundo {
    public static void main(String[] args) {
        System.out.println("Hola, Mundo!");
    }
}
```

Beneficios que aporta Python

- Comunidad grande y activa

- Amplia documentación y recursos:
 - docs.python.org
 - [stackoverflow.blog -> getting-started-with-python](https://stackoverflow.blog/2020/01/23/getting-started-with-python/)
- Soporte a través de foros y comunidades en línea:
 - stackoverflow.com/questions/tagged/python
 - www.reddit.com/r/learnpython
 - github.com/python
 - ...

The screenshot shows the Python documentation homepage. At the top, there are links for About, Downloads, Documentation, Community, Success Stories, and News. The main content area features a dark background with white text, displaying Python code examples such as list comprehensions and enumerate functions. To the right, there's a sidebar with links to Docs, Audio/Visual Talks, Beginner's Guide, Developer's Guide, FAQ, Non-English Docs, PEP Index, Python Books, and Python Essays.

The screenshot shows the GitHub repository page for Python. It features the Python logo icon, the word "Python", and a brief description: "Repositories related to the Python Programming language". It displays 20.7k followers and a verified badge. Below this, there are tabs for Overview, Repositories (83), Projects (33), Packages, and People (127).

The screenshot shows a post in the r/learnpython subreddit. The title is "How Can I Make A Timer Count Down Well Still Allowing For User Input?". The post includes a user's question about creating a timer that counts down while allowing user input, followed by a code snippet in Python:

```
# Ask User For Authentication Code
user_input = str(input("Please Enter Authentication Code\n"))
```

The screenshot shows the Stack Overflow page for questions tagged with [python]. The main heading is "Questions tagged [python]". The page includes a sidebar with links for Home, Questions (which is selected), Tags, Users, Companies, LABS, and Jobs. The main content area provides a brief overview of Python as a programming language and lists 2,199,506 questions. Navigation links at the bottom include Newest, Active, Bountied (13), Unanswered, More, and Filter.

Beneficios que aporta Python

- **Bibliotecas y frameworks robustos**

- Extensa colección de bibliotecas para diversas tareas (NumPy, Pandas, Matplotlib).
 - funcionalidades preconstruidas.
 - soluciones especializadas.
 - añaden nuevas capacidades al lenguaje.
- Frameworks populares para desarrollo web (Django, Flask).
 - base predefinida con componentes reutilizables y diseños ya establecidos.



desarrollo rápido de aplicaciones web



operaciones matemáticas avanzadas



generación de gráficos a partir de datos



manipulación y análisis de datos

Beneficios que aporta Python

- **Portabilidad**

- Compatible con diferentes sistemas operativos (Windows, Linux, macOS).

- **Versatilidad**

- Usado en diferentes áreas como:

- desarrollo web
- ciencia de datos:
 - limpieza y preparación de datos con Pandas + operaciones matemáticas con NumPy.
 - búsqueda de patrones en los datos gracias a gráficos y diagramas creados con Matplotlib y Seaborn.
- inteligencia artificial:
 - creación y entrenamiento de modelos de aprendizaje automático con Scikit-learn.



Instagram.

Necesita procesar enormes cantidades de datos y gestionar un número aún mayor de interacciones entre usuarios cada segundo.

Google

Aprovecha una combinación de Django con Python para procesar grandes conjuntos de datos y realizar análisis de los mismos.



Spotify.

Apuesta por Python tanto para servicios backend como para aprendizaje automático. En el apartado frontend también se combina con Django.

Beneficios que aporta Python. Versatilidad.

- automatización y Scripting:
 - Scripts para automatizar tareas repetitivas:
 - automatizar la descarga de imágenes desde una URL.
 - Gestión de sistemas y automatización de procesos:
 - un script que verifique regularmente el estado de los servidores y notifique si algo sale mal.
- desarrollo de software y prototipos:
 - desarrollo rápido de aplicaciones y prototipos:
 - aplicación de lista de tareas pendientes.
 - herramientas como PyQt y Tkinter (incluido en la biblioteca Python estándar) son útiles para diseñar interfaces gráficas de usuario.

```
import requests
import os

def descargar_imagen(url, carpeta_destino):
    try:
        # Realiza una solicitud HTTP para obtener la imagen
        response = requests.get(url)

        if response.status_code == 200:
            # Extrae el nombre del archivo de la URL
            nombre_archivo = os.path.basename(url)
            # Crea la ruta completa para guardar la imagen
            ruta_guardado = os.path.join(carpeta_destino, nombre_archivo)
            # Guarda la imagen en la carpeta especificada
            with open(ruta_guardado, 'wb') as archivo:
                archivo.write(response.content)
            print(f"Imagen descargada y guardada en {ruta_guardado}")
        else:
            print(f"Error al descargar la imagen. Código de estado: {response.status_code}")
    except Exception as e:
        print(f"Error al descargar la imagen: {str(e)}")
```

```
class Tarea:
    def __init__(self, descripcion, completada=False):
        self.descripcion = descripcion
        self.completada = completada

class ListaTareas:
    def __init__(self):
        self.tareas = []

    def agregar_tarea(self, descripcion):
        tarea = Tarea(descripcion)
        self.tareas.append(tarea)

    def marcar_completada(self, indice):
        if 0 <= indice < len(self.tareas):
            self.tareas[indice].completada = True

    def mostrar_tareas(self):
        for i, tarea in enumerate(self.tareas):
            estado = "✅" if tarea.completada else "❌"
            print(f"{i + 1}. {estado} {tarea.descripcion}")
```

Instalación

Windows

Linux

MacOS



Antes de instalar

Comprobar si tenemos Python

Abre una consola y escribe el comando `python`, si está instalado, verás un respuesta que incluirá el número de versión.

```
C:\Users\maria>python
Python 3.12.3 (tags/v3.12.3:f6650f9, Apr 9 2024, 14:05:25) [MSC v.1938 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> |
```

Verificar que versión necesitamos

Asegúrate de descargar la versión de Python correcta para tu sistema operativo. Accede a la página oficial de descargas de Python www.python.org/downloads donde encontrarás la versión más reciente.



Windows

- Descarga la última versión de Python
- Ejecuta el instalador
- Sigue las instrucciones del asistente



También puedes instalar python a través
de Microsoft Store



⚠ Asegúrate de seleccionar la opción de
"Agregar Python a PATH" para que puedas
ejecutar Python desde cualquier línea de
comandos.

Linux

- A través del gestor de paquetes

Debian/Ubuntu

```
sudo apt install python3
```

Fedora/CentOS

```
sudo dnf install python3
```

Arch Linux

```
sudo pacman -S python3
```



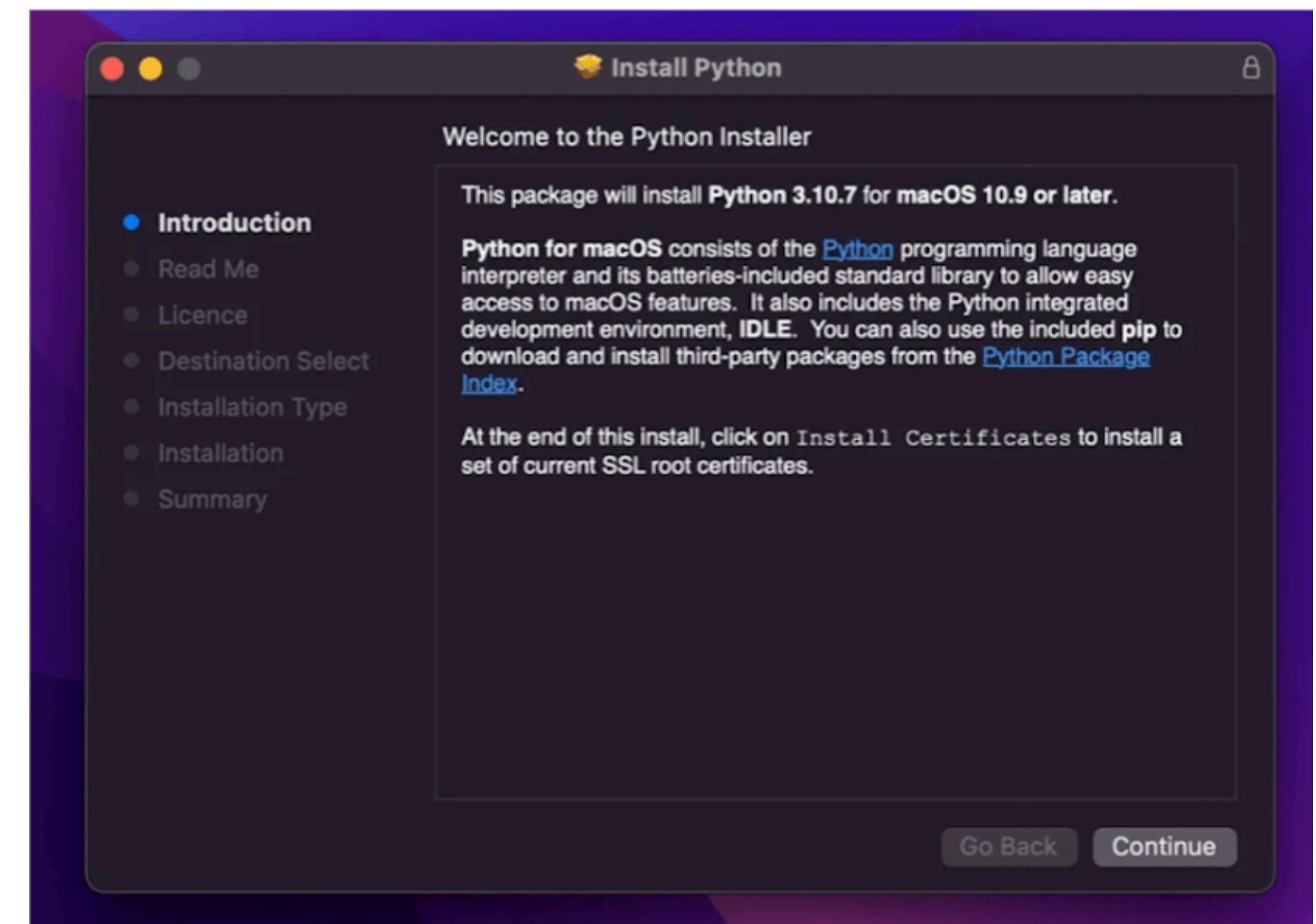
Métodos opcionales

- Desde el código fuente
- Mediante PPA

Consulta la guía de instalación

MacOS

- Descarga la última versión de Python
- Ejecuta el instalador
- Sigue las instrucciones del asistente



Verifica la instalación

Windows

```
python3 --version
```

Linux

```
python3 --version
```

MacOS

```
python3 --version
```



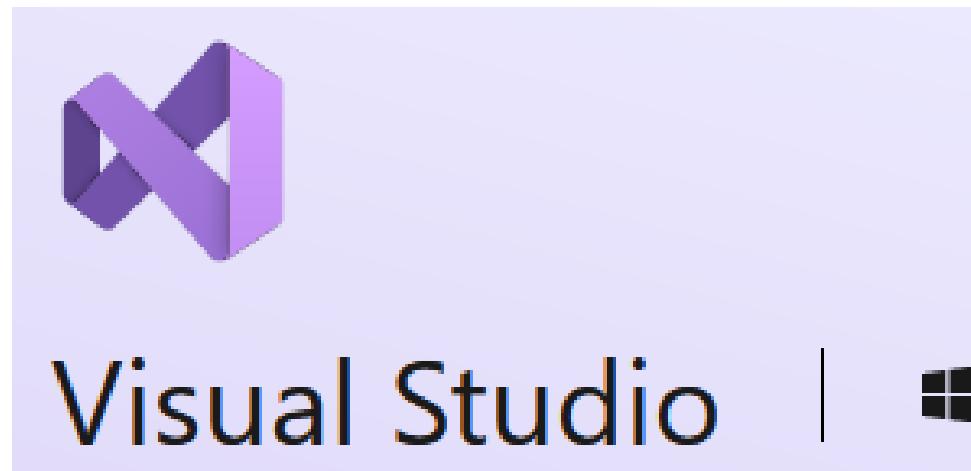
Para más información consulta la
guía de instalación

IDEs para Python

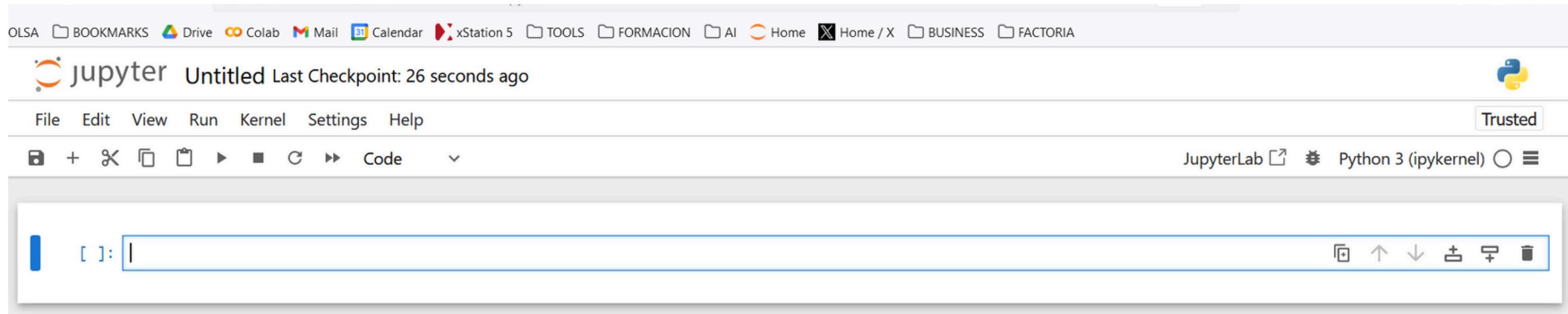


Configuración del Entorno. IDEs

Un Entorno de Desarrollo Integrado (IDE) es un software que brinda a los desarrolladores las herramientas que necesitan para escribir, editar, probar y corregir código en un único lugar



Jupyter Notebook



Su interfaz web aporta gran capacidad de visualización y facilidad para combinar código y documentación. La ejecución a nivel de celda de código ayuda a depurar prototipos. Posibilidad de incluir texto enriquecido (Markdown)

Instalación (Windows, MacOS y Linux, en todos los casos desde el terminal):

- Crear Entorno Virtual (opcional):

Crear: `python -m venv nombre_del_entorno`

Activar:

Windows: `nombre_del_entorno\Scripts\activate`

macOS/Linux: `source nombre_del_entorno/bin/activate`

- Instalar Jupyter Notebook: `pip install notebook`
- Iniciar Jupyter Notebook: `jupyter notebook`
- Instalar JupyterLab : `pip install jupyterlab`
- Iniciar JupyterLab: `jupyter lab`



Pycharm



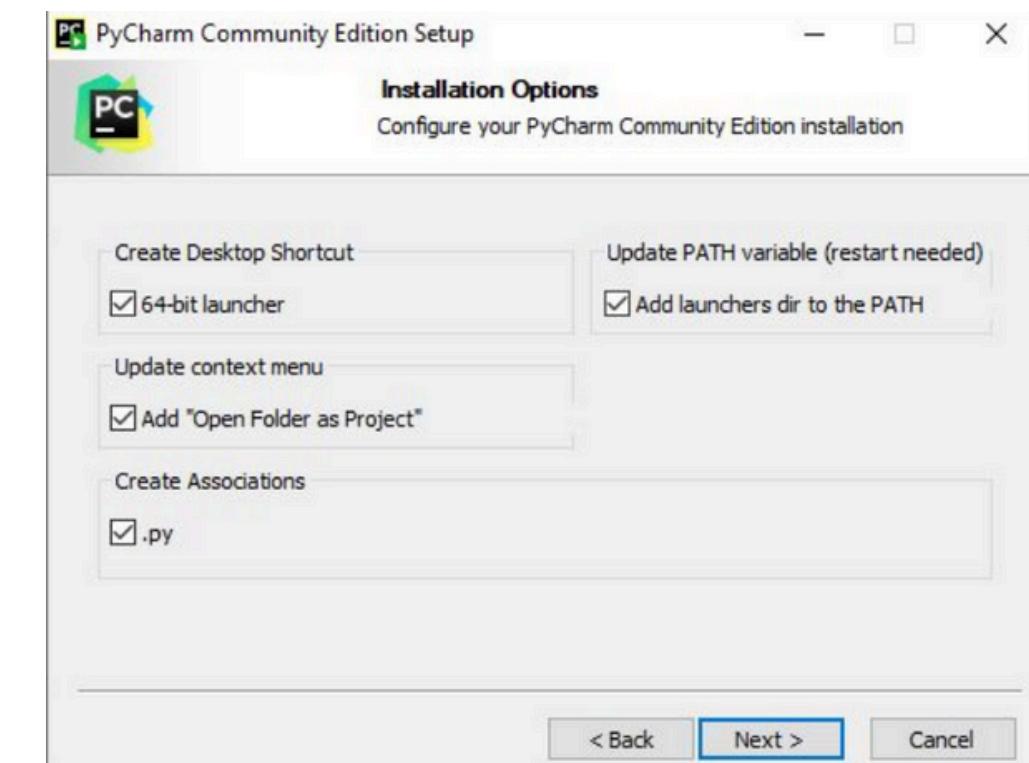
<https://www.jetbrains.com/es-es/pycharm/>

<https://www.jetbrains.com/help/pycharm/installation-guide.html>

Instaladores estándar para Windows (.exe), Linux (tarball) y MacOs

A screenshot of the PyCharm IDE interface. The main window shows a Python file named "paginator.py" with code related to pagination. A code completion dropdown is open over the word "Paginator", listing several suggestions: "Paginator", "Paginator", "Paginator", "Paginator", "Paginator", "Paginator", "Paginator", "Paginator", "Paginator", and "Paginator". The PyCharm logo is visible in the top right corner of the interface.

- Especializado en Python.
- Recomendado para proyectos complejos en su versión Profesional
- Análisis de código, testing y debugging avanzados
- Intensivo en el uso de recursos del ordenador
- Desarrollo web: integración con Django y Flask
- Completo y personalizable. Modo Zen.
- Amazon Q y GitHub Copilot plugins



Introducción a Pycharm: <https://www.youtube.com/watch?v=OoxLZILN-uI>

Visual Studio

Descargas

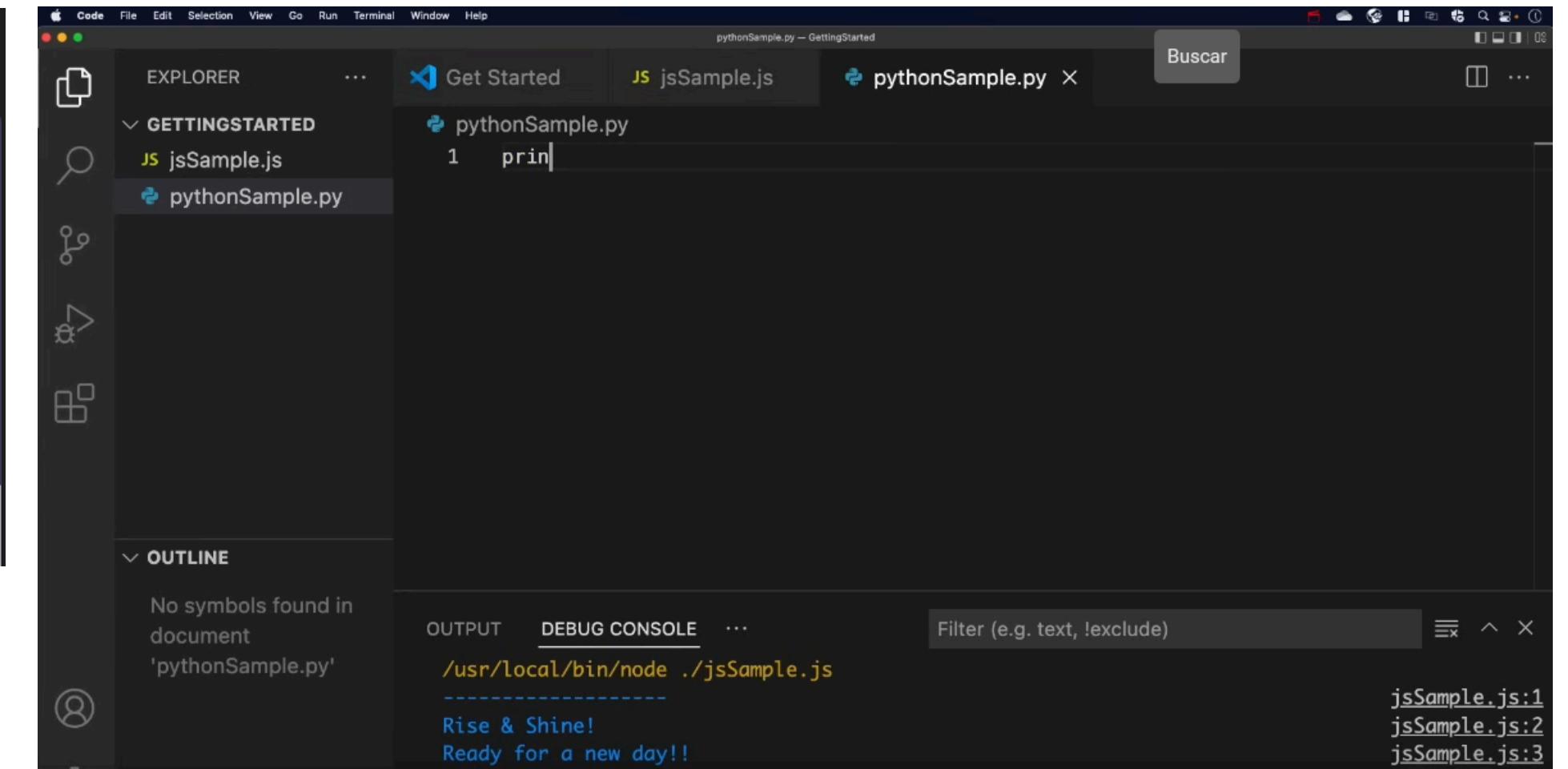
Visual Studio 2022 |

El IDE más completo para desarrolladores de .NET y C++ en Windows para crear aplicaciones web, en la nube, de escritorio, móviles, servicios y juegos.

[Comunidad](https://visualstudio.microsoft.com/es/downloads/) [Profesional](#) [Enterprise](#)

<https://visualstudio.microsoft.com/es/downloads/>

<https://code.visualstudio.com/docs/setup/linux>



Instaladores estándar para Windows, Linux (Debian y Ubuntu) y MacOs

- Intuitivo y personalizable, ofrece multiples extensiones y lenguajes de programación
- Autocompletado inteligente
- Live Share, colaboración en tiempo real
- Robusta integración con Git
- Integración con Github Copilot y Amazon Q

Introducción a VS : <https://www.youtube.com/watch?v=B-s71n0dHUk&t=66s>



Otros IDEs

Anaconda

Excelente para la creación de entornos virtuales. Incluye ya Python, R y Jupyter

- Incluye Spyder, un IDE ligero y poderoso para Python, del que proporciona ya una versión
- Instaladores estándar para Windows, Linux y macOS
- Paquetes incluidos por defecto: Pandas, Numpy, Tensorflow , Pytorch etc



IDLE

El entorno integrado de desarrollo y aprendizaje (IDLE) es el IDE de Python instalado por defecto.

Se ha desarrollado solo con Python y Tkinter . Tiene las siguientes características:

- Funcionamiento en Windows, Linux y macOS
- Ventana de *shell* para ejecutar comandos y mostrar el resultado
- un editor de textos de varias ventanas que proporciona resaltado de la sintaxis del código y compleción automática del código



Primeros Pasos con Python: Ejecución de Scripts Simples



¿Qué es un script en Python?

Un script en Python es un archivo de texto que contiene código escrito en lenguaje Python. Estos archivos suelen tener la **extensión .py**

Los scripts se ejecutan para realizar tareas específicas y se utilizan ampliamente para automatizar procesos, realizar cálculos, manipular datos y más.



Creación de un Script Simple

1 Abrir un Editor de Texto o IDE:- Puede ser cualquier editor de texto como Notepad, Visual Studio Code, PyCharm o Jupyter Notebook.

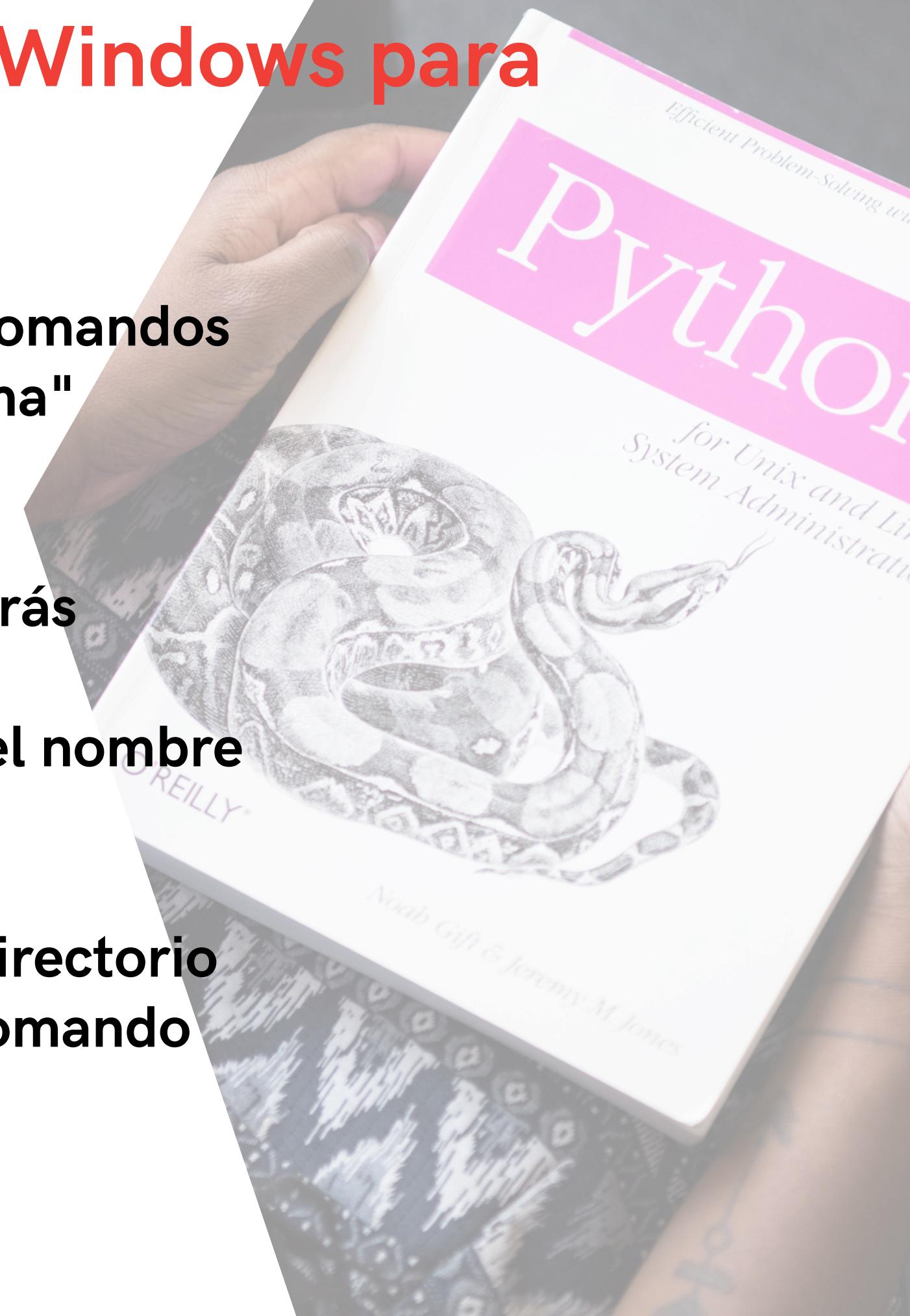
2 Escribir el Código: Aquí tienes un ejemplo de un script simple en Python que imprime "Hola, Mundo!" en la consola.

```
# Ejemplo de script simple en Python
print("Hola, Mundo!")
```

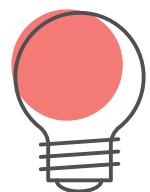
3 Guardar el Archivo:- Guarda el archivo con la extensión .py. Por ejemplo, hola_mundo.py

Uso de la línea de comandos de Windows para ejecutar scripts Python

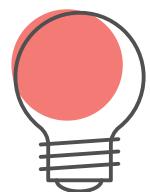
- En Windows, puedes acceder a la línea de comandos a través de la aplicación "Símbolo del sistema" (cmd.exe).
- Si has instalado Python correctamente, podrás ejecutar scripts desde cualquier ubicación simplemente escribiendo `python` seguido del nombre del archivo.
- Sin embargo, es recomendable navegar al directorio donde se encuentra el script utilizando el comando `cd` (cambiar directorio) antes de ejecutarlo.



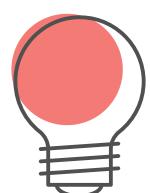
Codificación de caracteres en scripts Python



Python admite diferentes codificaciones de caracteres, como UTF-8, Latin-1, etc.



Si tu script contiene caracteres especiales (como acentos o símbolos), debes especificar la codificación adecuada al comienzo del archivo.



Puedes hacerlo agregando la línea `# -*- coding: utf-8 -*-` o `# coding: utf-8` al inicio del script.

